# Assessment of Complex Adaptive System Parameters Using a Learning Classifier System

by Jose Miguel Avalos Chumpitazi

B.S in Systems Engineering, December 2002, Universidad Nacional de Ingenieria
M.B.A. in Business Administration, December 2013, University of Florida

A Dissertation submitted to

The Faculty of
The School of Engineering and Applied Science
of the George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

January 10, 2019

Dissertation directed by

Michael Grenn
Professional Lecturer of Engineering Management and Systems Engineering

Blake Roberts
Professional Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Jose Miguel Avalos Chumpitazi has passed the Final Examination for the degree of Doctor of Philosophy as of September 27, 2018. This is the final and approved form of the dissertation.

**Assessment of Complex Adaptive System Parameters Using a Learning Classifier System**

Jose Avalos

Dissertation Research Committee:

Michael Grenn, Professional Lecturer of Engineering Management and Systems Engineering, Dissertation Co-Director

Blake Roberts, Professional Lecturer of Engineering Management and Systems Engineering, Dissertation Co-Director

Shahram Sarkani, Professor of Engineering Management and Systems Engineering, Committee Member

John Bischoff, Professional Lecturer of Engineering Management and Systems Engineering, Committee Member

Amir Etemadi, Professional Lecturer of Engineering Management and Systems Engineering, Committee Member

Thomas Mazzuchi, Professor of Engineering Management and Systems Engineering & of Decision Sciences, Committee Member

© Copyright 2018 by Jose Avalos
All rights reserved

## Dedication

The author expresses his gratitude and dedicates this Ph.D. degree to his wife Catherine and his daughters Alexandra and Angelina for all their love, support and patience during the past few years.

The author also thanks his parents Rosalvina and Santos and his brothers and nephews. These family members have also been his inspiration and support during this process.

**Abstract of Thesis**

**Assessment of Complex Adaptive System parameters using Learning Classifier System**

This study is inspired by the current tendencies of technological changes. The main purpose of this study is to analyze the factors that impact the generation of change requirements, methods of measuring those changes and the impact of those changes on the evolution of a system. Although many books and conference reports have considered the changeability of complex adaptive systems (CASs) and have studied the stability of software system requirements, there has been no in-depth study to assess which factors most impact the generation of new requirements in CASs or the relationships among CAS changeability parameters, such as fitness, agility, flexibility, robustness and adaptability. This study analyzes the impact of CAS changeability when an external agent requires a change and analyzes how such a change affects CAS evolution. The author first reviews the general concepts of changeability and CAS, and then, the author analyzes their relationship. A model using an extension of a learning classifier system (XCSF) is presented and evaluated to meet the objectives of this research, which include 1) analyzing the factors that impact the generation of new requirements, 2) accurately assessing the impact of changeability on telecommunications-based CAS components and their evolution, and 3) gaining insight into the impact of changes on CASs to improve the engineering of such systems in the future.

## Table of Contents

**Dedication** ..... **iv**

**Abstract of Thesis** ..... **v**

**List of Figures** ..... **viii**

**List of Tables** ..... **x**

**Glossary of Abbreviations** ..... **xi**

**Chapter 1 – Introduction** ..... **1**

1.1. Background ..... 1

1.2. Problem Statement ..... 1

1.3. Thesis Statement ..... 2

1.4. Research Questions ..... 3

1.5. Hypothesis ..... 3

1.6. Limitation ..... 3

1.7. Scope ..... 4

1.8. Overview of the Dissertation ..... 5

**Chapter 2 – Literature Review** ..... **8**

2.1 Technology Changes and Telecommunication ..... 8

2.2 Complex Adaptive System ..... 12

2.3 Telecommunication Technology as CASs ..... 16

2.4 Classifier Systems ..... 18

2.5 Requirement Stability ..... 25

2.6 Requirement Design ..... 28

2.7 Changeability Assessment ..... 34

2.8 Fitness ..... 41

**Chapter 3 – Model and Methods ..... 45**

3.1 Model ..... 45

3.2 Simulation Process ..... 47

3.3 Simulation Architecture (Based On Java XCSF) ..... 61

3.4 Hypothesis ..... 65

3.5 Statistical Analysis Tools ..... 66

**Chapter 4 – Results ..... 69**

4.1 Classifier Systems Results ..... 69

4.2 Impact on Change - Results ..... 71

4.3 Changeability and Fitness - results ..... 76

**Chapter 5 – Discussion and Conclusions ..... 84**

5.1 Regarding the Model ..... 84

5.2 Regarding Change Impact ..... 85

5.3 Regarding the Fitness and Changeability Aspects ..... 85

5.4 Future Research ..... 86

5.5 Future Applications ..... 86

**Bibliography ..... 89**

**Appendix A – List of Additional Incidents ..... 95**

**Appendix B – Eclipse Java Class Main Descriptions ..... 96**

## List of Figures

Figure 1-1: Parameters to be analyzed in the dissertation..... 5
Figure 2-1: S-curve of technology..... 9
Figure 2-2: Telecommunication as the act of communicating instantaneously at a distance ..... 11
Figure 2-3: Telecommunication business processes ..... 11
Figure 2-4: Telecommunication systems..... 12
Figure 2-5: Framework of an engineered CAS..... 14
Figure 2-6: Adaptive model of a telecommunication network..... 17
Figure 2-7: Underlying dynamic relationships involving CASs..... 17
Figure 2-8: Original LCS system..... 19
Figure 2-9: ZCS system..... 20
Figure 2-10: XCS system..... 20
Figure 2-11: XCSF process..... 21
Figure 2-12: Requirement changes ..... 26
Figure 2-13: Volatility of requirements versus duration of the project ..... 26
Figure 2-14: Manufacturing versus service implementation..... 28
Figure 2-15: Process of requirement maturity..... 31
Figure 2-16: Lambert's W function ..... 33
Figure 2-17: Assessment of requirement stability with the tolerance factor..... 34
Figure 2-18: Example of changeability process..... 35
Figure 2-19: MATE process flow..... 36
Figure 2-20: A changeable black box..... 36
Figure 2-21: Aspects of changeability..... 37
Figure 2-22: Aspect – principle – correlation matrix..... 37
Figure 2-23: Flexibility and adaptability definitions ..... 38
Figure 2-24: Non-linearity of perceived utility with regards to benefits..... 38
Figure 2-25: Definitions of robust, scalable and modifiable ..... 40
Figure 2-26: Landscape with maximal interdependence..... 43
Figure 2-27: Representation of evolution in three dimensions ..... 44
Figure 3-1: Proposed model to analyze the parameters in this dissertation..... 46
Figure 3-2: Relationship between changeability and classifier systems..... 47

Figure 3-3: Telecommunication change..... 48
Figure 3-4: Assessment process..... 48
Figure 3-5: Telecommunication service model ..... 49
Figure 3-6: Simulation process..... 50
Figure 3-7: Histogram of total incidents..... 52
Figure 3-8: MATLAB execution for the Lambert's function..... 55
Figure 3-9: General algorithm aligned to XCSF..... 61
Figure 3-10: Java XCSF flow ..... 62
Figure 3-11: Structure of java classes developed based on JavaXCSF ..... 64
Figure 3-12: Print screen of the JavaXCSF in Eclipse ..... 65
Figure 4-1: Error tendency for real and simulated data..... 70
Figure 4-2: Fitness over time for real and simulated data..... 70
Figure 4-3: Minitab test analysis ..... 72
Figure 4-4: Normal probability plot..... 72
Figure 4-5: Variance – residual analysis..... 72
Figure 4-6: Impact analysis..... 73
Figure 4-7: ANOVA analysis..... 73
Figure 4-8: RMI vs technology and performance..... 74
Figure 4-9: RMI vs technology and cost..... 75
Figure 4-10: RMI vs technology and tradeoff..... 75
Figure 4-11: Residual plot for flexibility and fitness ..... 77
Figure 4-12: Flexibility and fitness relationship..... 78
Figure 4-13 Residual plot for adaptability and fitness ..... 79
Figure 4-14 Adaptability and fitness relationship..... 79
Figure 4-15: Residual plot for adaptability and robustness..... 81
Figure 4-16: Robustness and fitness relationship..... 81
Figure 4-17: Residual plot for agility and fitness..... 82
Figure 4-18: Agility and fitness relationship..... 83
Figure 5-1: Final objective of the dissertation..... 88
Figure B-1: Telco service java main methods..... 96
Figure B-2: Main classifier classes..... 97

## List of Tables

Table 2-1: XCSF: update procedure with RLS..... 24

Table 3-1: Product information sample..... 50

Table 3-2: Original telecommunication data sample…..... 52

Table 3-3: Data translated to classifier system..... 53

Table 3-4: Sample input data for the requirement ..... 56

Table 3-5: RMI assessment ..... 56

Table 3-6: Example of the calculation of flexibility..... 56

Table 3-7: Example of the calculation of adaptability……..... 57

Table 3-8: Incidence sequence ……..... 59

Table 3-9: Interval of time for a problem “problem with the network fiber” ..... 59

Table 4-1: VIF values for the variable ..... 76

Table 4-2: Regression – analysis of variance for the requirement..... 76

Table 4-3: Model summary for the requirement ..... 76

Table 4-4: Lack of fit analysis for flexibility ..... 78

Table 4-5: Model summary – fitness and adaptability ..... 80

Table 4-6: Analysis of variance – fitness and adaptability ..... 80

Table 4-7: Lack of fit for robustness ..... 81

Table 4-8: Model summary – fitness and agility ..... 83

Table 4-9: Analysis of variance – fitness and agility..... 83

## Glossary of Abbreviations

CAS: Complex adaptive system

Telco: Telecommunication company (Telecommunication system)

LCS: Learning classifier system

XCS: Extension classifier system

XCSF: Function classifier system

ZCS: Zeroth level classifier system

GA: Genetic algorithm

DMTG: Distributed management task force

ITU-T: International Telecommunication Union-Telecommunication

[N]: Array of population

[M]: Array of match set

$cl$: Classifier of a classifier system

RLS: Recursive least square

FP: Number of function points that represents requirements

ΔFP: Change in the number of function points

SDR: System design requirements

IOR: Input and output requirement

TYR: Technology requirement

PR: Performance requirement

CR: Cost requirement

TR: Tradeoff requirement

STR: System test requirement

RMI: Requirement maturity index

$R_T$: Total number of requirements

$R_C$: Total number of requirement of changes

$r_{max}$: Maximum point of requirement stability

W(t): Lambert function

MATE: Multi-attribute trade space exploration

TRIZ: Russian acronym that means theory of inventive problem solving

FX: Flexibility

H(P): Measure of entropy based on the probability P for a specific value

AM(t): Measure of agility at time t

Kbps: Kilobits per second

BIT_i: Classifier value for the attribute “i”

TS: Telecommunication service

VIF: Variable inflation factor

# Chapter 1 – Introduction

## 1.1. Background

This dissertation is based on two concepts. The first concept is John H. Holland's concept of complex adaptive systems (CASs), which outlines characteristics that are very common among all systems, such as the presence of a large number of interacting components, the performance of nonlinear activities, the changeability of the system and the system's ability to learn and adapt from experience [23][24]. CASs are composed of agents and based on the behavior of the above characteristics, John H. Holland introduced the concept of classifier systems that represent agents in the form of "IF condition THEN action." The second concept involves changes in systems, especially in technology systems.

## 1.2. Problem Statement

Currently, technological changes are occurring faster than ever before. The "s-curve" of technology, described by Foster (1986) [1], has been shrinking in recent years, meaning that the requirements for change have grown quickly compared with the growth in the early 2000s. Additionally, as Fricke and Schultz mentioned, a system must be designed not only to meet customer or market needs but also to allow future changes through its entire lifecycle [2]. Thus, this study of changeability is critical for every system. As Ross mentioned, all customers and stakeholders desire changeable systems; however, they are not always willing to pay for this feature [34]. Arteta and Giachetti mentioned in their research that "dealing with change is one of the most fundamental challenges facing organizations today" [46]. They also

mentioned that the main drivers of change in organizations include available technology, the accelerated speed of technological development, access to technology, globalization, job skill shifts, environmental factors, and the increasing diversity of customer expectations. A proposed solution to ease stakeholder concerns and address the rapid technological changes is the analysis of the: (1) factors that generate new requirements, (2) changeability, (3) system evolution and (4) relationships among these factors. This paper quantitatively analyzes the input, output, technology performance, cost and tradeoff factors, which were described by Wymore, and the impact of these factors on the design requirements for a telecommunication complex adaptive system (CAS).

### 1.3. Thesis Statement

Previous relevant studies [3]–[7] have (sometimes separately) analyzed changeability and CASs. Maeshiro considers whether the level of robustness positively or negatively affects a system, but he does not include CASs in his analysis [3]. Edson analyzes how a project team, which is considered a CAS in the paper, adapts to change using a multilevel analysis method [42]. This dissertation will build upon these studies and present a model to analyze the relationship between changeability and CASs. These and other previous studies show that various factors influence the generation of requirements and a relationship occurs between the evolution of any system and the changeability parameters. Therefore, the main objective of this dissertation is to provide stakeholders with a detailed study of changeability that can provide insights into changeability, describe the factors with the greatest impact on changeability and discuss the ways in which the value of changeability changes as a system evolves.

These objectives can inspire stakeholders to ask how changes are generated and executed.

## 1.4. Research Questions

Based on the previous discussion, this dissertation will answer the following questions: which of the requirement factors described by Wymore (cost, performance, tradeoff or technology) has more and less of an impact on the generation of new requirements? Does a relationship occur between changeability and CAS evolution measured by the fitness? Do flexibility, adaptability, robustness and agility affect fitness?

## 1.5. Hypothesis

This dissertation will enhance system engineer knowledge through the evaluation of two hypotheses: (1) there are factors that affect, at different levels, the generation of new requirements and consequently the stability of the requirements of the system, and (2) there is a relationship between changeability and system evolution. The first hypothesis will use the concepts of Wymore that describe which factors (e.g., I/O, technology, performance, cost, tradeoff and system testing) must be included in system design requirements. The second hypothesis will use the concept of fitness, which describes the evolution of a system. The concepts of robustness, agility, flexibility and adaptability will also be used.

## 1.6. Limitation

In this dissertation, the analysis is based on telecommunication services. Chapter III and IV provide additional details on telecommunication services and their underlying technical information. Customers are the main actors of change in this

dissertation. Other external factors that could trigger a change, such as unpredicted natural disasters, are not considered in this dissertation.

Regarding the factors that influence changeability, this dissertation based its analysis on Wymore, who considers cost, performance, technology, tradeoff and quality as the factors that influence changeability. This dissertation does not consider quality due to the limitations of this information. In Chapter 4, the telecommunication stakeholder trade off costs and performance are assumed to be equal.

Finally, regarding changeability, only four aspects considered the main aspects of changeability by Fricke and Schultz [2] are considered in this dissertation.

## 1.7. Scope

Figure 1-1 describes the parameters analyzed in this dissertation. The center of the figure represents the complex adaptive system (CAS), which is the main concept of this dissertation. All other concepts will be related to this concept. The left side of the graph shows the factors that generate new requirements for the CAS. The top side shows the fitness of the system that represents numerically how the CAS evolves. The bottom side shows the changeability parameters that will be studied in this dissertation. Finally, the graphic shows the results of these three assessed values, which represent the emergence and feedback of the system to continue learning and evolving.

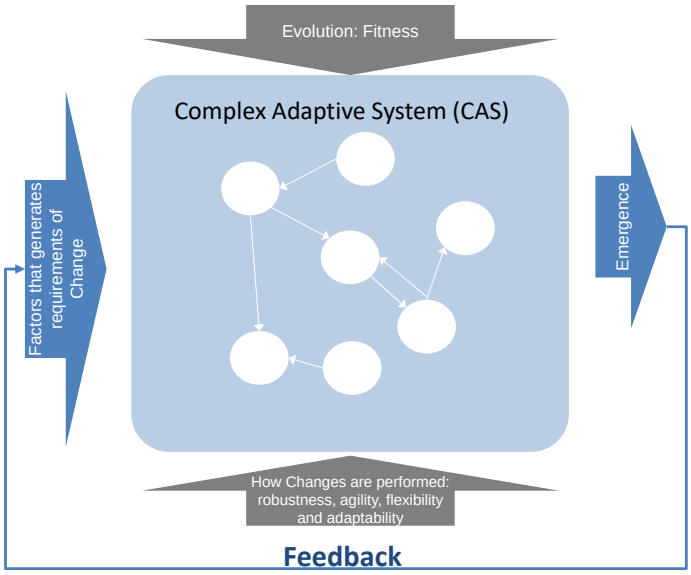


Figure 1-1: Parameters to be analyzed in the dissertation

## 1.8. Overview of the Dissertation

Chapter 2 will start by reviewing diverse studies on recent technological changes and their relationship with telecommunications. The main objective of Chapter 2 is to present different research with the objective of explaining the concepts in Figure 1-1. First, to represent the center of Figure 1-1, concepts of CASs will be introduced. This section will include different concepts of CASs, beginning with Holland's concepts. Holland was the pioneer in studying CASs. Then, referring to the left side of Figure 1-1, this chapter describes in detail the factors of the system design requirements described by Wymore that are measured during the initiation and evolution of a system. This dissertation will use requirement stability, which is a concept commonly used in software projects, to examine telecommunications systems by measuring the impact on the generation of new requirements. Requirement stability is a good concept that allows the author of this dissertation to determine which factors have greater or lesser impact on the generation of new

requirements. Referring to the bottom of Figure 1-1, the concept of changeability will be discussed, and as a result of this discussion, four critical concepts of changeability (i.e., flexibility, agility, robustness and adaptability) will be studied in detail. This chapter will discuss numerical assessments of these critical changeability concepts. This chapter will conclude with a discussion of fitness as a measure of system evolution. Fitness is represented at the top of Figure 1-1.

Chapter 3 will explain the telecommunications model to be analyzed in this dissertation. This model is based on the two system layers that comprise a telecommunications system: (1) the service layer that describes which product a customer has requested and (2) the technical layer that describes which resources are used to technically satisfy customer requests. The parameters of both layers will be used to describe the components of the system. This chapter will also describe the method of modeling the telecommunication system as classifier system using a sample of data to demonstrate system evolution. The classifier systems will receive as inputs a sample of information from the evolving telecommunications system. Finally, this chapter describes in detail the hypothesis of this dissertation and the statistical concepts that will be used to evaluate the hypothesis.

Chapter 4 describes the results of the simulation of the classifier system. Several simulations have been performed, and one sample of data is used to evaluate the hypothesis. This chapter starts with a demonstration of the performance of the model presented in this dissertation in which the fitness and error of the model are compared for simulated and real data. As was described previously, the main objectives are (1) demonstrating the existence of parameters that, to some extent, affect the generation of requirements and (2) demonstrating the presence of a relationship between the

evolution assessed by the fitness value and the changeability assessed by fitness, adaptability, agility and adaptability values. To accomplish those objectives, this chapter will present numerical and graphical statistical results.

Chapter 5 will discuss the conclusions and future recommendations. The author starts by discussing the novelty and good performance of the model described in this dissertation. This discussion is based on the different simulations performed using the proposed model. Then, the author discusses the main factors that influence the requirement stability in CASs. In addition, the relationship between fitness and changeability will be discussed. Finally, this chapter will give future system engineers insights on future research on the concepts described in this dissertation.

Figure 1-1 provides the baseline for the concepts that will be discussed conceptually in Chapter 2, modeled and simulated in Chapter 3, discussed in terms of the results of the simulation in Chapter 4, and discussed in the conclusions and future recommendations in Chapter 5.

# Chapter 2 – Literature Review

This chapter will review the concepts needed to achieve the dissertation's objectives described in Figure 1-1. The concepts of technology, CASs, design of requirements, changeability and fitness will be elaborated upon in detail in this chapter.

## 2.1 Technology Changes and Telecommunication

In recent years, technology has been changing quickly. Foster introduced the concept of the "s-curve" of technology, which is shown in Figure 2 [1]. The "s-curve" shows the effort required to improve performance over time. The slope of this s-curve has been decreasing over time. The bottom section of this s-curve shows that the new technology just entered the market. Then there is a period of takeoff in which the product is very well accepted by the market and there is a lot of production. The top section of the s-curve indicates the stabilization of the product. After this period of stabilization, a new product replaces this existing product. The timeline for completing all the processes of the s-process has been shrinking in recent years [72]. Understanding the ways in which systems change, the parameters that initiate change and the process of change is critical for stakeholders. This "s-curve" is particularly relevant for telecommunication systems.

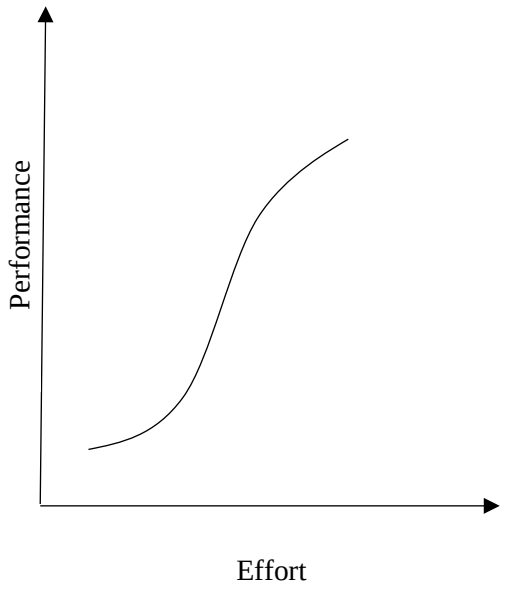


Figure 2-1: S-curve of technology

The word telecommunication is derived from the word “tele”, which means “far-off”. In general, this means communications at a distance. The initial concept for telecommunication was the telephone. Figure 2-2 is a classical view of how telecommunication works based on its definition [67]. At the top of the figure is the network used to connect the sending entity to the receiving entity. At the top of the figure are all facilities needed to transmit or connect the entities. In the middle of that figure is all information that flows during that process of connection, and at the bottom is the general process for a telecommunication system. Finally, at the bottom of the figure there are two concepts: media and service. The media “are the agencies or means that facilitate actions at a distance”. These can be real-time media, such as television messages, and recorded media, for example, a recording of a voice. There are many types of media: print, visual, audio, etc. The service is the final product for the customers and can be on voice, data or video [67]. To support these previous

concepts, telecommunication companies define their processes and systems to support the company. There have been different best practices that different organizations have been working on in recent years (DMTG, ITU-T, etc.). The best- known organization at the moment is TMForum.

Figure 4 shows the high-level business processes for a telecommunication system [67]. The process starts with a sales process in which account executives meet with the customers to sell the telecommunication company any product. Once the sales are accomplished, there is a process of provisioning of the service. Then, there are two final processes. The billing process processes all information in call detail record (CDR) format, and the customer support process gives customers a better experience while using the company's services [67]. Figure 2-4 shows the general architecture for automatization for a telecommunication system [68]. Starting from the top is the interface with the customers and end users, then all the applications that the company must use to support the processes previously described, and finally the enterprise layers, which is the foundation for the telecommunication organization. The support systems for the processes are generally called BSS/OSS (business support systems and operational support systems) in telecommunication companies [68]. Due to continuous changes in telecommunications, different companies have included concepts such as cloud, big data, analytics or artificial intelligence as part of the services they offer. Thus, the components described in Figure 2-2 have changed during the past few years. However, the components of Figure 2-2 will be studied in this dissertation.

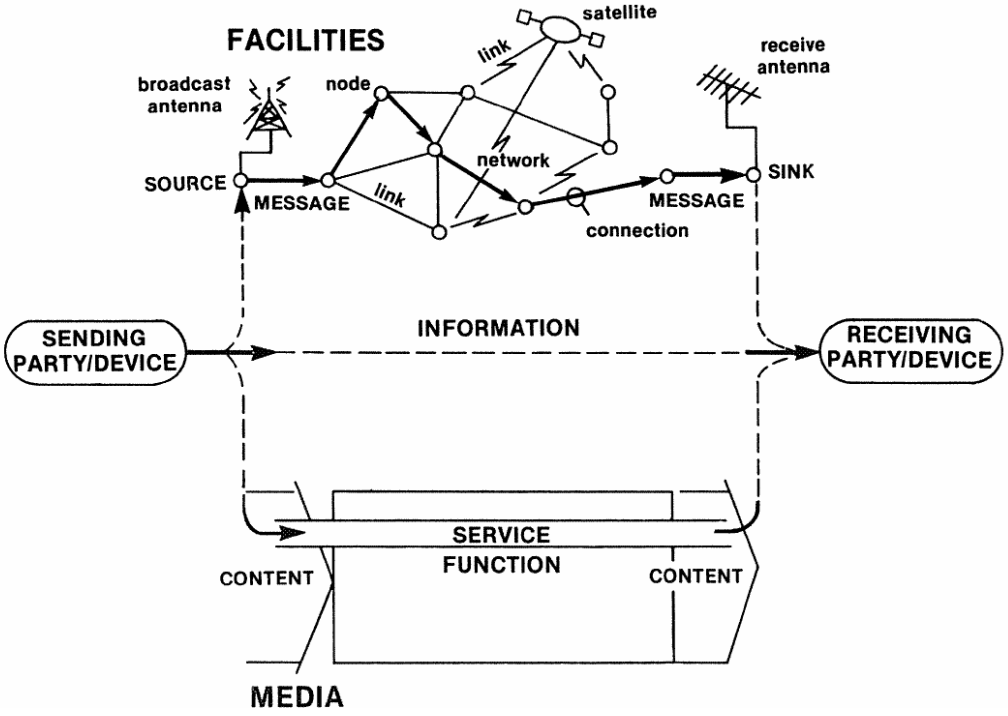


Figure 2-2: Telecommunication as the act of communicating instantaneously at a distance

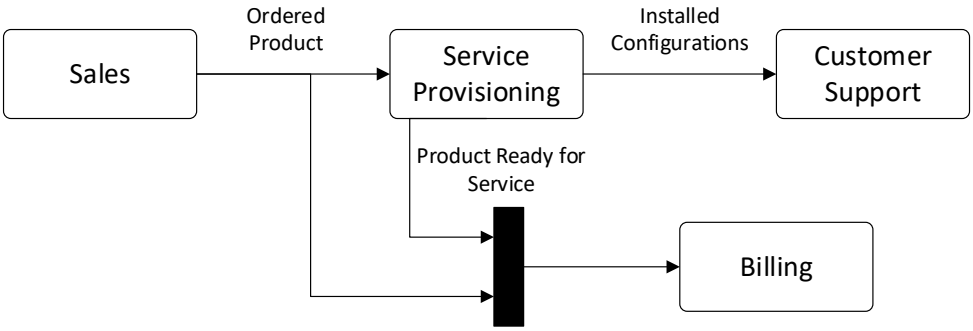


Figure 2-3: Telecommunication business processes

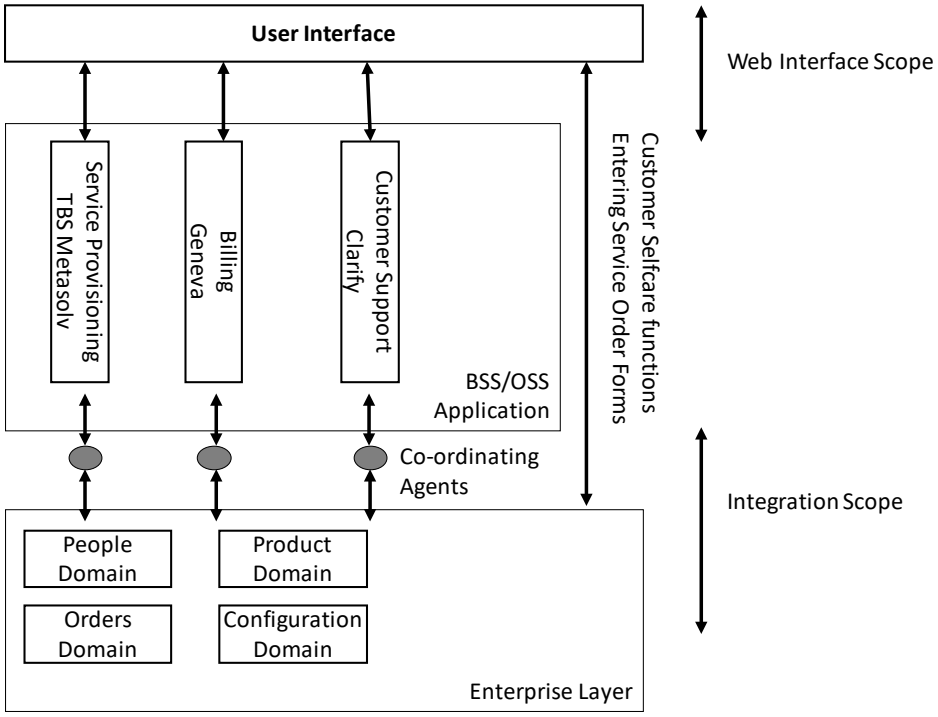


Figure 2-4: Sample of a telecommunication system architecture

## 2.2 Complex Adaptive System

CAS is the essential concept in this dissertation. There have been many studies of CASs. Two pioneering researchers will be mentioned in this section: John Holland and Stuart Kauffman. Holland proposed the theory of adaptive systems, and he defines CASs as "systems that have a large number of components, often called agents, that interact and adapt or learn" [43]. According to him, a CAS has four main features [43]:

- Parallelism: A CAS is composed of various agents that coexist and evolve together. Each of these agents works in parallel and produces signals and thus are analogous to biological cells.
- Conditional action: This concept is the basis of classifier systems because each agent behaves according to conditions and actions (IF/THEN structure).

When the “THEN structure” is performed, the system can generate a signal. Each of the signals can change over time as the system receives feedback and adapts accordingly.

- Modularity: Instead of the CAS system working individually to handle all situations, CAS agents can generate different groups – or what Holland called “sub-routines” – and work as a block to handle various and different situations. For example, biological cells can generate different types of molecules or amino acids based on distinct proteins
- Adaptation and evolution: CAS agents change often as a system attempts to maximize performance. This change is the basis for fitness, which will be described later. In the description of this feature, Holland introduced the concepts of credit assignment and rule discovery. For Holland, the credit assignment problem "arises because overt information about performance (payoff, reward, reinforcement, or the like) is often irregular and partial", and the rule discovery problem "arises when it becomes obvious that some of the agent’s rules are ineffective or detrimental". These concepts are mainly used in classifier systems

Stuart Kauffman studied CAS concepts in his book “At home in the universe”. His study was more oriented to the fitness landscape, which will be discussed in the next sections [63]. In addition to fitness, he studied CASs from the network perspective, i.e., the relationship among its components. He also studied the evolution of the system, which is a concept closely related to fitness.

In addition, with some similarity to this dissertation, Haghnevis and Askin proposed a framework of all factors to consider when there is a change in the CAS.

This framework is summarized in Figure 2-5 [49]. Flexibility and robustness, which will be analyzed in this dissertation, are included in this framework.

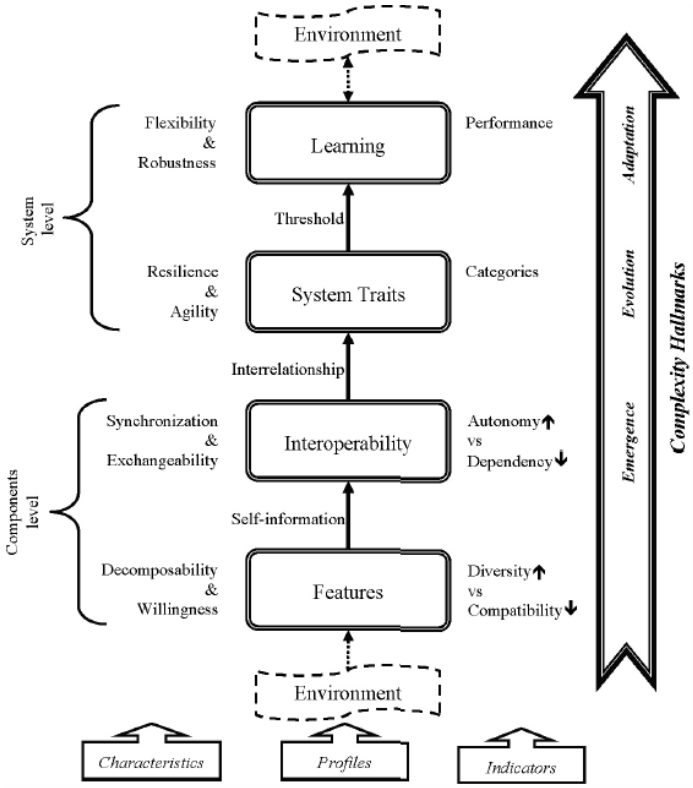


Figure 2-5: Framework of an engineered CAS

Choi et al. examined the three main characteristics of CASs as described in Figure 2-5: [48].

- Internal mechanisms: Internal mechanisms basically describe the following internal properties of a CAS. (1) Agents and schema. Agents describe all the entities that are part of the CAS, and the schema describes how the CAS functions. CAS function follows a Pareto principle, which basically indicates that the majority of the agents define the behavior of the CAS. (2) Self-organization and emergence. Self-organization represents the behavior of each agent and also the group of agents that reorganize themselves, resulting in the

emergence of new behavior in the CAS. (3) The network connectivity and its critical level. These concepts describe how agents are connected and what is the critical level of connectivity to keep agents linked [48].

- Environment: The environment describes all the agents and other entities that are not part of the CAS in the analysis. The environment has two components: (1) dynamism, which describes the constant change occurring in the environment, and (2) rugged landscapes, which will be discussed later in the fitness section, but basically is the search of the system for the optimum point compared with other systems around it [48]
- Coevolution: Coevolution implies that a CAS is always in disequilibrium due to its nature. This characteristic comprises three concepts: (1) quasi-equilibrium and state change because the CAS is always affected by the dynamism of its environment; (2) nonlinear changes because there is no way to predict the behavior of the CAS; in other type of systems, the linearity of their behavior enables the prediction of future behavior; and (3) non-random future. This concept is very important and is widely applied in this dissertation. The non-random future concept may seem contradictory to other CAS characteristics. However, even though it is difficult to predict the future behavior of the CAS, it is also correct to assume that its behavior follows a pattern that allows the construction of an architecture that researchers can use to study the CAS in detail. These characteristics have allowed, for example, the creation of concepts to model CASs, such as learning classifier systems (LCSs) [48].

Finally, McDaniels, Lanham and Anerson also studied CASs, and their concepts

will be applied to evaluate the telecommunications properties of CASs; they mentioned that, “although no real consensus exists on the set of characteristics that define a CAS, the following set of five key characteristics summarizes the major concepts from the literature: (a) diverse agents that learn, (b) nonlinear interdependencies, (c) self-organization, (d) emergence, and (e) coevolution” [22].

### 2.3 Telecommunication Technology as CASs

This dissertation simulates the sample data of a telecommunications system, and this section conceptually explains why telecommunication systems (TelCos) are CASs. TelCo connectivity is strongly dependent on the rapid changes in technology and customer needs. There are several studies that have analyzed TelCos as CASs: Wildberger demonstrated a network system similar to a CAS, but instead of nodes, the system had a power grid; with this, he demonstrated that a network is a CAS [17]. Moreover, Antoniou and Pitsillides provided an explanation of telecommunication networks as CASs [19]. Here, sample data are considered from a TelCo that offers point-to-point connectivity services to their customers, such as IP data and Internet. These services are delivered using the TelCo network. Guoliang and Tao also studied the telecommunication network as a CAS. They noted that the telecommunication network always follows nonlinear interaction with self-organization [70]. As shown in Figure 2-6, they described the process of adaptation of a telecommunication network. Finally, Choi et al. explained how supply chains are CASs. The model on which that they based their report is described in detail in Figure 2-7 [48].

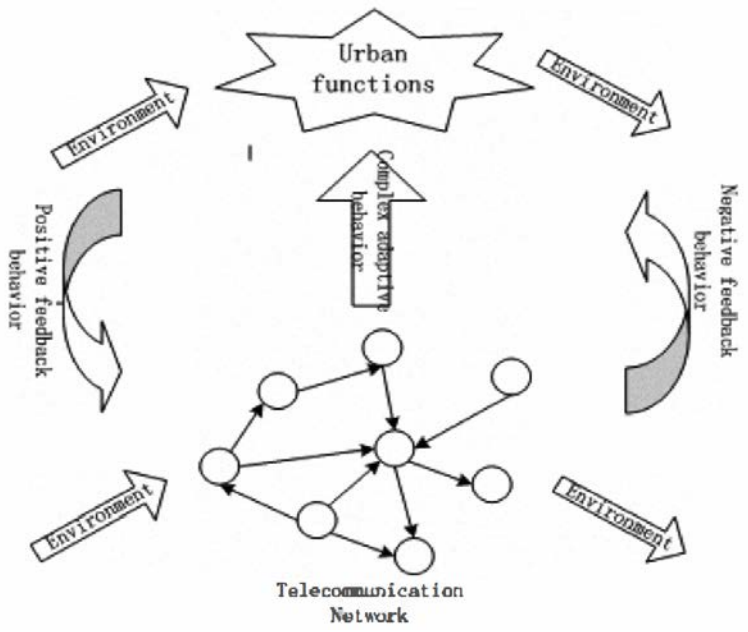


Figure 2-6: Adaptive model of a telecommunication network

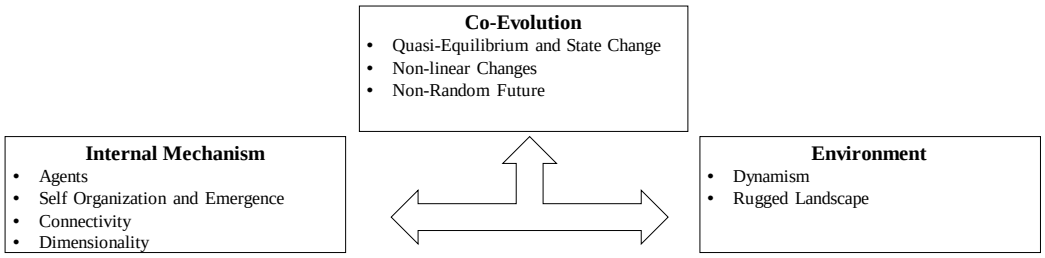


Figure 2-7: Underlying dynamic relationships involving CASs

As was described previously, the author will analyze the following aspects of a CAS to evaluate telecommunication systems. The following analysis is based on [44]:

1) Diverse agents that learn: Because each node in the network is similar to an agent in a CAS, each of these nodes learn from the experience of each user or customer who uses this node.

2) Nonlinear interdependencies: A telecommunications network does not have linear dependencies, as each node evolves and changes over time. Conceptually, the

following concepts explain the dynamism of network systems: (1) congestion control, (2) fault tolerance and attack tolerance, and (3) decentralized and distributed operation [21].

3) Self-organization: There are various perturbations that cause a network to evolve and self-organize because of the structural complexity of the TelCo network, such as connection diversity, complications due to node diversity, and random failures or change requests [21].

4) Emergence: Most TelCo networks have complex topologies, and there are special requirements for TelCo networks. Thus, models with various types of network topologies have emerged, such as small-world networks and scale-free networks. These topologies retain the most significant properties that have emerged over time to satisfy those requirements [21].

5) Coevolution: As explained previously, telecommunications networks evolve for various reasons, such as the external environment, technology and new needs from their users or customers [21].

### 2.4 Classifier Systems

In his book “Hidden Order”, Holland introduced the concept of a learning classifier system (LCS) to model CASs and the genetic algorithm (GA). He described the characteristics of CASs and proposed to represent this system as a set of rule-based agents [25]. The behavior of each agent is managed by rules, and each rule is represented as “IF condition THEN action” [45]. There have been different applications of classifier systems. Some examples are the optimization of data mining, the optimization of an aircraft, optimization of the UK market, and road traffic optimization [45].

In the original LCS classifier system, the rule and action system has only three

characters, {0, 1, #}, where # acts as a wildcard. For example, when the system has the condition and action of #101 and 001, respectively, and the system receives an input 1101, then the action will be 001. Figure 2-8 shows the original LCS system [45]. The system receives a message from the environment, and all the conditions that match the population become part of match set M. When the system has different rules to apply, there is a process of bidding to select the rule. A bid is a function of two parameters: the number of nonwildcards used in the rule and the fitness, which is related to a higher reward for the system [45].

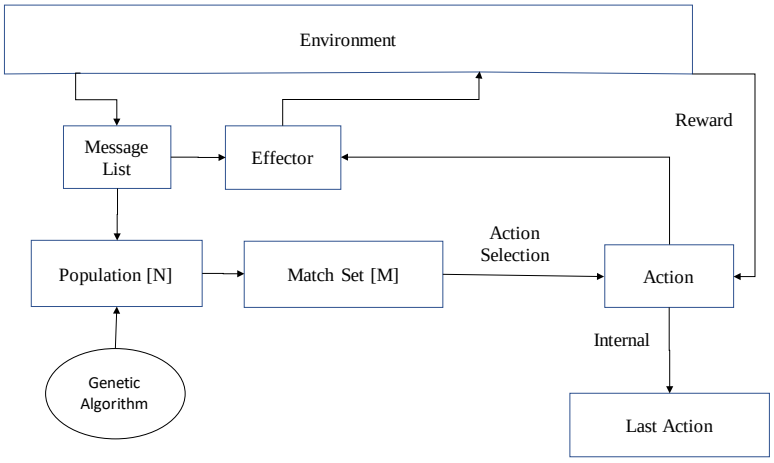


Figure 2-8: Original LCS system

Wilson introduced the concept of a zeroth level classifier system (ZCS) and removed bidding (using only fitness to select the action of the system) and the message list. Figure 2-9 shows a summary of this concept [45].

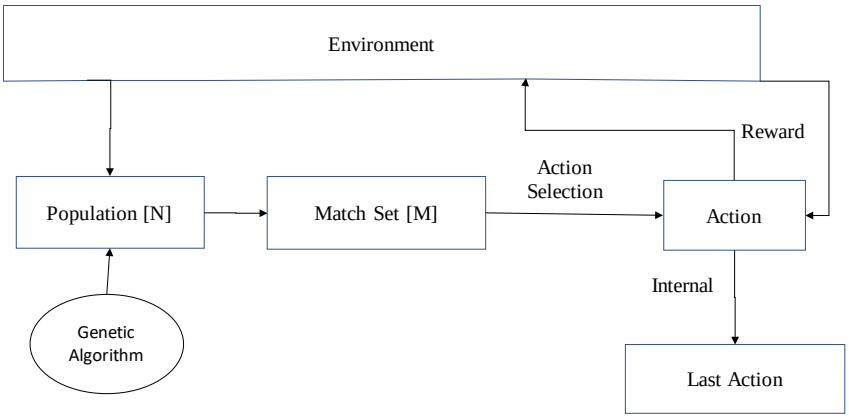


Figure 2-9: ZCS system

Wilson also introduced XCS, which is a better version of a classifier system. One of the main characteristics of this classifier is that fitness is based on the accuracy of the prediction of the system and not on the payoff the system receives [45]. Figure 2-10 shows how XCS works.

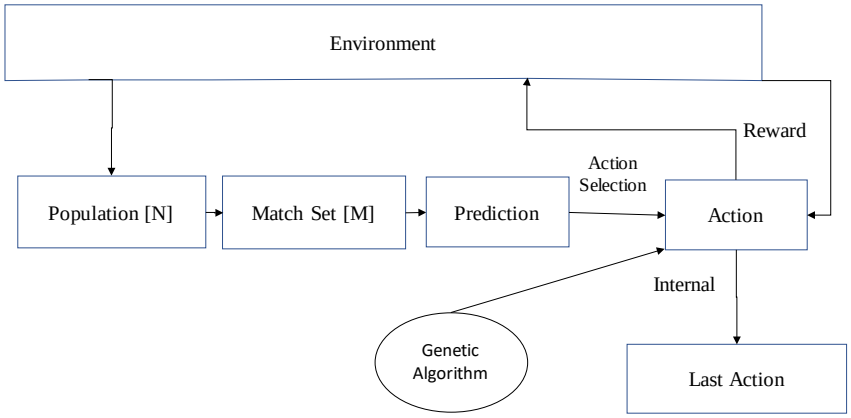


Figure 2-10: XCS system

Finally, the classifier system for function approximation (XCSF) was introduced. The main novelty of this new classifier system is that it accepts real input value [26]. XCSF is an extension of XCS in three ways [23]:

(i) XCS accepts only 0, 1, and # values as inputs, whereas XCSF accepts any numerical value.

(ii) The classifiers include the concept of vector weight (w) to accept any numerical value.

(iii) The weights are updated through the XCSF process.

Butz considers that XCS and XCSF are the most well-known architectures that have been described in the literature. The XCSF model is applied in this dissertation. Figure 2-11 graphically shows how the XCSF model works [27]. Lanzi *et al*. described how the XCSF process works [28].

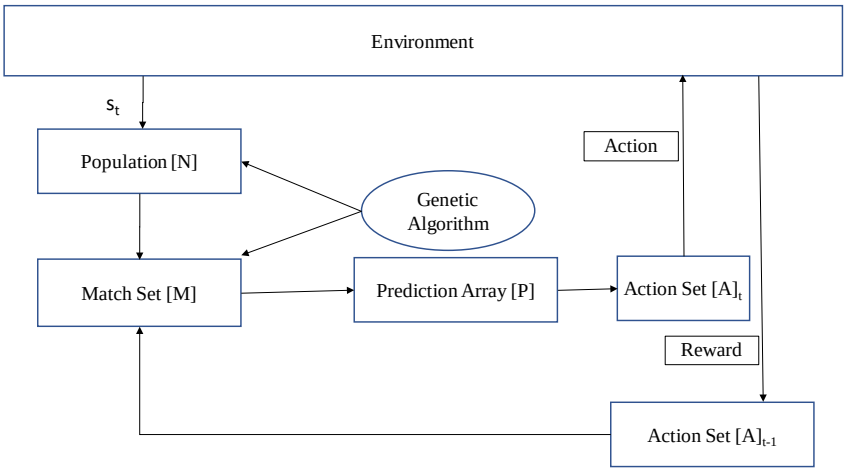


Figure 2-11: XCSF process

In general, the XCSF process proceeds as follows: when the system is initialized, it generates intervals $int_i = (l_i, u_i)$, which have the value: $l_i = s_t(i) - rand_1(r_0)$ and $u_i = st(i) + rand_1(r_0)$ for the lower and higher values of each interval, respectively, where $r_0$ is a constant value, and $rand_1$ is a random value between 0 and $r_0$. When the systems start, the weight vector, which represents the weight of each classifier, is initialized with zero values. After the system starts, a system change can be triggered by the environment. Similar to XCS, XCSF generates a match set [M] that is a subset of the population [N]. To be part of the match set, the values of the events should be part of the intervals

described previously, meaning that input $s_t$ should have a value between $l_i$ and $u_i$ [28].

Equation 2.1 shows the formula used to assess the predicted value as the result of the match set process [M]. Additionally, similar to XCS in XCSF, the fitness-weighted average uses the system-predicted reward. The predicted reward is calculated using equation 2.2. Vector weight w is important for assessing the predicted value of each classifier [28]:

$$cl.p(st) = cl.w_0 \times x_0 + \sum_{i>0} cl.w_i \times st(i) \quad (2.1)$$

$$P(s_t, a_i) = \frac{\sum_{cl\epsilon[M]|a} cl.p(s_t) \times cl.F}{\sum_{cl\epsilon[M]|a} cl.F} \quad (2.2)$$

Loiacono and Wilson described the variables as follows: "$s_t$ as the vector that represents the event input, cl.p($s_t$) as the predicted value, cl.$w_i$ as the weight of cl, $x_0$ as a constant input, cl as the classifier, [M]|a as the subset of classifiers in [M] with action a, and cl.F as the fitness of cl" [28].

Action array [A] stores all the classifiers that execute the selected action. This action is sent to the environment, and the system receives feedback from the environment to execute event $s_{t+1}$. Once that feedback is received, the weight vector is recalculated in two steps. First, a delta weight ($\Delta w_i$) is calculated using equation 2.3; in this work, the recursive least squares (RLS) algorithm is used as part of the XCSF model, and the delta weight is calculated based on this algorithm [29]. Second, using the delta weight, all weights are calculated using equation 2.5 [28]:

$$\Delta w_i = \frac{\eta * s_{t-1}(i)}{|s_{t-1}(i)|^2} \times (r - cl.p(s_{t-1})) \quad (2.3)$$

$$r = \mathrm{r}_{t-1} + \max_{a \in A} P(s_t, a_i) \quad (2.4)$$

$$cl.w_i = cl.w_i + \Delta w_i \quad (2.5)$$

where $\eta$ is a constant parameter called the correction rate, r is the reward of the system,

which is calculated in equation 2.4, and $r_{t-1}$ is the previous reward.

A classifier in XCSF represents a function that is applied only to the inputs that match the classifier condition. Other important parameters to assess the performance of the XCSF model are prediction error, accuracy and fitness. Prediction error represents the deviation of the predicted value of the system with respect to the real data. As its name suggests, accuracy represents the accuracy of the XCSF model [28]. Fitness, which is an important parameter in this research, represents how good the classifier "fits" the evolving system. Prediction error is updated as follows [28]:

$$cl.\varepsilon = cl.\varepsilon + \beta * (|r - cl.p(s_{t-1})| - cl.\varepsilon) \quad (2.6)$$

Accuracy (k) and relative accuracy (k') are calculated based on the following equations, respectively:

$$k = \begin{cases} 1 & if\ \epsilon \leq \epsilon_0 \\ \alpha(cl.\varepsilon/\varepsilon_0)^{-v} & Otherwise \end{cases} \quad (2.7)$$

$$k' = \frac{k \times num}{\sum_{cl \in [A]}(cl.k \times cl.num)} \quad (2.8)$$

Numerosity is the number of times a classifier is present in the population of the XCSF model [28]. Each action has an associated fitness that is calculated using equation 2.9. In this study, constant values α, β, and $\varepsilon_0$ are assigned to represent accuracy factor, learning rate, and error threshold, respectively [28].

$$cl.F = \text{cl}.F + \beta * (k' - \text{cl}.F) \quad (2.9)$$

There have been several studies about the type of functions to assign XCSF. Every time there is a learning process, each variable $x$ of the classifier must learn to approximate function $f(x)$. Therefore, for each $x$ as an input in the classifier, there is an associated value $\hat{f}(x)$ as the expected reward; however, the real value that the system obtains from the data in which the system will learn is $f(x)$. Then, the learning

process starts in the XCSF system, where there is an accuracy process in which evolved approximate function $\hat{f}(x)$ learns to approximate target function $f(x)$[28].

The recursive least square (RLS) algorithm is used in this dissertation, and it is very well explained in Table 2-1. This process is slightly different from the previously described process. Vector X has the information of input Matrix V, which is composed of the values $(X_t^T X_t)^{-1}$[28], where t is the step of the execution of the algorithm. This vector is very important because it helps update the weight vector. In RLS, the value of β is not constant, but updated according to step 7 in Table 2-1. The classifier and the accuracy are updated in step 8 and step 9, respectively, which is also slightly different from previously described Formulas 2.9 and 2.7. Finally, the weight vector is updated in step 11. As explained previously, Formulas 2.1 through 2.9 describe the general process. RLS is a function that, in general, follows the same process. There are other functions, such as Kalman and NLMS, that are not within the scope of this dissertation.

Table 2-1: XCSF: update procedure with RLS

| Step | Comment |
|---|---|
| 1: procedure UPDATE PREDICTION(cl, s, P ) | |
| 2: error ← P - cl.p(s); | Compute the current error |
| 3: x(0) ← $x_0$; | Build $\vec{x}$ by adding $x_0$ to s |
| 4: **for** i ∈ {1,..., $\|s\|$} **do** | |
| 5: x(i) ← s(i); | |
| 6: end for | |
| 7: $\beta_{rls}$ ← 1+ $\vec{x}$ × cl.V × $\vec{x}^T$; | Compute matrix update rate $\beta_{rls}$ |
| 8: cl.V ← cl.V − $\beta_{rls}^{-1}$ × cl.V × $\vec{x}^T$×$\vec{x}$× cl.V ; | Update the classifier cl.V related to V |
| 9: $\vec{k}^T$ ← cl.V × $\vec{x}^T$ ; | |
| 10: **for** i ∈ {0,... , \|s\|} **do** | Update classifier weights |
| 11: cl.$w_i$ ← cl.$w_i$ + $\vec{k}(i)$× error; | |
| 12: **end for** | |
| 13: end procedure | |

## 2.5 Requirement Stability

Bush & Finkelstein studied the changes in the requirements of complex systems. This dissertation applies these concepts to CASs and assesses and analyzes the factors that initiate new requirements in CASs, where change is used to adapt to the environment [9]. Several telecommunications system scenarios will be modeled as CASs. Many studies of requirement stability and volatility are based on software projects [8][10]–[14]. Bush and Finkelstein discussed the impact of feasibility analysis on requirement stability; they consider feasibility in terms of the analysis of goals, performance, output and scenarios [8][9]. Costello and Liu analyzed several metrics for requirement engineering, including requirement volatility, traceability, completeness, defect density, fault density, consistency and integrated progress. Requirement volatility, which is described as a measure of the additions, deletions and modifications of requirements, is considered in this dissertation [10]. Guo described a variable, ΔFP, that describes all changes in requirements; to show all the types of requirements, he performed the analysis illustrated in Figure 2-12, where (a) shows a stable environment where changes decrease over time and (b) and (c) show unstable environments in which changes in the number of requirements do not decrease [11].

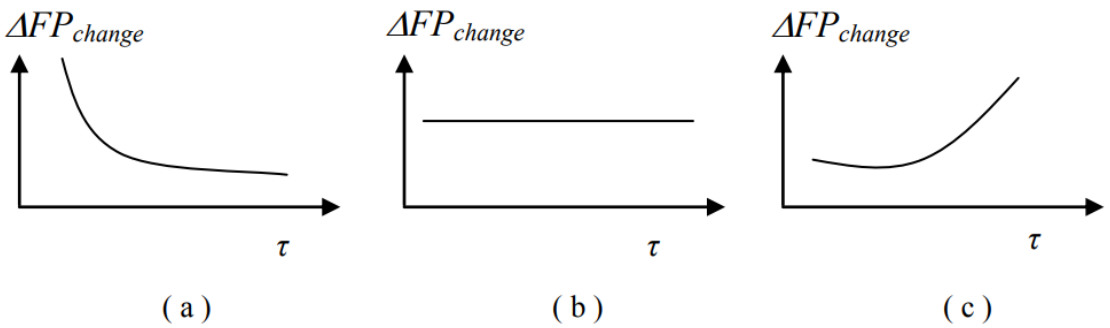


Figure 2-12: Requirement changes

Kulk and Verhoef studied in detail the relationships among software project workload, implementation time and volatility rate; they depicted the relationships among those variables as shown in Figure 2-13 [12]. This study helps further the analysis in this dissertation, as the author can assess that the acceptability of volatility rate depends on the implementation time of the software project [12].

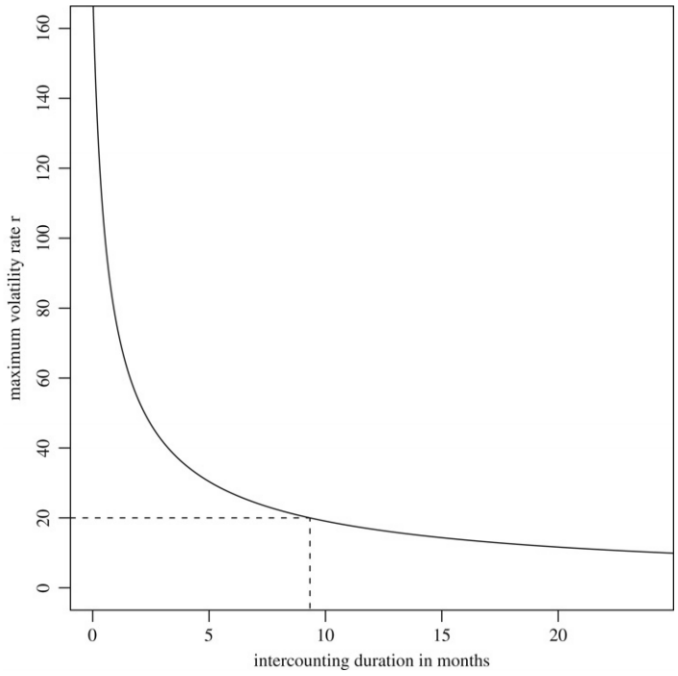


Figure 2-13: Volatility of requirements versus duration of the project

Loconsole and Börstler presented a real-world case to assess requirement stability [13], and Stark, Oman, Skillicorn and Ameele assessed requirement stability over several

software project releases [14].

This dissertation uses the previously described studies and applies them to CAS by modeling a telecommunications system. Fundamentally, software project concepts are applied in a telecommunications system because of the nature of their implementation. Software implementations and telecommunications system implementations share the same concept of service, which is very well described in [16]. The creation of products for a manufacturing process, which yields tangible products, follows a linear process. The creation of products for a service process, which applies to telecommunications and software, yields intangible products and follows a process that starts and finishes with the customer [15]. Telecommunications and software projects follow the same process for building networks and software to satisfy new customer requirements. Figure 2-14 describes graphically this difference between the concepts of the operational processes of manufacturing and technology (i.e., software and telecommunication) services [15].

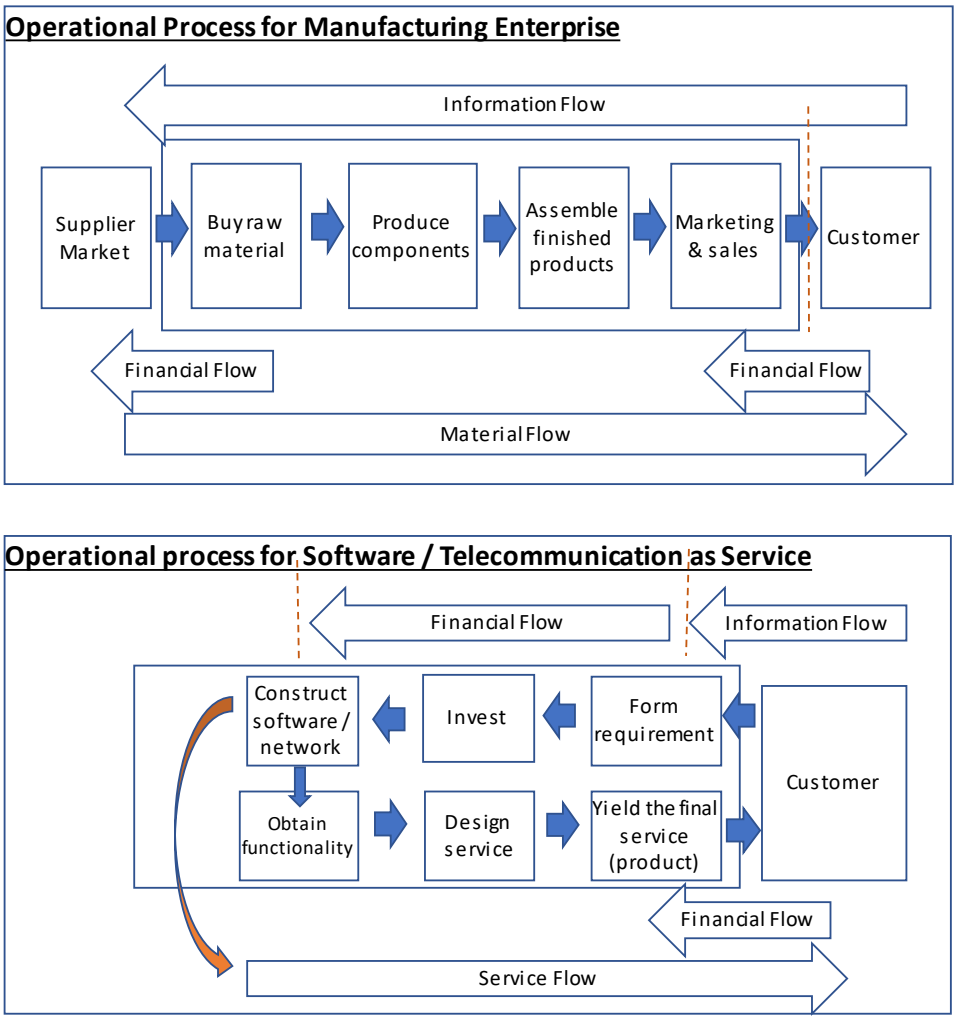


Figure 2-14: Manufacturing versus service implementation.

## 2.6 Requirement Design

Wayne Wymore's concept will be used as a baseline to analyze the main drivers that impact the customer's requirements for stability and volatility. Wymore mathematically defined that system design requirements (SDRs) have six components: input/output requirements, technological requirements, performance requirements, cost requirements, tradeoff requirements, and system test requirements [30]. He defined and provided a rigid mathematical framework. First, he defined the equation 2.10, which describes how the system transitions from one state to another [66]. Based on this definition, he proposed

equation 2.11 to describe the component of the system design for requirements (SDR).

$Z = (SZ, I\,Z, OZ, NZ, RZ)$ …… (2.10)

where Z is the system under consideration, SZ is the set of states of the discrete system Z, IZ is the set of inputs of the discrete system Z, OZ is the set of outputs of the discrete system Z, NZ is the set of next state function of the discrete system Z and RZ is the set of readout functions of the discrete system Z [66].

$SDR = (IOR, TYR, PR, CR, TR, STR)$ …. (2.11)

IOR is the I/O (functional) requirement, TYR is the technology requirement, PR is the performance requirement, CR is the cost requirement, TR is the tradeoff requirement, and STR is the system test requirement [30].

Wymore said that "when a solution is chosen in advance, the real problem is never stated, and the solution fails because the world is not seen as it is but as the problem solver wants it to be so that his or her solution will be the correct one" [66]. The author agrees with this statement and will use the fundamental concepts of Wymore to study them not in discrete mathematical way, but in a system that learns as represented by a CAS.

The input/output requirements basically define a space of system alternatives composed of a design and model for each input and output of the system. The input/output requirements define the behavior of the system, and respond to the following question: what function is the system being designed to perform [31]? An important complement of the input/output requirements is the technological requirements, which defines what is available to build the system in terms of hardware and software. Technological requirements respond to the following question: what is available to the system being designed in order for it to perform its function? Input/output requirements and technology requirements define the solution of the system [31]

Wymore defined performance requirements as the way a functional system is designed in terms of performance figures of merit, such as grade of services, response time, speed of service, capacity of service, quality of service, availability, reliability and maintainability. Performance requirements ask system engineers to “find the best functional design [they] can with respect to the performance” [31]. The cost requirement is the cost of the utilization of resources used to satisfy the other requirements. Cost requirements ask the system engineers to “find the most suitable system design [they] can with respect to the cost”. Tradeoff requirements specify the tradeoff between system costs and system performance. Tradeoff requirements ask system engineers to “find the most implementable system design [they] can with respect to the tradeoff between performance and cost”. Finally, testing requirements specify the methods and measures used to test the final system [31]. This dissertation does not consider testing requirements.

This dissertation will analyze the requirement components described by Wymore in terms of their impact on requirement stability. Noguira, Luqui and Bhattacharya analyzed several situations of requirement stability, and concluded that when requirement stability is not managed appropriately, there is a high risk of delivering low-quality products, which will negatively affect customer satisfaction [32]. Based on the definitions of birth rate and death rate, they recommend that projects should follow the process described in Figure 2-15 and conclude in the stable area [32]. Birth rate is the rate of new requirements incorporated into the system with respect to the total number of requirements, and death rate is the rate of requirements dropped with respect to the total number of requirements.

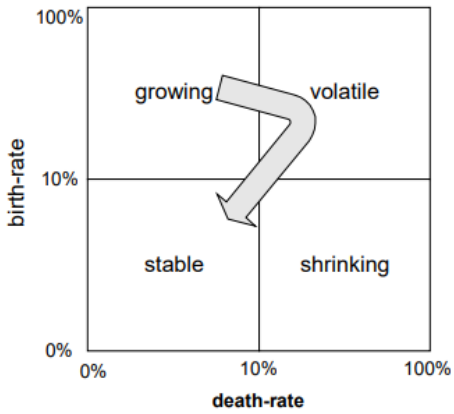


Figure 2-15: Process of requirement maturity

Most studies on requirement stability and volatility have focused on software development. This research will consider such previous studies as a baseline to measure the stability requirements of a telecommunication system as a CAS [8][10]–[14]. This research will analyze each node of the network as an element of a CAS and study how each element is generated, changed or removed over time by using a classifier system for simulation. Each origination or change is associated with a requirement; therefore, the elements of each requirement for change, as described by Wayne Wymore, will be analyzed.

The requirement maturity index (RMI) will be used to measure the impact on the stability or volatility of the requirement as high or low; requirement volatility and stability are highly related concepts that will be interchanged in this study. Nogueira, Luqui & Bhattacharya discuss the relationship between volatility and stability; they explained that stability is related to a low number of requirement changes and that volatility is related to a high number of requirement changes [32]. Stability is related to a low rate of change in requirements, and volatility is defined as change occurring very often. RMI will be calculated according to equation 2.12. This measure is

typically used to quantify the readiness of a software product, but in this case, it will be used to calculate the impact of requirement changes of a telecommunication system as a CAS [33].

$$RMI = \frac{R_T - R_C}{R_T} \ldots \quad (2.12)$$

Here, $R_T$ is the total number of requirements for a specified version, and RC is the total number of changes in that version. An RMI value close to 100% means that the system is more stable or less volatile; thus, changes to the existing system have been reduced.

To assess the stability of requirements, this dissertation will use a simple calculation of point of stability described in [12] which was previously introduced. Equation 2.13 shows how $r_{max}$ is calculated. RMI values lower than $r_{max}$ indicate that the system requirements are stable.

$$r_{max} = \left(e^{\frac{W(t)}{t}} - 1\right) X100 \quad \ldots\ldots\ldots (2.13)$$

t is the implementation time and W(t) is called Lambert's W or the omega function in honor of its creator, Johann Heinrich Lambert. This function is the solution to equation 2.14 [12]. For example, equation 2.15 is used to solve the Lambert function with a value equal to one. The solution of equation 2.15 is 0.567143290409. Figure 2-16 shows Lambert's function. As explained previously, t is the average implementation time, and therefore the solution in Figure 2-16 is considered with values of t higher than zero.

$$f(x) = x \times e^{X} \ldots..(2.14)$$

$$\Omega \times e^{\Omega} = 1 \ldots\ldots (2.15)$$

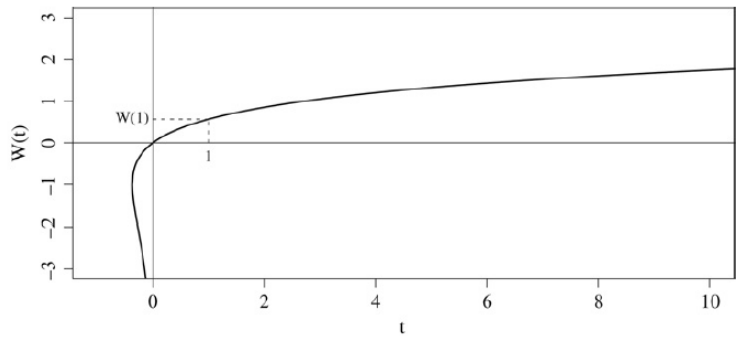


Figure 2-16: Lambert's W function

This dissertation will use $r_{max}$ values to assess the impact of requirement variables on requirement stability. Variables with RMI values higher than $r_{max}$ have a greater impact on the stability of requirements and therefore a greater impact on the generation of requirements. Finally, this dissertation discusses the tolerance factor, which was also discussed in [12]. This factor describes the extent to which stakeholders accept the demand for new requirements. Low p-values indicate that the stakeholders do not tolerate high volumes of requirements, whereas high p-values indicate a better tolerance for a high requirement demand [12]. Figure 2-17 shows the same concept as Figure 2-3 but includes different levels of tolerance. For example, for a project with a duration of 10 months a higher level of tolerance generates a system with higher maximum values of the growth rate of requirements. Formula 2.16 shows the assessment of the highest acceptable rate of growth of requirements, including the tolerance factor. In the formula, the tolerance factor is multiplied by time as part of Lambert's formula [12]. Chapter 3 will explain the level of tolerance that will be used in this dissertation, taking into consideration that software projects generally take longer than the implementation of a service in a telecommunication company. With respect to this point, the tolerance factor will be very useful to include.

$$r_{max} = \left(e^{\frac{W(p*t)}{t}} - 1\right) X100 \quad \ldots\ldots\ldots(2.16)$$

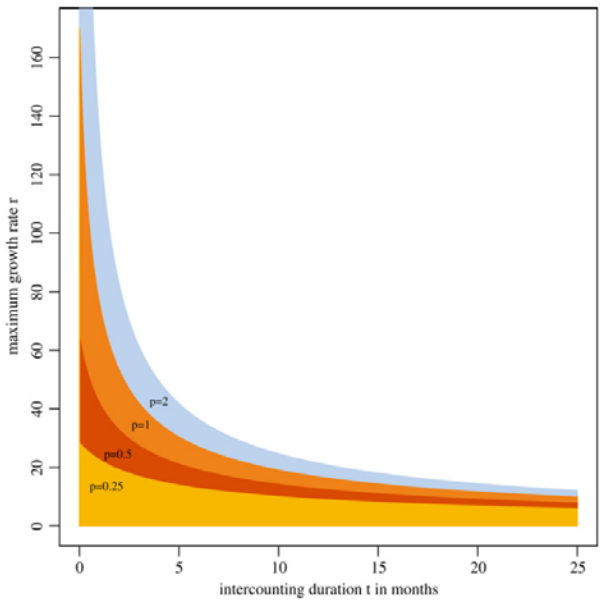


Figure 2-17: Assessment of requirement stability with the tolerance factor

## 2.7 Changeability Assessment

Regarding changeability, Ross *et al.* said that "change can be defined as the transition over time of a system to an altered state. If a system remains the same at time i and time i+1, then it has not changed" [34]. He also defined change agents, change effects and change mechanisms and described how to assess flexible, adaptable, scalable, modifiable and robust changes [34]. According to Ross, there are three elements of changeability: (1) the agent that initiates the change, (2) the mechanism of change that describes all the paths from the initial state to the end state of the change, which includes the cost or time to perform the change, and (3) the final effect of the change, which is the end state of the system. Figure 2-18 graphically shows these concepts, where the mechanism reflects only one of the many paths the system can follow to perform the change and the end state that reflects the final effect of the system [34].

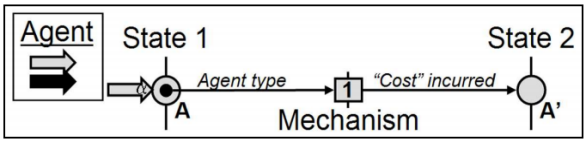


Figure 2-18: Example of changeability process

During his analysis of multi-attribute trade space exploration (MATE), which is shown graphically in Figure 2-19, Ross defined the concepts of attributes and design variables which he uses to define the concept of changeability from a different perspective. The entire process starts with a new need from the customer or stakeholder. At the top of this figure the author has the value that the stakeholder will have for this requirement. The quantifiable measure of this value is based on the attributes that ultimately will provide utility to the requirement. At the bottom of the figure, the designers of the solution translate the requirement into concept. The concept is parametrized using the design variables. There is a process of analysis between the design variables and attributes. Those design variables have an associated cost [47]. Figure 2-19 also shows the relationship between utility and cost. Figure 2-20 shows another perspective of changeability using the concept of design and attributes. The left side shows a black box, which represents a system. The black box has attributes that ultimately will be translated to value by customers and stakeholders. The decisional perception of the value is tied to a value assessment based only on the attributes. The experienced value has a different assessment because it is performed after some time of experience with the system. On the right side, Figure 2-20 presents a different box that represents different design variables and attributes. As noted in the figure, "The box is changeable if its design/attribute set or level of a given design/attribute changes". The box is a representation of the system [47].

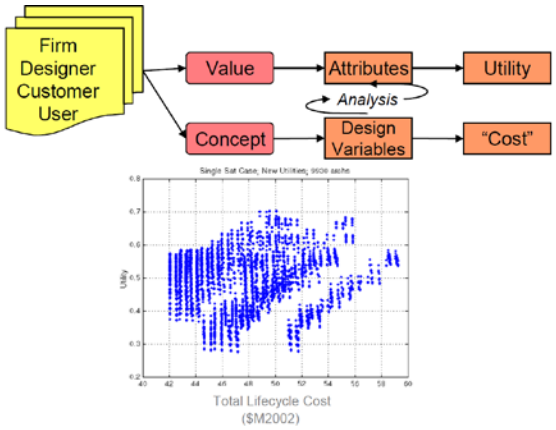


Figure 2-19: MATE process flow

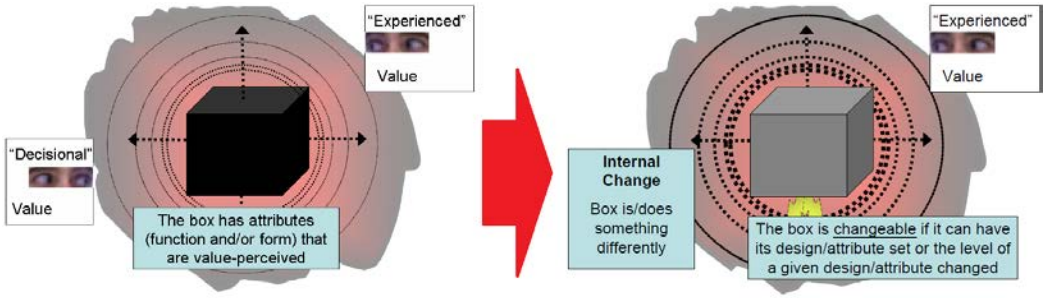


Figure 2-20: A changeable black box

The author will use the concept of changeability and the four aspects of changeability described by Fricke and Schultz (robustness, flexibility, adaptability and agility) as the most critical aspects of changeability. Figure 2-21 shows the aspects of changeability. Changeability and these four concepts will be elaborated later on this chapter. Fricke and Schultz mentioned that these four aspects "describe the degree of intelligence regarding the system ability either to be adapted, or to react to change itself and system architectures characterized by these attributes will yield great enhancements" [2]. During their description of those four aspects, they studied different fundamental research and described the TRIZ matrix shown in Figure 2-22 [64]. On the left side, the figure shows

the principles that help the four aspects accomplish the system strategy.

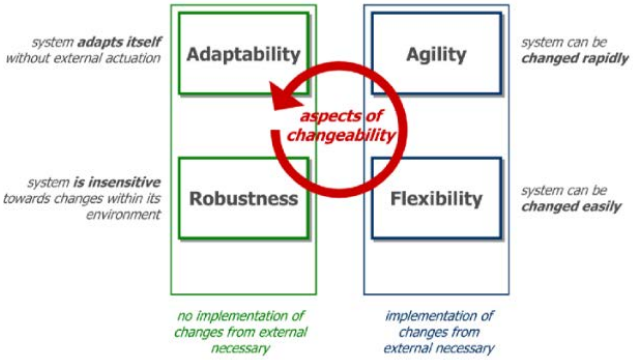


Figure 2-21: Aspects of changeability

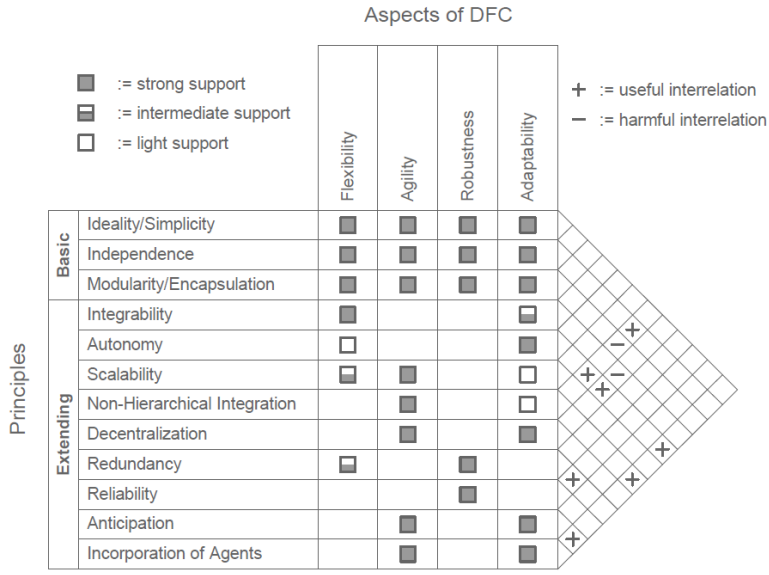


Figure 2-22: Aspect – principle – correlation matrix

The four aspects of changeability will now be explained in detail. Ross defined flexibility and adaptability very well, and Figure 2-23 shows them graphically [34]. A change is flexible if the agent that triggers it is external to the system, and a change is adaptable if it is triggered by an internal agent. For Schulz & Fricke, "flexibility represents the property of a system to be changed easily and without undesired effects", and “adaptability characterizes a systems capability to adapt itself toward changing environments to deliver its intended functionality" [64]. To assess adaptability and

flexibility, this dissertation uses the definition of Nilchiani and Hastings. They based their definition on the concept that any change in the system is associated with costs and/or benefits. Figure 2-24 shows that an increase in the benefits or an increase in the cost of the system is not always associated with an increase in utility [35]. Different customers or stakeholders can measure or perceive utility in different ways. Therefore, any flexibility measure – because of a change – must take into consideration the factors of utility, benefits and costs. Therefore, in their study flexibility was based on revenue, utility and cost concepts, and flexibility is represented as FX in the equation 2.18 [35].

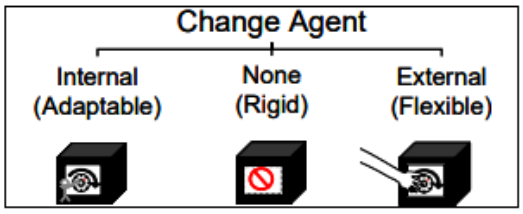


Figure 2-23: Flexibility and adaptability definitions

$$FX^{ta-tb} = \left(\frac{\sum_{i=1}^{n} R_i(p_i(t))_{flex}}{\sum_{i=1}^{n} R_{iconv}}\right) \prod_{i=n+1}^{m} \left(\frac{Ui(pi(t))_{flex}}{U_{iconv}}\right) \frac{\int_{ta}^{tb} dC_{conv}}{\int_{ta}^{tb} dC_{flex}} \quad (2.18)$$

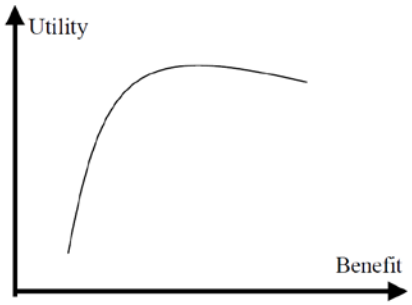


Figure 2-24: Non-linearity of perceived utility with regards to benefits.

Flexibility values over one indicate that a system is relatively flexible, and lower values indicate lower flexibility. Nilchiani and Hastings based their flexibility definition on "the discounted value of monetary benefits of function i of the system over its specified

lifetime (Ri), the system function i on analysis, the probability that a flexible change occurs to function i (pi(t)), the utility of the nonmonetary system function i for stakeholders (Ui), and the system cost (C)" [35]. Nilchiani and Hastings also proposed that flexibility is higher when the value to make the change is higher than the cost. Equation 2.19 describes this concept. This dissertation will use the same approach for the adaptability-assessed value—the only difference being, as described previously, that the external agent is the one that triggers the change [35].

$$FX^{ta-tb} = \frac{Benefit_1 - Benefit_0}{Cost_1 - Cost_0} \qquad (2.19)$$

For Schulz & Frick, "robustness characterizes systems, which are not affected by changing environments, that is robust systems deliver their intended functionality under varying operating conditions without being changed" [64]. According to Ross, "robustness is the ability to remain insensitive to change in parameters in spite of external changes", and he graphically described this definition in Figure 2-25 [34]. This concept can be very well contrasted with the changeability concept because, in general, very stable environments require high robustness, while very dynamic environments require high changeability, but not high robustness [3]. Equation 2.20 measures robustness, which is based on the direct and indirect risk of consequences due to system damage. Baker, Schubert and Faber defined "exposure as any event with the potential to cause damage to the system, damage as the reduced performance of the system components; and failure as the loss of functionality of the entire system" [36].

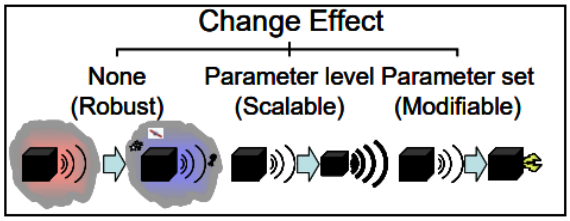


Figure 2-25: Definitions of robust, scalable and modifiable

$$RB = \frac{R_{DIR}}{R_{DIR} + R_{IND}} \qquad (2.20)$$

Here, $R_{DIR}$ and $R_{IND}$ are the direct and indirect risks of damage to the system, respectively. An indirect risk is the consequence of a direct risk.

The probability of failure and reliability will be used to calculate direct and indirect risk. Pham defined "reliability R(t) as the probability that a product or system operates properly over a specified period without failure, and unreliability F(t), which is a measure of failure, as the probability that the system fails before time t" [37] Mathematically, the concepts are described in equations 2.21 and 2.22, respectively:

$$R(t) = P(T > t), t \geq 0 \qquad (2.21)$$

$$F(t) = P(T \leq t), t \geq 0 \qquad (2.22)$$

where T is the failure time, and F(t) is the failure distribution function [37].

The Weibull distribution model is employed to describe the various types of failures that occur in telecommunications systems. In this research, the two-parameter Weibull distribution described by Pham in equation 2.23 is used "to calculate the probability that failure occurs after a certain time. In this equation, $\alpha > 0$, $\beta > 0$ and $\tau > 0$; $\alpha$ is a scale parameter, and $\beta$ is the shape parameter that determines the appearance or shape of the distribution" [37] [38].

$$F(t) = 1 - exp\left[-\left(\frac{t}{\alpha}\right)^{\beta}\right] \qquad (2.23)$$

There are several concepts of agility: for Schulz & Frick, "agility represents the property of a system to implement necessary changes rapidly" [64]. Arteta and Giatecchi defined agility “as the capability of a process to respond to unanticipated change” [46]. This is a response capability of the system for the preparedness in which the system responds via faster changes during continuous and unexpected changes over the system’s lifecycle. Ross defined agility as “the ability of a system to be both flexible and undergo change rapidly” [47]. To measure agility, this dissertation uses the concept of entropy described in equation 2.24. Entropy was initially proposed as a thermodynamic measurement of disorder [46]. Entropy is also one of the attributes of a self-organized system such as a CAS. In this dissertation, entropy represents a measure of uncertainty.

$$H(P) = K * \sum_{x \epsilon X} P(x) * logP(x) \qquad (2.24)$$

Shawky and Ali described each variable as follows: “P is the probability of a random variable X with a value x, and H(P) is a quantitative measure of the amount of uncertainty associated with X. H(P) is zero only if X is known with certainty (e.g., the probability of one outcome is 1, and the probability of all other outcomes is 0)” [20]. This dissertation measures agility based on the rate of entropy change over a period. During that period, a request for change can be made [44]. As described by Shawky and Ali, “higher agility corresponds to a higher rate of change of entropy from time t0 to time t1” [39]. Therefore, equation 2.25 is obtained:

$$AM(t) = -\sum_{i=1}^{n} P'_i(t) * (log_2 P_i(t) + 1.44) \quad \text{..... (2.25)}$$

where AM is the measure of agility at time t.

## 2.8 Fitness

Finally, this chapter describes fitness. There are different studies about fitness. Fitness landscape is a concept that is highly related to CASs, but it was used in the field of

biological evolutionary optimization and was very powerful in evolutionary theory. The concept helps predict and estimate system performance, especially for classifier systems. By considering all candidate solutions as a landscape, the algorithm will learn and determine the solution with the highest value among all possible solutions; the highest value in the search space represents the fitness of the solution [40]. As described in the previous section, Kauffman was a pioneer in describing CAS concepts and his main focus was the study of interdependence among the components of a CAS. Kaufman described that any organism has N components and that each component has binary values. Figure 2-26 shows two states of the same system. This system has three components (N=3). Kauffman defines the variable of dependency K which describes how many connections each component has [65]. Figure 2-26 shows examples where K=2 which represents the maximum number of connections for N=3. In the initial state each component fitness is calculated based on the average of each of its binary components. In Figure 2-26, the 0 and first component has a weight of 0.8. Then, the average for component 000 is calculated as the average of 0.8, 0.6 and 0.2. Finally, the system transitions to the end state, represented by the bottom of Figure 2-26. According to Kauffman, an individual binary component contributes to not only the fitness but also the values of the other two components (K=2) that contribute to the component.

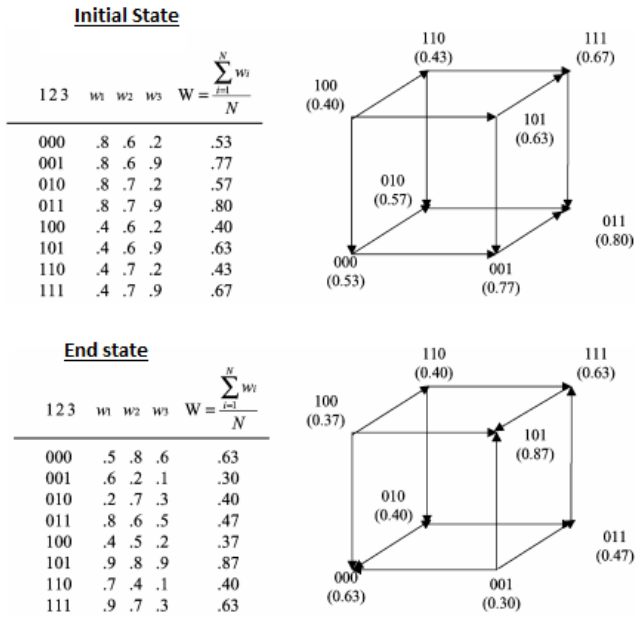


Figure 2-26: Landscape with maximal interdependence

Choi et al, provides a good description of how fitness works when looking for the optimal value of the system "Imagine a mountain climber situated in the middle of a mountain range, whose goal was to climb the highest peak in the range. If this person happened to be standing on the side of the highest mountain in the range and were cognizant of it, then climbing to the top would be a fairly straightforward task. If, however, the landscape consists of local hills and valleys, then the climber is faced with the difficult challenge of finding the top of the highest mountain." He continued: "When system components can be optimized individually without regard for one another the landscape is simple, but when individual components contribute in different ways, depending on the value/state of other components in a contingent or tightly coupled manner, the optimal state becomes difficult to find, as many local optima exist" [61]. Figure 2-27 shows graphically how fitness works. Fitness will help the CAS system to evolve with the target to move from population A to population B.

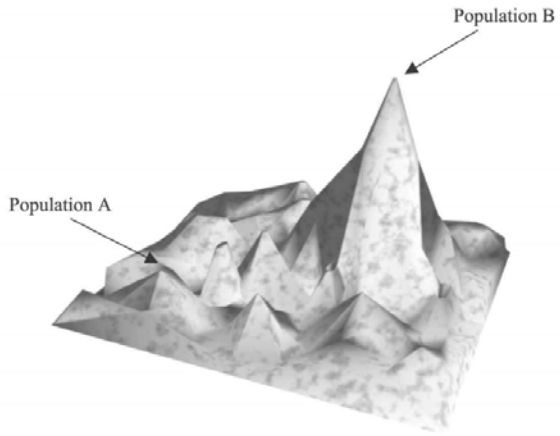


Figure 2-27: Representation of evolution in three dimensions

Choi also gave a good example of fitness that is very relevant to the concept of fitness that will be applied in this dissertation: “a purchasing manager tries to increase fitness by ensuring the acceptable cost, delivery, and quality of parts and raw materials purchased by engaging the supplier firms in a series of talks or training sessions” [48]. In this dissertation, the author will multiply price by active time and subtract cost to measure fitness. The system will require services to be active for as long as possible at the minimum cost; therefore, a high amount of active time, high revenue and low cost represent high fitness in CAS. This statement is supported by Li et al. They mentioned that fitness is evidenced by the profit or loss of the company. If the company performs well, then the profit and its fitness are higher. By contrast, when the company cannot, for example, fulfill demand, profit and fitness are reduced. Finally, they mentioned that the environment has a very strong influence on fitness [62].

# Chapter 3 – Model and Methods

This chapter will explain the model of telecommunications using XCSF. Additionally, the algorithm will be explained in detail. This description includes information on classes and their relationships. Finally, the hypothesis and statistical concepts used will be stated. Notably, the process of simulation used was based on the study of [44].

## 3.1 Model

Figure 3-1 graphically shows the model used in this dissertation. This model is based on all the concepts described in Chapter 2. On the left side, there is the baseline of information from the telecommunications system that will be inserted in the model. This model will translate this telecommunication entity information and the information to be taken by the classifier system. This information will include the parameters described in Chapter 2: changeability, fitness and requirement. Once the classifier system has this information, it starts to learn using the XCSF algorithm, which will simulate the CAS. At the end of each process, the system provides feedback to the environment. In each step of this process of learning and feedback, the system measures all of the parameters in the analysis.

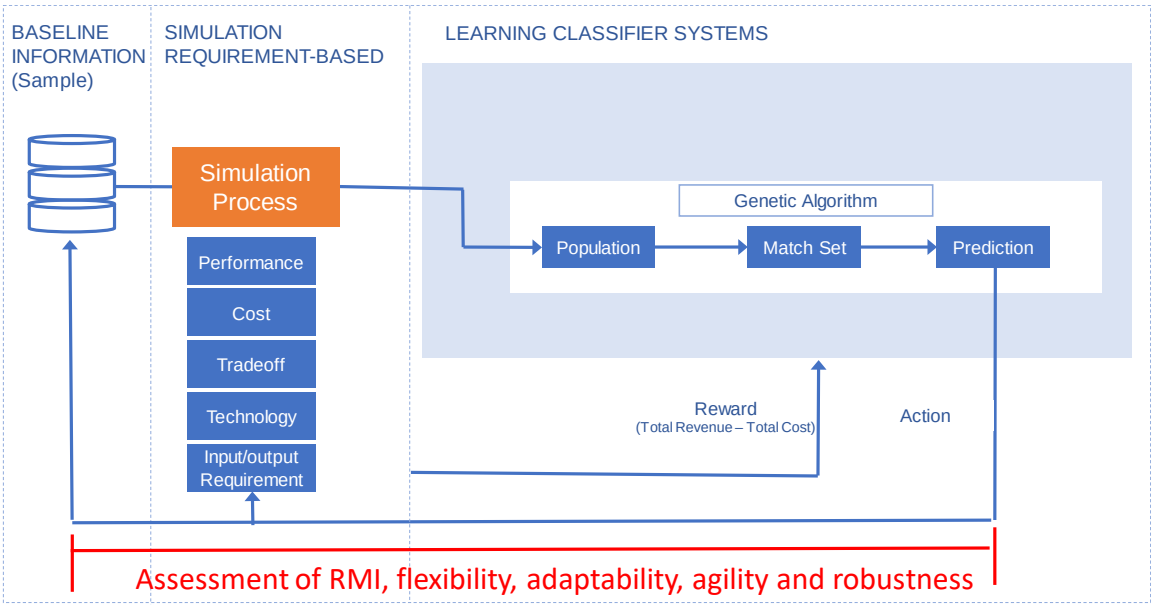


Figure 3-1: Proposed model to analyze the parameters in this dissertation

Figure 3-2 shows the relationship between the classifier systems and changeability parameters described in the previous chapter. The left side presents the changeability parameters discussed in the previous section. As Ross mentioned, the goal of any system is to balance the fulfillment of a need and keep the resources within a certain context. On the left side the author has the concepts of design variables and attributes described in the previous section [34]. On the right side are the classifiers discussed in the previous section. It is also important to note that as part of the genetic algorithm concept of classifiers, genotype and phenotype are included in this figure. Both concepts are frequently used in biology, in which the genotype is the internal structure of a system and the phenotype is the result of that structure. Marjan described the relationships between condition and action and between genotype and phenotype [69]. The population of classifiers, which is composed of conditions and actions, are called chromosomes by Marjan. The genotype defines the structure of a chromosome (i.e., the appearance of a chromosome or its gene structure) [69].

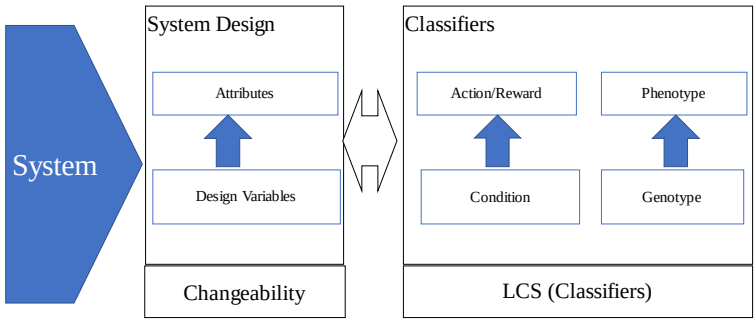


Figure 3-2: Relationship between changeability and classifier systems

## 3.2 Simulation Process

Figure 3-2 describes the method this study will use to assess the impact of the requirements for the changing telecommunications system. Baseline information is used to simulate a telecommunications system. This baseline uses the five parameters described by Wymore and other parameters during the XCSF simulation process. This information will be the input for the classifier system, and a reward will be obtained. Finally, feedback from the environment will be used to adjust the data. RMI and system changeability will be continually calculated.

Figure 3-3 shows how the model will work in a telecommunications environment. The initial requirements have three nodes, specific service information (i.e., service type and bandwidth) and specific technical information (i.e., location, equipment, port and card). After a system based on the initial requirements is implemented for a specific time, a new requirement is generated. Figure 3-3 shows the new requirement, which has four nodes, new service and new technical information. This research will measure the variables described by Wymore and analyze each of the variables, as well as additional variables that have a greater impact on the transition, to generate the new requirement, which transitions the system from the initial state to the final state. This dissertation will also

analyze the relationship between changeability and fitness.

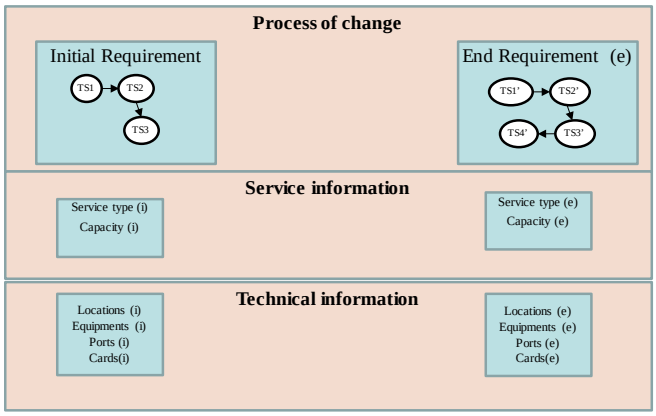


Figure 3-3: Telecommunication change.

Figure 3-4 shows the variables considered for measuring the baseline of the change. During that timeframe, an assessment of the initial and final requirements of the components defined by Wymoore will be performed [31]. Figure 3-4 shows the values to be calculated during the transition from initial requirements to final requirements. The monthly revenue and the total time the customer was with the company will be important metrics for assessing whether the requirement led to a positive or negative result. Subtracting these values from the cost will provide the system reward.

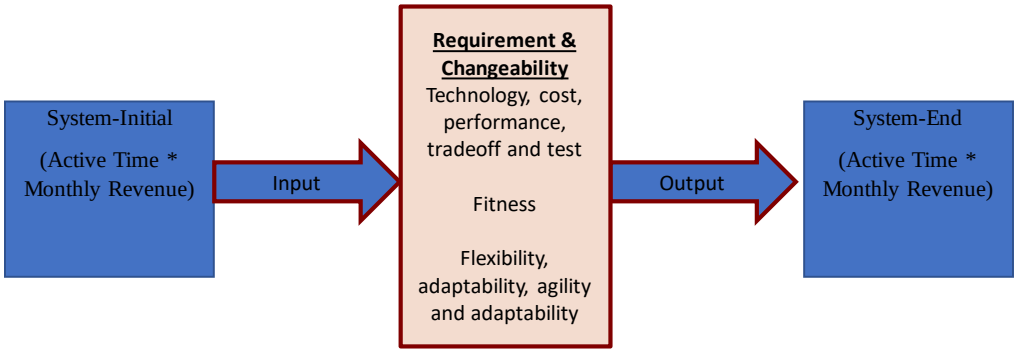


Figure 3-4: Assessment process.

In this study, a TelCo service (TS) is defined as a telecommunication component offered to connect its customers through the TelCo network to satisfy customer needs. TS

design attributes are defined as quantitative parameters that define TS. These attributes include (1) service type, (2) service capability, which is commonly measured by the bandwidth, (3) the device or equipment used to connect the service to the TelCo network, (4) service implementation time, (5) service activation date, (6) service deactivation date, (7) service implementation cost, (8) TelCo network incidents and (9) the revenue generated from the service. New TS requirements are the main drivers of the analysis conducted in this research. Figure 3-5 shows an example of three TSs for three customers connected to the TelCo network.

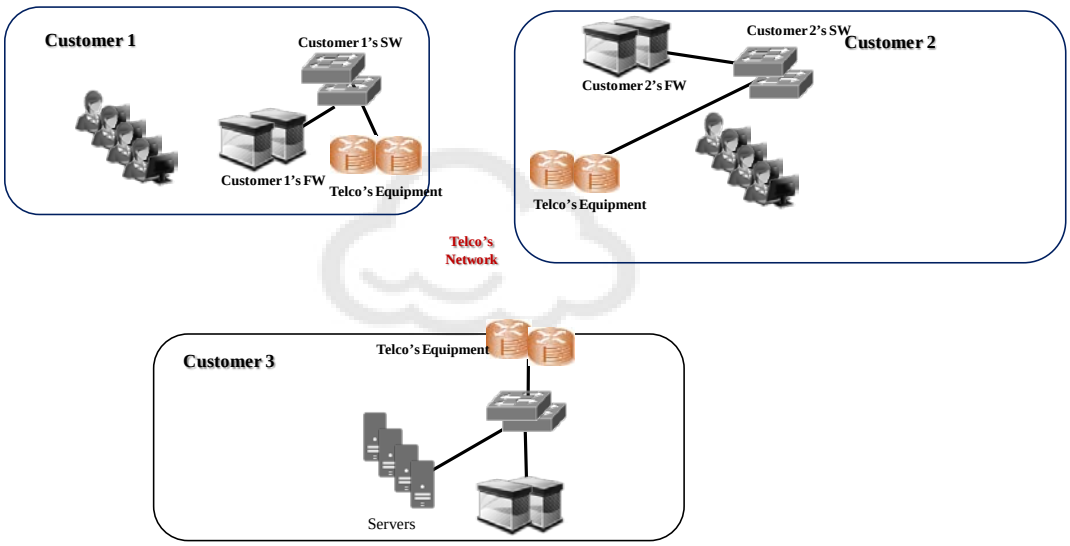


Figure 3-5: Telecommunication service model.

To analyze the factors of requirement stability in this research, a telecommunications model (TelCo) was created to consider various factors. Figure 3-6 shows the steps followed to create this definition.

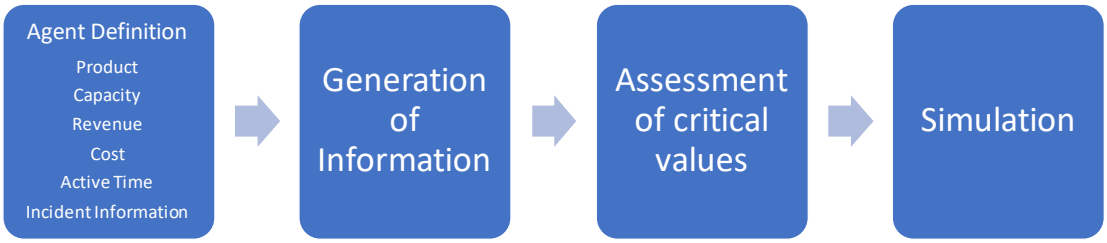


Figure 3-6: Simulation process.

All information generated in this model was based on TelCo sample data. The rules of the XCSF model are as follows:

1. Step 1: For the XCSF model to function, a baseline of information is needed to generate the simulation and learn from the simulation to improve the reward. All basic information used in this simulation is based on sample data from a functioning TelCo system. For example, the product information and attributes are based on TelCo product catalog sample data. Product selection is based on the probability that each product had been selected by the customer in the past. Table 3-1 shows a small sample of this information. This information is transferred to the simulation program architecture which is explained in the next section. Product cost information is based on historical information and is strongly dependent on the selected product.

Table 3-1: Product information sample

| Product Type | Bandwidth (Kbps) | Service | MRC | OTC |
|---|---|---|---|---|
| Local IP Data | 158720 | Additional Ports 155 MBps | $ 350.00 | $ 612.00 |
| Local Private Lines | 1536 | 1.5 Mbps | $ 1,760.00 | $ 2,200.00 |
| Domestic IP Data | 64 | Restricted service 64Kbps (8 Horas) | $ 2,151.25 | $ - |
| Internet Dedicated Access | 1024 | Internet Premium Upgrade 1024 Kbps | $ 3,670.00 | $ 612.00 |
| Domestic IP Data | 1024 | Satellite Connection 1024 Kbps | $ 19,362.84 | $ 15,714.27 |

2. Step 2: Based on the information described in step 1, distinct actions are executed to generate and translate the information to XCSF:

   a. MATLAB was used to generate random data. The incident information was selected randomly from past TelCo network experiences. The Kano model was used to generate the active time. According to this model, a direct relationship exists between performance and incidents. The number of incidents has an indirect relationship with performance, i.e., a large number of incidents negatively affects performance. Active time is a measure of customer satisfaction. Real data effectively show that this relationship exists; equation 3.1, which was derived by regressing the historical data of the active time of a service and the incidents generated during that active time, is used to describe that relationship. The p-value for this regression is 0.046.

$$Active\ Srv\ Time = 847 - 17.23 * AverageIncidentTime \quad (3.1)$$

   Figure 3-7 shows a distribution of incident time based on a sample of data. During the simulation process, this study assumes that, if the total number of incidents in a system is higher than 80, then the customer decides to disconnect the service, which corresponds to a penalty that negatively affects the reward and is taken as feedback by the system to improve future performance.

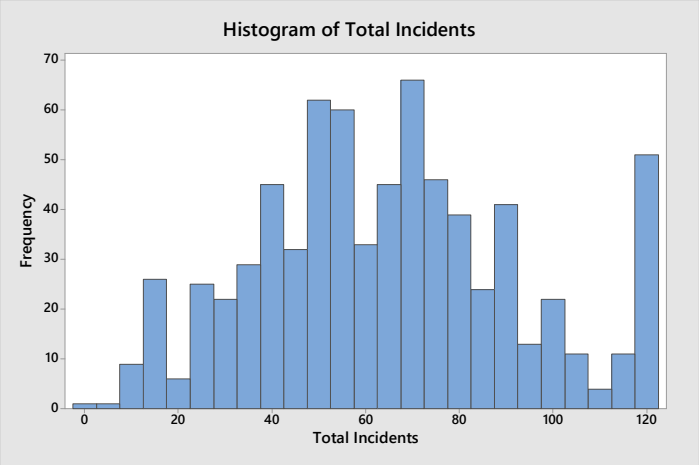


Figure 3-7: Histogram of total incidents.

b. Once the information is obtained from steps one and two, there is a translation of the information from the original values to the classifier values. Table 3-2 shows the original data are extracted from steps 1 and 2. Table 3-3 shows the same parameters, but with the values to be used in the XCSF model. As described previously, the data must contain values from zero to one.

TABLE 3-2: Original telecommunication data sample

| Service ID | Type of Service | Service | Bandwidth (Kbps) | Equipment | Incidents | Start Service Date | End Service Date | Total Revenue (USD) |
|---|---|---|---|---|---|---|---|---|
| 10122 | Internet Dedicated Access | Internet Premium 512 Kbps | 512 | | 0 | 12-JUL-01 | 10-JUN-02 | $3,733.56 |
| 10122 | Internet Dedicated Access | Internet Premium 1 Mbps | 1024 | CISCO 2611XM, SW: IP, DRAM: 128 MB, Flash: 32 MB, FE: 2 | 2 | 10-JUN-02 | 08-SEP-04 | $14,272.13 |
| 10122 | Internet Dedicated Access | Network Access 2 Mbps | 2048 | CISCO 2611XM, SW: IP, DRAM: 128 MB, Flash: 32 MB, FE: 2 | 2 | 08-SEP-04 | 20-APR-07 | $33,198.76 |
| 10122 | Internet Dedicated Access | Internet Carrier Class 4 Mbps | 4096 | CISCO 1841, IOS IP BASE, DRAM: 128 MB, Flash: 32 MB, FE: 2 | 1 | 20-APR-07 | 26-APR-09 | $44,226.85 |

TABLE 3-3: DATA TRANSLATED TO CLASSIFIER SYSTEM

| Product and Service Configuration | | | | | | | |
|---|---|---|---|---|---|---|---|
| **BIT_1** | **BIT_2** | **BIT_3** | **BIT_4** | **BIT_5** | **BIT_6** | **Capacity (Bandwidth)** | **Incidents** |
| 0 | 0 | 0 | 1 | 0 | 1 | 0.0025 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0.005 | 0.00023291 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0.01 | 0.00023291 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0.02 | 0.000116455 |

3. Step 3: Various values are assessed. Below is an explanation of how the values that impact the new requirements are assessed as well as the changeability parameters.

   a. The five Wymore parameters are assessed. Input and output will be represented by the initial and end states of the TS described at the beginning of this section. The input state is the state of the system before the change, and the end state is the state of the system after the change. Table 3-2 and Table 3-3 show an example of the input and output of the system. In Table 3-2, the first input is the initial service that was implemented on 01-July 2001; the output for this first change is the second row of Table 3-2, which was implemented on 10-June-2002.

      The performance has two components: the time taken by the TelCo system to respond to a customer and the time the service has been down. In Table 3-4, this is represented by the columns “Time to Activate”, which measures the total time to activate the service in days, “Direct Incidents”, which measures the average time the system was down in hours using a direct factor, and “Indirect Incidents”, which measures the average time the system was down in hours using an indirect factor. The technology is represented by the capacity of the system. A higher network capacity indicates a more advanced technology,

which is represented by the column “Capacity”, which measures the bandwidth of the circuit. Finally, the tradeoff is the average of the cost and the performance variables. For these variables, it is important to assess the magnitude of influence they have on the stability of the system. Table 3-4, which is related to Table 3-5, shows the calculation of RMI, which measures the number of new services and changes performed during the period of active service.

To assess the $r_{max}$ value described in Chapter 2, using equation 2.12, the author follows two steps. First, the average implementation time for the sample data is calculated. Table 3-4 shows a sample of information and the column Time to Activate was used as the parameter of time of implementation. To proceed with the calculation, the author uses the MATLAB function lambertw. Figure 3-8 shows the process for assessing this value in MATLAB. The author considers that the average time to implement a service is four months. In addition, the author uses the “p” value which is the tolerance factor described in Chapter 2. This factor is used to adjust for the difference between software and telecommunication projects. Software projects generally take longer, and therefore to adjust for this difference, the author considers 0.75 as a tolerance factor. According to Figure 3-8, the author will use 30% as the stability requirement value during the analysis in the next chapter. Different levels of requirement stability have been defined, but generally, values higher than 30% indicate means that there is not requirement stability. To reinforce the assumption described in this dissertation, factors that introduce requirement instability have a higher impact on the generation of requirements.

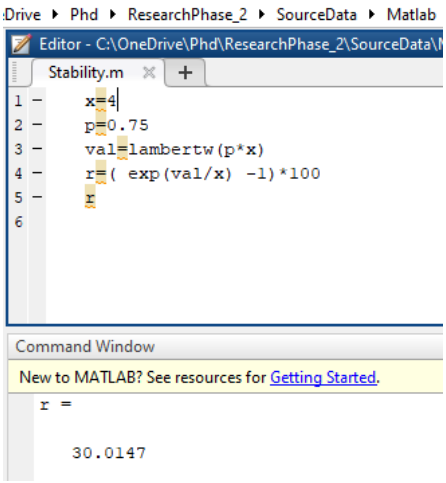


Figure 3-8: MATLAB execution of Lambert's function

b. The other values assessed are the changeability parameters. This dissertation uses the concept of Nilchiani and Hastings, as explained in Chapter 2, and the process of TS change to satisfy the new needs of customers. To assess flexibility, equation (2.11) is used. This equation has two parameters: "(1) the probabilistic revenue per cost over the flexible system over a specific time interval and (2) the probabilistic revenue per cost of the same system over the same period after a change has been performed" [44]. Nilchiani and Hastings mentioned that "if a system is studied in retrospect, as it is in this study, then the probabilities of specific events and changes implemented on a system become one" [35]. This definition is used to assign a probability of one. Table 3-6 is an example of how a system changes in a specific service. Service 10060 changes over time. The cost to implement the change and the revenue generated per service is shown in the "Total Cost" and "Total Revenue" columns, respectively. The flexibility of the first change is calculated as (($3,732.66 + 14271.13)/$3,732.66) × ((4,377.56 + 4,370.23)/4,377.56).

Table 3-4: Sample input data for the requirement

| Service | Revenue | Cost | Capacity (BW) | Time to Activate (Days) | Direct Incidents (hours) | Indirect Incident (hours) |
|---|---|---|---|---|---|---|
| International Private Lines | $ 210.00 | $3,014.56 | 512 | 26 | 30.41 | 48.63 |
| Red Privada Virtual Nacional | $ 560.00 | $ 180.00 | 512 | 66 | 21.47 | 10.59 |
| International IP Data | $1,015.00 | $1,104.38 | 768 | 119 | 38.48 | 23.43 |
| Domestic IP Data | $1,015.00 | $ 408.90 | 256 | 119 | 38.48 | 23.43 |
| Acceso Dedicado a Internet | $ 105.00 | $ 451.50 | 6144 | 15 | 12.02 | 76.6 |
| Local IP Data | $ 280.00 | $3,993.46 | 768 | 32 | 20.35 | 4.57 |
| Local IP Data | $ 453.87 | $ 619.85 | 64 | 52 | 20.88 | 19.64 |
| Acceso Dedicado a Internet | $ 256.61 | $2,561.10 | 512 | 28 | 20.27 | 34.49 |

Table 3-5: RMI assessment

| Service | Total New | Total Change | RMI |
|---|---|---|---|
| International Private Lines | 17 | 2 | 88.2353 |
| Red Privada Virtual Nacional | 18 | 2 | 88.8889 |
| International IP Data | 20 | 1 | 95.0000 |
| Domestic IP Data | 23 | 1 | 95.6522 |
| Acceso Dedicado a Internet | 22 | 1 | 95.4545 |
| Local IP Data | 30 | 1 | 93.3333 |
| Local IP Data | 75 | 3 | 96.0000 |
| Acceso Dedicado a Internet | 78 | 3 | 96.1538 |

Table 3-6: Example of the calculation of flexibility.

| Service ID | Flexibility | Total Revenue Generated (USD) | Total Cost | Type of Service | Service | Bandwidth (Kbps) |
|---|---|---|---|---|---|---|
| 10122 | 6.6395 | $3,733.56 | $ 4,377.56 | Internet Dedicated Access | Internet Premium 512 Kbps | 512 |
| 10122 | 2.4134 | $14,272.13 | $ 4,370.23 | Internet Dedicated Access | Internet Premium 1 Mbps | 1024 |
| 10122 | 0.6296 | $33,198.76 | $ 7,966.28 | Internet Dedicated Access | Network Access 2 Mbps | 2048 |

To assess adaptability, the definition of Nilchiani and Hastings described in Chapter 2 is again used. From the previous study, the author will define that “adaptability is the extent to which the TelCo internally changes its current infrastructure to cover new

customer requirements" [44]. The author will use equation 2.11 to assess the adaptability of the system; "the equation is the ratio of two parameters: (1) the probabilistic nonmonetary utility to cost over the adaptable system over a specific time interval and (2) the probabilistic utility to cost of another adaptable system over the same period of time" [44]. Nilchiani and Hastings also "provided examples of nonmonetary utility, such as increased resolution of images or higher processing capacity" [35]. For this dissertation, the number of services installed during a period is used as a measure of utility [44]. Additionally, the following concept is used: "if a system is studied in retrospect, as is the case here, the probabilities of specific events and changes on the system becomes one" [35]. For adaptability, the author analyzes how the telecommunication network has adapted over time. Therefore, cost is used to adapt the TelCo network, which is called "TelCo Infrastructure Projects". The author uses the number of services that have been installed during that timeframe. Table 3-7 shows an example of an investment by TelCo to adapt to a new need of the customers. This approach is an internal initiative by the company; therefore, the initiative is an adaptable task to satisfy new customer needs.

Table 3-7: Example of the calculation of adaptability.

| Project ID | Total Cost | Number of Services | Adaptability |
|---|---|---|---|
| 23797 | $2,390,536.64 | 184476 | 1.875279 |
| 22366 | $230,533.44 | 184716 | 0.601044 |
| 30739 | $32,132,492.78 | 183560 | 0.000249 |

In this dissertation, robustness is measured based on the work by Baker et al. [36], as described in Chapter 2, and assessed by dividing the direct risk by the direct plus indirect risk [36]. A scientific definition of risk was provided by the Royal Society [55]: "the probability that an adverse event occurs during a stated period of time, or

results from a particular challenge. As a probability in the sense of statistical theory, risk obeys all the formal laws of combining probabilities". Other studies, such as those of Mitchell [56] and Gillet [57], have developed a scientific perspective of risk: Mitchell [56] defined risk as "the probability of loss and the significance of that loss to the organization or individual", while Mitchell defined an equation to assess the probability of loss 'P' (lossn) and the significance of loss 'l' (lossn) for a specific event n [56]. Based on this, Harland et al. [58] derived equation 3.1. The author defines the events on any TS as any incident in that service. AXELOS defines an incident as "an unplanned interruption of a service or a reduction in the quality of service" [59]. The failure of a configuration item that has not yet affected service is also considered an incident. Table 3-8 shows the sequences of events over two specific incidents, i.e., 187155 and 193538, for the TSs. Appendix A shows a list of the main incidents used in this dissertation. The first incident shows that "bandwidth saturation" is the main problem (associated with direct risk) and that "operating system problem" and "network down" are the consequences (associated with indirect risks). The same condition applies for the second incident, where "media converter problem" is the first and main problem (associated with direct risk) and "network down" and "customer router problem" are the consequences (associated with indirect risks). Table 3-9 shows the incident sequence. Risk is calculated using equation 2.13. This equation is based on two concepts: (1) loss, which is the loss at the time service was impacted, and (2) probability of loss, which is the probability that the incident occurs. The author uses equation (2.16) to describes the density function of the probability of the risk. The MATLAB function Wblfit is used to assess the probability of failure initiated by the incidents described in Table 3-8. MATLAB defines Wblfit(data) as a "function that

returns the maximum likelihood estimates of the parameters of the Weibull distribution given the values in the vector data, which must be positive" [60]. The result is the calculation of parameters α and β described in equation (1.5). Equation 3.2 assigns the input vector based on the data of Table 3-9. Equation 3.3 shows the calculation of "parmhat", which contains the values of α and β.

$$data = [Input\ vector\ data] \quad (3.2)$$

$$[parmhat, parmci] = wvlfit(data) \quad (3.3)$$

Table 3-8: Incidence sequence

| Incident ID | Main Problems Detected | Problems in Network | Sequence Date |
|---|---|---|---|
| 187155 | Bandwidth Saturation | Network Down | 03-MAY-06 |
| 187155 | Operative System Problem | Network Down | 04-MAY-06 |
| 193538 | Media Converter Broken | Network Down | 19-MAY-06 |
| 193538 | Customer Router Problem | Network Down | 19-MAY-06 |

Table 3-9: Interval of time of a "problem with the network fiber"

| Sequence Date | Difference of Days |
|---|---|
| 27-NOV-06 02.07 PM | 77 |
| 02-DEC-06 08.56 AM | 102 |
| 02-DEC-06 09.22 AM | 102 |
| 10-APR-07 06.45 PM | 251 |
| 28-MAY-07 05.46 PM | 643 |
| 02-JUL-07 05.02 PM | 266 |
| 02-JUL-07 05.07 PM | 634 |

Equation (2.18) is used to assess telecommunications service agility, and the assessment of agility is based on Shawky and Ali [20] [44]. As explained in Chapter 2, agility is the ratio of the entropy increase in the telecommunications system [44]. The following steps will be performed to assess agility:

- First, the entities of the telecommunication system are defined; each TS is used as an entity.
- Second, a representative sample of data from the history of the data is used to represent how each entity has changed over an interval of time. Because changes do not often happen on a daily basis, a week is used as the unit of time to measure change [44]. Pi(tj) denotes the probability of changes to service i in week tj, where i takes values from 1 to n (where n represents the number of services analyzed), and j takes values from 1 to m (where m represents the number of weeks analyzed). As noted, a week is used as the unit of measurement. The author analyzes each change to the services in the TelCo and calculate Pi(tj) as the change at time tj. If all TSs are modified P times per week, then Pi(tj) is calculated as P/n [44].
- Third, equation 2.17 is used to calculate the entropy of the TelCo network, which represents the level of uncertainty of the TelCo network each week. Then, the entropy value and equation 2.18 are used to calculate agility [39].

4. Fourth, the process follows the XCSF simulation process described in Section 2.5. The XCSF learning process adopts the appropriate reward value for the system.

These steps are detailed in Figures 2-11 and 2-12. In summary, the information is obtained in a classifier format, and these values generate a reward value. The system learns from this experience and applies the best actions for future simulation processes. Additionally, the GA is applied to generate new values for the population to improve the reward values.

Figure 3-9 shows another perspective of the algorithm used in this dissertation but aligned to the XCSF process explained previously. It is important to note that step 3 assesses the required information in this dissertation.

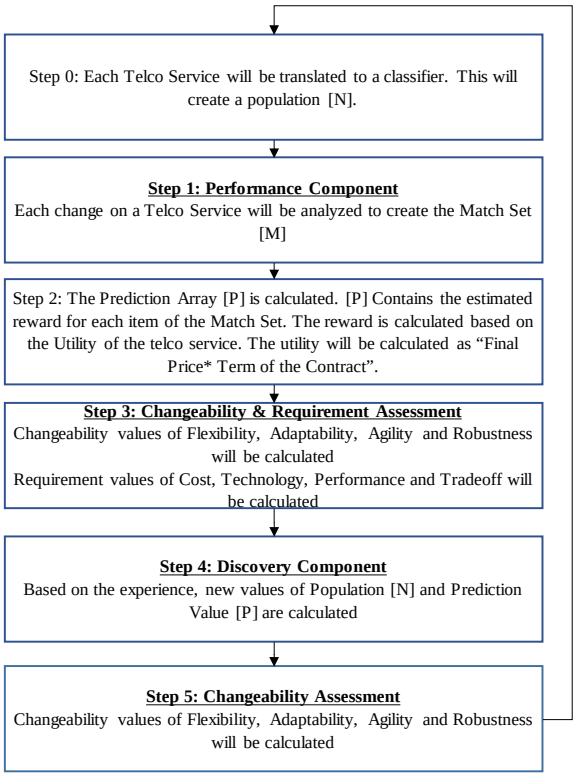


Figure 3-9: General algorithm aligned to XCSF

## 3.3 Simulation Architecture (Based On Java XCSF)

There are several approaches to simulate using programming the model previously described. Below are some of the frameworks that have been developed using XCS:

i. An XCS library based on C++ was proposed by Pier Luca Lanzi [51].

ii. An accuracy-based LCS was developed in Python by Ryan J. Urbanowicz [52]

iii. Martin Butz developed an algorithm based on Java [50]

In this dissertation, Java XCSF, as proposed by Butz, is used [50]. This model has the same basic concept of XCSF and is illustrated in Figure 3-10. To begin learning, the system obtains sample information from the telecommunication system, matches the information according to the intervals that the system initially generates, predicts a new

reward value, updates the weight vector predictors, evolves new conditions using the GA and informs the listeners of the results [53].

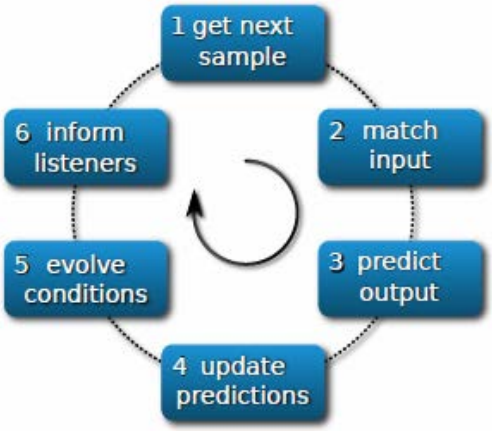


Figure 3-10: Java XCSF flow

Figure 3-11 shows the structure of the classes developed in this dissertation. This simulation program was a modification of the original JavaXCSF. Figure 3-12 shows the Eclipse program in which the program was developed [54]. Below are explanations of the actions and functions of each class during the simulation. More detailed explanations for several the methods of these classes are provided in appendix B.

- The TelcoConnection class is connected to the database that initially takes the sample of information. Then, the system learns from this baseline, and the database is updated according to this learning. This class contains functions for extracting, deleting and updating the information to be used during the simulation process.
- TelcoService is the representation of any telecommunication service as described in Chapter 3. Attributes of bandwidth, cost, revenue, etc. are defined in this class. Each of these attributes are mapped to the information extracted using TelcoConnection. This information is sent to the rest of the classes to perform the

simulation. This class also calculates some of the values that do not come from the database, such as, the active service described previously.

- The XCSF class is connected to the TelcoService class. The primary function of this class is to coordinate the learning process of the system. Once the information is obtained, this class acts as the main executive by connecting to the rest of the classes and performing some of the statistics needed during execution. This class also calculates system agility and adaptability. The agility calculation is performed by obtaining the execution time and the initial number of services that are performed during the simulation, which are used to calculate agility. The adaptability is calculated by controlling the information of new services and the cost to implement those new services.
- The XCSSet class has the main objective of creating sets of information that are used by the initial population and the match set. Per the definition of XCSF, both sets of information have the same structure of information. The population receives directly the simulated or real information. The match set has more functions and processes. This class has methods for matching the information from the population, covering when the population does not cover the learning, and controlling to update the information.

  The main objective of all these classes is to improve the performance, primarily with a focus on the fitness of the system.
- The Classifier class controls the process for updating, matching and updating the classifier information. This class also controls the conditions and predictions of the classifiers. Different methods of conditions and predictions are available.

- The PerformanceEvaluator class assesses the performance of the system. XCSF controls not only the fitness but also other performance parameters such as predicted error, error, fitness and numerosity. This class is also used to assess the records of all changeability parameters (agility, flexibility, adaptability and robustness) and information related to new or changes in requirements (RMI, cost, performance, tradeoff and technology). Because this class records all of this information, the author uses the output of this class to perform the statistical analysis.
- The condition class, as was explained on the XCSF concepts, determines the classifier match.
- The prediction class assess new predictions of the system given the input information. One of the main functions of this class is to map the input to predicted output values. According to the definition of XCSF, the prediction can be updated based on feedback from the system.

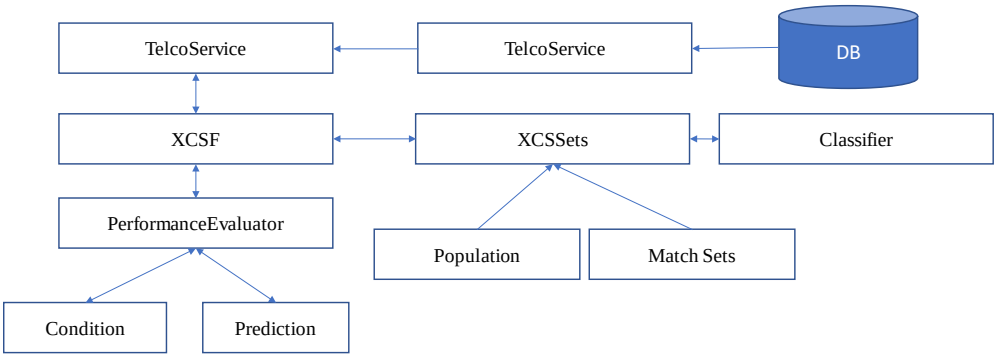


Figure 3-11: Structure of java classes developed based on JavaXCSF

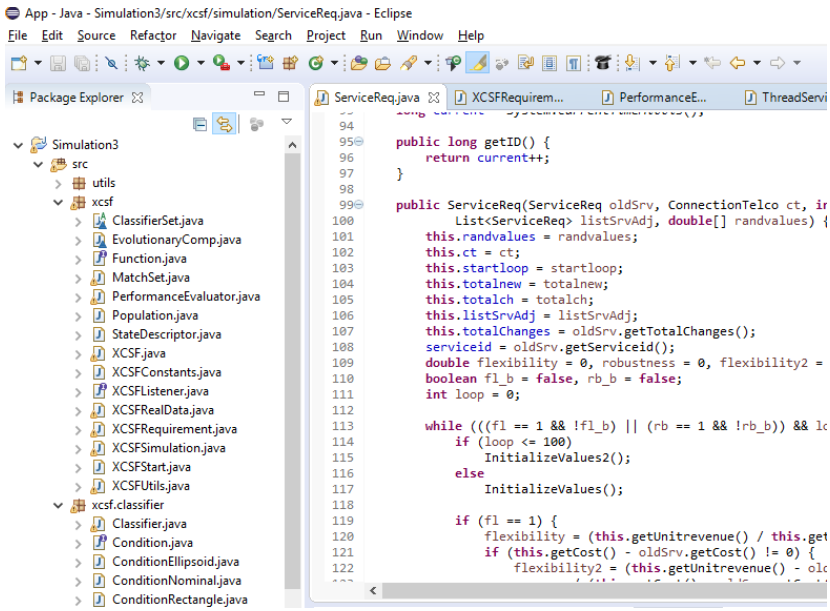


Figure 3-12: Print screen of the JavaXCSF in Eclipse

## 3.4 Hypothesis

The first step of examining the hypothesis is demonstrating that the model is performed using the appropriate telecommunications dataset previously described.

With respect to requirement impact, stakeholders and customers seek to understand the extent to which certain factors influence the generation of new requirements in a system and the effect of these factors on requirement stability and volatility. Due to the dependency on the various variables described by Moore, regression analysis and surface and contour plots will be analyzed in this study to address the problem statement. Assuming equation 3.4, the null hypothesis is described by equation 3.5 and the alternative hypothesis is described by equation 3.6. One objective of this dissertation is to analyze the impact on RMI. The next section explains the statistical tools used to prove this hypothesis.

$$ImpactNR = \beta_0 + \beta_1 * Performance + \beta_2 * Technology + \beta_3 * Tradeoff + \beta_4 * Cost \quad (3.4)$$

$$H_{10}: \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0 \quad (3.5)$$

$H_{1a}: \beta_i \neq 0$; where i:1,2,3,4 (3.6)

Regarding the relationship between fitness and changeability, this dissertation uses the approach described in [44]. The main objective of this research is examining how changeability affects CASs by studying the impact of agility, flexibility, adaptability and robustness on CASs.

Formulas (3.7) – (3.14) show the null and alternative hypotheses for flexibility, adaptability, robustness and agility respectively. These hypotheses are based on the research questions described on Chapter I. For flexibility:

$H_{20}: Fitness \neq F(Flexibility)$ (3.7)

$H_{2a}: Fitness = F(Flexibility)$ (3.8)

For adaptability:

$H_{30}: Fitness \neq F(Adaptability)$ (3.9)

$H_{3a}: Fitness = F(Adaptability)$ (3.10)

For robustness:

$H_{40}: Fitness \neq F(Robustness)$ (3.11)

$H_{4a}: Fitness = F(Robustness)$ (3.12)

For agility:

$H_{50}: Fitness \neq F(Agility)$ (3.13)

$H_{5a}: Fitness = F(Agility)$ (3.14)

It is important to note that for each hypothesis, each variable of changeability will be analyzed separately, which means, for example, that the relationship between flexibility and fitness will be analyzed separately from the relationship between fitness and agility.

## 3.5 Statistical Analysis Tools

This dissertation simulates and uses the following concepts from statistics to test the previously described hypotheses [77]:

- Scatterplots are used to visualize the tendencies of different variables during the analysis. For example, the tendencies of fitness and error are primarily visualized using scatterplots. Once the author confirms some tendencies on the variables, the relationships are analyzed using regression and related concepts.
- Normal probability and residual plot of variance are used to verify the normality assumptions. This concept is used in this dissertation for two purposes. The first is for ANOVA analysis to analyze the strength of the variables. Second, normality analysis is used to analyze nonlinear relationships, such as the relationship between fitness and agility, which will be explained in the next chapter.
    - A normal probability plot shows the relationship between the residuals and the probability. The tendency must follow a straight line to validate the assumption of normality.
    - A residual plot of variance is used to demonstrate that the distribution shows a homogeneous error variance to validate the normality assumption.
    - The Ryan-Joiner analysis, which is similar to the Schapiro Wilk, analysis, is performed to evaluate the normality of the residual values
- Factorial design using ANOVA is also used in this dissertation. The main objective of this analysis is to review if there is an effect from one or more variables and a response variable. This analysis is mainly used to analyze the effects of requirement parameters on the RMI.

- Contour plotting is used to analyze the relationship between a response and two continuous variables. The objective is to define different levels for the response variable to evaluate the relationship. This plot is primarily used to analyze the relationship between the Wymore variables of requirements and RMI. The maximum value of the requirement stability calculated previously is very important for defining the scale of RMI during the contour plot analysis.
- Regression is also used in this dissertation. This analysis is used to analyze the relationship, for example, between RMI and requirement parameters. The coefficient of determination ($R^2$) and p-value are also used to illustrate the strength of the relationships between variables in the analysis.
- As part of the regression, a variance inflation factor (VIF) analysis is performed to confirm the lack of multicollinearity among the variables.
- It is also important to note that for nonlinear regression, which is used to analyze the relationship between agility and fitness, the author uses the p-value and not $R^2$ to demonstrate the strength of the relationship.

# Chapter 4 – Results

The results of this simulation will incorporate some information from [44]. There are three steps in this chapter. First, the tendencies of the results for real and simulated data in the error of the prediction and fitness are presented. Second, the results for the factors with a greater impact on requirement stability are described. Third, the results for the changeability parameters and fitness are presented.

## 4.1 Classifier Systems Results

For an XCSF simulation process to be acceptable, its error must be below 0.01, which is the threshold error value. Figure 4-1 shows the error values throughout the simulation process; during all parts of the simulation process, all values are below 0.01. Classifiers with a prediction error below this value are considered accurate. The XCSF model for the TelCo system expects that its prediction error falls below the predefined threshold value [41]. Figure 4-1 shows the tendency for real and simulated data. Most data are close to zero. Therefore, the simulation and real data are valid for the XCSF model. The general tendencies for both values are nearly identical, with some variance of the values but always less than 0.01. As expected, in general, the value of real data is higher than that of simulated data. Simulated data yields better predictions because they are based on an established model, whereas real data have higher margin errors.

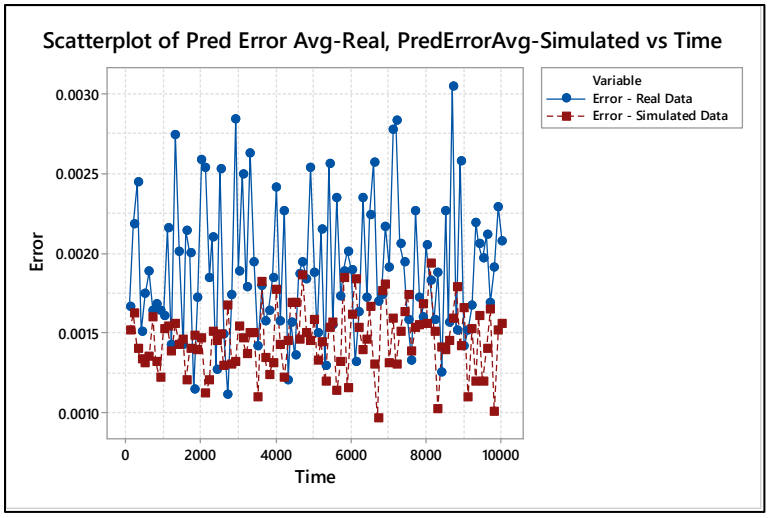


Figure 4-1: Error tendencies for real and simulated data.

Figure 4-2 shows the trends of fitness over time for two sets of information: simulated data and real data. The initial value for both datasets is 0.01. The values for both datasets have some level of variance. For the real data, the values are between approximately 0.01 and 0.16, and for the simulated data, the values are between approximately 0.10 and 0.14. The higher value in the real data is due to the randomness in the simulated data. As explained previously, in general, the simulated data are randomly generated based on historical information and on a telecommunications process.

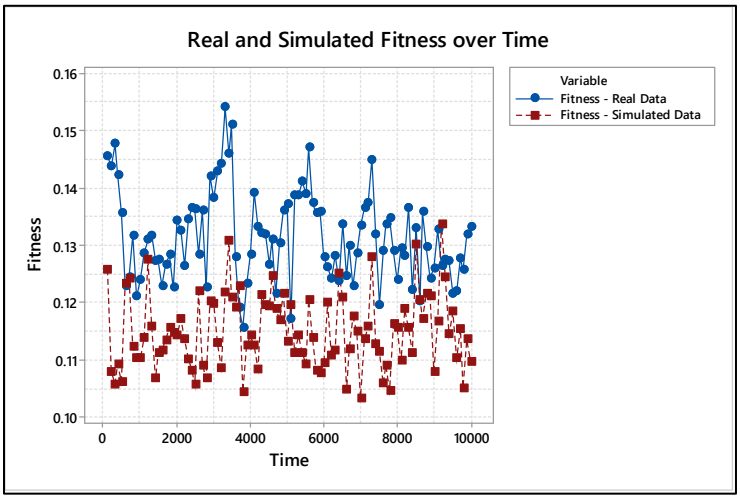


Figure 4-2: Fitness over time for real and simulated data.

## 4.2 Impact on Change - Results

One of the performed simulations confirms the concept described by Wymore, which describes the relationship between requirements and technology, performance, cost and tradeoff. Tradeoff is a function that depends on cost and performance, and therefore, it is not included in this initial analysis, as it would generate a correlation. The first objective is to analyze the normality assumptions of the variables. Figure 4-4 shows the normal probability plot, which demonstrates that the residual values of the variable in analysis are normally distributed. This graph shows that the residuals follow a fairly linear distribution; only the two extreme tails of the diagram, which represent 5% of the data, do not follow the straight line. Figure 4-5 shows the variance, which demonstrates that the residual follows a constant variance. This graph shows that the values are not perfectly distributed but do not follow any pattern. In addition, the values are evenly distributed around the mean of all the values. Based on the previous explanation, the author understands that the diagrams do not follow a perfect normal distribution, but the closeness of both graphs to a normal distribution allows the author to provide some insights to continue the analysis. To demonstrate the normality, the author uses the hypotheses described on equations 4.1 and 4.2

$H0: Data\ follow\ a\ normal\ distribution$ … (4.1)

$Ha$: Data do not follow a normal distribution … (4.2)

If p-value ≤ α, then he data do not follow a normal distribution (reject H0). If p-value > α, then the author cannot conclude that the data do not follow a normal distribution (fail to reject H0). Figure 4-3 shows the Ryan-joiner test, which is similar to the Shapiro-Wilk test. According to Figure 4-4, the p-value = 0.062 > 0.05.

Therefore, this author will accept the normality requirement from the ANOVA analysis.

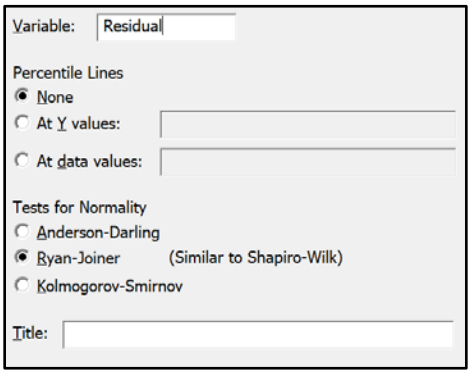


Figure 4-3: Minitab test analysis

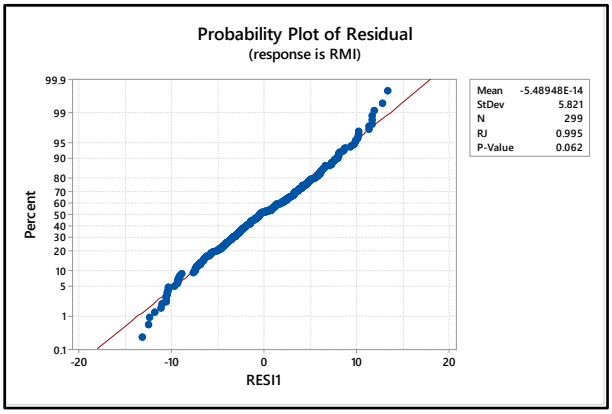


Figure 4-4: Normal probability plot.

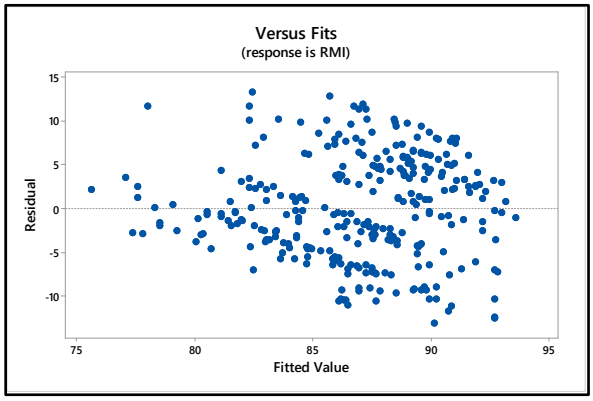

Figure 4-5: Variance – residual analysis

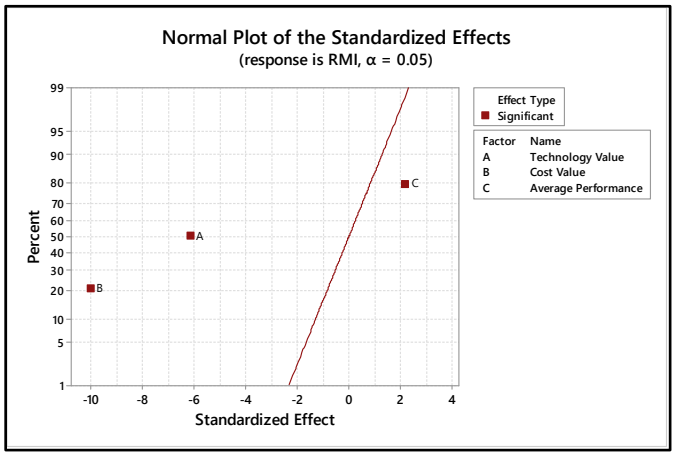


Figure 4-6: Impact analysis

Figure 4-6 helps the author to show the magnitude of each of the variables in the analysis. Figure 4-6 shows that at the 0.05 level, the variables A, B and C, which represent the technology, cost and performance, respectively, have a significant effect on the RMI. The combination of technology and performance also has a significant effect on the RMI. Finally, Figure 4-7 shows the result of the ANOVA analysis. Because all single variables have p-values of less than 0.05, the author concludes that the three variables have a significant effect on the RMI. As explained previously, the figure also shows that the combination of technology and performance has a significant effect on the RMI.

**Analysis of Variance**

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Model | 3 | 3867.8 | 1289.28 | 37.66 | 0.000 |
| Linear | 3 | 3867.8 | 1289.28 | 37.66 | 0.000 |
| Technology Value | 1 | 1308.5 | 1308.52 | 38.22 | 0.000 |
| Cost Value | 1 | 3453.1 | 3453.14 | 100.87 | 0.000 |
| Average Performance | 1 | 161.9 | 161.87 | 4.73 | 0.030 |
| Error | 295 | 10099.0 | 34.23 | | |
| Lack-of-Fit | 264 | 9617.4 | 36.43 | 2.35 | 0.003 |
| Pure Error | 31 | 481.6 | 15.53 | | |
| Total | 298 | 13966.8 | | | |

Figure 4-7: ANOVA analysis.

The next step in this analysis is to show the effect of each of the two variables on the RMI using contour plot analysis. Figure 4-8 shows the relationship among the RMI, technology and performance. In this case, performance is the average of the two parameters previously described. There is a tendency to have more changes when there is high technology and low performance; a different situation is observed for low technology and low/medium performance.

Figure 4-9 shows the relationships among the RMI, technology and cost. The figure shows a stable area with low technology and cost values. There is a tendency to have more changes when technology is high, and cost is low/medium; the same situation is observed when technology is low and cost is high.

Figure 4-10 shows the relationships of the RMI with technology and tradeoff. These results suggest that there is a tendency to have many changes when technology is high and tradeoff is low. The same situation occurs when technology is low and tradeoff is high. Requirement stability is present when technology and tradeoff are both low.

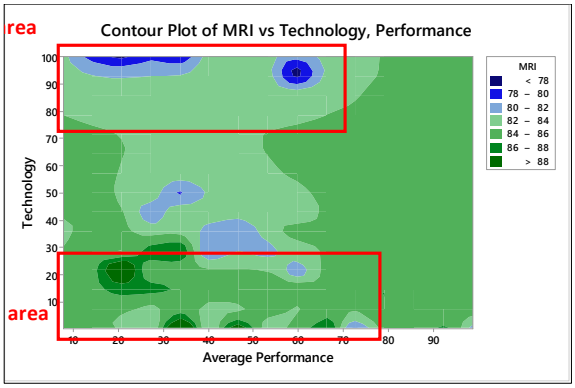


Figure 4-8: RMI vs technology and performance.

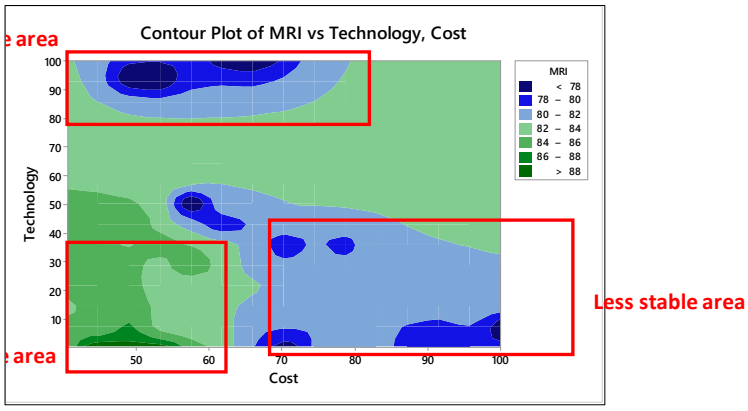


Figure 4-9: RMI vs technology and cost.

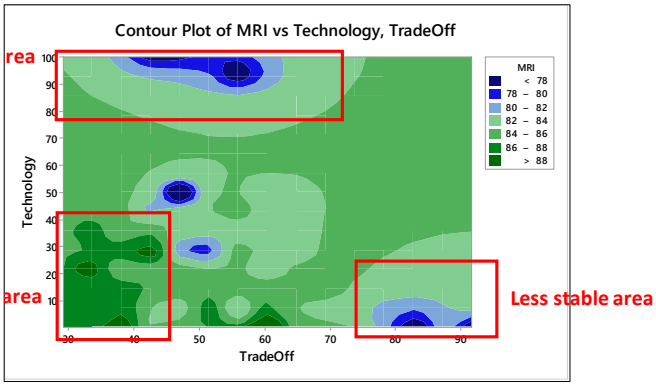


Figure 4-10: RMI vs technology and tradeoff.

Finally, a regression analysis is performed. Table 4-1 shows the VIF values. VIF values equal to one mean that multicollinearity does not occur among the variables in the analysis. Because the values are closer to one, the author concludes multicollinearity does not occur and the author can proceed with the regression. The regression analysis produces the result shown in equation 4.3, Table 4-2 and Table 4-3. Performance has two components (time to respond and average incident time). The $R^2$ for this regression is 22.19%, and the p-values for all variables are less than 0.05 except for the Performance2 variable, which has a p-value of 0.282. Based on this information, the null hypothesis described in Chapter 4 is rejected, and therefore the

author accepts that there are relationships of technology, cost and performance with RMI.

Table 4-1: VIF values for the variables

| Term | Coef | SE Coef | T-Value | P-Value | VIF |
|---|---|---|---|---|---|
| Constant | 105.69 | 1.54 | 68.45 | 0 | |
| Technology | -0.1162 | 0.0206 | -5.63 | 0 | 1.19 |
| Cost | -0.3413 | 0.0239 | -14.3 | 0 | 1.00 |
| Performance 1 | 0.00735 | 0.00682 | 1.08 | 0.282 | 1.10 |
| Performance 2 | -0.0253 | 0.0113 | -2.25 | 0.025 | 1.27 |

$$RMI = 105.69 - 0.12 * Technology - 0.34 * Cost + 0.0074 * Performance1 - 0.0253 * Performance2 \quad (4.3)$$

Table 4-2: Regression – analysis of variance for the requirement

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Regression | 4 | 10051 | 2512.74 | 56.87 | 0 |
| Technology | 1 | 1402.3 | 1402.32 | 31.74 | 0 |
| Cost | 1 | 9032.3 | 9032.29 | 204.43 | 0 |
| Performance 1 | 1 | 51.3 | 51.27 | 1.16 | 0.282 |
| Performance 2 | 1 | 222.7 | 222.74 | 5.04 | 0.025 |
| Error | 775 | 34241.5 | 44.18 | | |
| Lack-of-Fit | 664 | 32494.6 | 48.94 | 3.11 | 0 |
| Pure Error | 111 | 1746.9 | 15.74 | | |
| Total | 779 | 44292.5 | | | |

Table 4-3: Model summary for the requirement

| S | R-sq | R-sq(adj) | R-sq(pred) |
|---|---|---|---|
| 6.647 | 22.69% | 22.29% | 21.88% |

## 4.3 Changeability and Fitness - results

### i. Flexibility and fitness

The first step in the analysis of the relationship between flexibility and fitness is to assess the normality probability assumption. Figure 4-11 shows a residual plot with a nonlinear pattern. Under this analysis, the author proceeds to perform

nonlinear regression analysis. Figure 4-12 shows the relationship between the fitness and flexibility of the simulated data. The first value of fitness during the simulation process is 0.01. Through the process of evolution of the CAS system using XCSF, the system achieves a fitness value of approximately 0.109. The relationship between fitness and flexibility is shown in Figure 4-12. The S value is an important measure of goodness of fit for a nonlinear model because $R^2$ does not provide a good measure of nonlinear relationships. The S value is 0.00891, which is relatively low and represents an accurate fit. Fitness and flexibility start from zero and go toward a stable value. Equation (4.4) shows the relationship. According to this formula, there is a rapid increase in the fitness value (0.05) when the system is not flexible (values less than 1). When the system is flexible (value equals 1), fitness increases at a very small rate (0.002). According to [44], "this is what we expect for any system, i.e., when flexibility increases, the system is more adaptable. However, after a certain level of adaptation is attained, flexibility has a small impact on fitness" [44].

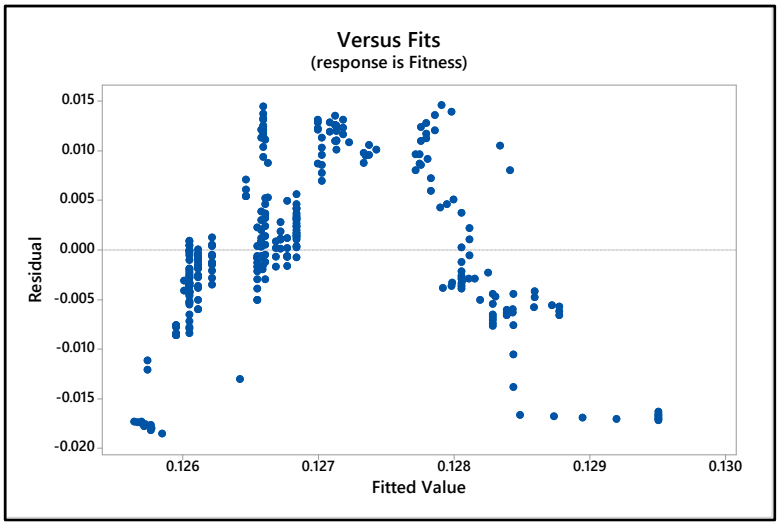


Figure 4-11: Residual plot for flexibility and fitness

$$Fitness = 0.137426 - 0.0587665 * e^{-2*Flexibility} \quad (4.4)$$

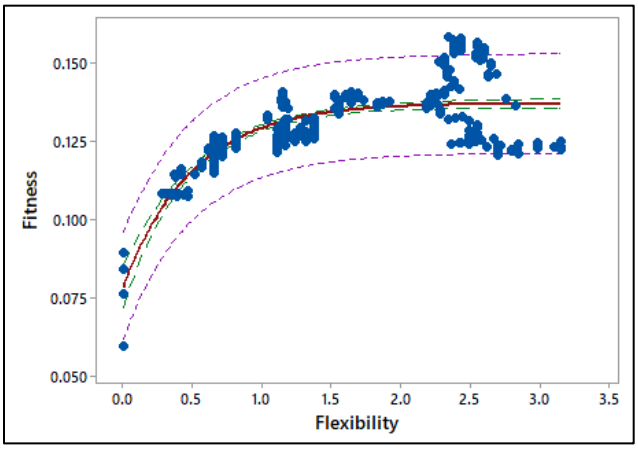


Figure 4-12: Flexibility and fitness relationship

Table 4-4: Lack of fit analysis for flexibility

| Source | DF | SS | MS | F | P |
|---|---|---|---|---|---|
| Error | 355 | 0.0230253 | 0.0000649 | | |
| Lack of Fit | 93 | 0.021576 | 0.000232 | 41.94 | 0 |
| Pure Error | 262 | 0.0014493 | 0.0000055 | | |

### ii. Adaptability and fitness

In this analysis, the author also starts with the normality assumption of the residual. Figure 4-13 shows a residual plot with a nonlinear pattern. Based on this analysis, the author assumes the linear relationship and proceeds with the nonlinear regression analysis. After scatterplot analysis, the author proceeds to perform cubic relationship analysis of adaptability and fitness and obtains a low S value of 0.0111124, which indicates a good regression value. The relationship is shown graphically in Figure 4-14, and the regression analysis is shown in detail in Table 4-5 and Table 4-6. For adaptability, the system starts at a low value. When the system is adaptable (corresponding to a value equal to one), there is little variability

in fitness. Most fitness values are achieved when the system is adaptable. Equation (4.5) shows the relationship. There is an increase in fitness at a rate of 0.059 until an adaptability of 0.3 is attained. Thereafter, there is a decrease in fitness at a rate of −0.011 until an adaptability of 0.6 is attained. Finally, for adaptability values exceeding 0.6, there is an increase at a rate of 0.097. This result, i.e., a positive relationship between fitness and adaptability, was expected [44].

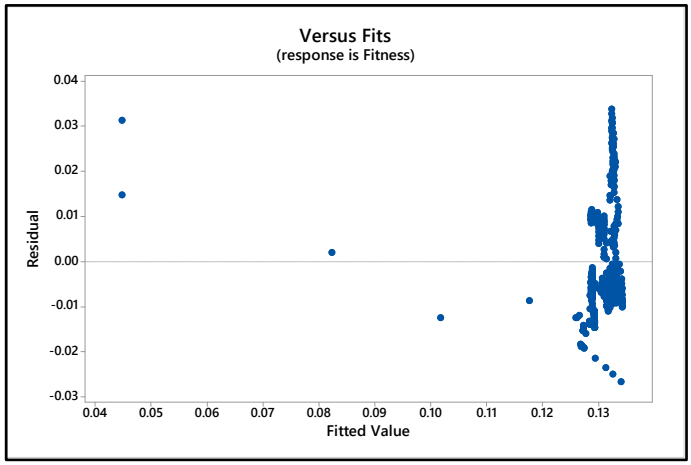


Figure 4-13 Residual plot for adaptability and fitness

$$Fitness = 0.06788 + 0.1433 * Adaptability - 0.3615 * Adaptability^2 + 0.2736 * Adaptability^3 \quad (4.5)$$

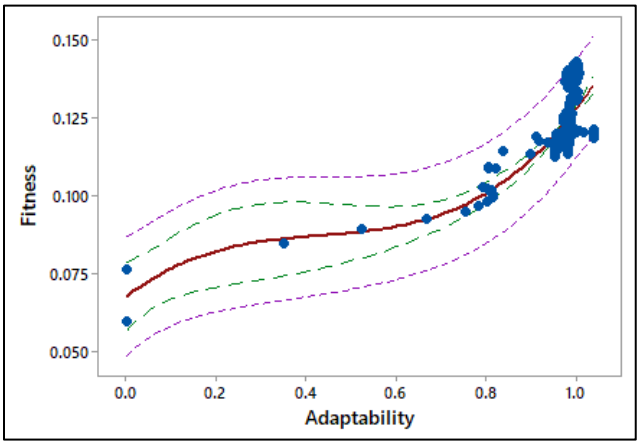


Figure 4-14 Adaptability and fitness relationship.

Table 4-5: Model summary – fitness and adaptability

| S | R-sq | R-sq(adj) |
|---|---|---|
| 0.0079704 | 52.21% | 51.68% |

Table 4-6: Analysis of variance – fitness and adaptability

| Source | DF | SS | MS | F | P |
|---|---|---|---|---|---|
| Regression | 3 | 0.0189436 | 0.0063145 | 99.4 | 0 |
| Error | 273 | 0.0173431 | 0.0000635 | | |
| Total | 276 | 0.0362868 | | | |

### iii. Robustness and fitness

This analysis also starts with Figure 4-15 which shows a residual plot with a nonlinear pattern. Therefore, the author accepts the nonlinear analysis of these variables. Figure 4-16 shows a good relationship between fitness and robustness. The S value for this relationship is 0.0072004, which indicates that there is a good relationship between the two variables. A positive relationship exists between these two variables; equation (4.6) shows the relationship. According to this formula, there is a slow increase in fitness (rate of 0.08), even when the systems are robust (values exceeding 0.5). Moreover, fitness increases at a very small rate (below 0.001) when robustness is higher than 0.7. The new rate of increase in fitness is 0.006 [44].

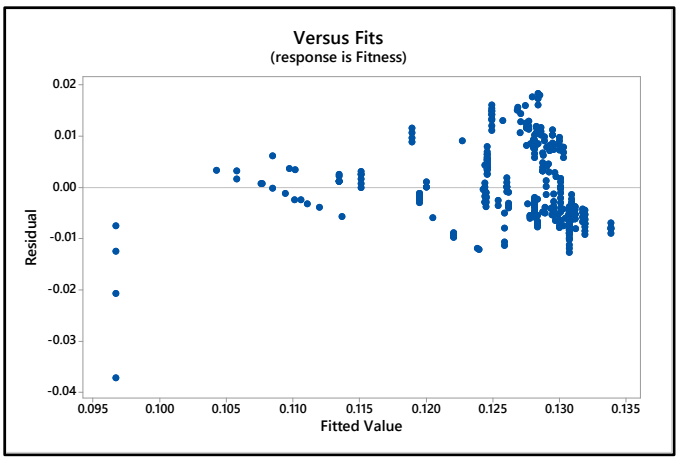


Figure 4-15: Residual plot for adaptability and robustness

$$Fitness = 0.134771 - 0.0576978 * e^{-3.64992 * Robustness} \quad (4.6)$$

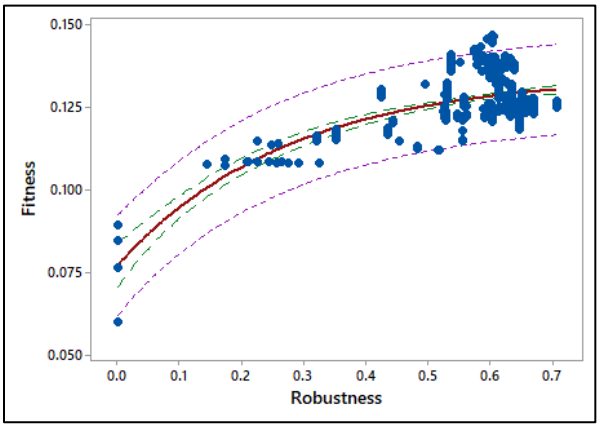


Figure 4-16: Robustness and fitness relationship.

Table 4-7: Lack of fit for robustness

| Source | DF | SS | MS | F | P |
|---|---|---|---|---|---|
| Error | 329 | 0.0160265 | 0.0000487 | | |
| Lack of Fit | 101 | 0.0149862 | 0.0001484 | 32.52 | 0 |
| Pure Error | 228 | 0.0010403 | 0.0000046 | | |

### iv. Agility and fitness

Finally, for the agility and fitness analysis, the author also performs a residual

analysis as illustrated in Figure 4-17 which shows a residual plot with a nonlinear pattern. This figure shows that the assumption is acceptable for nonlinear regression analysis. Figure 4-18 shows a good relationship between fitness and agility. The system starts with relatively low agility. As agility increases, fitness also increases. When agility increases further, fitness decreases. The S value for this relationship is 0.0039932, which indicates a good nonlinear relationship; equation (4.7) shows the relationship. Tables 16 and 17 show the detailed information about the regression analysis. Fitness increases until agility reaches 0.12 at a rate of 0.083. Thereafter, there is a decrease in fitness at a rate of 0.0081. Finally, there is a rapid increase when agility reaches 0.48 at a rate of 0.014. According to these results, when system agility increases, fitness also increases. However, at some point (agility value of 0.12), agility has a negative effect on fitness until agility reaches 0.48 (the fitness decreases at a rate of 0.081). For agility values exceeding 0.48, there is an increase in fitness; the new rate of increase is 0.069 [44].

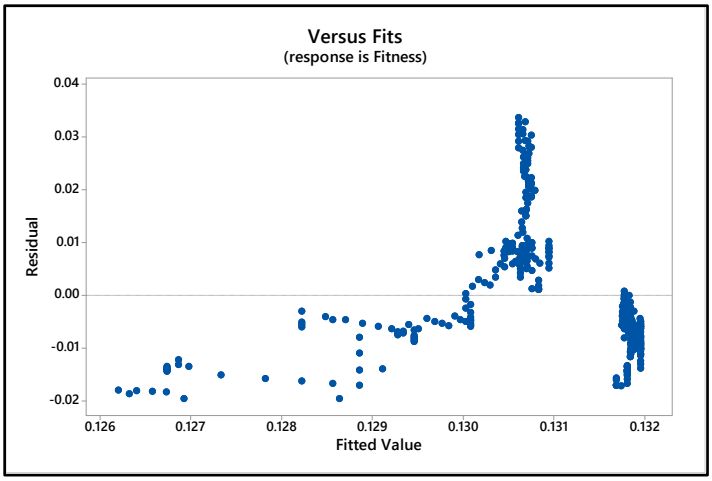


Figure 4-17: Residual plot for agility and fitness

$$Fitness = 0.125 + 0.1938 * Agility - 1.074 * Agility^2 + 1.235 * Agility^3 \quad (4.7)$$

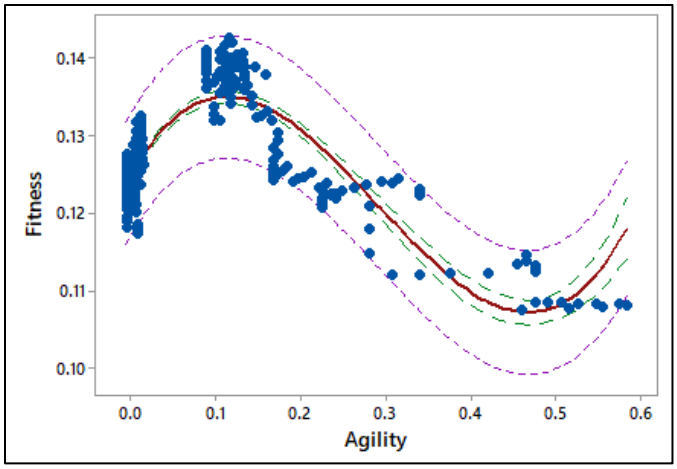


Figure 4-18: Agility and fitness relationship.

Table 4-8: Model summary – fitness and agility

| S | R-sq | R-sq(adj) |
|---|---|---|
| 0.0039932 | 73.71% | 73.45% |

Table 4-9: Analysis of variance – fitness and agility

| Source | DF | SS | MS | F | P |
|---|---|---|---|---|---|
| Regression | 3 | 0.0135473 | 0.0045158 | 283.19 | 0 |
| Error | 303 | 0.0048316 | 0.0000159 | | |
| Total | 306 | 0.0183788 | | | |

# Chapter 5 – Discussion and Conclusions

As described in the introduction of this dissertation, this study is based on previous relevant studies that have analyzed or partially analyzed requirement stability, changeability and CAS or analyzed these parameters separately. This dissertation developed a new model that considers the calculation of the impact of requirement stability, changeability parameters and fitness by simulating a telecommunication system. This simulation is based on the classifier system as representative of CASs. This dissertation presents results showing the factors with the greatest impact on the requirement stability and the relationship between changeability parameters and CAS behavior. This new approach provides insights into the relationship between factors with a greater impact on the generation of requirements and the evolution and dynamism of a CAS. Finally, this dissertation presents an understanding of behavior in a technology environment, where new requirements emerge from the environment and the system evolves to adapt to the environment continuously. This dissertation uses some information from a previous study performed by the author [4].

## 5.1 Regarding the Model

This dissertation models the behavior of a CAS using a classifier system with the objective of analyzing the relationship with changeability. The Java XCSF algorithm and sample telecommunication data were used to model the evolution of a telecommunication system modeled as a classifier system and to calculate its fitness value. All the changeability indicators (fitness, error, flexibility, adaptability, robustness and agility) and requirement indicators (RMI, cost, performance, technology and tradeoff) were calculated during the process of simulation. To validate

the model, historical data of a TelCo were collected and used to generate the agent interactions. The results demonstrated that a telecommunication system can be modeled as a CAS using XCSF because the error is always below the error threshold. The error thresholds suggest that the model is appropriate for the telecommunication system and may be applicable to other types of CAS.

### 5.2 Regarding Change Impact

The requirement for closeness to normal helped the author perform the analysis supporting the conclusions in this section. This study provides the system engineering community a general view of the parameters that should be considered when analyzing new requirements. This study also allows the magnitude of the influence of a parameter to be visualized for consideration during the lifecycle of a system. This study applies to telecommunications as part of a technology system but can be applied to any type of technology as long as the technology can be considered a CAS. High technology affects the stability of requirements for customers. This variable does not depend on performance, cost or tradeoff, indicating that, when new technology is implemented, more requirements are generated.

### 5.3 Regarding the Fitness and Changeability Aspects

Based on the simulation results presented in this study, the author can draw the following conclusions. (1) Flexibility and robustness have positive relationships with the fitness. However, there is a point of stability. For flexibility, the stability is achieved when the system reaches the point of flexibility (equals 1) and robustness at which the system is more robust (higher than 0.5). (2) Adaptability has a positive relationship with fitness until the point at which the system is adaptable (adaptability value equals one); thereafter, there are few variances of fitness. (3) Agility and fitness

initially have a positive relationship, but when the system becomes more agile, i.e., when more changes are implemented in the system, agility and fitness have a negative relationship.

## 5.4 Future Research

Aspects of changeability values other than those used in this dissertation can be used in future research. In Chapter 2, the author mentioned that Fricke and Schultz proposed the correlation matrix. This matrix includes other aspects of changeability, such as redundancy and scalability, that can be used in future research. In addition, the method and model used in this dissertation provide a guideline for future analysis in the context of other technological industries. Another aspect to be included in future research is the inclusion of an assessment of testing, which was outside the scope of this research as an aspect to generate new requirements and consequently impact the requirement stability of the system. The research described in this dissertation is a starting point toward achieving the objective of providing a more detailed simulation model that can address future questions from customers and stakeholders.

## 5.5 Future Applications

Chapter 4 explained other algorithms that different researchers have used to represent classifier systems. Future research could include the application of these algorithms. Python is now applied in different concepts of machine learning. The approach in this dissertation takes into consideration not only the actual evolution of a CAS as represented by the CAS but also what factors have the greatest impact on the generation of change (through requirements) as well as all aspects of changeability. This approach enables systems engineers and other stakeholders to gain useful insights into the evolution of a system under a given set of conditions and

achieve a better view of what generates new requirements and how or where the system could be more robust or more adaptable. The management field is also a good application of the method described in this dissertation; each member of a team is an agent who adapts to new functions to achieve new strategies, and the environment in which management performs today requires greater agility and flexibility, for example, to create new products.

The author wants to conclude this dissertation with an explanation of how CASs are impacted by the generation of requirement and the relationship with changeability. Figure 5-1, which is based on Figure 1-1, shows these relationships. On the left side, the environment that has a direct interaction with the evolving CAS is shown. However, as part of the environment, four factors are considered in this dissertation. On the left side the author has the four factors studied in this dissertation. Longer arrows indicate a greater impact on requirement stability. At the bottom of this graph, the relationship of each of the four aspects of changeability with fitness is shown. This graph can give stakeholders and customers better visibility of all parameters studied in this dissertation.

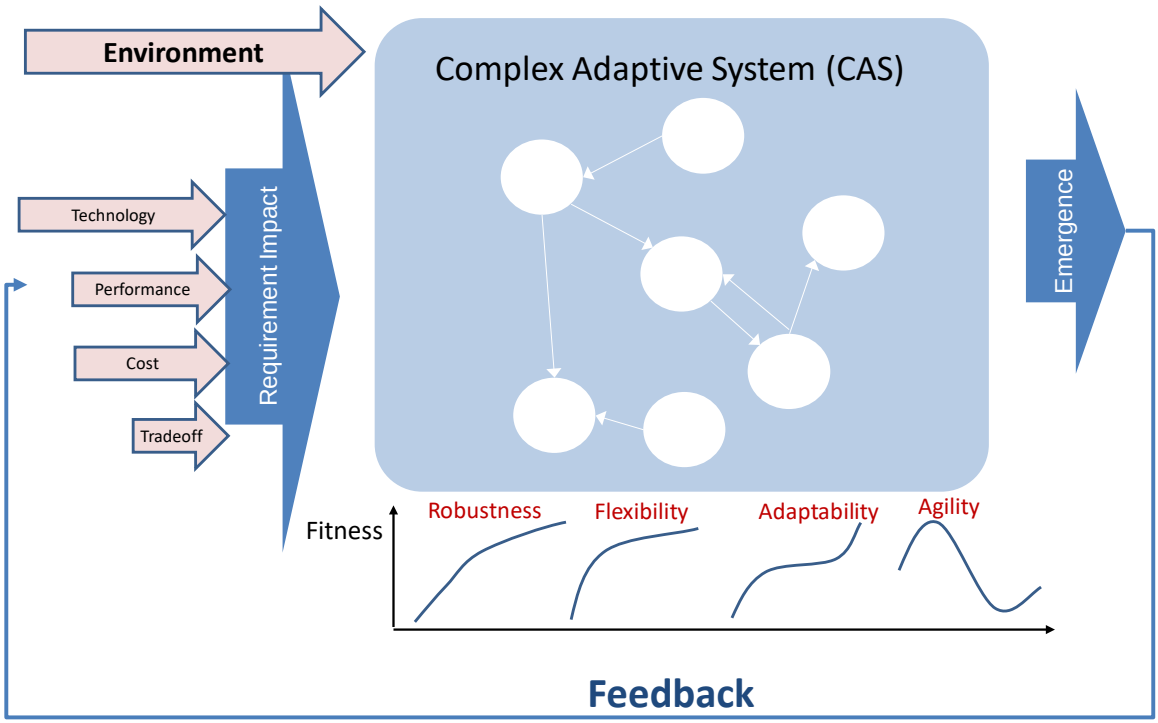


Figure 5-1: Final objective of the dissertation

## Bibliography

[1] FOSTER, R. (1986), Innovation: The Attacker’s Advantage, Summit Books, New York.

[2] E. Fricke and A. P. Schulz, "Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle," Syst. Eng., vol. 8, no. 4, pp. 342–359, Oct. 2005.

[3] T. Maeshiro, "The importance of the robustness and changeability in evolutionary systems," in SMC'98 Conf. Proc. 1998 IEEE Int. Conf. Syst., Man, and Cybern. (Cat. No.98CH36218), San Diego, CA, 1998, pp. 2342–2347.

[4] M. C. Edson, "A complex adaptive systems view of resilience in a project team," Syst. Res. Behavioral Sci., vol. 29, no. 5, pp. 499–516, Sept. 2012.

[5] M. Tilebein, "A complex adaptive systems approach to efficiency and innovation," Kybernetes, vol. 35, no. 7/8, pp. 1087–1099, Aug. 2006.

[6] B. M. Khoshroo and H. Rashidi, "Towards a framework for agile management based on chaos and complex system theories," in 2009 16th Annu. IEEE Int. Conf. Workshop Eng. Comput. Based Syst., San Francisco, CA, USA, 2009, pp. 291–292.

[7] S. D. Pathak, J. M. Day, A. Nair, W. J. Sawaya, and M. M. Kristal, "Complexity and adaptivity in supply networks: building supply network theory using a complex adaptive systems perspective," Decision Sci., vol. 38, no. 4, pp. 547–580, Dec. 2007.

[8] Bush, D., & Finkelstein, A. (2002, May). Environmental scenarios and requirements stability. In Proceedings of the international workshop on principles of software evolution - IWPSE '02 (pp. 133–137). Orlando, Florida: ACM Press.

[9] Bush, D., & Finkelstein, A. (2003, September). Requirements stability assessment using scenarios. In Proceedings 11th IEEE international requirements engineering conference (pp. 23–32), Monterey Bay, CA.

[10] Costello, R. J., & Liu, D. B. (1994). Metrics for requirements engineering (Maters thesis). California State University, Long Beach.

[11] Guo, J. (2010). Towards automatic analysis of software requirement stability. International Journal on Advances in Information Sciences and Service Sciences, 2, 33–42. https://doi.org/10.4156/aiss.

[12] Kulk, G. P., & Verhoef, C. (2008). Quantifying requirements volatility effects. Science of Computer Programming, 72, 136–175. https://doi.org/10.1016/j.scico.2008.04.003.

[13] Loconsole, A., & Borstler, J. (2005, December). An industrial case study on requirements volatility measures. In 12th Asia-pacific software engineering conference (APSEC'05). Taipei, Taiwan: IEEE.

[14] Stark, G. E., Oman, P., Skillicorn, A., & Ameele, A. (1999). An examination of the effects of requirements changes on software maintenance releases. Journal of Software Maintenance: Research and Practice, 11, 293–309. https://doi.org/10.1002/(SICI)1096-908X(199909/10)11:5<293::AID-SMR198>3.0.CO;2-R.

[15] Qi, J., Shu, H., & Kan, K. (2006). OSS and ERP systems: Likeness, difference and reference. In A. M. Tjoa, L. Xu & S. S. Chaudhry (Eds.), Research and practical issues of enterprise information systems (pp. 457–466). Boston, MA: Springer US.

[16] Tien, J. M. (2008). Services: A system's perspective. IEEE Systems Journal, 2, 146–157. https://doi.org/10.1109/JSYST.2008.917075.

[17] S. Forrest and M. Mitchell, "Adaptive computation: the multidisciplinary legacy of John h. Holland," Commun. ACM, vol. 59, no. 8, pp. 58–63, Jul. 2016.

[18] J. H. Holland, Signals and Boundaries: Building Blocks for Complex Adaptive Systems. Cambridge, MA: MIT Press, 2012.

[19] T. Maeshiro, "The importance of the robustness and changeability in evolutionary systems," in SMC'98 Conf. Proc. 1998 IEEE Int. Conf. Syst., Man, and Cybern. (Cat. No.98CH36218), San Diego, CA, 1998, pp. 2342–2347.

[20] M. C. Edson, "A complex adaptive systems view of resilience in a project team," Syst. Res. Behavioral Sci., vol. 29, no. 5, pp. 499–516, Sept. 2012.

[21] P. Antoniou and A. Pitsillides, "Understanding complex systems: A communication networks perspective," Cyprus Univ., Nicosia, Cyprus, Tech. Rep. TR-07-01, Feb. 2007.

[22] R. R. McDaniel, H. J. Lanham, and R. A. Anderson, "Implications of complex adaptive systems theory for the design of research on health care organizations," Health Care Manage. Rev., vol. 34, no. 2, pp. 191–199, Apr. 2009.

[23] Holland, J. H. (1995). Hidden orderhow adaptation builds complexity (No. 003.7 H6).

[24] Holland, J. H. (2012). Signals and boundaries: Building blocks for complex adaptive systems. Mit Press.

[25] R. J. Urbanowicz and J. H. Moore, "Learning classifier systems: a complete introduction, review, and roadmap," J. Artificial Evol. Appl., vol. 2009, pp. 1–25, Jun. 2009.

[26] J. Kacprzyk and W. Pedrycz, Springer Handbook of Computational Intelligence. Berlin Heidelberg: Springer, 2015.

[27] M. V. Butz, "Learning classifier systems," in Springer Handbook of Computational Intelligence, J. Kacprzyk and W. Pedrycz, Eds., Berlin, Heidelberg: Springer, 2015, pp. 961–981.

[28] P. L. Lanzi, D. Loiacono, S. W. Wilson, and D. E. Goldberg, "Generalization in the XCSF classifier system: analysis, improvement, and extension," Evol. Comput., vol. 15, no. 2, pp. 133–168, Jun. 2007.

[29] P. L. Lanzi, D. Loiacono, S. W. Wilson, and D. E. Goldberg, "Prediction update algorithms for XCSF: RLS, kalman filter, and gain adaptation," in Proc. 8th Annu. Conf. Genetic Evol. Computation, Seattle, Washington, 2006, pp. 1505–1512.

[30] Wymore, A. W. (1993). Model-based systems engineering. Boca Raton: CRC Press.

[31] Chapman, W. (2018). Engineering modeling and design. New York, NY: Routledge.

[32] Nogueira, J. C., Luqi, & Bhattacharya, S. (2000, June). A risk assessment model for software prototyping projects. In Proceedings 11th international workshop on rapid system prototyping. RSP 2000. Shortening the path from specification to prototype (Cat. No.PR00668) (pp. 28–33), Paris, France.

[33] Anderson, S., & Felici, M. (2002, August). Quantitative aspects of requirements evolution. In Proceedings 26th annual international computer software and applications (pp. 27–32), Oxford, UK.

[34] A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Defining changeability: reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value," Syst. Eng., vol. 11, no. 3, pp. 246–262, Jun. 2008.

[35] R. Nilchiani and D. Hastings, "Defining software flexibility in space systems," in Space 2004 Conf. Exhibit, AIAA SPACE Forum, San Diego, CA, 2004, pp. 5857.

[36] J. W. Baker, M. Schubert, and M. H. Faber, "On the assessment of robustness," Structural Safety, vol. 30, no. 3, pp. 253–267, May 2008.

[37] H. Pham, System Software Reliability. London: Springer Science & Business Media, 2007.

[38] H. Pham, Springer Handbook of Engineering Statistics. Berlin, Germany: Springer, 2006.

[39] D. M. Shawky and A. F. Ali, "A practical measure for the agility of software development processes," in 2010 2nd Int. Conf. Comput. Technol. Develop., Cairo, Egypt, 2010, pp. 230–234.

[40] Merz, P., & Freisleben, B. (2000). Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. IEEE Transactions on Evolutionary Computation, 4, 337–352. https://doi.org/10.1109/4235.887234.

[41] P. O. Stalph and M. V. Butz, "Current XCSF capabilities and challenges," in Learning Classifier Systems, J. Bacardit, W. Browne, J. Drugowitsch, E. Bernadó-Mansilla, and M. V. Butz, Eds., Berlin, Heidelberg: Springer, 2010, pp. 57–69.

[42] Edson, M. C. (2012). A complex adaptive systems view of resilience in a project team. Systems Research and Behavioral Science, 29(5), 499-516.

[43] Holland, J. H. (2006). Studying complex adaptive systems. Journal of Systems Science and Complexity, 19(1), 1-8.

[44] Avalos, J., Grenn M. & Roberts B. (2018). Assessment of Complex Adaptive System Changeability using a Learning Classifier System. IEEE Systems Journal

[45] Bull, L. (2004). Learning classifier systems: A brief introduction. In Applications of Learning Classifier Systems (pp. 1-12). Springer, Berlin, Heidelberg.

[46] Arteta, B. M., & Giachetti, R. E. (2004). A measure of agility as the complexity of the enterprise system. Robotics and computer-integrated manufacturing, 20(6), 495-503.

[47] Ross, A. M. (2006). Managing unarticulated value: changeability in multi-attribute tradespace exploration (Doctoral dissertation, Massachusetts Institute of Technology, Engineering Systems Division).

[48] Choi, T. Y., Dooley, K. J., & Rungtusanatham, M. (2001). Supply networks and complex adaptive systems: control versus emergence. Journal of operations management, 19(3), 351-366.

[49] Haghnevis, M., & Askin, R. G. (2012). A modeling framework for engineered complex adaptive systems. IEEE Systems Journal, 6(3), 520-530.

[50] P. O. Stalph and M. V. Butz, "Documentation of JavaXCSF," Würzburg Univ., Germany, COBOSLAB Report Y2009N001, Oct. 2009

[51] The XCS Library. (n.d.). Retrieved from http://xcslib.sourceforge.net/

[52] Accuracy-based Learning Classifier Systems for Python 3 https://pypi.org/project/xcs/

[53] Stalph, P. O., & Butz, M. V. (2010). JavaXCSF: the XCSF learning classifier system in Java. ACM SIGEVOlution, 4(3), 16-19.

[54] Butz, M. V. (2007). The XCSF classifier system in Java. ACM SIGEVOlution, 2(2), 10-13.

[55] N. Pidgeon, C. Hood, D. Jones, B. Turner, and R. Gibson, "Risk perception," in Risk: Analysis, Perception and Management: Report of a Royal Society Study Group. London, UK: Royal Society, 1992, pp. 89–134.

[56] V.-W. Mitchell, "Organizational risk perception and reduction: a literature review," Brit. J. Manage., vol. 6, no. 2, pp. 115–133, Jun. 1995.

[57] J. E. Gillett, Hazard Study and Risk Assessment in the Pharmaceutical Industry. Florida, US: Interpharm Press, 1996.

[58] C. Harland, R. Brenchley, and H. Walker, "Risk in supply networks," J. Purchasing Supply Manage., vol. 9, no. 2, pp. 51–62, Mar. 2003.

[59] AXELOS, ITIL Service Design. AXELOS, 2011

[60] Mathworks. (2017). Weibull Parameter Estimates. Available: https://www.mathworks.com/help/stats/wblfit.html, Accessed on: Dec. 1, 2017

[61] McCarthy, I. P. (2004). Manufacturing strategy: understanding the fitness landscape. International Journal of Operations & Production Management, 24(2), 124-150.

[62] Li, G., Ji, P., Sun, L. Y., & Lee, W. B. (2009). Modeling and simulation of supply network evolution based on complex adaptive system and fitness landscape. Computers & Industrial Engineering, 56(3), 839-853.

[63] Kauffman, S. A. (1997). At home in the universe. Mathematical Social Sciences, 1(33), 94-95.

[64] Schulz, A. P., & Fricke, E. (1999, November). Incorporating flexibility, agility, robustness, and adaptability within the design of integrated systems-key to success?. In Digital Avionics Systems Conference, 1999. Proceedings. 18th (Vol. 1, pp. 1-A). IEEE.

[65] Fleming, L., & Sorenson, O. (2001). Technology as a complex adaptive system: evidence from patent data. Research policy, 30(7), 1019-1039.

[66] Mabrok, M. A., Elsayed, S., & Ryan, M. J. (2016). Mathematical framework for recursive model-based system design. Nonlinear Dynamics, 84(1), 223-236.

[67] Carne, E. B. (2012). Modern telecommunication. Springer Science & Business Media.

[68] Snoeck, M., & Michiels, C. (2002). Domain modelling and the co-design of business rules in the telecommunication business area. Information Systems Frontiers, 4(3), 331-342.

[69] B. Marjan, "An application of classifier systems to the reduction of finite state machines," M.S. thesis, Lehigh Univ., Pennsylvania, PA, 1992.

[70] Guoliang, L., & Tao, Q. (2010, March). The complex adaptiveness of telecommunication networks and urban functions. In Advanced Computer Control (ICACC), 2010 2nd International Conference on (Vol. 1, pp. 107-110). IEEE.

[71] Ott, R. L., & Longnecker, M. T. (2015). An introduction to statistical methods and data analysis. Nelson Education.

[72] Langer, A. M., & Yorks, L. (2013). Strategic IT: Best practices for managers and executives. John Wiley & Sons.

# Appendices

# Appendix A – List of Additional Incidents

Below is a list of the incidents used in this dissertation

| # | List of Incidents |
|---|---|
| 1 | Cable Stolen |
| 2 | Bandwidth Saturation |
| 3 | Operative System Problem |
| 4 | LAN Customer Problem |
| 5 | Customer Energy Problem |
| 6 | Customer Telephony Problem |
| 7 | Customer Router Problem |
| 8 | Media Converter Broken |
| 9 | Customer Central Telephony Problem |
| 10 | Logical Interface Blockage |
| 11 | Problem with Carrier Routing |
| 12 | Customer Manipulation and/or Shutdown of Equipment |
| 13 | PCs and Application |
| 14 | Modem Problem |
| 15 | Modem Configuration Problem |
| 16 | Software Router Problem |
| 17 | Contractor Problem |
| 18 | Carrier Local Problem |
| 19 | Broken Power Converter |
| 20 | Voice Port Blocking |
| 21 | UTP Patch Cord Problem |

## Appendix B – Eclipse Java Class Main Descriptions

Figure B-1 shows the methods for the Telco Service class, which represents a specific service. ConnectionTelco establishes connectivity to the database. The Service() method initializes the class by establishing connectivity to the database. This method also has variables that include the “oldSrv” variable, which obtains information from an old service. This variable is mainly applicable when there is a change in the service. The Initialize method assigns values to the service database for this specific instance of the service. This information is critical because it will generate values related to the changeability and generation of requirements. The getID method generates a specific identifier for the instance of the service. This value is important because it links the values of services and therefore facilitates an analysis of the tendencies of the changeability values. The InitializeProductValues class obtains product information and translates it as classifier information.

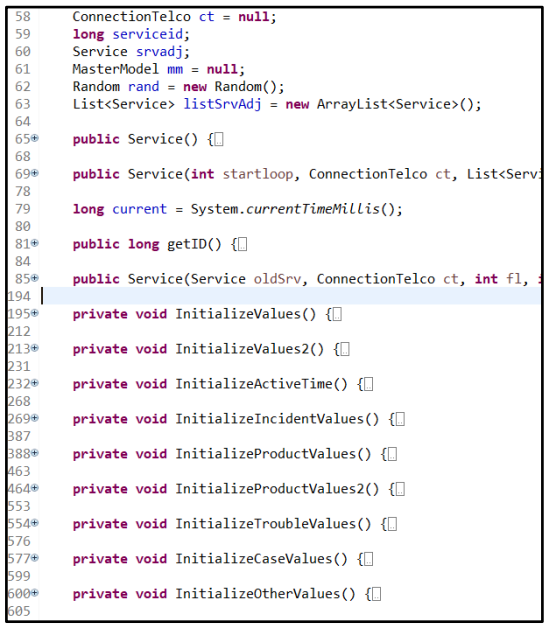


Figure B-1: Telco service java main methods

Figure B-2 shows the main classes of the classifier system. One of the main methods is adaptability. As previously explained in this dissertation, this method obtains information from an internal agent to adapt the general classifier system. This method generates information on new investments and also counts the number of services that the system has generated due to that investment. The runSingleExperiment_Simulation calls the TelcoServices and generates the new service instance. This method also relates the services that are part of a change and saves information on the flexibility, agility and robustness of the system. The runSingleExperiment_Service obtains all the information of the service and assesses the different formulas that are part of the classifier system. This class calls different methods and classes to assess the fitness, apply the GA and evolve the system.

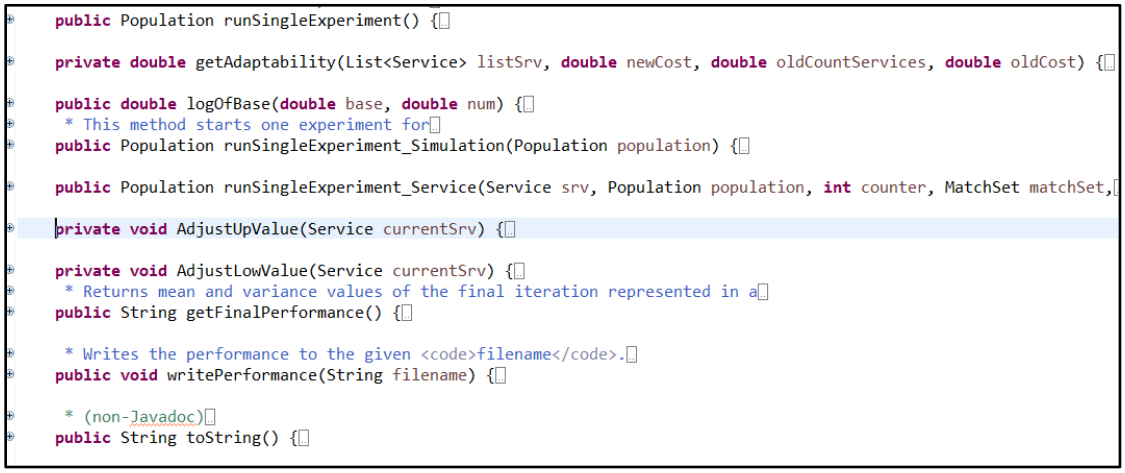


Figure B-2: Main classifier class