University of Crete
School of Sciences and Engineering
Computer Science Department

Simulating Value Networks in a competitive world using Vensim tool and suggesting competition strategies

by Konstantinos Tsikrikas

Master's Thesis

Heraklion, February 2011

University of Crete
School of Sciences and Engineering
Computer Science Department

Simulating Value Networks in a competitive world using Vensim tool and suggesting competition strategies

by Konstantinos Tsikrikas

*A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science*

Author: ____
Konstantinos Tsikrikas, Computer Science Department

Board of enquiry:

Supervisor: ____
Christos Nikolaou, Professor - University of Crete

Member: ____
Dimitrios Plexousakis, Professor – University of Crete

Member: ____
Kostantinos Maggoutis, Researcher – FORTH-ICS

Approved by: ____
Aggelos Bilas, Professor – University of Crete, Chairman of the Graduate Studies Committee

Simulating Value Networks in a competitive world using Vensim tool and suggesting competition strategies

Konstantinos Tsikrikas

Master's thesis

University of Crete – School of Sciences and Engineering
Computer Science Department

Abstract

In recent years, globalization of most economies and rapid technological progress lead most industries to standardization of their corporate operations in order to solve any coordination problem. The concept of standardization is to make a situation that each party can contribute in making a mutual profit though mutual decision making. Companies and industries organize their fundamental structure into service systems capitalizing on the advantage of standardization. The service systems consist of complex social and technical resources that work together and interact in order to create value are named as value systems. In this thesis, we present an e-business model that can be used in order to analyze the value of each partner taking into consideration the value that is created through development of relationships among partners. We build and simulate an application of this model using Vensim tool in order to study the value of each partner in terms of business competition. Taking into consideration the correlation between Game Theory and value networks we propose various scenarios that define strategies of specific partners which has as common goal the maximization of their value. Towards bridging the gap between Service Network Analysis & Prediction Tool (SNAPT) and Vensim tool we implement a plugin which integrates to SNAPT the functions of simulation and prediction of service network's value.

Προσομοίωση δικτύων αξιών σε ένα ανταγωνιστικό κόσμο χρησιμοποιώντας το εργαλείο Vensim και πρόταση στρατηγικών ανταγωνισμού

Κωνσταντίνος Τσικρικάς

Μεταπτυχιακή εργασία

Πανεπιστήμιο Κρήτης - Σχολή Θετικών & Τεχνολογικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Περίληψη

Τα τελευταία χρόνια, η παγκοσμιοποίηση των περισσότερων οικονομιών και η ταχεία τεχνολογική πρόοδος οδηγεί τις περισσότερες βιομηχανίες στην τυποποίηση των εταιρικών δραστηριοτήτων τους, προκειμένου να επιλύσουν οποιοδήποτε πρόβλημα συντονισμού. Η έννοια της τυποποίησης, είναι η δημιουργία μιας κατάστασης όπου κάθε μέλος της μπορεί να συνεισφέρει, προκειμένου για την αποκομιδή κοινού κέρδους μέσα από την κοινή λήψη αποφάσεων. Οι εταιρείες και οι βιομηχανίες οργανώνουν την βασική δομή τους σε συστήματα υπηρεσίας αξιοποιώντας το πλεονέκτημα της τυποποίησης. Τα συστήματα παροχής υπηρεσιών αποτελούνται από σύνθετους κοινωνικούς και τεχνικούς πόρους που συνεργάζονται και αλληλεπιδρούν με σκοπό τη δημιουργία αξίας ονομάζονται ως συστήματα αξιών. Στην παρούσα εργασία, παρουσιάζουμε ένα μοντέλο ηλεκτρονικού επιχειρείν που μπορεί να χρησιμοποιηθεί για την ανάλυση της αξίας του κάθε εταίρου, λαμβάνοντας υπόψη την αξία που δημιουργείται μέσω της ανάπτυξης των σχέσεων μεταξύ των εταίρων. Κατασκευάζουμε και δημιουργούμε μια εφαρμογή αυτού του μοντέλου χρησιμοποιώντας το εργαλείο Vensim για τη μελέτη της αξίας του κάθε εταίρου από την άποψη του ανταγωνισμού μεταξύ των επιχειρήσεων. Λαμβάνοντας υπόψη την αντιστοιχία μεταξύ της Θεωρίας Παιγνίων και των δικτύων αξίας προτείνουμε διάφορα σενάρια που καθορίζουν τις στρατηγικές των συγκεκριμένων εταίρων, η οποία έχει ως κοινό στόχο τη μεγιστοποίηση της αξίας τους. Προσπαθώντας να γεφύρωσουμε το χάσμα μεταξύ μιας γενικής πλατφόρμας για την σχεδίαση, ανάλυση, εποπτεία και βελτιστοποίηση Δικτύων Υπηρεσιών (SNAPT) και το εργαλείο Vensim υλοποιήσαμε ένα plugin το οποίο ενσωματώνει στο SNAPT τις λειτουργίες της προσομοίωσης και την πρόβλεψης της αξίας του δικτύου υπηρεσιών.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my supervisor, Professor Christos Nikolaou for his contribution, advices and support during the process of this thesis. I would also like to express my appreciation to the other members of my committee, Professor Dimitrios Plexousakis and Kostas Maggoutis for their valuable comments and feedback.

Very special thanks to Professor Marina Bitsaki for her valuable contribution.

Finally, I would like to dedicate this work to my family and my close friends. Their love and endless support have earned my respect.

1. Introduction

Over the last decades due to globalization of regional economies, companies have obtained the ability to outsource the functions of production and distribution of their products worldwide. Simultaneously, technology is advanced towards enhancing the interoperability among companies worldwide. Within this context, industries have been led to standardization of their corporate operations in order to solve any coordination problem. The concept of standardization is to make a situation that each party can contribute in making a mutual profit though mutual decision making. The specific concept is implemented through services and this is the main cause of maintenance and development of services in worldwide economy. The execution of a service process requires an individual or organization which provides the services and one or more customers which consume the specific service. Companies and industries organize their fundamental structure into service systems capitalizing on the advantage of standardization.

Each service system has the ability to interact with other service systems. As a result, service systems form a network of service systems that can be named as service networks. The service systems consist of complex social and technical resources that work together and interact in order to create value are named as value systems. The networks of value systems that interact with other value systems are named value networks. There are various studies which have focalized in definition, creation and configuration of value networks [3], [4], [7], [9]. There are also other studies which propose methods in defining the relationships among the networks and emphasize on visualization of value networks. The authors of reference 18 propose an approach of a flow model that includes business entities and the exchanges among them. Each business entity creates value and may be a value sub-network by itself. The authors distinguish the exchanges into offerings and revenues. Offerings can be goods or services while revenues are money. This flow model is depicted by nodes and arcs. The main goal of the specific model is value computation analysis of each partner taking into account the value that is created by the relationships that are developed among partners through offerings and revenues. The authors of reference 18 propose an application of this model in the repair service system as part of the automotive industry (chapter 3).

The first contribution of this thesis is the proposition of a different approach of the previous e-business model. The difference is based on value computation of various partners. We also apply our e-business model to repair service system as part of the automotive industry. Repair service system comprises several partners that are collaborated in order to create value. We focus on partner that manufactures the brand-name cars and is named as Original-Equipment-Manufacturer (OEM). The primary goal of this thesis is to propose the strategies that OEM has to follow in order to maximize its value in repair service system.

The step towards achieving this goal is to build, simulate and analyze this repair service system. Thus, we choose Vensim tool because is a visual modeling tool that allows us to conceptualize, document, simulate, analyze, and optimize the specific example of our model. We have the ability to define various strategies that based on reality in order to achieve the maximization of OEM value in repair service system. A further thought about strategies and business competition leads us to enhance our model with one more repair service system. In this way, we define a second repair service system which includes its OEM. The main goal of this OEM is to maximize its value in repair service system that is belongs to. We define these repair services systems in our model in order to study the behavior of each OEM under competition. The behavior of partner is based on its strategy given that its competitor has a specific strategy at each time. This situation is similar to any game of Game Theory [20]. It is a study of applied mathematics that attempts to study the strategic behavior in situations in which each partner attempt to make the most successful decision given that its competitor made a specific decision. There is a solution concept of a game of Game Theory in which no player has an incentive to deviate unilaterally from its choice given that its opponent have already made a specific decision. This solution concept is a steady state which in named Nash equilibrium [23]. Taking into consideration we construct various scenarios based on reality in which each OEM follows a strategy given that its competitor chooses a specific strategy. The second contribution of this thesis is to propose scenarios that have Nash equilibrium. In these scenarios, the strategy of each OEM leads to maximization of its value under conditions of competition.

The last part of this thesis concerns with SNAPT[1] which is implemented by TSL[2]. The Service Network Analysis & Prediction Tool (SNAPT) is a Model-Driven Architecture

---

[1] Service Networks Analysis & Prediction Tool
[2] Transformation Services Laboratory Website: http://www.tsl.gr/

(MDA) platform that enables modeling and analysis of Service Networks. SNAPT was developed in a modular way, to host future task-specific, extended Service Network models. SNAPT incorporates the Service Network and a KPI model, facilitates cost - revenue analysis and includes an extensive feature set to achieve its partial goals. The core Service Network model could be expanded in many ways to facilitate both analysis and implementation of complex Services. SNAPT should be in the center of these emerging methodologies. SNAPT could obtain additional features such as value analysis and prediction through connection with simulation platforms or tools. Towards connecting SNAPT with a simulation tool we facilitate a connection between SNAPT and Vensim tool. The connection between these tools is made capitalizing the fact that SNAPT is implemented on the concept of plugins. They are independent components that provide additional functionality to SNAPT. The last contribution of this thesis is the implementation of a plugin in SNAPT that provides the functions of value analysis and prediction to any service network model of SNAPT. The name of the specific plugin is SNAPT2Vensim because its functionality based on transformation of a service network model of SNAPT into model of system dynamics which can be visualized and simulated by Vensim tool.

The rest of the thesis is organized as follows. In chapter 2 we discuss the background theory and approaches about value networks and correlation between Game Theory and value networks. In chapter 3 we present the specifications of two simulation tools. First, we discuss about iThink tool and then we present Vensim tool in more detail. In chapter 4 we discuss about e-business model of reference 18 and we present our e-business model. We discuss about Nash equilibrium and reaction functions. In addition, we introduce the suggested competition scenarios of our model that has Nash equilibrium. In chapter 5 we present the implementation of SNAPT2Vensim in order to achieve a connection between SNAPT and Vensim. Chapter 6 includes the conclusions of the thesis and discuss about future work.

2. Background

2.1 Value Networks

Over last twenty years, companies have abandoned traditional way of transaction that is goods-oriented and follow a service-oriented one. This transformation is based on the evolution of service science starting from *value chain*. Is also known as value chain analysis and is a concept that popularized and described by Michael Porter [2]. According to Michael Porter, the concept of *value chain* is based on definition and examination of value that is added from any activity that is involved in a business in order to create a competitive advantage. It is a view that includes various exchanges of firm by considering flow of goods or services that starts from raw materials and reaches consumption of them as a unit of analysis. However, the current environment is more complex and dynamic than before and requires inter-organizational exchanges. The concept of value chain has been substituted from the concept of a network view of high level exchanges among different organizations or firms. Value network model has been defined as an attempt to organize inter-organization relationships [6]. In addition, the concept of *Value Network* has been a great source of research for many researchers [7], [8], [9], [10]. As a result of research are various views of *value networks* that are conflicted and interesting for perspective analysis. Below we will discuss about the previous work that has already been done presenting Verna Allee model, e3-value model [11], c3-value model [11].

2.1.1 Verna Allee approach for defining and modeling Value Networks

According to Verna Allee[3], a value network is a group of relationships that has a web oriented schema. The participants of this group will be individuals, organizations or groups. Each participant is connected to at least one other participant and generates a large network of relationships. As a result of each participant connection is not only a

[3] http://www.vernaallee.com/

network of relationships, but also a generation of economic value or other benefits via transactions among two or more participants. Verna Allee approach includes two primary types of value or mediums of exchange as you can see in value network of figure 1. The first type of value exchange is named tangible and is depicted by solid lines in figure 1. Tangible are all exchanges that include goods or services, including all transactions involving contracts and invoices, return receipt of orders, request for proposals, confirmations or payment. In this type are also included products that represent knowledge or services and generate revenue. The second type of value exchange is named intangible and is depicted by dashed lines in figure 1. Intangible exchanges are separated into two primary subcategories: benefits, knowledge. On the one hand, intangible knowledge exchanges include strategic information, planning knowledge, process knowledge, technical know-how, collaborative design, policy development, etc., which flow around and support the core product and service value chain [13]. On the other hand, intangible benefits are advantages of favors that can be offered from one person to another. An intangible exchange may include knowledge offering or experience offering. However, the approach of role of knowledge in exchanges of a network that includes business relationship is negligible. An example of an intangible exchange may be inspired from politics. Someone may offer political support or someone may offer himself as a volunteer and spend his time to take part in a charitable organization. Another example of intangible exchange is technical advice that may be offered by a firm to another one.

The main purpose of a value network is to generate economic success which represents value (benefits) for its participants. In figure 1, participants are depicted as a circle. Each participant takes part in a value network by using his product or knowledge into tangible or intangible exchanges that generates value for others members of value network.

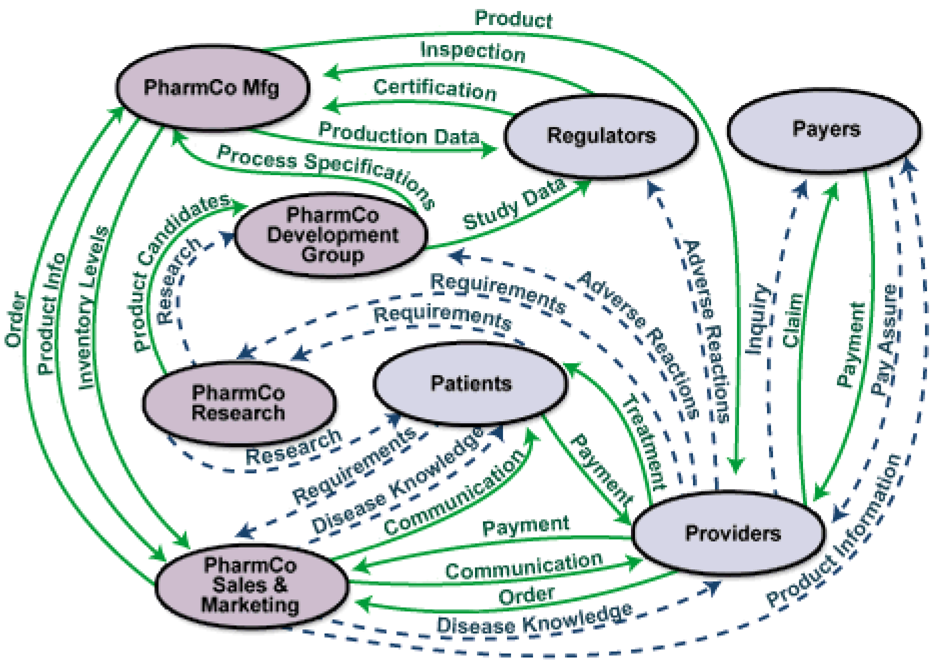


Figure 1. A value network modeled by Verna Allee

According to Verna Allee, a successful network comprises participants that offer and receive value taking part into any type of value exchange. The success of a value network is depends on success of each participant that takes part in it. Usually, we have only focused on the tangible business transactions without having in mind the contribution of intangible business transaction in a value network. Each intangible business transaction such as benefit or knowledge contributes in building stable and successful relationships among participants in a value network. This is very important issue because does not exist any value network which has not intangible transactions. Thus, the appreciation of intangible business transactions gives us the chance of building a value network which makes value from every kind of transaction. In addition, the way of making value from every transaction either tangible either intangible, make the results of every transactions being exploited by value network as it is shown in figure 1.

2.2.1 e3-value approach for defining and modeling Value Networks

ARA model has a name that is consisted of the first of three words: Actor, Resources and Activities. This model in industrial networks builds on this opinion and describes a conceptual model for the analysis of interactions of company. Actors could be companies or firms or individuals that are perform various activities using a set of resources. So, e3-value methodology is based on ARA [13] model view. In e3-value network, Actor is an entity that has its economical independency and it represents a company, or an organization, or a customer. The object of exchanges among the Actors is named *Value object*. A *value object* may be a good, or a service or money which means that at least one Actor could get economic value from any exchange. Each exchange is done through *Value Port* which is a connection point between the Actor and the outside world. A Value port connects to a *Value exchanged* creating a pipe which could trade a value object in the future. *A value interface* is a group of value ports. If an Actor has a motivation of a potential profit then it will perform a *value activity*. So, e3-value is a modeling framework that it has taken part into various modeling engagements [10].

2.3.1 c3-value approach for defining and modeling a Value Network

According to [18], c3-value model [17] is based on e3-value modeling techniques focusing on strategic analysis. The modeling approach of c3-value defines a strategic analysis that based on three factors: customer, capability, and competition in order to emphasize the meaning of competition that is defined by *VRIN* characteristics. VRIN is a claim of a resource-based view of the firm (RBV) that the competitive advantage is obtained by strategic resources that are valuable, rare, inimitable, and non-substitutable [17]. Actually, c3-value model defines a distinguished version of value objects which is separated in two categories: the primary category is referred to value objects that focus on intended business of an Actor. The secondary category is referred to object values that strengthen the value that delivered by the primary value object. Nevertheless, c3-value modeling approach it is a powerful technique in strategic analysis.

2.2 Game Theory and Value Networks

Game Theory is a study of applied mathematics that tries to capture the behavior of two or more individuals that are interdependent and interact between each other [20] [21] [22]. The capture of behavior rely on three dimensions: predict, explain and prescribe. The study of Game Theory comprises situations that every individual have to anticipate what decision the others will do from its own decision. Every individual has its own strategy and interacts with the other units that belong to the same situation. Anyone could think that Game *Theory* is related to *games* but it is important to separate term named *Games* from term named *Theory* because *games* are taxonomy of strategic situations in social science depicted as a periodic table of elements in chemistry. So, an example of a situation in Game Theory might be a game that involves one individual or a group of individuals that have a common purpose, to be a winner. Each game has at least two players and each player has its own strategy in order to win a game. In each turn of a game each player has a number of choices. The decision-making depends on the choices of other players. Every player's strategy focuses on making a decision assuming that will be better for him when he knows the decision of his opponents. If players are human beings then each of their behaviors is unpredictable and is usually based on introspection and guesses rather than careful observation of how people actually play in game. A great example of a game is auctions. The bidders assume the rules of each as given auction and the way in which value an object, such as an oil lease or a painting. Certainly, there are cases or situations where individuals may be not people but organizations or nations. In these cases, the decision-making analysis that is analyzed of Game Theory is more complex and depends on the interactions that the participants have.

Game Theory has been used in economics. Market is an activity that forms part of the economy. Applying the previous example about a game and the players that are taking part into it, we consider that the game is the whole market and the players are providers and consumers. Obviously according to rules of market there are interactions among providers and consumers. The behavior of consumers will affect the behavior of providers. Nevertheless, the behavior of providers will surely affect the behavior of consumers. Due to the rules of market each individual, as provider or as consumer is acting based on his strategy in order to success. Usually, a provider aims to raise the price of products that he sells, while a consumer tries to find the lowest price in the market about the product which he is willing to buy. Thus, consumers have different target from providers and this is the first point of their interactions. If a provider decides to increase the price of his products, then the consumers will change their strategic behavior and decide to start searching another provider who may sell his product at a lower price. So, provider may change his strategic behavior and decrease the price of products in order to raise the sales and simultaneously his profits. By this way, the strategic behavior of consumers may be affected by the change of price and consumers decide to buy products from the previous provider. This is a situation that comprises multiple interactions among consumers, providers, other participants of the market which strategic behavior can be analyzed by Game Theory. In addition, the cases of monopolist or oligopolies are very interesting and can also be analyzed by Game Theory. According to *Aumann* in 1987 "*game theory is a sort of umbrella or 'unified field' theory for the rational side of social science, where 'social' is interpreted broadly, to include human as well as non-human players (computers, animals, plants)*". So, a more abstract thought can lead us to realize that a market except from an activity that forms a part of economy is a social network. Market has participants as we mentioned before that have different financial or social status and interact between each other. Usually, the interactions between providers and consumers are exchanges of any type of goods, services and information. Providers offer a good or a service or information to consumers and earn revenue. Consumers buy a good or a service or information from providers. Both of these interactions create economic value for the participants that take place into them. Nevertheless, this social network can be named value network because has the features that we have already describe about value network above. If Game Theory can analyze the strategic behavior of a situation that is represented by a network that belongs to social science, then Game Theory will be able to analyze the strategic behavior of participants of a value network. The dependencies that exist in a value network determine the interactions among the participant of the specific value network. Let us use an example of a value network in order to describe how game theory can be applied on value networks. Thus, a simple example is about firms that sells books. Any consumer can order a book online or from any cooperating shop. If any consumer buys a book of a firm then he will give economic value at this interaction between consumer and provider. If the sales of this firm are increased then the strategic behavior of the competitor firm will be affected. Potentially, the competitor firms will change his strategic behavior and decrease prices of its products or change the way of business marketing. Then, such a change may affect the strategic behavior of a portion of consumers who may be flattered by lower prices than before and be willing to buy from the specific firm. However, there are more participants that are involved in these interactions such as a transportation firm which delivers the order from any firm to the customer. The strategic behavior of a transportation firm is also affected by any changes of the strategic behavior of every participant that interacts with it. We can also consider the owners of cooperating shops that interact with consumers. They are also affected by any changes in the market. Game theory analyzes strategic behaviors about all participants that we presented in this example. Obviously, firms are competitive, providers are competitive, and transportation firms are competitive. Actually, game theory is attempting to analyze and predict the complexity of the decision that will be made by participants due to competition existence.

3. Value Network modeling & simulating tools

In chapter 2, we referred to definition and modeling of value networks that has already been done from specific researchers. An e-business modeling approach is presented by later researching studies [3] [10] [11] [13]. This approach includes a model with specified relations and e-business scenarios apart from defining value. Obviously, the evolution of a simple value network definition is e-business models that involve relationships among participants (economic entities or business units) and e-business scenarios [10]. In reference 17, the authors present an approach of a flow model for offerings and revenues with economic entities (business units) as nodes. They consider that each node (economic unit) is a fundamental part of value creation and simultaneously may be a sub-value network (or sub-service system) by itself. The connection between economic entities will be done through arcs that represent offerings and revenues. The authors of reference 18 applied their e-business model to repair service system as a fundamental part of the automotive industry. The primary contribution of this thesis is the proposition of our e-business which that is based on previous model (see chapter 4). The second contribution of this thesis is the designing, building and simulation of our model in order to examine the behavior of each partner. We needed a software tool that has the appropriate features in order to build, simulate and analyze the behavior of our model over time. We studied two software tools that have almost these features. The first software tool is named iThink[4] (Modeling &

[4] http://www.iseesystems.com/softwares/Business/ithinkSoftware.aspx

Simulation Software for Business) and the second is named Vensim[5] (Ventana Simulations). We preferred to use the second software tool rather than the first one. The first reason is that the first have been already used in reference [23] and the second reason is that Vensim tool provides more features in simulating and predicting than iThink tool. Below we will present iThink tool in brief and then Vensim tool in more detail.

3.1. iThink tool (Modeling & Simulation Software for Business)

IThink tool provides a convenient way of making decisions and leading to business improvement without any risk. The concept of iThink is based on creating an easy "guide" for a business team in creating models that are able to simulate business process and scenarios. In addition, it points out any potential conflict and simultaneously offers opportunity to fix unpredictable problems. It is a software tool that is based on Systems Thinking and its models may show to user outcomes that will be unpleasant, or costly if they are real. Thus, it is a helpful tool in indentifying crucial points of business performance without losing any unpredictable case. The power of iThink tool is based on connecting interdependencies among processes and problems. It focuses on evaluate the different components of each model and manage to keep the whole interactions enable. So, it makes easier to understand the variables that impact your business from your business team and enables teams to work together ensuring that any potential decisions will be completely implemented.

Comparing iThink with traditional ways of business planning such as spreadsheets and other linear approaches, it provides completed views of the entire system showing the operations in more detail. The basis of these views is dynamic modeling and Systems Thinking, while any point of a system is included and can be examined in any order. In this way, you do not make decisions that may have negative consequences on processes that seem to be unrelated.

[5] http://www.Vensim.com/

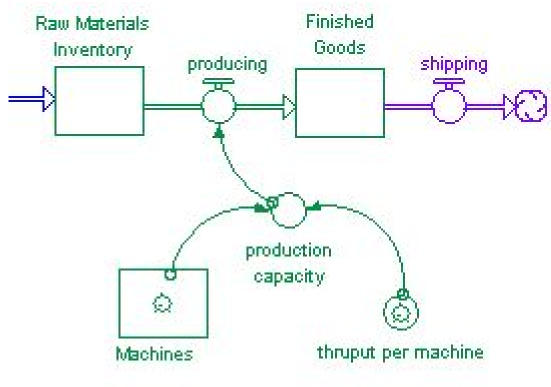


Figure 2. An example of a model in iThink tool

In figure 2 is shown an example of a model designed in iThink tool. Is depicted a flow of operations that are modeled in constructing a business process successfully. There are various features that iThink provides in mapping or modeling. It provides a graphical interface which is icon-based that makes simple the model construction. In addition, Stock and Flow diagrams which supports the common language of Systems Thinking and provides a sense of prediction. It provides *Causal Loop Diagrams* which presents the whole causal relationships and enhanced stock types that enable discrete and continuous processes with support for queues, ovens and enhanced conveyors. It also offers model equations which are automatically generated creating a great advantage to user. A special category is the *Built-functions* which facilitates operations that have mathematical, statistical and logical type. The implementation of *Arrays* provides an advantage of repetition of any model structure. The existence of *Modules* supports a more complex level of model construction. Any process that is represented in a model can be simulated in iThink tool by "run" application. The simulation part provides sensitivity analysis of the process and reveals the crucial points and optimal conditions. The results of any simulation are presented as graphs, tables, animations, QuickTime movies, and files. Certainly, exists a way of importing dynamic data into a model through Microsoft Excel files or exporting dynamic data from iThink tool writing into Microsoft excel files.

3.2. Vensim tool (Ventana Simulations)

Vensim[6] is the software tool that we base the construction of our model. Its name is inspired from the words: ***Ven***tana and ***Sim***ulations. Vensim is a visual modeling tool that allows the user to conceptualize, document, simulate, analyze and optimize models of dynamic systems. It provides a flexible and convenient way of building simulation models from causal loop or stock and flow diagrams [24]. Because of this we can say that Vensim tool uses the System Dynamics [25] which is an approach to understanding the behavior of complex systems over time using feedback loops and stocks and flows. Actually this is the main difference between System Dynamics and the other approaches to studying complex systems. Briefly, the modeling in Vensim tool is based on connecting variables with arrows and creating relationships among these system variables and building flows. An Equation Editor contributes in successful construction of a complete simulation model. It also provides features for analyzing the result of simulations. If a built model that have been made by a user is ready for simulation, then Vensim tool will provide to user all the appropriate features to explore the behavior of the model during the time.

3.2.1 Main Features

The interface that Vensim uses seems to be as a workbench and a set of tools. Workbench is the main window of Vensim, which includes the Title Bar, the Menu, the Toolbar, and the Analysis tools, as it is shown in figure 3. If a model is opened in Vensim, then the Sketch tools and the Status Bar will also appear.

The Title Bar shows two significant items: the model that is open (in figure 3: pop.mdl) and the Workbench Variable (in figure 3: Population). The second item is the variable in the model that the user has selected and for which user wants more information, for example the dynamic behavior of that variable. The variable selection can be done by clicking on variable's name at Workbench. The Menu Bar includes the buttons that provide the appropriate options about setting a model. Below Menu Bar, we find Main

[6] http://www.Vensim.com/

Toolbar menu that provides buttons which represent common operations such as save, open, and print. Main Toolbar menu apart from commonly used buttons provides buttons that represents simulation features (set up a simulation, choose a runname, Simulate, Synthesim, Reality Check). Below Main Toolbar we find the Sketch Tools menu. This menu includes menu items that are related to Sketch Drawing Area. Each menu item of this toolset provides a drawing feature to user. Some of them represent a type of model variable, one of them a drawing arrow. The Sketch Drawing Area is actually a panel where user can draw and design a model. At the left bottom side of Vensim User Interface we find Analysis Tools bar which consists of menu items that are used in analysis of a model. Analysis always follows simulation of a model and any analysis menu item can be used after running of model simulation. The bottom bar is named Status Bar and includes buttons for changing the status of selected objects such as the variable color, the box color, surround shape, arrow color, arrow width etc.

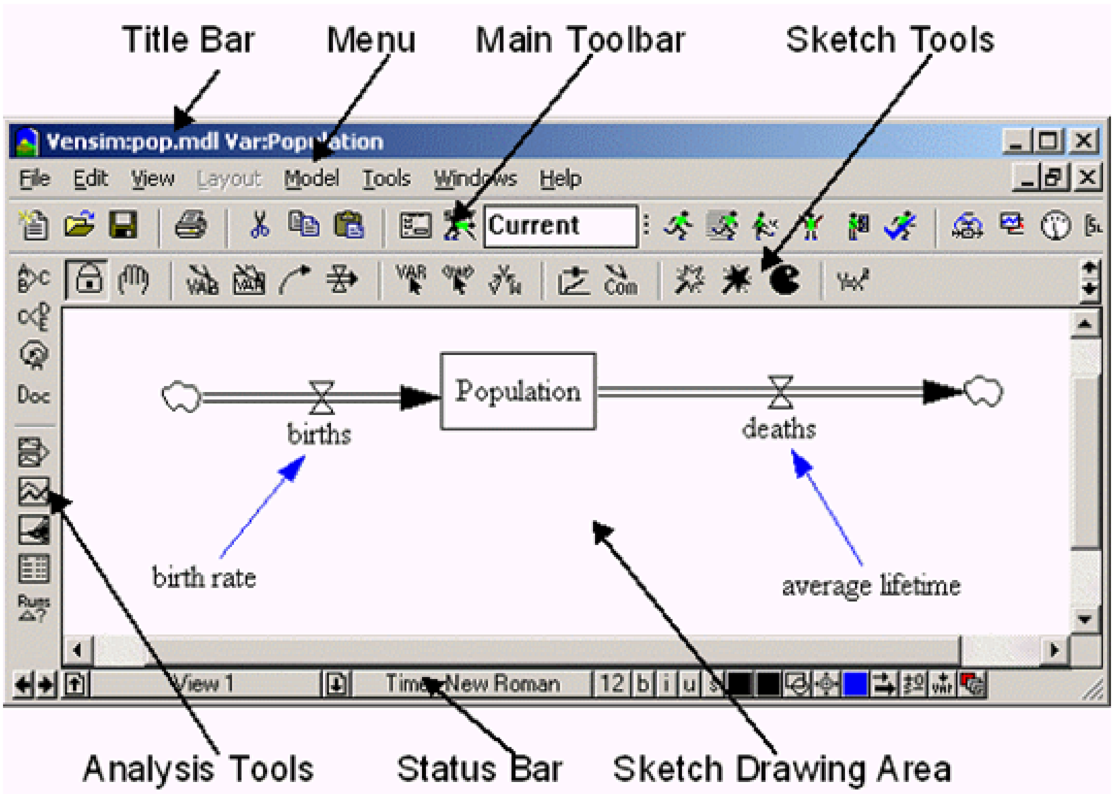


Figure 3. The Vensim User Interface

3.2.2 Designing and building a model in Vensim tool

At the beginning of building a model, we have to set the time bounds of a model. We refer to setting such as *initial time*, *final time*, *time step*, and *units for time* that determine the bounds of simulation. For example if we want to run a simulation of a model for one year, then we will set initial time equal to zero, final time equal to twelve, step time equal to one and units for time equal to month. So, we can simulate a model during a year and each time step represents a month. Such an example is shown in figure 4.

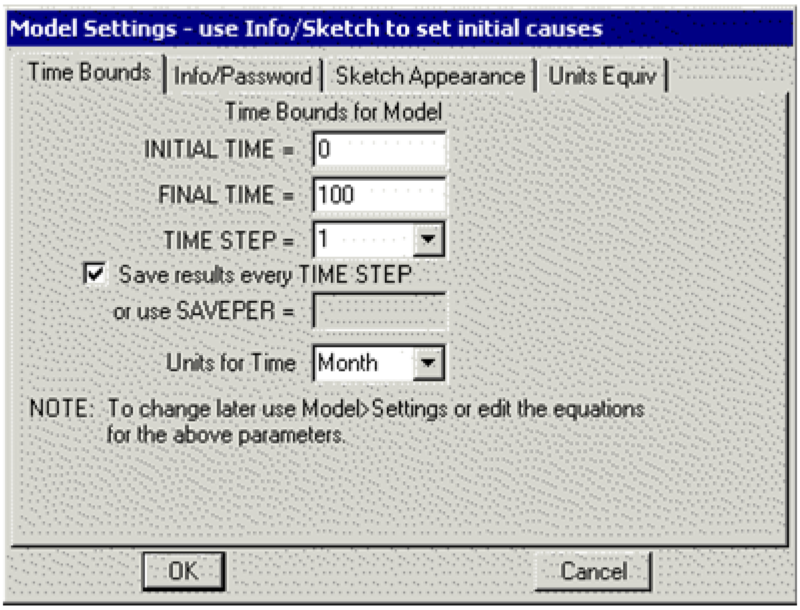


Figure 4. Model settings dialog box

We continue with the model building in Sketch Drawing Area as is depicted in figure 3. Generally, a model in Vensim consists of variables and arrows. We select a variable taking into account that a variable is and does can be determined by the way is defined or used. There are no forward declarations of variables and no prefixes that indicate a type of a variable. The selection of each variable type is related to what we want each variable to do in the model. For example, if we want a variable which value remains stable during the simulation time, then we will select a ***constant*** variable. The value of each variable is computed through its equation (see below). Apart from constant variables, there are ***auxiliary*** variables which are dynamic and are computed from other variables at a given time. Because of this any auxiliary variable has an expression that involves other variables in its equation. Auxiliary is the most numerous variable type. The equation of an auxiliary variable consists of an expression that involves other variables. Another type of variable is ***Data***. The variables of this type have values that change over time, but are independent from other variables. They use an empty equation to indicate the independence from other variables. The next type of variables is ***Initial***. It is almost the same as constant apart from that it is the result of the combination of different variables at initialization time. The most dynamic type of variable is ***Level***. As you can guess from its name represents levels. These variables have *INTEG* equation because the value of a *Level* variable at a specific time is known as a result of integration from previous times. Another type of variable is ***Lookup***. It contains variables that are nonlinear functions with linear parameters (x- and y-axis values). Variables of this type are defined in equations beginning with a left parenthesis (and ending with a right parenthesis). Below is figure 5 which shows the equation editor of a variable. The circle indicates both sides of the equation and the arrow at the bottom indicates the box that defines the type of variable. Except from the value of the correlated variables there are several already built functions that we can use in equations. Some of them are: *RANDOM UNIFORM, RANDOM NORMAL, IF THEN ELSE, ABS* etc.

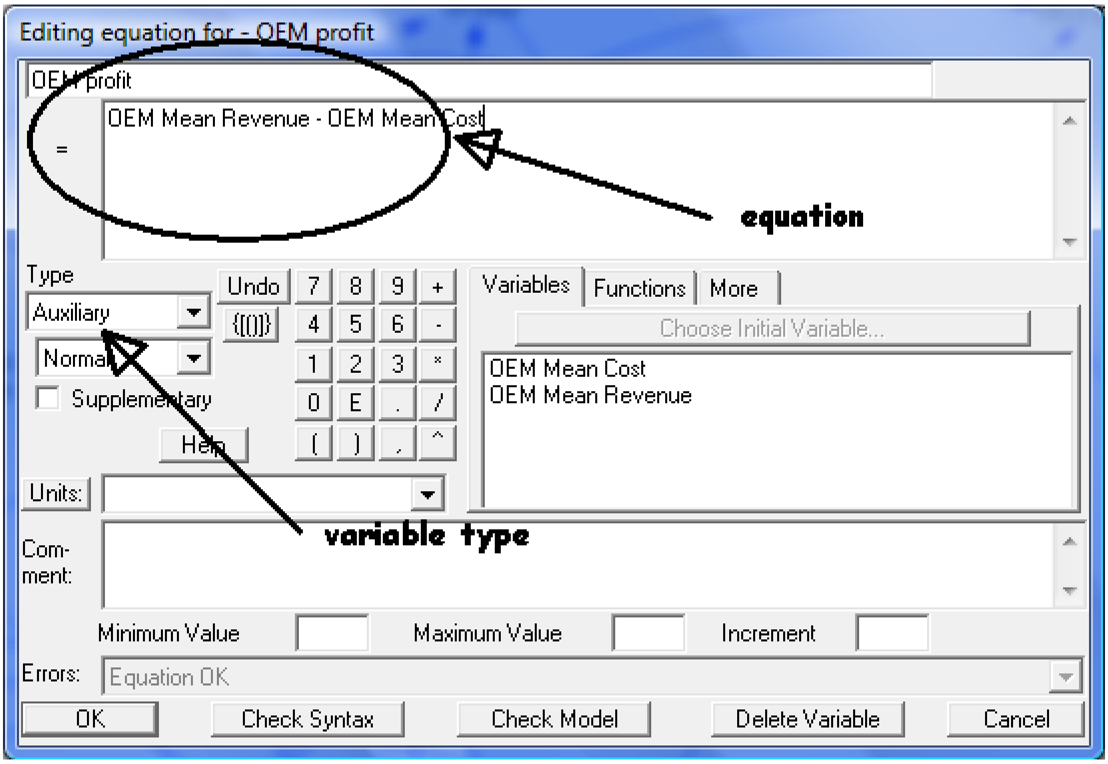


Figure 5. Equation Editor Dialog box

The interactions among variables are defined by arrows. They are arrows with a single arc. Actually, the interactions among variables are dependencies. An arrow that begins from a variable and reaches another one defines that the equation of second variable must contain the value of the first variable. Using arrows we can define the dependencies among variables and it will work as we want. Apart from single arrows Sketch toolset contains one more type of arrow which named ***Rate***. It is an arrow which can only begin from a level variable and reach to level variable also. Simultaneously, the variable that is reached by this arrow must contain the value of the first level variable in its equation. In practice, Rate arrow shows the flow from first variable to second and actually marks their dependency. An example of such an arrow is shown in figure 6 where the ***Rate*** arrow represents the incoming customers of the market. The level variables that are connected by this arrow are level variables. The "source" variable is named *Potential* Customers and represents the future customers. The "destination" variable which is named *Customers* represents the current customers of a firm. Nevertheless, the rate arrow is showing the flow of people from potential customers to customers.

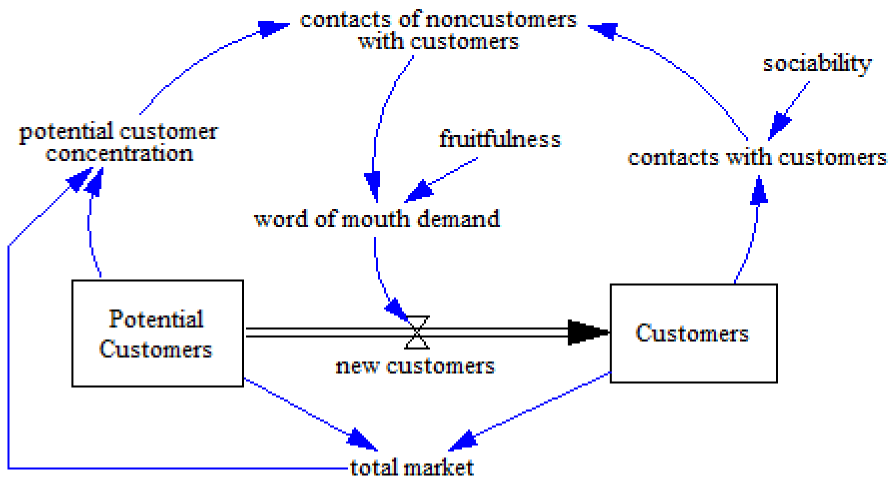


Figure 6. Example of a ***Rate*** arrow

At the beginning of this section we referred to feedback and loops. Let us do a further presentation about these features. We begin with *Causal Loops* which will be described using an example. We assume that the example of figure 7 is generated from reality. The variables of this flow diagram are auxiliaries except from *Average Sales* which is a level variable. As you can see all variables apart from *time to average sales* are connected with curved arrows creating a circle or a loop. In simulation time, if *sales* change as soon as *price* does, *price* changes as soon as *indicated price* does, and *indicated price* changes as soon as *sales* does, then *sales* effectively depends on itself. But such a system is not casual. However, this system can be causal if we recognize that the variable *indicated price* is not adjusted based on current *sales*, but rather on average *sales* over the previous days, weeks, months and years. The box that contains the variable *Average Sales* is a level variable which means that its value at a particular time is known as a result of integration from previous times. *Feedback loops* are almost the same as *Causal Loops* because feedback loops requires the existence of at least one level variable.

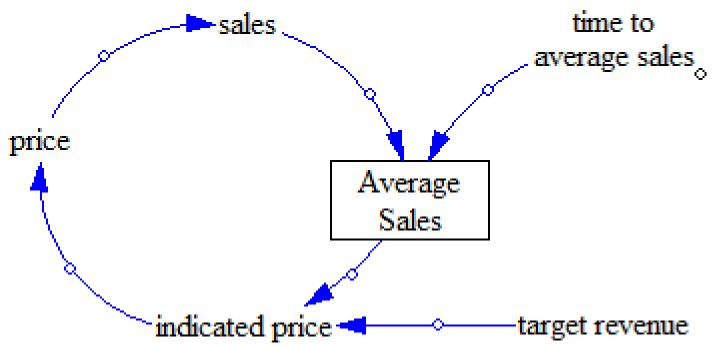


Figure 7. An example of a Causal Loop

3.2.3 Model Simulation & Analysis

A well-specified and complete model in Vensim is ready for simulation. That means there are no errors in the specific model. The simulation process starts when we push the appropriate button and changes the appearance of the model diagram. In figure 8 is depicted the diagram in simulation process of model in figure 6 below. As you can see, in each variable has been drawn a wavy blue line that represents the changes of value over simulation time. The definition of simulation time becomes from the dialog box in figure 4 above. Through simulation process we can draw several conclusions about the behavior of each variable and the entire model at extension. It is a significant advantage because we can predict what will happen in the future if we follow a specific strategy. In simulation process we can also try various experiments given that we have the same model even if we did initializations. As you can see in figure 8, two variables that are named *fruitfulness* and *sociability* are both constant variables. Because of this we can change the value of them in simulation process and see how it affects the behavior of the other variables automatically. So, we can predict if our plan will be successful using the simulation process.

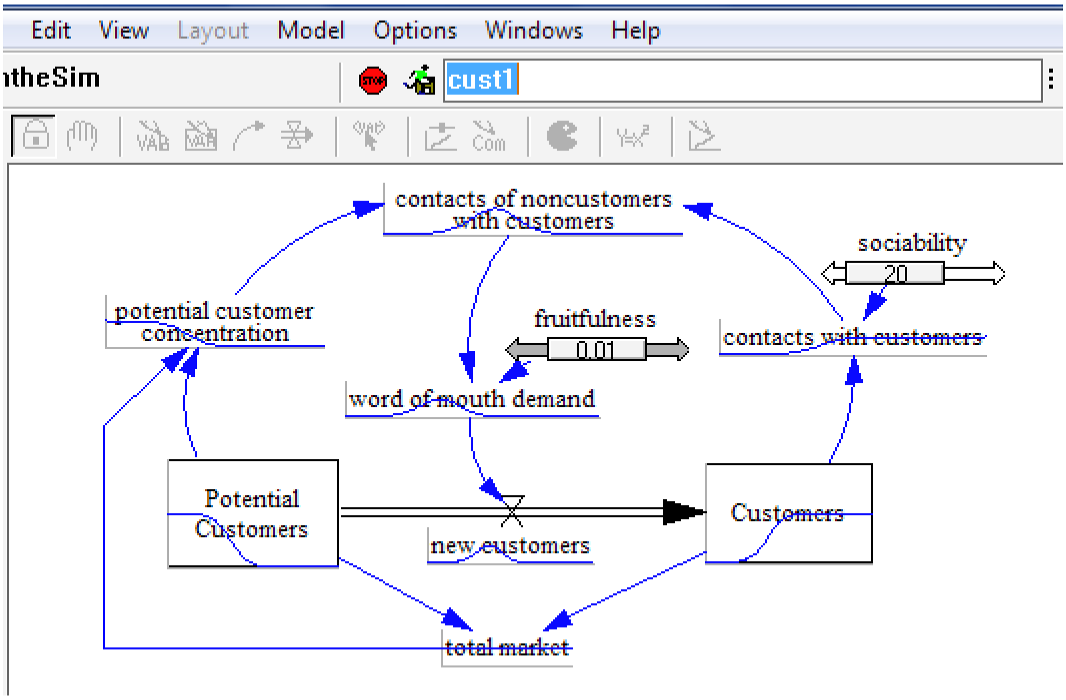


Figure 8. Example of a simulation process

We can also have a more detailed analysis of diagram in simulation time. This can be using analysis tools. We presented the analysis tool in figure 3 before. The main concept of analysis tools is analytical representation of behavior of each variable. This representation can be graphical, or numerical, vertical or horizontal. An additional feature is a comparison among the graphs of two of more variables. In figure 9 is shown a graphical representation of behavior of variable *Customers*. The x-axis represents the number of customers while y-axis represents the simulation time which has as unit the month. Through the observation of graph we can draw a conclusion about behavior of variable *Customers*.

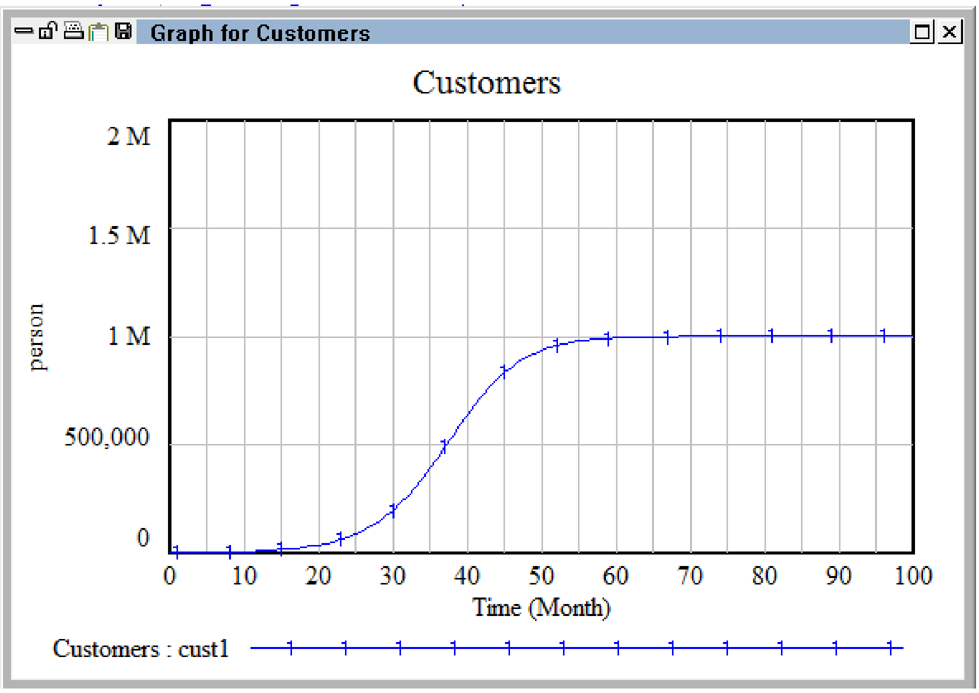


Figure 9. Graphical representation in simulation process

We can also examine the value of this variable in vertical or horizontal style over simulation time, as is shown in figure 10. The upper window is showing the values of variables: *Customers* and *Potential Customers* in horizontal provision. This way is convenient in comparing the values of a variable with the values of another in order to examine the behavior of the model.

Table

Time (Month)	0	1	2	3
"Customers" Runs:	cust1			
Customers	1000	1199.8	1439.47	1726.95
"Potential Customers" Runs:	cust1			
Potential Customers	1e+006	999800	999561	999273

Table Time Down

Time (Month	"Customers"	Customers	"Potential	Potential Customers
0	Runs:	1000	Customers"	1e+006
1	cust1	1199.8	Runs:	999800
2		1439.47	cust1	999561
3		1726.95		999273
4		2071.75		998928
5		2485.24		998515

Figure 10. A Table that includes values over simulation time and below a Table Time Down with time running down

The lower window is almost the same as the upper. The basic difference is based on the form of value depiction. In Table Time Down the values have horizontal provision which may help more or less in examination. The results of each simulation process will not be lost after the end of simulation process. At the beginning of each simulation process is generated a file that contains the results of this process. It is a dataset file of Vensim and every user can set its name properly. Thus, we can run several times the same dataset without building again the same model.

4. Designing & Building e-business model using Vensim tool

The secondary contribution of this thesis which is presented in this section, deals with the design of our model that is based on e-business model in reference 18 into Vensim tool. This e-business model is applied in an example of a repair service system as part of an automotive industry. We propose our e-business model which is based on fundamental structures of previous e-business model but it is different about the value computation of specific partners. We applied our model in the same example as e-business model in [18] but we use different equations of value computing in the value chain, as we will discuss below. However, we need to go further and examine the behavior of the entire repair service system over time. Most of all we need to know how we can maximize the value of car manufacturer in the value chain. That means which strategy must be followed by car manufacturer in order to make the maximum profit. So, we decided to use Vensim tool in order to construct our model and then examine its behavior over time. By this way we could find the most profitable strategy about car manufacturer. We were not satisfied because of lack of competition. Thus, we decided to extend our model and comprise a second repair service system that belongs to a new car manufacturer. We assumed that these two car manufacturers are competitors and they have common customers. So, we wanted to find the most profitable strategy of each car manufacturer at the same simulation time. Due to the existence of competition, the strategy that is followed by each car manufacturer affects the strategy of its competitor. We assumed that this competition could be an application of Game Theory [20]. So, we constructed various scenarios that represent strategic games. Each strategic game includes a specific strategy for each firm depending on the strategy of its competitor at the specific time. The secondary contribution of this thesis is that through these strategic games we tried to find they have Nash Equilibrium [26] among these two firms of our model. In this chapter we also present these scenarios and discuss about each of them.

4.1 Defining e-Business model

In this section, we describe the e-business model of [18] which is the basis of our model, including its formal structure and its fundamental concepts. The main structure of this e-business model is a flow model that is constituted from nodes and transfer objects. The nodes represent economic entities or economic units. The transfer objects represent economic offerings and revenues which are the object that flows through arrows among economic units. An economic unit or entity is the fundamental unit of value creation. The primary types of individuals are Sellers and Customers. In service system the sellers produce the raw materials and they sell them to customers who are the consumers of the offering. A fundamental unit of this flow model is time. The flows of model change with time and simultaneously values change with time. Thus, there is a definition of time units in order to capture the variation of several flows of the model. The value computation of each flow is captured as time progresses from time unit to time unit.

We assume that the exchanges are distinguished to tangible and intangible as is referred in [3]. In tangible exchanges there is revenue for the producer or seller and a cost for the buyer of consumer of a good. They have a physical structure. However, intangible exchanges play an important role in the creation of value in service systems. The structure of intangible exchanges is not physical and do not generate revenue directly such as a service. However, intangible exchanges improve the relationships among the involved partners through the affection of their properties such as satisfaction index that is a comfortable way to gauge the relationship value. These relationships are characterized by the rate of transactions. These are the roles of trust and risk that are imported into this e-business model. As the relationships in real life the relationships between business participants are based on trust and are depending on the rate of transactions. Thus, the success of such relationships is based on the past experience of their transactions. The trust level is high as soon as the level of a relationship is high. When the level of relationship is low then the level of risk is high and will be reduced when the level of relationship is improved. The level of risk indicates how unreliable the behavior of a partner to another one is. When the offerings are reliable then the level of trust is high. Because of this we can say that the level of trust confirms the reliability of offerings. But the maintenance of a certain level of relationship costs. This kind of costs is named relationship costs and comprises gifts, visits, free services. Certainly, each one of them must be modeled as an offering of provider to customer in order to improve the level of customer satisfaction index. Only a higher relationship cost in the current period could improve a lower level of customer satisfaction index.

4.2 Applying e-Business model in case study : Repair Service System

In this section we present the application of our e-business model in a case study of a repair service system. Original Equipment Manufacturer (OEM) is the name of car manufacturer of this case study. Each citizen who owns an OEM brand-name car arrives for repairs at any dealerships of the OEM. Afterwards, the technicians of the dealership diagnose the problem of the car that muse be repaired. Diagnosis may require an order of new parts from OEM and the appropriate repairs. The process of ordering parts may be time-consuming. This may happen if technicians need to ask for advice from expert technicians who are available from the OEM. Some advices may be information about warranty-covered parts, new parts, etc. So, the order will be after all of these processes. Each dealership has a parts manager who has access to parts catalog, checks first if the necessary parts are available locally. If they are not available locally, then checks the availability of OEM and supplier inventories. Finally, parts manager orders the necessary parts. Each parts manager has two choices in ordering parts. We also assume that dealer's technicians need an hour every day in performing searches for failure parts. We also assume that parts manager waste about one half hour in checking parts catalogs, or supplier inventories. The parts manager of each dealer has two possible choices of ordering parts. The first choice is the third-party suppliers (TPSs) and the second one is certified supply-chain suppliers (SCSs). The orders from certified supply-chain suppliers (SCSs) are made through OEM by dealer's parts manager. The costs of car repair are paid from OEM if service and parts are covered by warranty. If the services and parts are not covered by warranty, then they will be paid by the car owner. The dealer's technicians take technical advices from the expert technicians of OEM for free. OEM makes a collection of information about the new parts, warranty and failure symptoms of each month. OEM uses this information to generate new parts catalogs and mail them to its suppliers and dealers. Certainly the delays of all these processes extend the repair time. Due to this fact the level of customer satisfaction index is becoming lower. The number of dealer's customers will be reduced as long as the level of customer satisfaction index is low. Customer satisfaction index indicates how much satisfied are the customers of OEM and probably affects potential customers to buy or not brand-name cars of OEM. If this satisfaction index is low, then it is a negative effect on the overall service system. In opposite case, a high level of customer satisfaction index indicates increased sales for the brand name of OEM.

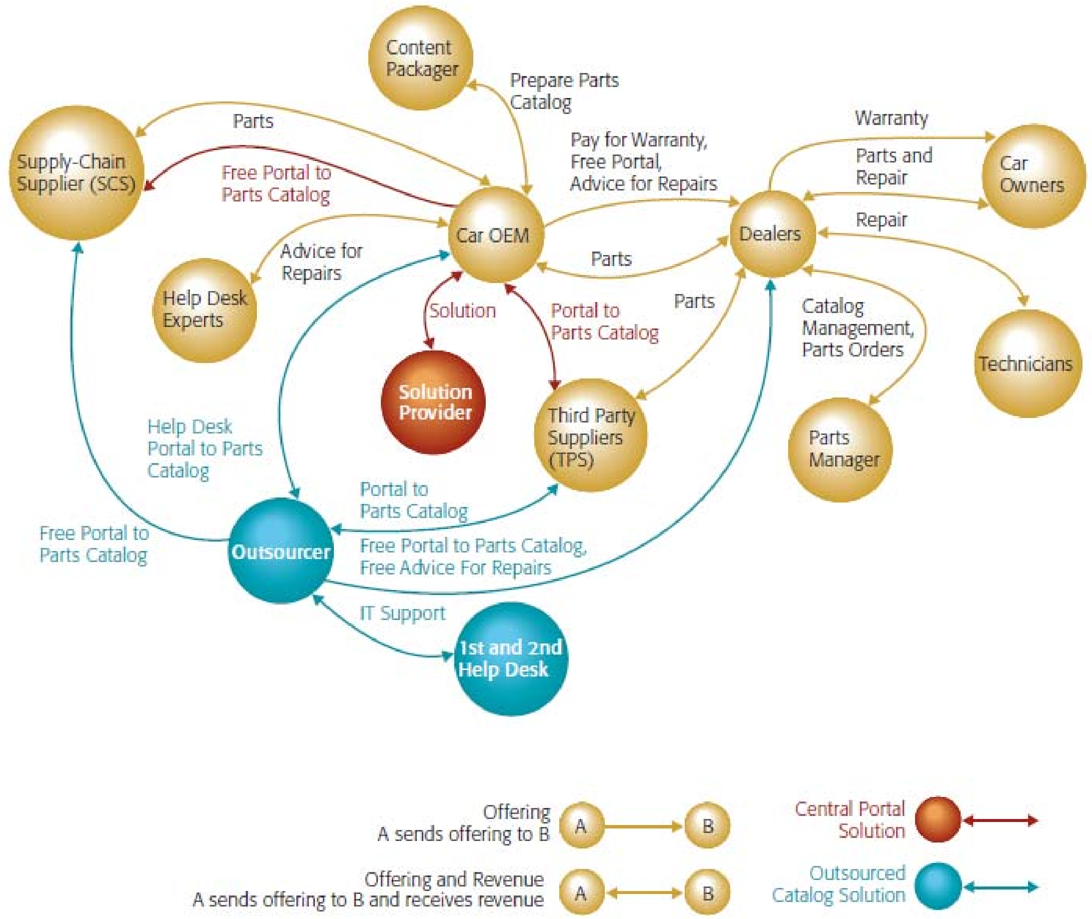


Figure 11. A flow of offerings among the various partners (shown as circles) in the repair service system

In figure 11, it is shown a flow model that includes several partners and the flows of offerings among the partners. The flows of offering are depicted by an arrow with a single arc which determines its direction. The flows of offering that includes revenue are depicted by an arrow that has two arcs with opposite directions. One of these arcs represents the flow of offering and the second one the flow of revenue. Each partner of the flow model is represented by a circle. Technicians, parts manager and the help desk are represented by a circle and we consider that they are economic entities. The reason of this consideration relies on that each of them is offering their labor as a service to the service system.

As we have already mentioned, each partner cooperate with others partners in order for the service system has successful performance. Certainly, the objective of the whole service system is to sell services and goods. So, apart from the definition of the formal structure of this e-business model is important to have an estimation of the value that each partner creates in participating to service system. This estimation will be quantitative and will be able to estimate how this value changes with time in combination with the changes that may be happen in the business process. Thus, in reference 18 is presented a value computation about the partners of the whole repair service system. The value computation of each partner comprises equations of value, revenue, costs, and satisfaction index including the factor of time.

The authors of reference 18 attempts to find a strategy that OEM can follow in order to increase the value of its service system. Certainly, there are several strategies that can OEM follows to accomplish this. One strategy may be related to increase of customer satisfaction in order to increase the number of customers and so the sales. Another strategy may be related to reduction of repairs that are covered by warranty and improve the quality of car parts. Thus, if the parts have higher quality than before, then the car repairs will be reduced. The basic form of repair service system contains a content packager which prepares the parts catalogs and mails them to suppliers and dealers. So, the authors suggest a transformation of repair service system which includes a solution provider which contributes in interaction of partners' information systems. The solution provider is a central portal which is operated by the OEM. This portal provides access to information about parts in every partner of service system. The solution provider is depicted in figure 11 by the red circle. So, this solution requires upgrading the IT infrastructure of OEM. This upgrade increases the costs of OEM but it eliminates the mailing costs. This transformation also affects the value of other partners.

The central portal increases the partners' interoperability and reduces the repair time because dealers' technicians do not waste time in searching about available parts. They indentify and order parts more quickly than before. The level customer satisfaction index is higher and leads to more sales in the future. In this transformed repair service system, the value of OEM is calculated by an equation that includes the calculation of solution provider's value.

The second transformation that the authors suggest includes a replacement of previous central portal of OEM with an outsourcer who provides an electronic catalog system. The maintenance of this electronic catalog system is a service that is provided by the outsourcer. In figure 11, the outsourcer is depicted by a blue circled area. We observe some changes that occurred by the new transformation. The solution of outsourcer eliminates the OEM's costs of purchase and maintenance of central portal. The outsourcer offers the portal as a service and is paid by OEM yearly. As you can see in figure 11, the outsourcer manages the labor of help desk experts. Thus, if we compare the last two models then we will observe that the value of OEM may increase or reduce. This variation of OEM's value depends on the agreements that will be done between OEM and outsourcer or OEM and solution provider. This transformed repair service system also provides a modified equation that calculates the value of OEM and includes the value of the outsourcer.

4.3 Value Computation

Based on the last repair service system model of previous section we want to provide a framework that determines various strategies which can be followed by each partner in order to optimize its value or total value of service system. We implement a model that is able to examine how each strategy affects each partner's value over time. Specifically, we need a model which is able to capture the strategic behavior of its participants and compute their value each time taking into account the value of previous times. So, our model is a system dynamics model that is visualized, conceptualized, documented, simulated, analyzed, and optimized by Vensim tool. We focus on strategies that OEM can follow in order to optimize its value. According to this concept we modify the flow model of figure 11 and we add a second repair service system of another OEM. Through this addition we try to study the strategic behaviors that each OEM follows under the existence of real competition. We consider various scenarios where each of them has specific strategy of each OEM which attempting to optimize its value (see section 4.6). In this section we present our model analytically. We also present the equations which we use in the calculations of properties of each partner and we show you various figures of the visualization of each partner.

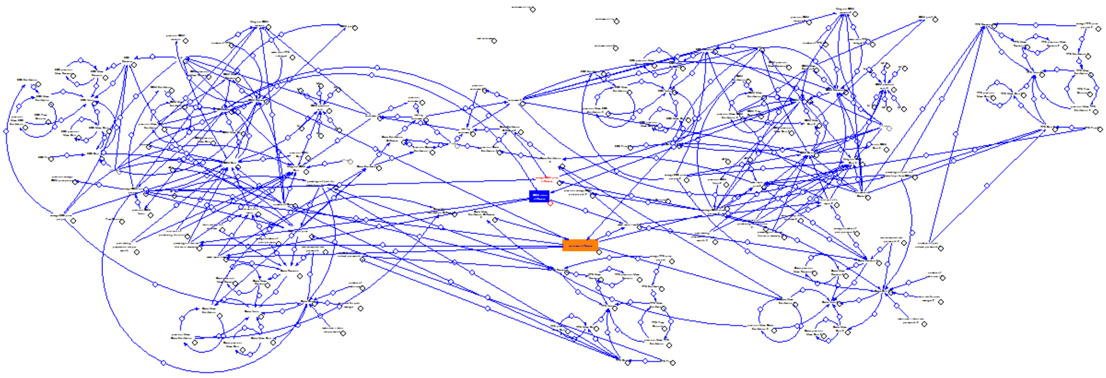


Figure 12. A General view of our model including both repair service systems

In figure 12 is shown a general view of our model that contains two repair service systems. As we have already mentioned, Vensim tool provides variables and arrows that determine the dependencies among the variables. We follow these features to visualize our model. We assume that each of partners which are depicted in figure 11 could be visualized by more than one variable in our model. Each of them may represent its revenue, or value, or costs and simultaneously encapsulates an equation that computes its value over. We assume that the time unit is month. Now we present you the view of each partner in our model and the value computation of each partner's using equations.

4.3.1 Dealer

The first partner whom we present is dealer who makes money by repairing the customers' cars. The repair comprises the new parts which replace the faulty parts in customers' cars, the fault diagnosis and the labor involved in parts replacement. The cost of a car repair is paid from OEM in the case that the repair is covered from warranty. In different case, dealer is paid from the car's owner. In figure 13 it is shown a view of Dealer from our model. The variable *total cost of a repair* represents the value of total cost of each repair and contains the values of variables that we have already mentioned such as *average number of parts per repair*, *labor rate by the customer*, *main repair price*, and *main repair time.* The equation that calculates the total cost of each repair is:

Total cost of a repair = labor rate by the customer * mean repair time + mean repair price * average number per parts

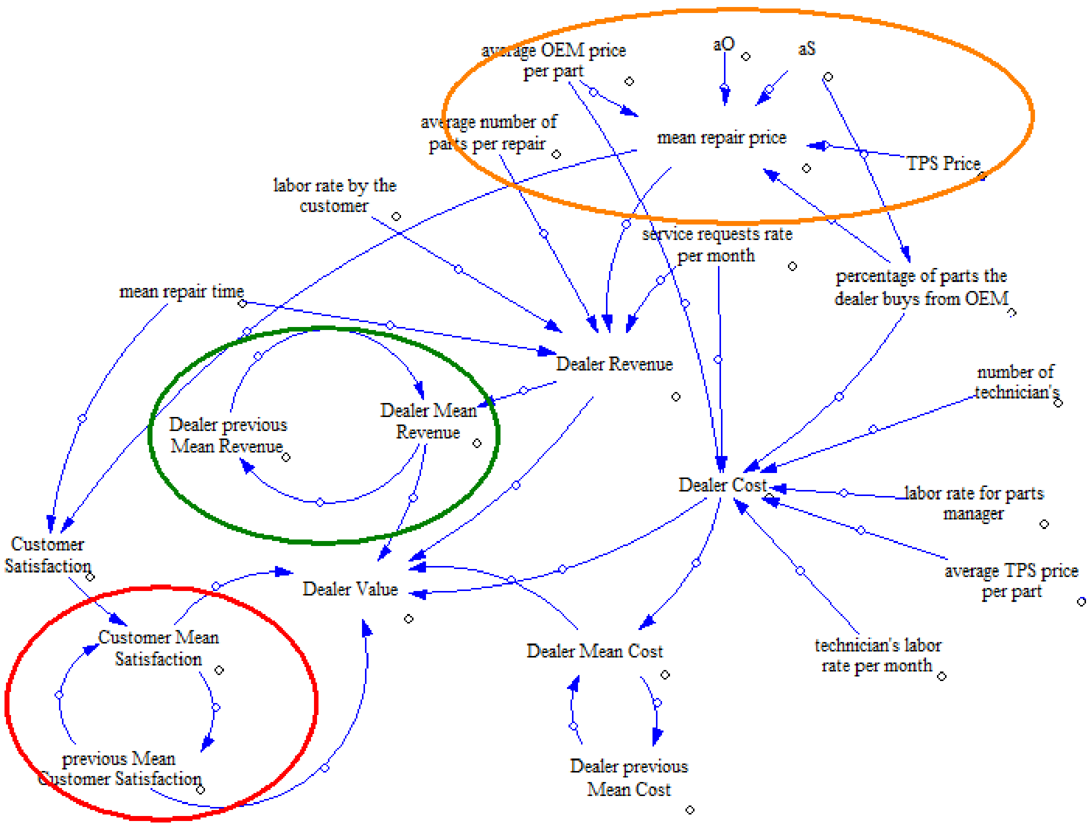


Figure 13. A view of dealer in our model

Total cost of a car repair comprises the cost of the dealer that performs the repair. So, the dealer's cost involves the technician's labor which is purchased by dealer and the parts' manager labor which is also purchased by dealer at a rate per month. The variable that is named *service requests rate per month* represents the average number of repair services that each dealer undertakes per month. As we have already referred, parts manager of each dealership works on each repair searching for the desired parts. The search begins from the dealer's inventory where is stocked the parts that is most preferred based on the past repairs. If these parts are not available in dealer's inventory, then parts manager will order these parts from Third Party Supplier (TPS) or from OEM. Usually, the average price per part of TPS is smaller than the average price per part of

OEM. If the desired parts of each repair are not available in dealer's inventory, then the repair time will be extended and the customer satisfaction will be affected negatively. Thus, the equation that calculates the dealer's cost per month is:

Dealer cost = service requests rate per month * (percentage of parts the dealer buys from OEM * average OEM price per part + (1- percentage of parts the dealer buys from OEM) * average TPS price per part) + labor rate for parts manager + number of technician's * technician's labor rate per month

A further analysis of equation of total cost of a repair includes the definition of equations that compute *mean repair price* and *mean repair time*. The computation of mean repair price involves the total number of parts that dealer buys for car repairs per month. Dealer has more than one choice in buying parts for car repairs. So, we have to separate the total number of parts that dealer buys into the percentage of parts that dealer buys from TPS and the percentage of parts that dealer buys from OEM. The percentage of parts that dealer buys of OEM is multiplied with the average price per part of OEM which is increased by a factor that corresponds in this price. Actually this increase of price is the profit of each dealer. The percentage of parts that dealer buys of TPS is multiplied with the average price per part of TPS which is increased by a factor that corresponds in this price. The view of mean repair price in our model is shown in an orange circle in figure 13. The equation of mean repair price is:

Mean repair price = percentage of parts the dealer buys from OEM * ((aO * average OEM price per part) + average OEM price per part) + (1- percentage of parts the dealer buys from OEM) * ((aS * TPS Price) + TPS Price)

We consider that mean repair time is constant over time. We assume that the value of mean repair time is one hour and a half of hour. The mean repair time comprises the time that technicians needs to research about the failure; the time that parts manager needs to order the necessary parts and also the time to execute the car repair. Actually, the time that is needed to execute the repair is the only one that is useful and cannot be reduced. However, the time that technicians needs to research and the time that parts manager needs to order the necessary parts are delays. Because of this each, dealer is willing to reduce in order to reduce the labor costs or the time to do the order of parts and simultaneously increase customer satisfaction. If customer satisfaction is increased then the dealer will expect to have more customers in the future. Thus, we define the equation that calculates the value of customer satisfaction index:

Customer Satisfaction Index = EXP ((-0.049 * mean repair price) - (0.2788 * mean repair time))

Now, we can compute Dealer's revenue if we based on total cost of a repair multiplied with service requests rate per month as you can see in figure 13. The equation of Dealer Revenue computation is below:

Dealer Revenue = (labor rate by the customer * mean repair time + mean repair price * average number per parts) * service requests rate per month

Let us continue with calculation of dealer's value which relies on dealer's revenue and cost per month. We compute the value of each partner using the equation: ***Value = Revenues – Cost*** + (***Expected Revenue*** – ***Expected Cost)*** (1). We have already discussed about the first and the second term of this equation. The last two terms represent the expectation of each dealer that based on their relationship with customers. A dealer expects potential sales of parts or services if its customers are satisfied from its past sales of parts or services. Hence, the expected revenue of each dealer is computed by the mean customer satisfaction index and the revenues of past times. The expected cost of each dealer is also computed by the mean customer satisfaction index and the cost of past times. The dealer's revenue and cost of past times using equations are represented by *mean Dealer Revenue* and *mean Dealer Cost* respectively. The mean Dealer Revenue is computed by this equation:

Mean Dealer Revenue = a * Dealer Revenue + b * previous Mean Dealer Revenue

According to this equation, the Mean Dealer Revenue is calculated by dealer's revenue of each time and the dealer's revenue of past times, where $a > 0$, $b < 1$ and $a + b = 1$ [27]. This equation will be executed recursively in simulation process of Vensim and we model it as is shown in the green circle in figure 13. We use level variables which provide us the use of recursion. So, the mean dealer revenue of each time will be the previous mean dealer revenue of the next time in simulation process. The mean dealer cost is computed by the same equation as the mean dealer revenue.

Now, mean customer satisfaction index is computed by almost the same way. Firstly, we have to define the equation that calculates the customer satisfaction index of each time. We consider that the basic factors in which customer satisfaction index based on are mean repair price and mean repair time. Based on our research we define this equation which calculates Mean Customer satisfaction index at each simulation time.

Mean Customer Satisfaction Index = a * previous Mean Customer Satisfaction Index + b * Customer Satisfaction Index (2)

As you can see in figure 13, the red circle contains the modeling of this recursion. We use level variables again. Nevertheless, we are ready to define the equation which computes the dealer's value including its expectation about future sales of parts or services. Actually, we define that:

- *Expected revenue* **=** (**Mean Customer Satisfaction Index / previous Mean Customer Satisfaction Index) * (Mean Dealer Revenue)** (3)
- *Expected cost* = (**Mean Customer Satisfaction Index / previous Mean Customer Satisfaction Index) * (Mean Dealer Cost)** (4)

Summarizing the equations (1), (2), (3), (4) we can define the equation that computes the each Dealer's value including its expectations:

Dealer Value = Dealer Revenue - Dealer Cost + (Mean Customer Satisfaction Index / previous Mean Customer Satisfaction Index) * (Mean Dealer Revenue - Mean Dealer Cost)

4.3.2 Original Equipment Manufacturer (OEM)

Original Equipment Manufacturer (OEM) is a dealers' partner that sells them certified car parts and also provides to dealers' technicians information about car repair. We consider that OEM sells its parts at an average price per part (p_o). We also consider that the number of parts which are ordered per month is represented by variable ***f*** and the percentage of parts that dealer buys from OEM is represented by variable ***a***. However, we have already defined a variable that represents the percentage of parts that dealer buys from OEM before. Thus, we combine these variables defining a new variable that represents the average number of repairs that each customer needs per month. The name of this variable is ***af***. In addition we assume that there is an access rate which is charged to each TPS and it is paid to OEM. Summarizing these variables we are able to define the equation which calculates the OEM revenue which is also depicted in figure 14:

OEM Revenue = number of Dealers * af * average OEM price per part + (access rate a TPS is charged * number of TPS's)

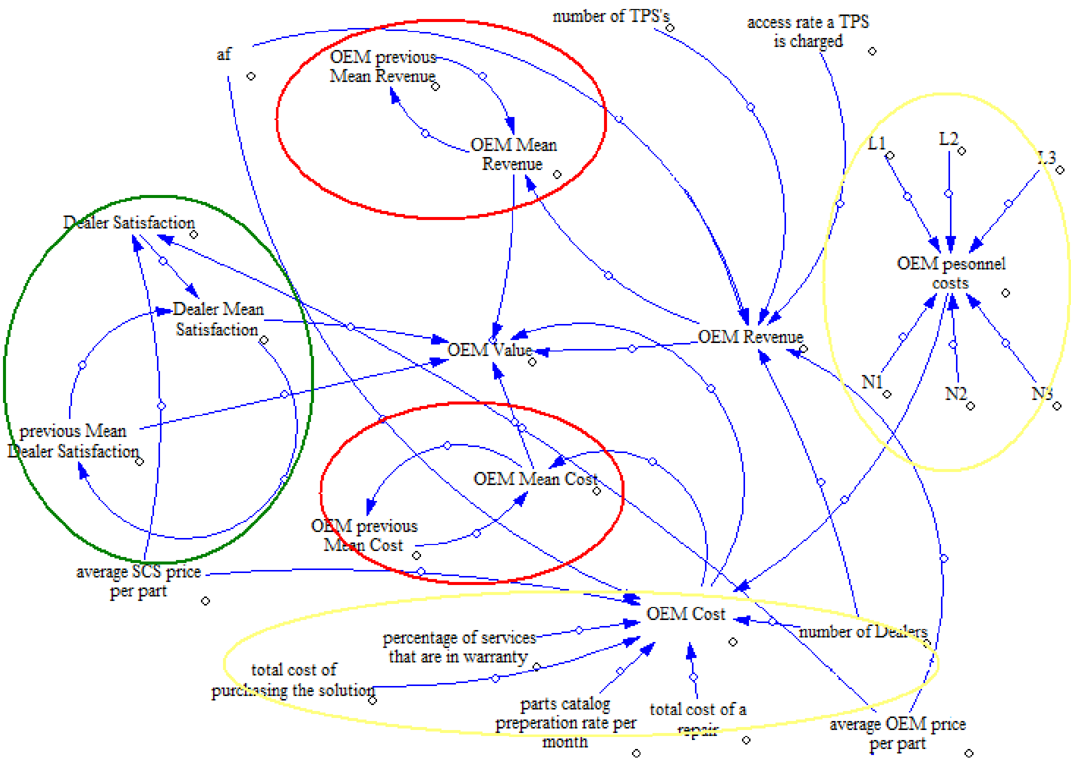


Figure 14. A view of OEM in our model

Although, OEM purchases raw material from SCSs in order to construct its certified parts and sell them depending on requests that are made by dealers per month. Each of these raw materials is purchased by OEM in average SCS price per part. In addition, the repairs that are covered from warranty are paid by OEM. Thus, we define a variable that is named *percentage of services that are in warranty* which how many car repairs are paid by OEM and not by each car owner. The parts catalog is replaced by a central portal which gives access to anyone in repair service system to up-to-date information about parts for free except from TPSs. This central portal is purchased by OEM in price that is defined in variable *annual rate for purchasing the service of solution* which is divided by 12 because we compute how it costs for OEM per month. The OEM also pays the labor which is offered by its personnel. The OEM personnel consist of three levels of employees that are classified depending on their experience and knowledge. In figure 14 you can see the views both of OEM personnel costs and OEM Cost in yellow circles. Below we define the equations that compute the OEM personnel costs and the OEM Cost.

OEM personnel costs = (L1 * N1) + (L2 * N2) + (L3 * N3)

OEM Cost = (number of Dealers * af * average SCS price per part) + (number of Dealers * af * percentage of services that are in warranty * total cost of a repair) + parts catalog preparation rate per month + (annual rate for purchasing the service of solution/12) + OEM personnel costs)

The relationship between OEM and dealers is the same as the relationship between dealer and customers respectively. The value of dealer satisfaction index is fully dependent from the value of average OEM price per part. If OEM increases the value of average OEM price per part then the dealer satisfaction index will be reduced. We are based on this rationale and we define the equation that computes the value of dealer satisfaction index:

Dealer Satisfaction Index = EXP (- 0.002 * average OEM price per part)

Due to the expectations of OEM of future sales attempts to increase dealer satisfaction offering technical advice for free. In addition, OEM upgrades its IT infrastructure and purchases a central portal that gives access to dealer's parts manager in parts catalog of OEM. So, dealer's parts manager can be informed about any part quickly. Then, the mean repair time will be reduced. As a result the customer satisfaction index will be increased and simultaneously the dealer satisfaction index will be increased.

The estimation of these expectations can be made by taking into account the sales and the dealer satisfaction index of past times. So, we separate these expectations into two terms: ***expected revenue*** and ***expected cost***. The first term represents the revenues that OEM expects to make in the future. The second term represents the cost that OEM expects to pay in the future. The computation of expected revenues is based on mean dealer satisfaction index and mean OEM revenue. Similarly, the computation of expected revenues is based on mean dealer satisfaction index and mean OEM revenue:

- *Expected revenue* **=** (**Mean Dealer Satisfaction Index / previous Mean Dealer Satisfaction Index) * (Mean OEM Revenue)**
- *Expected cost* = (**Mean Dealer Satisfaction Index / previous Mean Dealer Satisfaction Index) * (Mean OEM Cost)**

The model view of *mean dealer satisfaction index* is depicted into a green circle in figure 14. The computation of this variable based on its value of past times. We define the variable *previous mean dealer satisfaction index* in order to use the value of *mean dealer* *satisfaction index* of previous time. Below is the definition of the equation that computes the value of *Mean Dealer Satisfaction Index*, where a > 0, b < 1 and a + b = 1 [27].

Mean Dealer Satisfaction Index = a * previous Mean Dealer Satisfaction Index + b * Dealer Satisfaction Index

We use the corresponding equations in order to compute the *Mean Dealer Revenue* and *Mean Dealer Cost*. In figure 14 is shown the model view of these equations which is contained into a red circle.

Mean Dealer Revenue = a * Dealer Revenue + b * previous Mean Dealer Revenue

Mean Dealer Cost = a * Dealer Cost + b * previous Mean Dealer Cost

Based on our assumptions we generate a general definition of OEM value is: ***Value = Revenues – Cost*** + (***Expected Revenue*** – ***Expected Cost)***. So the final equation that calculates the value of OEM including the *expected revenue* and *expected cost* is:

OEM Value = (OEM Revenue - OEM Cost) + (Dealer Mean Satisfaction / previous Mean Dealer Satisfaction) * (OEM Mean Revenue - OEM Mean Cost)

4.3.3 Supply-Chain Supplier (SCS)

A Supply-Chain Supplier (SCS) is the partner who sells raw materials for car parts construction to OEM. Each SCS sells raw materials at *average SCS price per part*. The revenue of each SCS depends on the number of car repairs that are undertaken by dealers (af) and the number of dealers. Thus, the equations that compute the revenue and cost of SCS are below.

SCS Revenue = (af * average SCS price per part * number of Dealers)

SCS Cost = number of Dealers * af * SCS Price

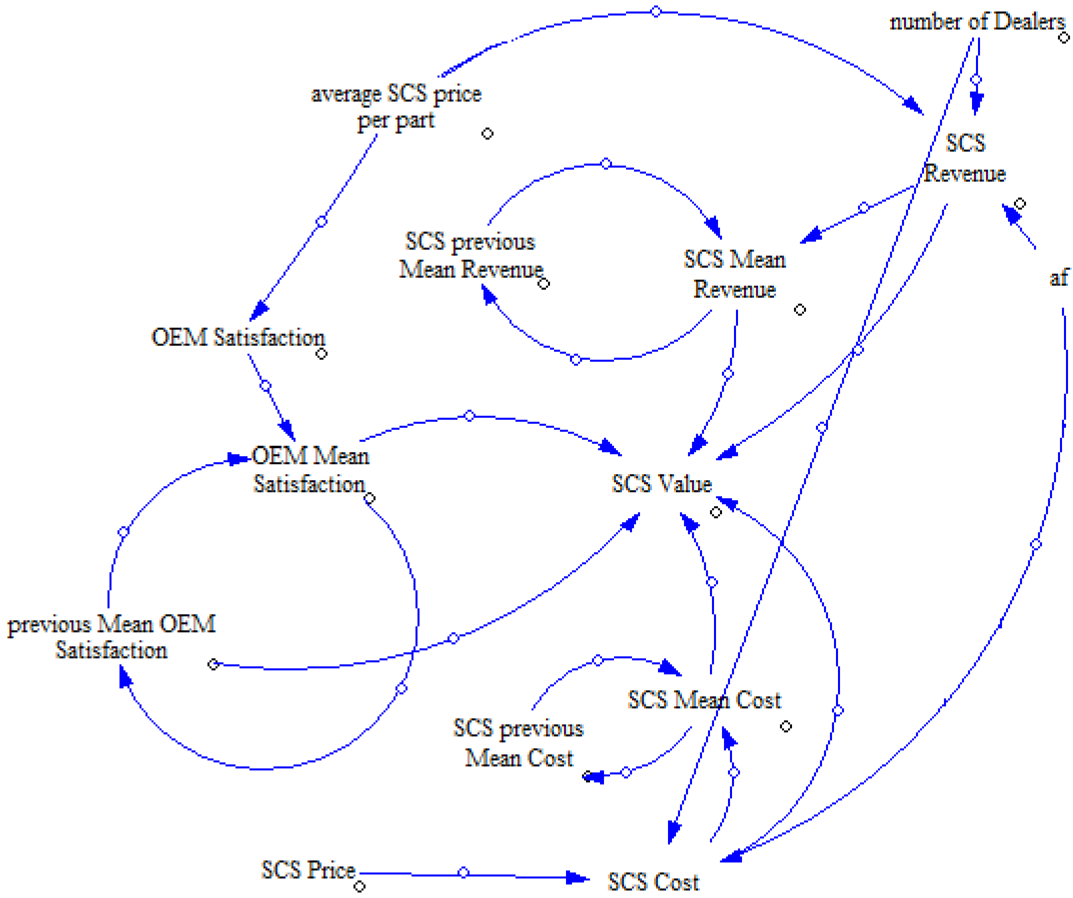


Figure 15. A view of SCS in our model

In figure 15, it is shown the model view of SCS which is similar to model views of previous partners. We compute SCS value taking into account that SCS sees value in its relationship with OEM. So, the relationship between each SCS and OEM is the same as the relationship between the OEM and Dealers respectively. Nevertheless, the value computation of SCS is similar to value computation of the previous partners.

SCS Value = SCS Revenue - SCS Cost + (OEM Mean Satisfaction * (SCS Mean Revenue - SCS Mean Cost)/ previous Mean OEM Satisfaction)

4.3.4 Third-Party Supplier (TPS)

A Third-Party Supplier (TPS) sells parts to dealers that have lower quality than the parts which are sold by OEM. These parts are not certified by OEM. The average price in which TPS sells each one from its parts is lower than the average price in which OEM sells each one from its parts. So, the equation that computes the revenue of a TPS per month is:

TPS Revenue = number of Dealers * (1 - percentage of parts the dealer buys from OEM) * average TPS price per part * numbers of parts ordered per month

The calculation of TPS cost is almost the same as the calculation of TPS revenue. The only difference between these two calculations is that the equation that computes the TPS Cost includes the price of the raw materials that are bought by TPS.

TPS Cost = number of Dealers * TPS Price * (1 - percentage of parts the dealer buys from OEM) * numbers of parts ordered per month

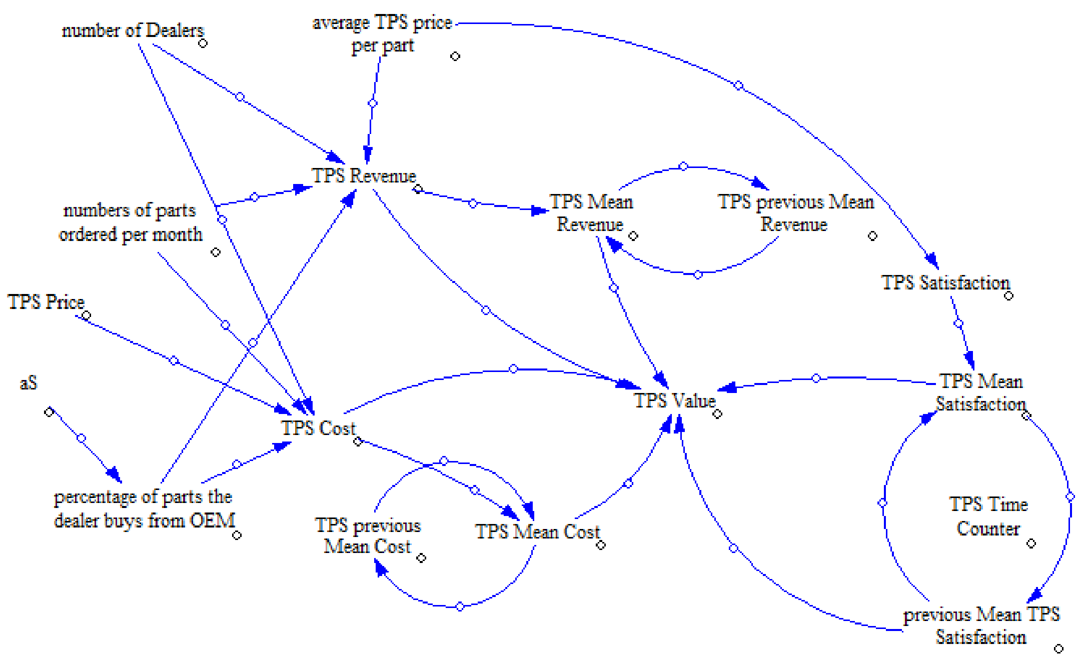


Figure 16. A view of TPS in our model

In figure 16, it is shown the model view of TPS which is similar to model views of SCS. We compute TPS value taking into account that TPS sees value in its relationship with dealer. So, the relationship between each TPS and Dealer is the same as the relationship between the OEM and Dealer respectively. Nevertheless, the value computation of TPS is similar to value computation of SCS as is shown in the following equations.

TPS Value = TPS Revenue - TPS Cost + (TPS Mean Satisfaction / previous Mean TPS Satisfaction) * (TPS Mean Revenue - TPS Mean Cost)

4.4 Nash Equilibrium and reaction functions

As we mentioned in chapter 2, Game Theory is a study of applied mathematics that tries to capture the behavior of two or more individuals that are interdependent and interact between each other. This study also comprises situations that every individual have to anticipate what decision the others will do from its own decision. We assume that such a situation could be a game that each of its players chooses the best available action. However, the best available action of each player is dependent on the actions of the other players. Each player chooses an action based on the actions that the other players will choose. This means that each player has to form a belief about the potential actions of the other players. The belief of each player is derived from its past experience and knowledge playing the game. This experience leads each player to make a prediction about the way that the other players behave in game process. Another assumption is that each player treats each play of the game in isolation. Each player does not restrict his actions from the behavior of specific players because his experience leads him to predict the action of typical opponents, not any specific set of opponents. A typical example of such a game is the interaction between buyers and sellers. These partners interact between each other and the first pairs of their choices may be considered as random. There are many cases where buyer and seller interact only once.

Summarizing, we define a theory that combines two components. The first component based on the assumption that each player acts in any game according to his rationale, given that he has experience and knowledge about past actions of his opponents. The second component based on the assumption that each player predicts the potential actions of his components correctly. Thus we present you the following definition: "A Nash equilibrium is an action profile $\boldsymbol{a^*}$ with the property that no player $\boldsymbol{i}$ can do better by choosing an action different from $\boldsymbol{a_i}$, given that every other player $\boldsymbol{j}$ adheres to $\boldsymbol{a_j}$" [26]. So, Nash equilibrium [28] represents a steady state where no player of any play of a game has a reason to change his action. At this state, each player's choice is the best given that his opponent chooses the specific action at any play of a game.

A further analysis of the concept of Nash equilibrium will be done through a presentation of a more specific example that follows. Assume that there are two firms in an industry, and let us call them Hybrid and Automatic. Imagine they produce cars. As we are considering that the industry is fictitious, let us make an assumption and imagine they can choose to produce either 75 or 100 cars per year. According to their quantity of productions, they will earn their corresponding profits. The matching between production and profits of the companies is represented by the payoff matrix in Table 1, below.

		Hybrid: 75 cars	Hybrid: 100 cars
Automatic	75 cars	$200m, $200m	$150m, $225m
	100 cars	$225m, $150m	$175m, $175m

Table 1. The payoff matrix about Automatic and Hybrid. The left number is profit to Automatic, the right number is profit to Hybrid

If both companies produce 75 cars each, then they each have $200 million profit. We consider that the total supply is restricted and the price is high. If both companies produce 100 cars each and the industry supply remain high, then the market price will be decreased drastically. As you can see in the Table 1, the profits of both companies are just $175 million. The best production that each company can do is to produce 100 cars when the competitor company produces 75 cars. The company which produces more cars will make $225 million in profit, while the other will make $150 million respectively.

Thus, we will research each cell of the Table 1 if there is a pair of values that leads to Nash equilibrium as we already mentioned above. Consider the lower left cell, where Automatic is producing 100 cars and make $225 million in profit. Automatic calculates that if it reduces the car production to 75 cars, then it also reduces its profit to $200 million. However, Hybrid calculates that if it produces 75 cars, then it makes $150 million (given that Automatic produces 100 cars) in profit. If Hybrid changes its strategy and choose to produce 100 cars, then its profits raises to $175 million. We can observe that when Automatic produces 100 cars and Hybrid produces 75 cars, only Automatic achieves the maximum profit. We cannot observe the same about the Hybrid because it achieves more profit when it produces 100 cars than when it produces 75 cars. As a result of this observation is that the lower left cell cannot be Nash equilibrium, as we mentioned above.

Let us continue the research about the left upper cell. Automatic produces 75 cars and makes $200 million, the same as Hybrid. Thus, if Hybrid changes its strategy and decides to raises its production to 100 cars, given that Automatic continues to produce the same number of cars as before, then Hybrid makes $225 million in profit. At the same time, Automatic reduces its profit to $150 million. Therefore, the new strategy of Hybrid is profitable for itself because it raises its profit and simultaneously decreases the profit of Automatic, its competitor. Let us try the opposite process and assume that Hybrid keeps its production stable and Automatic decides to raises its production from 75 cars to 100 cars. Studying the left lower cell, we observe the opposite than in previous cell. Now, the Automatic has larger profit than Hybrid and we understand that the new strategy of Automatic it was much more profitable than its competitor, Hybrid. Thus, the left upper cell cannot be Nash equilibrium, because its weakness to be optimal for both of companies in profit.

At right upper cell, Automatic produces 75 cars and Hybrid produces 100 cars. If Hybrid changes its strategy and decides to produce 75 cars, while Automatic keeps its production in 75 cars. It is noticeable that after this change the profit of Hybrid will be reduced from $225 million to $200 million, while Automatic will be rose its profit from $150 million to $200 million. Therefore, the new strategic of Hybrid cannot be profitable for itself. On the contrary, the strategic of Automatic is more profitable than the other's. If Hybrid follows another strategic and keeps its production stable, when Automatic changes its plans and increases its production form 75 cars to 100 cars, then the results according to Table 1 will be different from those before. According to this assumption, Automatic will raise its profits from $150 million to $175 million, while Hybrid will reduce its profits from $225 million to $175 million. Certainly, this assumption cannot be profitable for both companies. Thus, if we try to make an aggregation between the last two assumptions, we will conclude that the right upper cell cannot be Nash equilibrium.

So far, we research three cells of the Table 1 and none of them can be Nash equilibrium. Let us try the last one, which is the right lower cell. At this cell, Automatic produces 100 cars and makes $175 million in profit, while Hybrid also produces 100 cars and it also makes $175 million in profit. Therefore, both companies have the desire to raise their profits because of the rules of competition. Firstly, we make an assumption that Automatic changes its plans and decide to reduce its production from 100 cars to 75 cars in order to raise its profit (given that Hybrid keeps its production stable). According to Table 1 the profit of Automatic will be reduced, after these action and certainly cannot be a profitable strategy for the company. Thus, Automatic does not has an incentive to deviate unilaterally form its initial choice. Secondly, we make an assumption about the strategic of Hybrid, given now that Automatic will remain stable and continues to produce 100 cars. The assumption also includes that Hybrid decides to reduce its production from 100 cars to 75 cars. If we see Table 1, the profit of Hybrid will decrease from $175 million to $150 million. This progress shows us that neither Hybrid has an incentive to deviate unilaterally from its initial choice. Hence, only when both companies are producing 100 cars does neither company want to change choice. Therefore, the right lower cell can be Nash equilibrium, where neither company has an incentive to deviate unilaterally from producing 100 cars.

Above, we presented an example in which two competitive firms that choose among two appropriate strategies, which made the example simpler. In this example, we assumed that the companies could not have cooperated. If this could have happened, then their profits would have been much higher. The cooperation outcome between two companies involves an agreement about a restricted production, which would be in our example 75 cars. Thus, the earning profit would be $200 million. That kind of game that the Nash equilibrium leads the companies to lower payoffs than the cooperative outcome is also known as prisoner`s dilemma [29]. Applying prisoner`s dilemma to our example, we discover that there some possible outcomes. Firstly, the companies could each choose to produce 100 cars which as we mentioned above, is a Nash equilibrium. Secondly, they could collude and produce 75 cars each. Nevertheless, if the companies collude, there is a main problem. Even if they could manipulate to secretly collude, both companies have an incentive to cheat on the agreement. Collusion is very risky and it is difficult to maintain as it is. Each company may break the agreement and raise its production over the limit that the companies agreed.

However, let us change the example slightly. Imagine that Hybrid entered the industry first. Thus, it could have built a factory which is capable to produce 100 cars a year. Therefore, when Automatic enters the industry, it asks how many it should build. Logically, Automatic will choose to produce 75 cars. In this example, Hybrid has a first-mover advantage [30]. By selecting first the strategy, Hybrid gets to select which Nash equilibrium the industry selects. So, it gets to keep the lion's share of the profits for itself.

Above, we created Table 1 which is payoff matrix and unfortunately has a major limitation of choices. In reality, Hybrid and Automatic could have more than 2 choices in production. They could also choose 80, 90,110, and so on. However, if we companies have more than possible choices, we will create payoff matrixes with more rows and columns. But, the matrix would quickly get large as the number of choices decreased. Thus, it is impossible to create such an abstract matrix.

A different way of approaching equilibrium is to develop the idea of reaction functions [31]. A reaction function is an equation that shows us which is the optimal choice for each company, given each possible choice the other company has already made. If we create a specific example, where there are two companies and their reaction functions been already defined, we will represent the reaction functions graphically. Generally the reaction functions are downward-sloping for an intuitive reason, for example: if Hybrid produces more, the industry price will reduce and then the optimal quantity of output for Automatic must decline. Below, we will describe an experiment showing graphically the reaction functions and also the Nash equilibrium, if it exists.

4.5 Competition Scenarios

According to the assumptions in section 4.3, our model contains two repair service systems. Each of them has a model view that is shown in figure 11. The main difference between these value networks is that the OEM of the second service system has lower value than the OEM of first service system. The customers of our model have to choose between these two repair systems in order to repair their car. As is shown in figure 15, ***OEM1*** cloud represents the OEM partner of the first repair service system while ***OEM2*** cloud represents the OEM partner of the second repair service system. The cloud which is named ***Customers1*** represents the customers who pay for their car repair a dealer that collaborates with ***OEM1*** while ***Customers2*** pay for their repair car a dealer that collaborates with ***OEM2***. If a customer form ***Customers1*** is not satisfied enough from the services of Dealer1 then he will choose to collaborate with a dealer which belongs to the second repair service system. If a customer form ***Customers2*** is not satisfied enough from the services of Dealer2 then he will choose to collaborate with a dealer which belongs to the first repair service system. Each customer offers revenue to collaborating dealer which offers revenue to cooperating OEM. So, each customer is very significant about revenues and value of each OEM. Assuming that the goal of each OEM is the maximization of its value, it is obvious that they are competitors.

We have already defined the equations that compute the value of each partner of our model, so we know how each OEM can maximize its value. However, we do not know how each OEM can maximize its value given that another OEM has exactly the same goal as itself. We have to study the behavior of each OEM attempting to maximize its value given that the opponent OEM attempts to do the same. Each of them has a strategy in order to be competitive and finally achieve its goal. Due to competition each OEM chooses its strategy having in mind which will be the strategy of its competitor. As a result, the primary goal of this thesis is the research about how and when each OEM achieves its main goal given that they are competitors and act simultaneously. A further though leads us to combine our goal with the concept of Nash equilibrium. This means that we research the cases where both OEM can achieve their goal simultaneously even if they are competitors. Thus, we assume various scenarios where each one of them includes specific strategy about each OEM. Each strategy determines the possible actions of specific OEM given that its competitor will choose a specific action. So, the main goal of this thesis is to examine the possible pairs of actions of each scenario and show which pair is a Nash equilibrium. Such a pair of actions of a specific scenario will show us that the strategies of each OEM lead them to achieve their goal under the conditions of competition. We apply all these scenarios to our model using Vensim tool simulation and analysis because we want to study each of them per month. Below we present you the rationale of each scenario and also the results of simulation in Vensim.

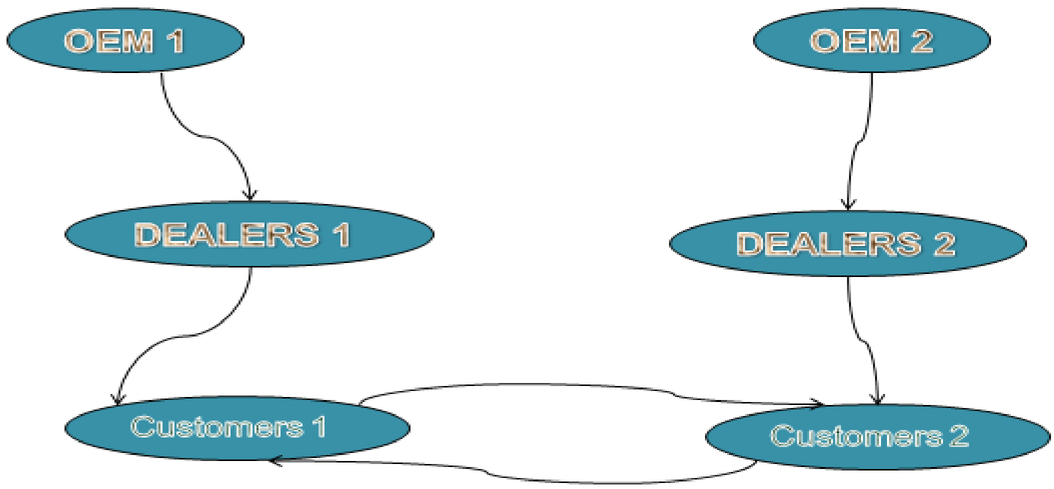


Figure 17. A general model view of competition between two repair service systems

4.5.1 First Scenario

As we already mentioned, customers are very important for each OEM because both OEM have the same goal which is the maximization of their value in service system. A customer continues his collaboration with an OEM if he is satisfied from the services which are offered to him. This means that each customer offers more revenues to OEM and so the value of the specific OEM is increased. Nevertheless, each OEM aims to keep the customer satisfaction index in high level. So, each OEM must focus its strategy on the factors that affect the customer satisfaction index. As we already presented, customer satisfaction index is affected by mean repair price and mean repair time. In this scenario, we assume that mean repair time is stable for both repair service systems and has a value that is defined by our experience in value networks. So, the concept of this scenario is focused on mean repair price which is affected by the average price that OEM sells each one from its parts to dealers. Thus, the key of each OEM is the average OEM price per part. In our Vensim model the average price that OEM sells each one from its parts is defined as an ***auxiliary*** variable. So, the strategy of each OEM is focused on the criteria that determine when OEM has to increase or reduce the value of this variable. We assume that the first criterion is the difference of OEM value between each current and previous time. The second criterion is the difference of the number of parts that are ordered at each simulation time. Below we present you the algorithm of this strategy.

```
IF (OEM1 VALUE < PREVIOUS OEM1 VALUE)

        IF (ORDERS FOR PARTS OF OEM1 < ORDERS FOR PARTS OF OEM2)

                OEM1 PRICE = OEM1 PRICE - offset

        ELSE IF (ORDERS FOR PARTS OF OEM1 > ORDERS FOR PARTS OF OEM2)

                OEM1 PRICE = OEM1 PRICE + offset

ELSE IF (OEM1 VALUE > PREVIOUS OEM1 VALUE)

        OEM1 PRICE = OEM1 PRICE
```

As you can see, this algorithm includes a specific type of conditions (***IF THEN ELSE***) which help to distinguish the cases of a negative and positive of OEM value difference. This computation based on the rationale that OEM examines the difference of its value between each current and previous time in order to evaluate its current strategy. If the difference is positive, then OEM will not change its average price per part. But if the difference is negative then OEM will examine the difference of the number of its parts that are ordered each time concerning the orders of previous time. In the case that this difference is negative, OEM will not change its average price per part. In the opposite case, OEM checks if the average price per part is bigger than 113 units. This specific number is used because is the lowest price in which OEM can sell its parts to dealer without increasing its costs and reducing its revenues. So, if this condition is true, then OEM will reduce its average price per part as much as OEM considers that is appropriate. If the same condition is false, then OEM will not change the specific price. In addition, the equation that computes the value of average price that OEM of the first repair service system sells each one from its parts is:

average OEM1 price per part = IF THEN ELSE(OEM1 Value - previous OEM1 Value >= 0 , 0 , IF THEN ELSE(parts ordered > previous parts ordered , 0 , IF THEN ELSE(average OEM1 price per part > 113 , - OFFSET, 0)))

It is obvious that the primary criterion of the specific OEM is the variability of its value over each current and previous time. It is very significant for each OEM to increase its value or keep it in high level. Thus, if the value of OEM has been increased concerning the previous time, then OEM considers that it should not change anything about the specific price. But if the value of OEM has been reduced concerning the previous time, then OEM has to examine if it receives fewer orders than its competitor. In this case, OEM realizes that has to do something to attract more orders from dealers. That means that its collaborating dealers must have more customers. Thus, OEM reduces the average price that sells each one from its parts in order to increase the dealer satisfaction index of its cooperating dealers and simultaneously increase the satisfaction index of their customers. Nevertheless, these dealers will be more attractive for the customers and then the orders for parts of this OEM will be increased. However, in the case that the orders for parts of the first OEM are more than the orders for parts of the second OEM and also the value of the specific OEM has been reduced concerning the previous time, then OEM will prefer to increase its average per part. This means that OEM aims to increase its value even if it may reduce the satisfaction index of its collaborating dealers which means fewer customers for them and finally fewer orders for OEM parts than before. Similarly, the OEM of the second repair service system has the same strategy as the previous OEM. So, the only difference between these equations is the offset which depends on each OEM decision.

average OEM2 price per part = IF THEN ELSE(OEM2 Value - previous OEM2 Value >= 0 , 0 , IF THEN ELSE(parts ordered > previous parts ordered, 0 , IF THEN ELSE(average OEM2 price per part > 113 , - OFFSET, 0)))

The simulation of this scenario generates results which show us the behavior of each OEM over the simulation time. We define the simulation time at 60 months in order to examine the behavior of each OEM each month for 5 years. We also assume that the initial value of average price per part of the first OEM is 200 while the initial value of average price per part of the second OEM is 190 because it is smaller than In this way we can draw a conclusion about the existence of Nash equilibrium between the values of these two OEMs.

So, in figure 16 we present the graphs of two variables of each OEM over simulation time. We compare the value of the first OEM with the value of the second one and the average price per part of the first OEM with the average price per part of the second OEM horizontally. In addition, we examine the contrast between the value and the average price per part of each OEM in order to examine how the average price per part can affect the value of each OEM.

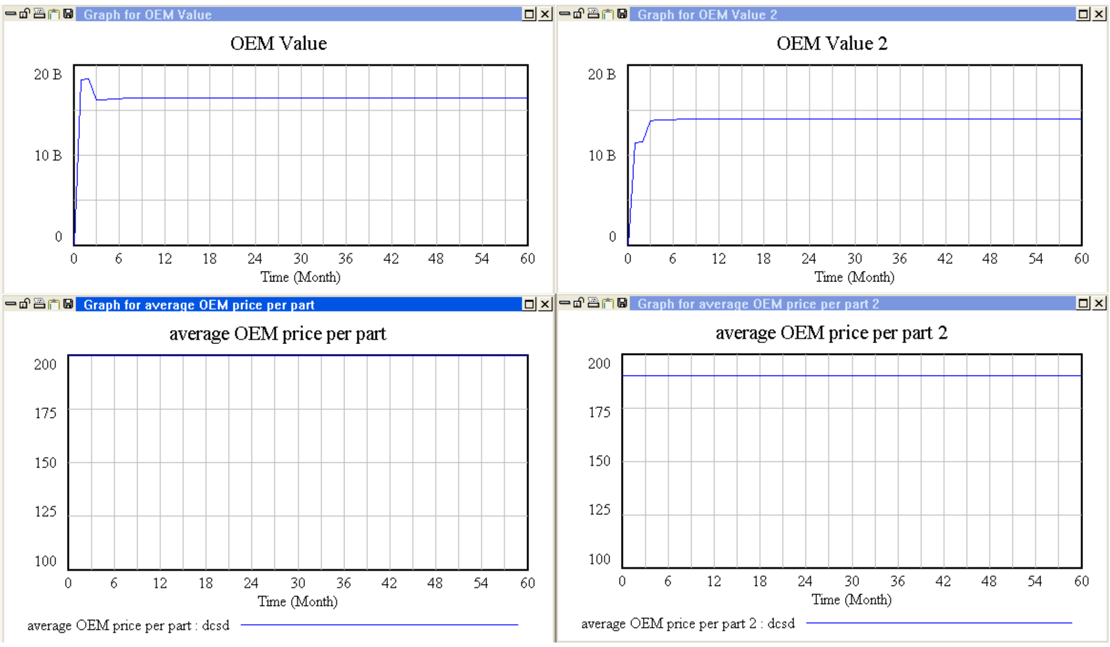


Figure 18. The graphic representations of value and average price per part of each OEM

As you can see in figure 18, the average price of each OEM does not change from its initial value over the whole simulation time. This means that according to its strategy neither the first OEM nor the second OEM is willing to change its average price per part in order to increase its value. However, the value of each OEM is changed over simulation time. In Table 2 is shown the values of variables that are related to the specific scenario. So, this table comprises the value, the average price per part and the number of orders of each OEM. It is observed that at the fourth month of simulation process the number of orders for parts of the first OEM is reduced and at the same time its value is reduced concerning the previous month. This means that the number of orders for parts of the second OEM is increased and simultaneously the value of this OEM is increased.

Table Time Down

Time (Month	OEM Value	"average OEM price per part" Runs: dcsd	average OEM p	"af" Runs: dcsd	af	"OEM Value 2" Runs: dcsd	OEM Value 2	"average OEM price per part 2" Runs: dcsd	average OEM p	"af 2" Runs: dcsd	af 2
0	1.26153		200		18000		1.26722		190		12000
1	1.83307e+010		200		18000		1.13425e+010		190		12000
2	1.84755e+010		200		18000		1.14289e+010		190		12000
3	1.60925e+010		200		15580		1.38167e+010		190		14420
4	1.61709e+010		200		15580		1.38809e+010		190		14420
5	1.62301e+010		200		15580		1.39291e+010		190		14420
6	1.62739e+010		200		15580		1.39648e+010		190		14420
7	1.63059e+010		200		15580		1.39906e+010		190		14420
8	1.63289e+010		200		15580		1.40092e+010		190		14420
9	1.63453e+010		200		15580		1.40225e+010		190		14420
10	1.63569e+010		200		15580		1.40319e+010		190		14420
11	1.63652e+010		200		15580		1.40386e+010		190		14420
12	1.6371e+010		200		15580		1.40433e+010		190		14420
13	1.63751e+010		200		15580		1.40466e+010		190		14420
14	1.63779e+010		200		15580		1.40489e+010		190		14420
15	1.63799e+010		200		15580		1.40505e+010		190		14420
16	1.63814e+010		200		15580		1.40516e+010		190		14420
17	1.63823e+010		200		15580		1.40524e+010		190		14420
18	1.6383e+010		200		15580		1.4053e+010		190		14420
19	1.63835e+010		200		15580		1.40534e+010		190		14420
20	1.63839e+010		200		15580		1.40537e+010		190		14420
21	1.63841e+010		200		15580		1.40539e+010		190		14420
22	1.63843e+010		200		15580		1.4054e+010		190		14420
23	1.63844e+010		200		15580		1.40541e+010		190		14420
24	1.63845e+010		200		15580		1.40542e+010		190		14420
25	1.63845e+010		200		15580		1.40542e+010		190		14420
26	1.63846e+010		200		15580		1.40542e+010		190		14420

Table 2. A table that includes values of specific variables over simulation time in first scenario

The following months of simulation process shows us that the number of orders for parts of each OEM remain stable. This happens because neither the first OEM nor the second one changes its average price per part. Thus, the satisfaction index of its collaborating dealers does not change. So, the same will be happen to the satisfaction index of these dealers' customers. In Table 2, the red circles indicate the changes that happen at the fourth month of simulation process. The green boxes indicate that the average value of each OEM remains stable and also the black arrows indicate that the number of orders for parts of each OEM is stable since the fourth month. A further thought of this stability that is observed leads us to draw a conclusion that these two competitors are at a state which neither the first OEM nor the second one has a reason to choose any action different from its opponent. As we already mentioned this state is named Nash equilibrium and this means that the specific strategy of each OEM leads to this state even if they are competitors and also different. In the following figure is shown how number of orders for parts of each OEM can affect the value of each OEM. As a result, the simulation process of this scenario shows that the OEMs of our model can be in Nash equilibrium, if each OEM uses a strategy that computes its average price per part based on estimation of its value and number of orders for its parts.

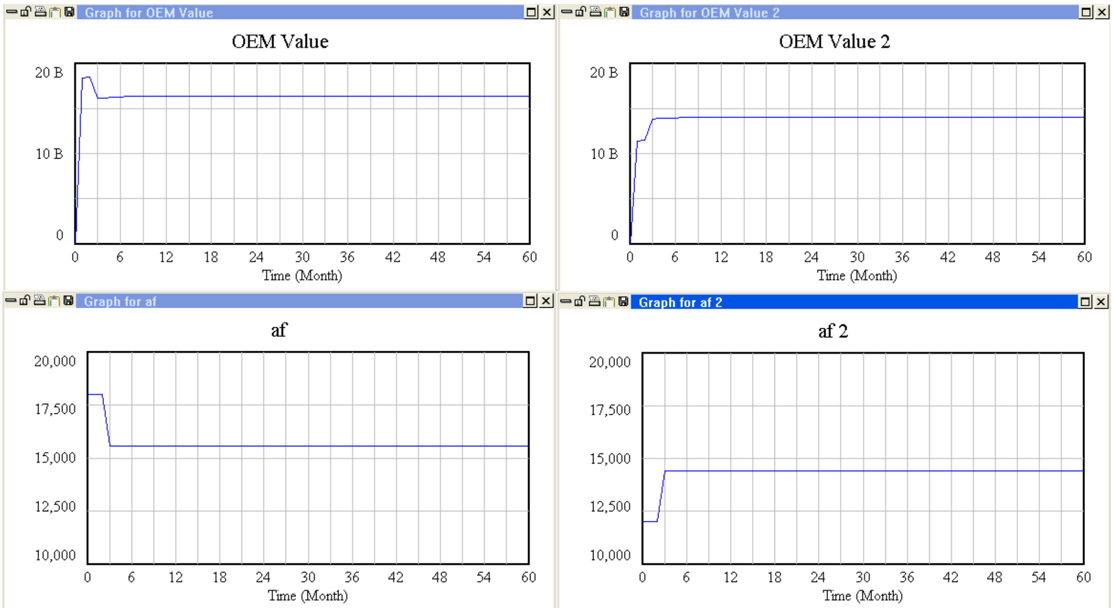


Figure 19. Graphs that represent the value and the number of orders for parts of each OEM in first scenario

4.5.2 Second Scenario

The second scenario is almost similar to the first one. Both OEMs have the same goal which is the maximization of its value. The OEM of the first repair system has bigger initial value than the second one. So, the main difference between the first and the second scenario is based on the strategy that is followed by each OEM. Original Equipment Manufacturer of the first repair service system follows a strategy that is based on the number of orders for its parts. This means that the computation of average price at which OEM sells each of its parts relies on the number of orders for its parts at each time concerning the corresponding orders at previous time. According to this strategy, if the number of orders is increased then OEM will not change the specific average price. In the opposite case, OEM will examine the estimation of its value concerning the value estimation of previous time. So, if the value of OEM is reduced then OEM will reduce its average price per part as much as it believes that is appropriate in order to increase its value in next month. But if the value of OEM is increased then OEM will not keep the specific average price stable. The algorithm of this strategy is defined below.

IF (ORDERS FOR PARTS OF OEM1 < ORDERS FOR PARTS OF OEM2)

```
        IF (OEM1 VALUE < PREVIOUS OEM1 VALUE)
                OEM1 PRICE = OEM1 PRICE - offset
        ELSE IF (OEM1 VALUE > PREVIOUS OEM1 VALUE)
                OEM1 PRICE = OEM1 PRICE
ELSE IF (ORDERS FOR PARTS OF OEM1 > ORDERS FOR PARTS OF OEM2)
        OEM1 PRICE = OEM1 PRICE
```

On the other hand, the strategy that is followed by OEM of the second repair service system is almost the same as previous strategy. The only difference between these strategies is based on the rationale of second OEM about the estimation of the average price in which it sells each of its parts. This means that in the case that the number of orders for parts of OEM is reduced concerning the specific orders of previous time and simultaneously the value of OEM is also reduced then OEM will reduce its average price per part as much as it considers that is appropriate according to the corresponding average price of the its competitor. For example, if its competitor sets the specific average price at 200 units, then the second OEM will set the corresponding price at 190 units. Below we define the algorithm of this strategy.

```
IF (ORDERS FOR PARTS OF OEM2 < ORDERS FOR PARTS OF OEM1)
        IF (OEM2 VALUE < PREVIOUS OEM2 VALUE)
                OEM2 PRICE = OEM1 PRICE - offset
        ELSE IF (OEM2 VALUE > PREVIOUS OEM2 VALUE)
                OEM2 PRICE = OEM2 PRICE
ELSE IF (ORDERS FOR PARTS OF OEM2 > ORDERS FOR PARTS OF OEM1)
        OEM2 PRICE = OEM2 PRICE
```

We simulate our model including this scenario and we get the results of simulation process. In Table 3 we can see the values of some significant variables of this scenario such as the value of each OEM, the average price in which each OEM sells its parts and the number of orders for parts of each OEM. We observe that OEM of the first repair service system has initially more orders than the OEM of the second repair service system as we already defined. The average price per part of the first OEM has higher than the average price per part of the second OEM. During the first three months of simulation process the situation is same as initially. At the fourth month of simulation process, the number of orders for parts of the first OEM is reduced concerning the previous month while the number of orders for parts of the second OEM is increased concerning the previous month. As a result, the value of the first OEM is reduced concerning its value of previous month while the value of the second OEM is increased concerning its value of previous month. However, the average price per part of the first OEM and the average price per part of the second OEM remain stable over the simulation process. This happens due to the rationale of each OEM's strategy which we already presented.

Table Time Down

Time (Month	OEM Value	"average	average OEM p	"af" Runs:	af	"OEM Value	OEM Value 2	"average	average OEM p	"af 2" Runs:	af 2
0	1.26153	OEM price	200	test	18000	2" Runs:	1.26722	OEM price	190	test	12000
1	2.42571e+010	per part"	200		18000	test	1.31231e+010	per part 2"	190		12000
2	2.44487e+010	Runs:	200		18000		1.3223e+010	Runs:	190		12000
3	2.12952e+010	test	200		15580		1.59857e+010	test	190		14420
4	2.1399e+010		200		15580		1.60599e+010		190		14420
5	2.14774e+010		200		15580		1.61158e+010		190		14420
6	2.15354e+010		200		15580		1.6157e+010		190		14420
7	2.15776e+010		200		15580		1.61869e+010		190		14420
8	2.1608e+010		200		15580		1.62084e+010		190		14420
9	2.16297e+010		200		15580		1.62238e+010		190		14420
10	2.16452e+010		200		15580		1.62347e+010		190		14420
11	2.16561e+010		200		15580		1.62424e+010		190		14420
12	2.16637e+010		200		15580		1.62478e+010		190		14420
13	2.16692e+010		200		15580		1.62516e+010		190		14420
14	2.1673e+010		200		15580		1.62543e+010		190		14420
15	2.16756e+010		200		15580		1.62562e+010		190		14420
16	2.16775e+010		200		15580		1.62575e+010		190		14420
17	2.16788e+010		200		15580		1.62584e+010		190		14420
18	2.16797e+010		200		15580		1.62591e+010		190		14420
19	2.16804e+010		200		15580		1.62595e+010		190		14420
20	2.16808e+010		200		15580		1.62598e+010		190		14420
21	2.16811e+010		200		15580		1.62601e+010		190		14420
22	2.16813e+010		200		15580		1.62602e+010		190		14420
23	2.16815e+010		200		15580		1.62603e+010		190		14420
24	2.16816e+010		200		15580		1.62604e+010		190		14420
25	2.16817e+010		200		15580		1.62605e+010		190		14420

Table 3. A table that includes values of specific variables over simulation time in second scenario

The values that are presented in Table 3 can be represented in graphs. As you can see in figure 20, the upper left graph of value of the first OEM is shown that the value increases until the third month. At the fourth month of simulation process is observed a fall of the value of the first OEM while the value of the second is increased as is depicted in the upper right graph. The lower left graph shows that the numbers of orders for parts of the first OEM is increased until the fourth month but the specific number is reduced at the fourth month while the number of orders for parts of its competitor is increased.

However, we observe that the value of both OEMs remains stable after the fourth month until the end of simulation process. Thus, no one of the competitors has an incentive to deviate unilaterally from its choice. This means that the strategy that is followed by each OEM given that its competitor chooses a specific strategy leads both OEMs to a state that is Nash equilibrium. Hence, we can say that even if the second OEM determines the value of its average price per part concerning the corresponding value of its competitor, then we are drawn to the same conclusion as in previous scenario.

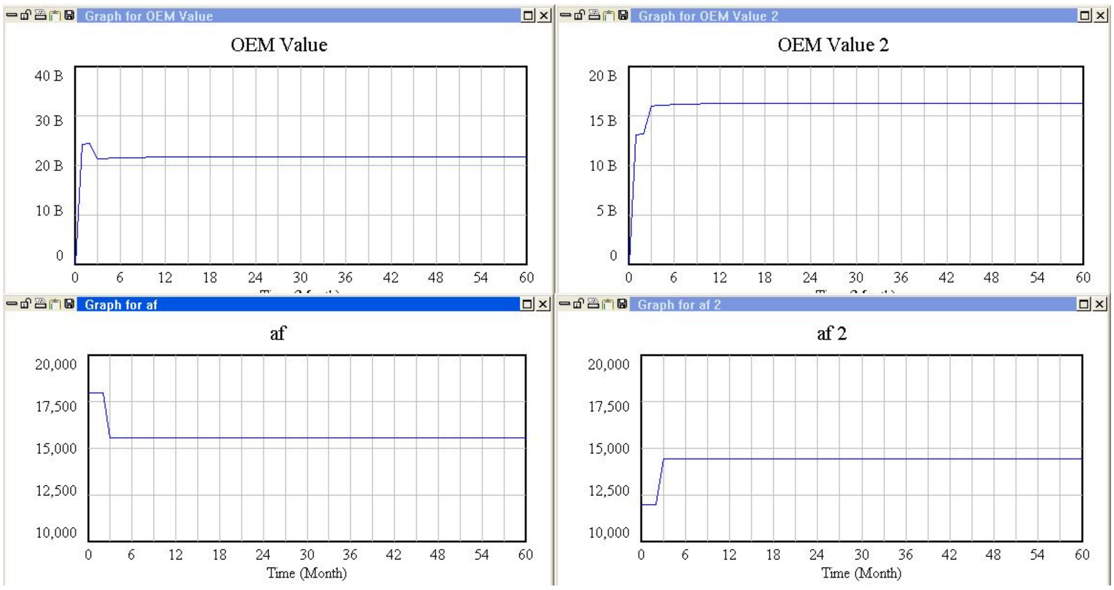


Figure 20. Graphs that represent the value and the number of orders for parts of each OEM in second scenario

4.5.3 Third Scenario

The third scenario includes a different rationale about the second OEM which is considered as weaker than first OEM. So, we assume that the first OEM does not change its rationale and follows the same strategy as in second scenario. But the second OEM follows a different strategy that is considered as more risky but will be profitable for itself. The main concept of this strategy is the variability of mean repair time. This means that it tries to increase the satisfaction index of collaborating dealers which will increase the satisfaction index of their customers. Thus, they probably attract customers from its competitor increasing its average price per part and simultaneously decreasing the mean repair time. In this way, dealers attempt to attract customers that are interested in a quick repair of their car despite of the fact that this will cost more money. Thus, in the case that the number of orders for parts of second OEM is decreased and simultaneously its value is also reduced, then OEM will increase its average price per part and reduce the mean repair price. The reduction of mean repair price is achieved though offerings of OEM to collaborating dealers such as access to central portal for free, technical advice for free, special offerings about specific parts. These actions increase the satisfaction index of dealers which means that dealers will continue to purchase parts from OEM in the future. As a result, the satisfaction index of customers that collaborates with these dealers is increased because of lack of waiting time. We assume that there is a highest limit of average price per part that helps OEM to avoid the financial disaster. We also assume that there is a lowest limit of mean repair price because each car repair has minimum time requirements. The algorithm of this strategy is presented below.

```
IF (ORDERS FOR PARTS OF OEM2 < ORDERS FOR PARTS OF OEM1)
        IF (OEM2 VALUE > PREVIOUS OEM2 VALUE)
                OEM2 PRICE = OEM2 PRICE + offset
                MEAN REPAIR TIME = MEAN REPAIR TIME - offset
        ELSE
                OEM2 PRICE = OEM2 PRICE - offset
                MEAN REPAIR TIME = MEAN REPAIR TIME + offset
ELSE IF (ORDERS FOR PARTS OF OEM2 > ORDERS FOR PARTS OF OEM1)
        OEM2 PRICE = OEM2 PRICE
        MEAN REPAIR TIME = MEAN REPAIR TIME
```

The values that are generated by this scenario simulation process are depicted in Table 4 below. The average price per part of the first OEM remains stable during the whole simulation process. Also the mean repair price of dealers that collaborates with the first OEM remains stable. Initially the number of orders for parts (see sixth column) of the second OEM is higher than the number of orders for parts of the first OEM (see second column). Thus, the value of second OEM (see third column) is lower than the value of the first OEM (see first column). This means that the second OEM will increase its average price per part and reduce the mean repair price of collaborating dealers in order to attract more customers. According to Table 4, we observe that such a fact happens at the fourth month of simulation process and is continued until the tenth month. During the next months we do not observe any changes about the average price per part, the number of orders for parts and the value of each OEM. Thus, no one of the competitors has an incentive to deviate unilaterally from its choice. This means that the strategy that is followed by each OEM given that its competitor chooses a specific strategy leads both OEMs to a state that is Nash equilibrium. So, we can draw a primary conclusion that the pair of strategies of this scenario leads them to Nash equilibrium.

Table Time Down

Time (Month	OEM Value	"af" Runs: test	af	"OEM Value 2" Runs: test	OEM Value 2	"average OEM price per part 2" Runs: test	average OEM p	"mean repair time 2" Runs: test	mean repair time	"af 2" Runs: test	af 2
0	1.26153		16000		1.27302		180		1.5		14000
1	2.41959e+010		16000		1.73799e+010		190		1.5		14000
2	2.43869e+010		16000		1.93908e+010		200		1.3		14000
3	2.3277e+010		15176		2.26575e+010		210		1.1		14824
4	2.33565e+010		15154		2.47389e+010		220		0.9		14802
5	2.34188e+010		15139		2.65955e+010		230		0.9		14787
6	2.34665e+010		15129		2.84486e+010		240		0.9		14777
7	2.35001e+010		15121		3.0295e+010		250		0.9		14769
8	2.35255e+010		15116		3.2139e+010		260		0.9		14764
9	2.35444e+010		15113		3.22305e+010		260		0.9		14761
10	2.35581e+010		15111		3.2298e+010		260		0.9		14759
11	2.357e+010		15111		3.235e+010		260		0.9		14759
12	2.35783e+010		15111		3.23872e+010		260		0.9		14759
13	2.35842e+010		15111		3.24136e+010		260		0.9		14759
14	2.35884e+010		15111		3.24324e+010		260		0.9		14759
15	2.35913e+010		15111		3.24456e+010		260		0.9		14759
16	2.35933e+010		15111		3.24549e+010		260		0.9		14759
17	2.35947e+010		15111		3.24615e+010		260		0.9		14759
18	2.35957e+010		15111		3.24661e+010		260		0.9		14759
19	2.35964e+010		15111		3.24693e+010		260		0.9		14759
20	2.35969e+010		15111		3.24715e+010		260		0.9		14759
21	2.35973e+010		15111		3.24731e+010		260		0.9		14759
22	2.35975e+010		15111		3.24742e+010		260		0.9		14759
23	2.35977e+010		15111		3.2475e+010		260		0.9		14759
24	2.35978e+010		15111		3.24756e+010		260		0.9		14759
25	2.35979e+010		15111		3.24759e+010		260		0.9		14759
26	2.35979e+010		15111		3.24762e+010		260		0.9		14759
27	2.3598e+010		15111		3.24764e+010		260		0.9		14759
28	2.3598e+010		15111		3.24765e+010		260		0.9		14759
29	2.3598e+010		15111		3.24766e+010		260		0.9		14759
30	2.3598e+010		15111		3.24767e+010		260		0.9		14759
31	2.3598e+010		15111		3.24767e+010		260		0.9		14759
32	2.3598e+010		15111		3.24767e+010		260		0.9		14759

Table 4. A table that includes values of specific variables over simulation time in third scenario

Apart from Nash equilibrium, there is also another significant point of these results. We refer to the fact that the second OEM manages to increase its value as much as to overcome the value of its competitor. Despite the fact that initial value of second OEM is lower than the initial value of the first OEM, second OEM manages to increase its value as much as it possible in order to overcome the value of its competitor. In figure 21 where is shown the graphs of both OEMs, it is obvious that the value of the second OEM is higher than the value of the first OEM since almost the seventh month of the simulation time.

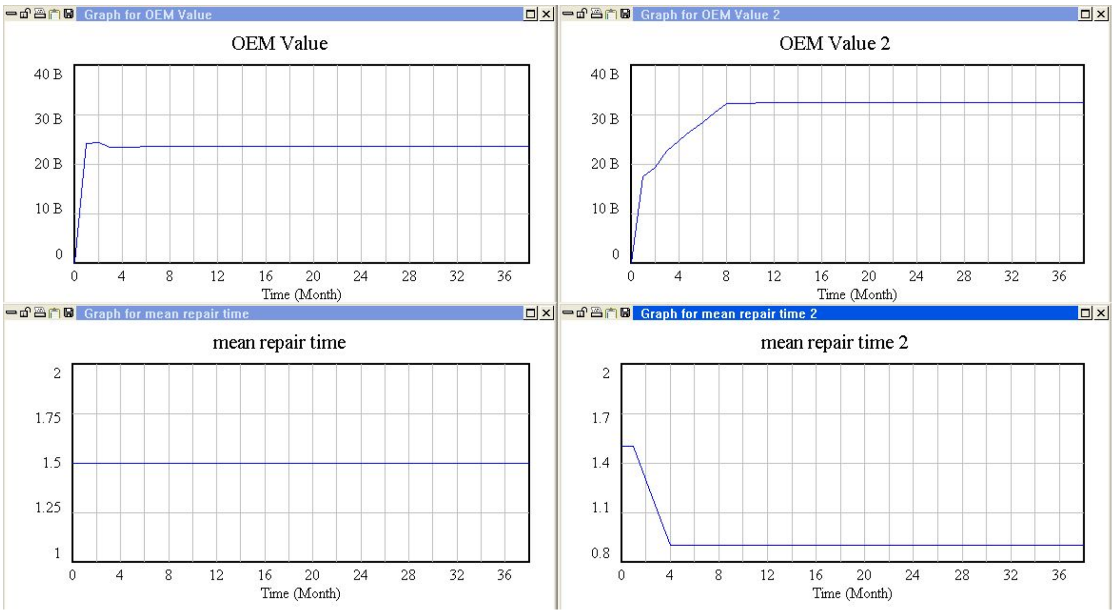


Figure 21. Graphs that represent the value of each OEM and the mean repair time of their collaborating dealers in third scenario

The lower graphs of figure 21 show the progress of the mean repair time of each repair service system. As you can see in the lower right graph, when mean repair time is reduced, the value of second OEM is increased rapidly. Certainly this happens in combination with increase of the average price per part of the second OEM. So, we can draw conclusion that the strategy which is followed by the second OEM is fully successful under conditions of competition because leads it to overcome its competitor even if it has lower value initially.

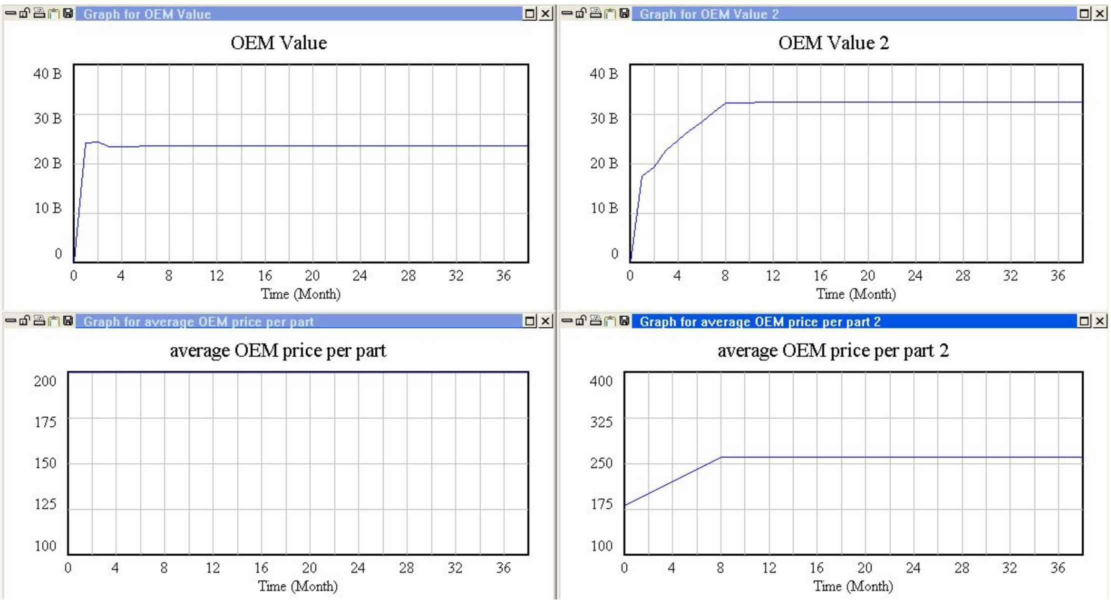


Figure 22. Graphs that represent the value of each OEM and the mean repair time of their collaborating dealers in third scenario

4.5.4 Fourth Scenario

In this scenario we try to study examine something different from previous scenarios. We attempt to study the behavior of the weaker competitor when its strategy is almost dependent on the strategy of its competitor. This means that the second OEM will be a "follower", a competitor which has a strategy that follows the strategy of its competitor. Thus, we begin with the presentation of the strategy of the first OEM and then we will present the strategy of the second OEM which is the "follower". The strategy of the first OEM is almost the same as in previous scenarios and it is also based on the determination of the average price per part.

At the beginning of this strategy, OEM checks its value at current time concerning its value of previous time. If this value is increased then OEM will not change anything about its average price per part. In a different case, OEM will check the number of orders for its parts at current time concerning the same number of previous time. If this number is increased then OEM decides to increase the average price per part in order to increase its value. This action may reduce the satisfaction index of collaborating dealer. These dealers will increase the mean repair price and then the satisfaction index of their customers will be reduced. As a result, each of these customers will prefer to repair his car to the other OEM. Because of this the number of orders that the specific OEM receives will be reduced. OEM does not care about that because it wants to increase its value. In the case that the specific number is reduced then OEM will reduce its average price per part in order to increase the satisfaction index of collaborating dealers. This means that the satisfaction index of their customers will be increased and this fact will affect the numbers of orders for parts of OEM positively. In this way, OEM attempts to increase its value. Below, we define the algorithm of this strategy.

```
IF (OEM1 VALUE < PREVIOUS OEM1 VALUE)
        IF (ORDERS FOR PARTS OF OEM1 < ORDERS FOR PARTS OF OEM2)
                OEM1 PRICE = OEM1 PRICE - offset
        ELSE IF (ORDERS FOR PARTS OF OEM1 > ORDERS FOR PARTS OF OEM2)
                OEM1 PRICE = OEM1 PRICE + offset
ELSE IF (OEM1 VALUE > PREVIOUS OEM1 VALUE)
        OEM1 PRICE = OEM1 PRICE
```

On the other hand, the strategy that is followed by the second OEM it is different from the previous one. As we already mentioned, the second OEM is "follower" which means that the strategy of specific OEM is based on the average price per part of its competitor. According to this strategy, OEM examines its current value concerning its previous value. If its current value is higher than the previous one, then OEM will not change its average price per part. In different case, OEM compares the number of orders for its parts with the number of orders for parts of its competitor. If the number of orders for its parts is higher than the number of orders for parts of its competitor, then OEM will set its average price per part equal to the average price per part of its competitor. This means that OEM set its average price per part as much as the average price per part of its competitor increased by a sum. The specific OEM determines this sum depending on current circumstances each time. In this way, OEM attempts to increase its value and simultaneously attempts to be more competitive than before, despite the fact that the number of orders for parts of this OEM may be decreased. In the case that the specific number is lower than the corresponding number of its opponent, OEM will set its average price per part equal to the average price per part of its opponent decreased by a sum. OEM also determines this sum depending on current circumstances each time. So, we characterize this competitor as "follower" because its strategy determines its average price per part according to average price per part of its competitor. Essentially, according to its strategy second OEM "follows" the first OEM. We define the algorithm of this strategy below.

```
IF (OEM2 VALUE < PREVIOUS OEM2 VALUE)
        IF (ORDERS FOR PARTS OF OEM2 < ORDERS FOR PARTS OF OEM1)
                OEM2 PRICE = OEM1 PRICE - offset
        ELSE IF (ORDERS FOR PARTS OF OEM2 > ORDERS FOR PARTS OF OEM1)
                OEM2 PRICE = OEM1 PRICE + offset
ELSE IF (OEM2 VALUE > PREVIOUS OEM2 VALUE)
        OEM2 PRICE = OEM2 PRICE
```

We simulate our model including this scenario and we get the corresponding results. The simulation process has a range of a hundred months as previous scenarios. In figure 23 is shown four graphs.

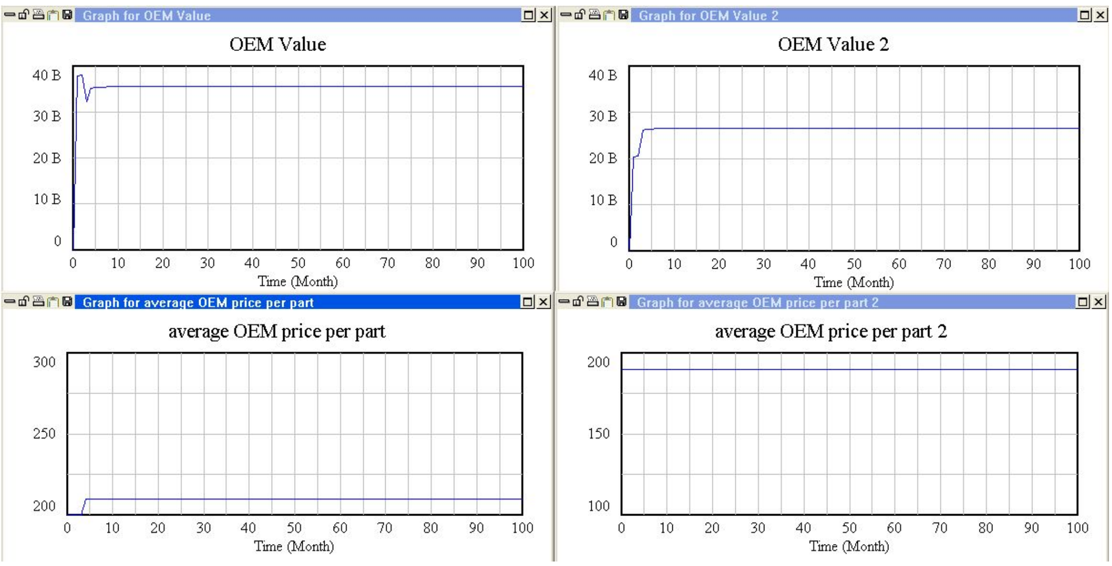


Figure 23. Graphs that represent the progress of value and average price per part of each OEM over simulation time in fourth scenario

The upper graphs depict the progress of the variable that represents the value of each OEM during the simulation time. The lower graphs depict the progress of the variable that represents the number of orders for parts of each OEM during simulation time.

Initially, the average price of the first OEM is higher than the average price per part of the second one and the first OEM receives more orders than the second one. At the third month the value of the first OEM is decreased because the number of orders that OEM receives is decreased while the number of orders for parts of its competitor is increased. Because of this fact, second OEM increases its average price per part in order to increase its value. Indeed, second OEM manages to increase its value at fourth month, as is shown in Table 5. During the remaining simulation time, we observe that the average price per part of each OEM and the number of orders for parts of each OEM remain stable.

Time (Month	OEM Value	"OEM Value	OEM Value 2	"average	average OEM p	"average	average OEM p	"af" Runs:	af	"af 2" Runs:	af 2
0	1.26153	2" Runs:	1.26722	OEM price	200	OEM price	190	test	25000	test	15000
1	3.78058e+010	test	2.02987e+010	per part"	200	per part 2"	190		25000		15000
2	3.81044e+010		2.04533e+010	Runs:	200	Runs:	190		25000		15000
3	3.21729e+010		2.6097e+010	test	200	test	190		20976		19024
4	3.51234e+010		2.62182e+010		210		190		20976		19024
5	3.52718e+010		2.63094e+010		210		190		20976		19024
6	3.53438e+010		2.64085e+010		210		190		20953		19047
7	3.54251e+010		2.64575e+010		210		190		20953		19047
8	3.54839e+010		2.64926e+010		210		190		20953		19047
9	3.5526e+010		2.65177e+010		210		190		20953		19047
10	3.5556e+010		2.65355e+010		210		190		20953		19047
11	3.55772e+010		2.65481e+010		210		190		20953		19047
12	3.55922e+010		2.6557e+010		210		190		20953		19047
13	3.56028e+010		2.65632e+010		210		190		20953		19047
14	3.56102e+010		2.65676e+010		210		190		20953		19047
15	3.56154e+010		2.65707e+010		210		190		20953		19047
16	3.56191e+010		2.65728e+010		210		190		20953		19047
17	3.56217e+010		2.65743e+010		210		190		20953		19047
18	3.56234e+010		2.65754e+010		210		190		20953		19047
19	3.56247e+010		2.65761e+010		210		190		20953		19047
20	3.56256e+010		2.65767e+010		210		190		20953		19047
21	3.56262e+010		2.6577e+010		210		190		20953		19047
22	3.56266e+010		2.65773e+010		210		190		20953		19047
23	3.56269e+010		2.65774e+010		210		190		20953		19047
24	3.56271e+010		2.65776e+010		210		190		20953		19047
25	3.56273e+010		2.65777e+010		210		190		20953		19047
26	3.56274e+010		2.65777e+010		210		190		20953		19047
27	3.56275e+010		2.65778e+010		210		190		20953		19047
28	3.56275e+010		2.65778e+010		210		190		20953		19047
29	3.56276e+010		2.65778e+010		210		190		20953		19047
30	3.56276e+010		2.65778e+010		210		190		20953		19047
31	3.56276e+010		2.65778e+010		210		190		20953		19047
32	3.56276e+010		2.65778e+010		210		190		20953		19047

Table 5. A table that includes values of specific variables over simulation time in fourth scenario

We also draw the same conclusion if we observe the graphs in figure 24. Comparing the upper left graph with the lower left graph we see that the value of the first OEM is decreased when the number of orders that received by the first OEM is decreased. On the other hand, the value of the second OEM is increased while the number of orders for its parts is increased at the same time. It is obvious, that these values of both OEMs remain stable from almost the sixth month until the end of simulation process. Due to this fact, we understand that no one of the competitors has an incentive to deviate unilaterally from its choice about its average price per part. This means that the combination of these strategies leads both OEMs to a steady state which is named as Nash equilibrium, under conditions of competition. In a further discussion about simulation results includes, we can say that the strategy that is followed by the second

OEM is more successful than the strategy of the first OEM, given that the first OEM chooses the specific strategy. In addition, the strategy of OEM is not successful enough given that the second OEM chooses the specific strategy.

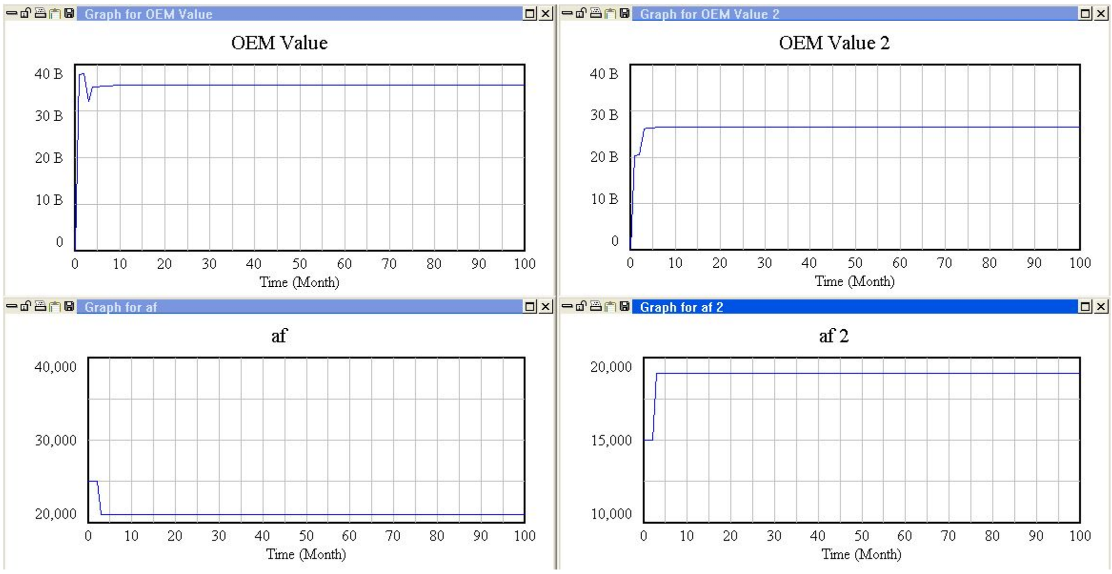


Figure 24. Graphs that represent the progress of value and number of orders for parts of each OEM over simulation time in fourth scenario

4.5.5 Fifth Scenario

The last scenario is different from previous scenarios because comprises one more repair service system. We assume that the previous original equipment manufacturers have similar values and are almost equivalent. However, the OEM of third repair service system has smaller value than the previous values. This means that third OEM has the role of "weaker" in this scenario. The main goal of this scenario is to examine the behavior of both equivalent competitors when a new competitor is added to the market. The initial assumptions in this scenario includes that OEM of third service system is collaborated with fewer number of dealers than its competitors. In addition, the specific OEM has fewer employers than its competitors. We also assume that the initial average price of third OEM is 185 while the average price per part of first OEM is 200 and the average price per part of second OEM is 195. The number of orders for parts of each OEM it is computed according to value of each OEM initially.

The strategy of each one is similar with the strategies of its competitors. According to rationale of this strategy, OEM examines its value at current time concerning its value at previous time. If its value is increased, then OEM will not change its average price per part. In opposite case, OEM compares the number of orders for its parts with the number of orders for parts of its competitors. If this number is lower than the number of both competitors, then OEM will decrease its average price per part in order to receive more orders in the future. In different case, OEM does not change its average price per part. Below we define the algorithm of this strategy.

```
IF (OEM1 VALUE < PREVIOUS OEM1 VALUE)
        IF (ORDERS FOR PARTS OF OEM1 < ORDERS FOR PARTS OF OEM2 OR OEM3)
                OEM1 PRICE = OEM1 PRICE - offset
        ELSE IF (ORDERS FOR PARTS OF OEM1 > ORDERS FOR PARTS OF OEM2 OR OEM3)
                OEM1 PRICE = OEM1 PRICE
ELSE IF (OEM1 VALUE > PREVIOUS OEM1 VALUE)
        OEM1 PRICE = OEM1 PRICE
```

After the definition of strategy, we continue with simulation process. In figure 25 is shown six graphs. The upper graphs depict the values of variables that represent the value of each OEM. The lower graphs depict the values of variables that represent the number of orders for parts of each OEM. As you can see in the upper left graph, the value of first OEM is increased in the first months of simulation process. At third month the value of first OEM is decrease because of reduction of orders for parts of first OEM. At the same time, the number of orders for parts of second OEM is also decreased while the corresponding variable of third OEM is increased. This happens because the average price per part of third OEM is lower than the average price per part of its competitors. This means that car owners prefer to collaborate with dealers that associated to third OEM. Thus, the value of third OEM is increased since the first months while the value of its competitors is decreased.

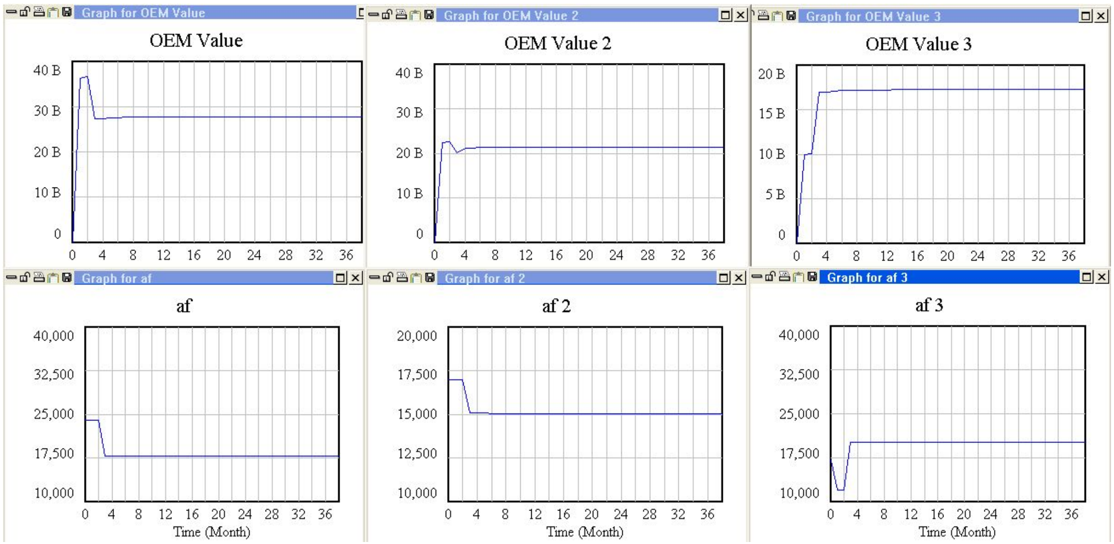


Figure 25. Graphs that represent the progress of value and number of orders for parts of each OEM over simulation time

In Table 6 is shown a table that comprises the values of specific variables of this scenario. As you can see Table 6 includes the values of each OEM's value over simulation time and also the number of orders that each OEM receives each month of simulation process. We mark with green boxes the range of values that remain stable during simulation process. This means that on one of three OEMs change its average price per part since fourth month as is shown in figure 26. In this figure are also depicted graphs that contrast the values of each OEM value with the average price per part of the corresponding OEM. Thus, if we combine the fact that the number of orders for parts of each OEM remains stable with the fact that average price per part of each OEM remains stable at same time, then we will draw a conclusion that no one of these competitors has an incentive to deviate unilaterally from its choice about its average price per part. So, the combination of strategies that are followed by competitors leads them to a steady state that named Nash equilibrium. We also observe that third OEM manages to increase its value during simulation process and be a more competitive opponent than at the beginning of process. Nevertheless, the specific strategy is more profitable for a weaker competitor than a stronger one.

Time (Month)	OEM Value	"OEM Value 2" Runs:	OEM Value 2	"OEM Value 3" Runs:	OEM Value 3	"af" Runs:	af	"af 2" Runs:	af 2	"af 3" Runs:	af 3
0	1.26153		1.27011		1.26722		24000		17000		17000
1	3.62936e+010	test	2.23323e+010	test	9.9359e+009	test	24000	test	17000	test	12000
2	3.65802e+010		2.24992e+010		1.00116e+010		24000		17000		12000
3	2.72679e+010		2.00402e+010		1.69292e+010		17778		15053		20170
4	2.74007e+010		2.11387e+010		1.70079e+010		17778		15053		20170
5	2.75011e+010		2.12165e+010		1.7067e+010		17778		15053		20170
6	2.75893e+010		2.12486e+010		1.71183e+010		17787		15035		20179
7	2.76434e+010		2.12904e+010		1.715e+010		17787		15035		20179
8	2.76824e+010		2.13205e+010		1.71728e+010		17787		15035		20179
9	2.77102e+010		2.13421e+010		1.71891e+010		17787		15035		20179
10	2.77299e+010		2.13574e+010		1.72006e+010		17787		15035		20179
11	2.77439e+010		2.13682e+010		1.72088e+010		17787		15035		20179
12	2.77538e+010		2.13758e+010		1.72145e+010		17787		15035		20179
13	2.77607e+010		2.13811e+010		1.72186e+010		17787		15035		20179
14	2.77656e+010		2.13849e+010		1.72214e+010		17787		15035		20179
15	2.7769e+010		2.13876e+010		1.72234e+010		17787		15035		20179
16	2.77714e+010		2.13894e+010		1.72248e+010		17787		15035		20179
17	2.7773e+010		2.13907e+010		1.72258e+010		17787		15035		20179
18	2.77742e+010		2.13916e+010		1.72265e+010		17787		15035		20179
19	2.7775e+010		2.13922e+010		1.72269e+010		17787		15035		20179
20	2.77756e+010		2.13927e+010		1.72273e+010		17787		15035		20179
21	2.7776e+010		2.1393e+010		1.72275e+010		17787		15035		20179
22	2.77763e+010		2.13932e+010		1.72277e+010		17787		15035		20179
23	2.77765e+010		2.13934e+010		1.72278e+010		17787		15035		20179
24	2.77766e+010		2.13935e+010		1.72279e+010		17787		15035		20179
25	2.77767e+010		2.13936e+010		1.72279e+010		17787		15035		20179
26	2.77768e+010		2.13936e+010		1.7228e+010		17787		15035		20179
27	2.77768e+010		2.13936e+010		1.7228e+010		17787		15035		20179
28	2.77769e+010		2.13937e+010		1.7228e+010		17787		15035		20179
29	2.77769e+010		2.13937e+010		1.7228e+010		17787		15035		20179
30	2.77769e+010		2.13937e+010		1.7228e+010		17787		15035		20179
31	2.77769e+010		2.13937e+010		1.7228e+010		17787		15035		20179
32	2.77769e+010		2.13937e+010		1.7228e+010		17787		15035		20179

Table 6. A table that includes values of specific variables over simulation time in fifth scenario

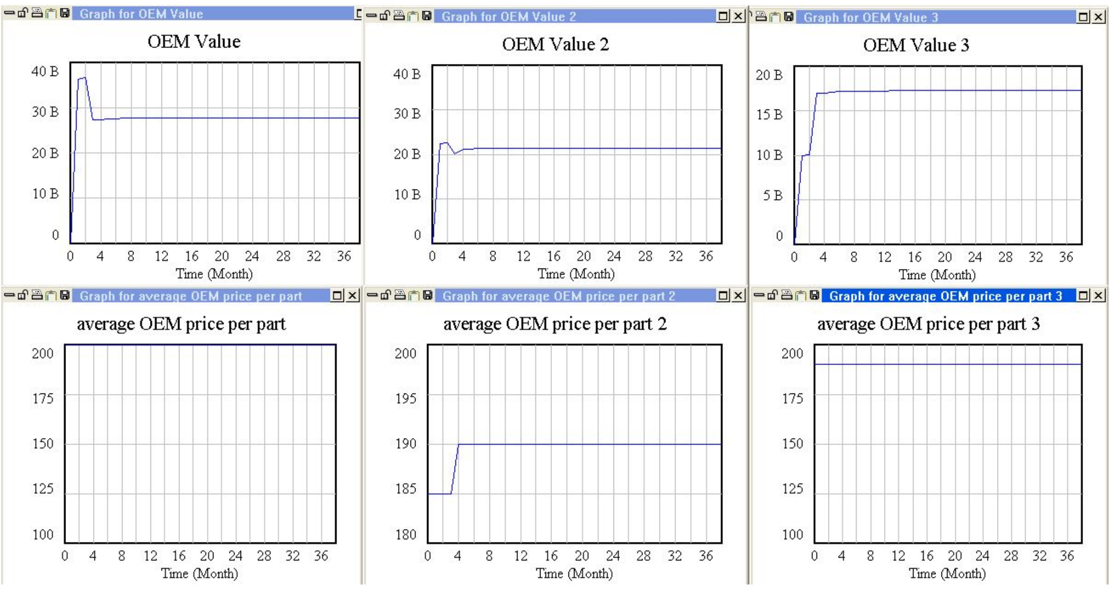


Figure 26. Graphs that represent the progress of value and average price per part of each OEM over simulation time

4.5.6 An experiment based on average price per part

We also implement an experiment that includes a provision on competition between two OEMs in order to draw conclusions about the strategy of each firm over time. There are different experiments based on entirely different assumptions and cover the entire spectrum of competition between these two firms. One of them is based on repair price that OEM sells its repair service to the car owner – customer. We researched the reactions (functions) of each OEM given that the competitive firm has already changed its strategy in order to maximize its sales. In addition, we will research if there is Nash equilibrium [26] among the main repair price of both firms.

However, we have to determine the main features of this experiment making a more detailed introduction. We assume that in the world there are two competitive firms, OEM 1 and OEM 2 which produce and repair cars. The customers – car owners have two possible choices in order to buy or repair their cars. We will focus much more on repair than sale of cars because of representing the service department of the firms. In more detail, we assume that at the beginning of the experiment both of firms have the same numbers of customers', which is represented in Vensim model by the variable named customers 1 and customers 2, respectively. At the specific experiment, we assume that the corresponding customers for each firm are 1500000. For the customers there are only two firms which sales or repairs cars in their world. If a customer is disappointed from one firm, then he will choose the other firm to repair his cars. This customer will be considered as a leaving customer about the first firm and will be considered as an incoming customer for the other firm. So, the number of customers in our Vensim model is always shared between these firms corresponds to variable named seed customers. This value of seed customers is also remains stable during the simulation process. Therefore, the values of customers 1 and customers 2 will be changed every simulation process step. If value of customers 1 will be increased, then value of customers 2 will be decreased and vice versa. The number of customers of each firm is completely affected from Dealer Satisfaction index of each firm. This index indicates how much satisfied are the customers of each firm. The definition of customer satisfaction has two main dependencies: a) main repair price, b) main repair time. At this experiment we will focus of the variability of the first dependency and we consider that the second one will remain stable and same for both firms. Main repair price depends in a large proportion on average OEM price per part, which is the price that the firms sales any parts of a car to the dealers. As we have already referred about the model, dealers are the organizations in which the customers communicate on repairing their cars. Dealers paid and may received complaints or their concerns by customers about the quality of their services or about the repair price. If main repair price is not satisfactory enough for customers, happens in a large proportion due to average OEM price per part because Dealers buy their car parts from OEM. So, OEM is the main adjuster of main repair price and that is one of two reasons which will examine the variability of average OEM price per part, at this experiment. Below, we will analyze the second reason.

The specific experiment also relies on the assumption that both firms offer services instead of products which are intangible. More specifically, as we mentioned at chapter 2 about the definition of tangible and intangible goods, both firms offer car's repair as a product which is intangible. So, we cannot count how many products have been sold by each firm. According to Verna Allee, there are tangible and intangible goods. In our model, every dealer repairs each customer's car. Dealers offer services to customers which belong to intangible goods. The concept of quantity now is replaced by the concept of service. Each OEM does not produce products, but it produces services. We also assume at this experiment that every customer represents one service. So, if any dealer has 100 customers that mean it can sell 100 car repair services. We assume that every customer owns one car, so we calculate the number of customers and the number of services, too. Thus, we calculate the current production of each firm. This is very important calculation because this variable in our model, affects the revenues and the costs of each firm. It also affects the costs and the revenues of its supplier, which adjust the value of OEM supplier. By extension, it affects the value of the firm which is one of our basic concepts. So, if a firm increases the number of its customers, then it will increase its production because each customer represents a car service. Thus, the firm has an incentive to increase its customers through the increase of dealer satisfaction index, which deflects the how satisfied are the customers. So, if OEM sales its parts to dealer in convenient prices for the dealer, then dealer will offer the car service in convenient price for the customer. It is a virtual chain which has OEM as a first ring and customers as the last one. If this happened, then customer will increase its satisfaction and simultaneously increase the reputation of OEM which supplies the specific dealer. So, as long the firm expands its reputation to our world as most customers would leave the rival firm to cooperate with it. As a result, the specific firm will increase its revenues, so its value too. Therefore, we reach to the main purpose of each firm. However, if all these cases happened and the number of its customers increases, then firm shall decline its average price per part, according to law of demand definition. Each firm has an incentive to increase their value as is possible. Thus, we will try to find which the optimal average price per part for each firm, given that the rival firm has a specific average price per part during each simulation process. If we will find any intersection point between graphical diagrams for both firms that means:

(i) The value of average OEM price per part 1 is optimal, given that average OEM price per part 2 has a specific value, so OEM 1 has no incentive to deviate to change its strategy,

(ii) The value of average OEM price per part 2 is optimal, given that average OEM price per part 1 has a specific value, so OEM 2 has no incentive to deviate to change its strategy.

After determining the concept of experiment, we will continue with the analysis of the experiment. As we mentioned above, we will rely on volatility of average OEM price per part which almost completely affects the main repair price.

Furthermore, relying on Nash equilibrium principle, we will research if there are any points, at least one, that both OEM maximize its own value. Thus, we will research which the maximum value of variable named OEM value 1 is, given that variable average OEM repair price per part 2 has a specific and stable value during the whole simulation process. After finishing this process, we will do the opposite experiment in order to examine the case of Nash equilibrium. Specifically, if this is true, then it will be a point of intersection between the graphical of both average OEM repair price per part. This point will be the optimal pair of average OEM repair price, where no firm has an incentive to deviate unilaterally from its value. Our target is to find such a point and a great proof is a graphical which represents it.

Therefore, we start the experiment assuming that the average OEM repair price per part 1 is equal to 120 and will be stable during the whole simulation process. Using a variable which actually represent the simulation time we will increase the average OEM repair price per part 2 by 20 every 2 months in order to make it as much real as we can. Because is not real that a firm increase its repair price every month. Duration of two months gives a little opportunity to each firm to examine its specific strategy and see the results. After the end of simulation process, we notice the value of average OEM repair price per part 2 where the OEM value 2 has maximum value. We will continue executing this simulation process increasing the value of average repair price per part 1 by 10. Finally, we will see the results of this part of experiment below. As we have already mentioned, the simulation range will be 48 months, so will have the chance to examine a quite satisfactory graphical diagram.

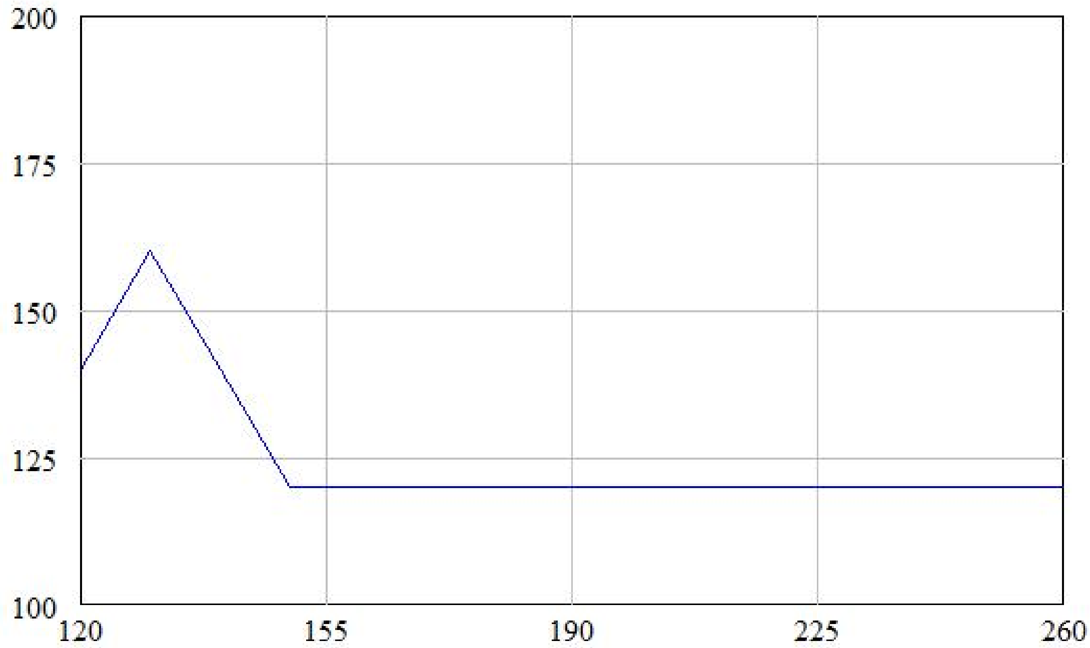
Below, Table 7 includes two columns. The first column represents the chosen values of average OEM price per part 1 and the second column represents the values of average OEM price per part 2, which maximize the OEM value 2 given that the average OEM price per part 2 which has the value of the first column at the corresponding row. (As we mentioned the first firm has the number 1 at the end of every variable of its model and the second firm has the number 2, respectively).

Average OEM price per part 1	Average OEM price per part 2
120	140
130	160
140	140
150	120
160	120
170	120
180	120
190	120
200	120
210	120
220	120
230	120
240	120
250	120
260	120

Table 7. The values of average OEM price per part 2, which maximize the OEM value 2 given each possible output choice of average OEM price per part 1 at the first column

If we try to study Table 7, we will easily understand that when we reach value of average OEM price per part 1 which is bigger or equal to 150, then value of average OEM price per part 2 remain stable at the value equal to 120. Therefore, our conclusions from Table 7 will be represented from a Graphical Diagram below.

Actually, the Graphical Diagram 1 confirms the conclusions of Table 7 and shows us a more integrated sensation about the progress of this experiment. We complete the first step in order to reach Nash equilibrium. Let us move to the second step and do exactly the same as before reversing the variables.



Graphical Diagram 1. X axis represents average OEM price per part 1, while Y axis represents average OEM price per part 2

Firstly, we create the corresponding table, which represents the results of simulation process of our model. Thus, we can see Table 8, which includes two and fifteen rows. The first column represents the chosen values of average OEM price per part 2 and the second column represents the values of average OEM price per part 1, which maximize the OEM value 1 given that the average OEM price per part 1 which has the value of the first column at the corresponding row. (As we mentioned the first firm has the number 2 at the end of every variable of its model and the second firm has the number 1 respectively).

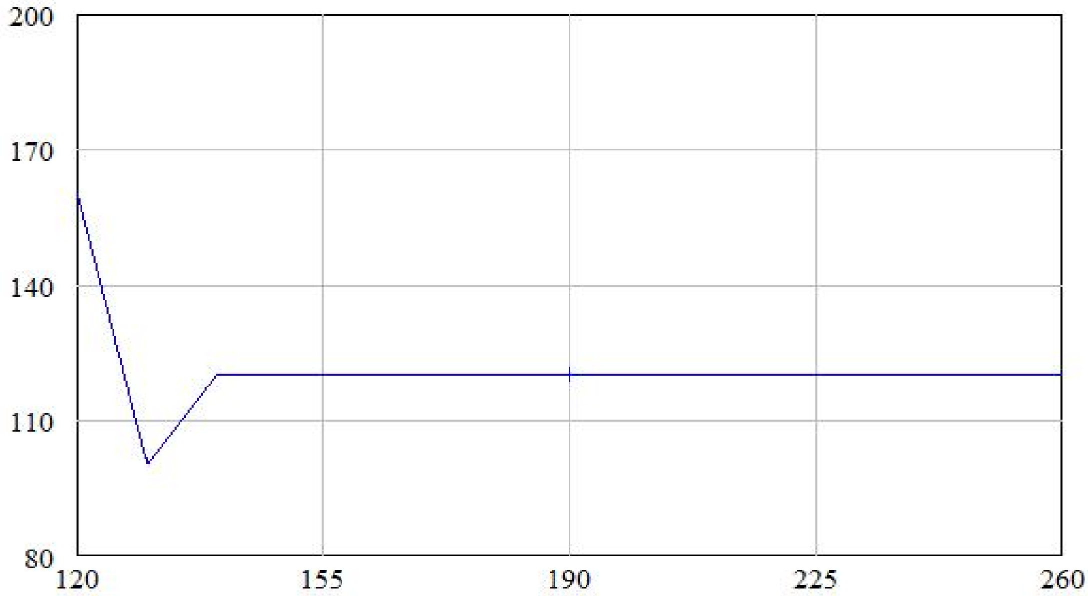
Average OEM price per part 2	Average OEM price per part 1
120	140
130	160
140	140
150	120
160	120
170	120
180	120
190	120
200	120
210	120
220	120
230	120
240	120
250	120
260	120

Table 8. The values of average OEM price per part 1, which maximize OEM value 1 given each possible output choice of average OEM price per part 2 at the first column.

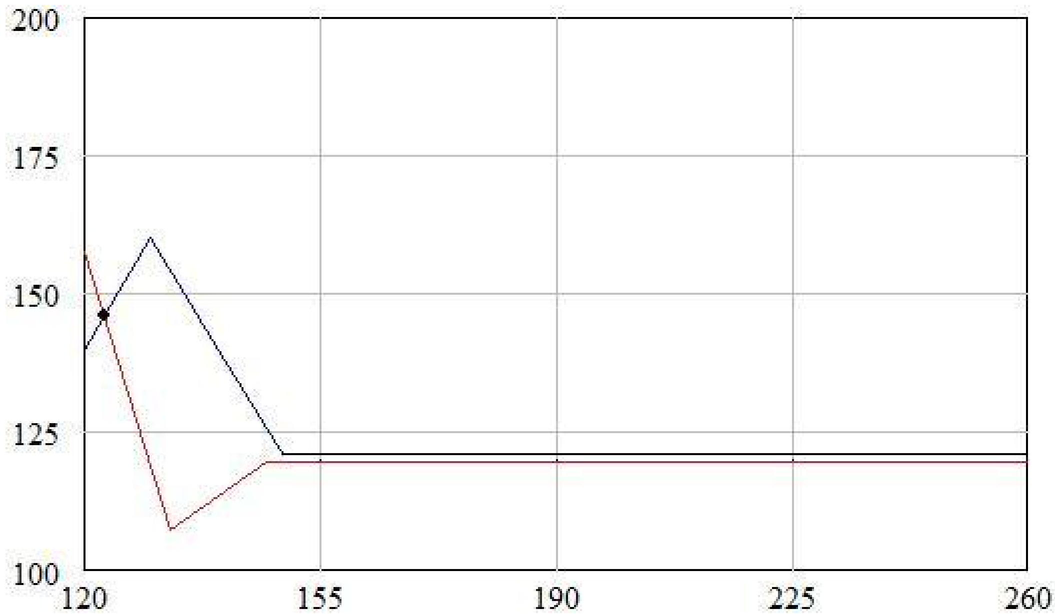
Looking at Table 8 carefully, we understand that there is a correspondence between these two experiments that have done by us. So, at the beginning of the experiment it is noticeable that the values of average OEM price per part 1 had instability.

Due to the fact that it could be expressed at the Graphical Diagram and it looks also like Graphical Diagram 1. However, we are close to find a point where two Graphical Diagrams are intersected. This point of intersection is our proof of being Nash equilibrium in this experiment. This finding will be better expressed through the following Graphical Diagram.

Now we have to compare the graphical diagrams and research if it possible to find an intersection between those two graphs. We create a new aggregated graphical diagram where both of the previous graphs will be viewed. It is about Graphical Diagram 3 below.



Graphical Diagram 2. X axis represents average OEM price per part 2, while Y axis represents average OEM price per part 1



Graphical Diagram 3. The graph with red color corresponds to the second step of this experiment and the other graph with the black color corresponds to the first step of this experiment

The Nash equilibrium is represented by the point of intersection of the two lines, which means that is the optimal value of average OEM price per part of both firms. So, the optimal average OEM price per part is 140.

Finally, at the end of this experiment we draw a conclusion that there is a pair of values that leads these firms to Nash equilibrium. This experiment is not based on competition of real world but it is a proof that Nash Equilibrium can be applied in our e-business model.

5. Exporting from Service Network Analysis & Prediction Tool (SNAPT) to Vensim tool

The Service Network Analysis & Prediction Tool (SNAPT) is a Model-Driven Architecture (MDA) platform that enables modeling and analysis of Service Networks. SNAPT was developed in a modular way, to host future task-specific, extended Service Network models. SNAPT incorporates the Service Network and a KPI model discussed in [33], facilitates cost - revenue analysis and includes an extensive feature set to achieve its partial goals. So, the core Service Network model could be expanded in many ways to facilitate both analysis and implementation of complex Services. SNAPT should be in the center of these emerging methodologies. To facilitate value analysis and prediction, SNAPT could be connected to well-established simulation platforms.

As we mentioned in previous chapter, Vensim is a software tool which is used for developing, analyzing, and packaging high quality dynamic feedback models. It seemed almost an ideal tool in simulating Service Network models of SNAPT. We face the difficulties in bridging the gap between SNAPT and Vensim tool capitalizing on advantage of extensibility of SNAPT we implemented a SNAPT extension that will transform service network models to dynamic systems modeled in Vensim. It will also be able to simulate and predict service network’s value taking into consideration factors like business competition. This SNAPT extension is actually a plugin and it is named SNAPT2Vensim. Obviously, its name is inspired from the connection of both software tools.

Before we start presenting the implementation of plugin, we mention to SNAPT's meta-model that followed when a service network is modeled. The functionality of SNAPT2Vensim is based on the adjustment of a SNAPT's service network model from a system dynamic model of Vensim. Firstly, we will present the features of service network meta-model that was based on the implementation of SNAPT that is proposed in [33] and also can be features of any system dynamics model that will be created through the usage of SNAPT2Vensim plugin. Secondly, we will be a more detailed presentation about the whole implementation of SNAPT2Vensim.

5.1 The Service Network meta-model of SNAPT

We consider that Service Network theory based on the concept that any business can be modeled as a service. Even if we don't deal with intangible goods but we deal with tangible goods such as products, it is the transport of the product that indicates the service offered to the End Customer. For example, in the car industry domain, we can model the process of manufacturing a car as a Service Network that includes all Business Entities incorporated and working together in order to make the delivery of a car. Thus, Service Networks can be a practical application of modeling any kind of businesses.

A second thought of car industry example gives us the opportunity to observe that if there is a group of entities, then each entity will can participate and cooperate with each other entity of the group through consuming or offering services. Thus, except car industry example there are many more examples that could be applicable by Service Network. Nevertheless, if an analyst wants to model a Service Network, then he will not face any restrictions or any limitations of the assumptions that would be made by the analyst. Certainly, the existence of a Service Network implies that there is a service or a group of services that an entity delivers to another entity.

According to service network meta-model of SNAPT which proposed in [33], *Business Entities* and *Services* are the fundamental parts of a Service Network. Services define the connection among any form of Business Entities of Service Network. In SNAPT's GMF editor, a Service is depicted by a double arrow with opposite directions. As you can see in Figure 1, any Participant or Enabler or End Customer are connected with another Business Entity using an arrow that has two opposite directions. Each direction defines a transaction among the correlated Business Entities. Thus, each double arrow has exactly one source and a target Business Entity. Service Offering is depicted by the dotted arrow, while Service Consumption is depicted by the solid arrow.

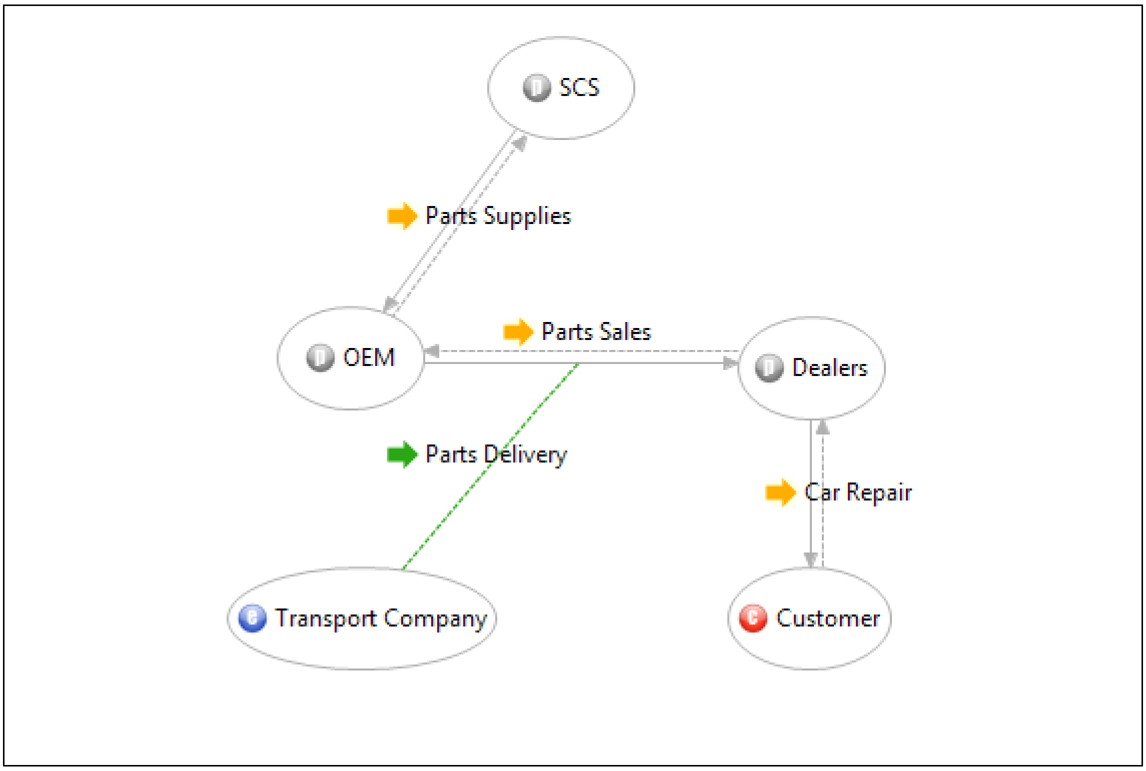


Figure 27. An example of SNAPT's Service Network designed in GMF editor of SNAPT. Except simple Services includes an Enablement Service

Service Offering relations correlate the source Business Entity, which is also the service provider, with the offered Service. Service Consumption relations analyze the connection between the Entity of Service Network which consumed the specific offered service, with the offered service.

There are various classes of Business Entities such as *Participant*, *End Customer*, *Enabler* and *Sub-Network*. Each class of Business Entities determines the features and functionality of the nodes of the Service Network. A Business Entity can be *Participant*, which is the simplest form of a Business Entity. So, any Business Entity not being Enabler or End Customer should be modeled as Participant. A Business Entity can also be *End Customer*, which is the Entity that can only consume services (the destination of the service transaction). Finally can be *Enabler*, which is the Entity that acts in order to enable the delivery of other services. As an example of an Enabler, we suggest a transport company which make the transport of a product that been already ordered by a customer through e-bay. Nevertheless, the transport company offers a service which actually enables the delivery of the product that customer have already ordered from e-bay. Due to the incompatibility of designing of a Sub-Network in Vensim tool, we will not comprise this form of Business Entity in the implementation of SNAPT2Vensim plugin. Business Entities also play a Role in the network, such as "Audi" which is the Business Entity with the Role of "car manufacturer". In SNAPT, every Business Entity and every Service of Service Networks has a name are uniquely identified by an identifier. The existence of Services and Business Entities in Service Networks generates costs and revenues. Revenues actually are the sums of the money earned by source Business Entity, which is the Service provider. Each Business Entity has total costs and total revenues which are assigned from the services that each Business Entity offers or consumes. There is another form of Service that is offered by an Enabler. As we presented above, an Enabler does not offer a Service, but delivers the Service. This form of Service is named *Enablement*.

5.2 Creating Vensim model using SNAPT2Vensim plugin

Generally, a Service Network model of SNAPT consists of nodes and links which determine the relation among the nodes. A system dynamics model in Vensim tool also consists of variables and arrows that represent the relations and specifically the dependencies among the variables. Actually, the main functionality of SNAPT2Vensim plugin is based on the automatic conversion from a Service Network model that is designed in SNAPT to a system dynamics model that is compatible in Vensim tool.

Firstly, we start with a fundamental part of a SNAPT's Service Network, Business Entity. As we mentioned above, a node of Service Network in SNAPT is modelled by a Business Entity. So, we need in Vensim tool a type of variable which can represent the most features of any class of Business Entity. In this way and taking into account the features of Business Entity, we chose *Variable-Auxiliary/Constant*. This type of variable has the most features that we need to model a Business Entity in Vensim tool. An *auxiliary/Constant* Variable comprises not only *constant variables*, *auxiliary* variables but also *Data* variables, *Level* variables and other types of variables.
A Constant variable does not change over time and also can be temporarily changed prior to simulating a model. An auxiliary variable is computed from Levels, Constants, Data and other auxiliaries. Auxiliary variables have no memory, and their current values

are independent of the values of variables at previous times. Thus, they are two types of variables in Vensim tool that can be used to model any type of Business Entity, either be a *Participant*, either be an *End Customer*, either be an *Enabler*. Nevertheless, we can store the whole information that is exported from SNAPT's Service Network about every type of Business Entity, into a constant or an auxiliary variable of Vensim tool. So, we modelled any type of Business Entity as a constant or auxiliary variable, in Vensim tool.

Another issue is information that we have to store about every Business Entity.

Firstly, we give the name of every Business Entity to the corresponding variable in Vensim tool. Actually, we are interested in value of every Business Entity. So, every variable in produced Vensim model will have after its name the word "*Value*", in order to emphasize the significance of Value. Certainly, any user can change the name of any variable.

Secondly, a variable in Vensim model represents a Business Entity, or an Enabler, or a Service of the specific SNAPT's Service Network model and simultaneously the value of each variable in Vensim model is computed by the equation: *Value = Revenues – Costs*. The left side of equation is the name of each variable and at the right side consists of the subtraction between *revenues* and *costs*. So, we use the information from Service Network model of SNAPT and we put the amount of revenues of each Business Entity, or Service, or Enablement as *revenues* of the specific variable in Vensim tool. Afterwards, we are taking into account the c*osts* of each Participant, or Enabler, or End Customer, or Service, or Enablement. The estimation of *costs* is more different than the estimation of revenues because of the implementation and semantics of Vensim tool. According to semantics of Vensim tool, every arrow that begins from one variable and reaches to another creates a dependency. This dependency requires that the equation of target variable must include the value of source variable that is actually the cost of target variable. In this way, we subtract the value of every source variable from the existing value of target variable. Thus, we have constructed the equations among the variables of Vensim model successfully. Certainly, the equation of every variable it is editable at any time by user.

The window that is shown in figure 2 is the equation editor that Vensim tool provides to user. The upper field includes the name of variable and it must be unique. The name of each variable is the left side of equation. At the right side of equation is the value of variable which is already computed as we described above.

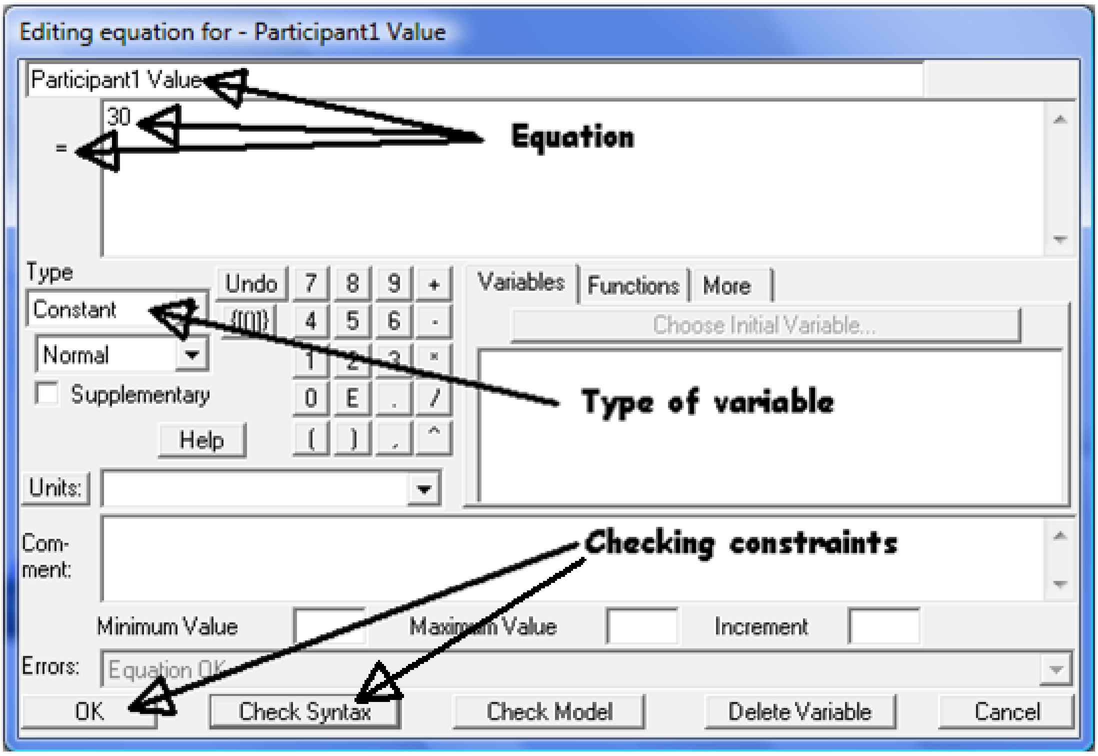


Figure 28. An example of equation editor of a variable in Vensim tool

Below those fields as you can see in figure 2, is a dropdown box that includes the options of types of variables. The initial selection as we mentioned above is constant variable but a user can change a variable of the created Vensim model using variable editor as we presented above.

We continue with another significant part of SNAPT's Service Network which is Services. As in SNAPT's Service Network also in Vensim tool each arrow has one source variable and a target variable. Certainly, a single arrow has direction from source variable to target variable. As we mentioned in previous chapter, each arrow creates a dependency between its source and target variable. Each dependency includes that the equation of the target arrow must comprise the value of source variable. In a Service Network of SNAPT, services are modelled by double arrows, as we presented above. The implementation of double arrow is based on the need to represent the direction of Service Offering and the direction of Service Consumption. But in Vensim tool, a double arrow is not implemented and it provides exclusive a single arrow. Nevertheless, each service is depicted by a variable which is the source of two arrows. The target of one of these arrows is the variable that represents the participant which offers the service and the target of the other arrow is the participant that consumes the service. In following figures is shown how a service of a Service Network model is depicted in the corresponding Vensim model. The double arrow of a service network model of SNAPT is substituted with two arrows that have the same source but opposite directions. In the example of figure 30, the value computation of Dealers comprises the value of Part Sales because Dealers buy parts of OEM and pay for them. So, the equation that computes the value of Dealers comprises the subtraction of the value of Parts Sales. The equation that computes the value of OEM comprises the addition of the value of Parts sales because OEM gets revenue from selling its parts.

Figure 29. The view of a service in SNAPT

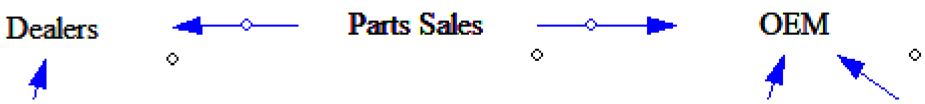


Figure 30.The view of the same service in Vensim tool

The significant part of this consideration is that in practise we have already computed the revenues of each Service Provider and simultaneously the costs of each Service Consumer through the execution of SNAPT2Vensim. Enablements are another issue that is important to discuss. As we mentioned above, enablement is another form of Service that offered by an Enabler. As you can see in figure 1, enablement is depicted as a dashed green line that starts from a Business Entity and ends to a Service double arrow. It is a way of visualizing that an Enabler does not offer a Service but delivers a Service. Actually, an Enabler offers another form of Service that facilitates a related Service to be successfully completed. In Vensim tool, we have a disadvantage about enablement. Does not exist any feature in order to design an enablement. Vensim tool does not provide any connection between arrows, single or double. Thus, we depict an enablement in Vensim tool as a variable which is a source of an arrow that is connected with the participant which consumes this service.

In figure 3 below, it is shown a Vensim model that was created from Service Network model of figure 1 using SNAPT2Vensim plugin of SNAPT. As you can see, the green dashed line that depicts the enablement offered by Transport Company to "Parts Sales" Service is substituted by a variable and the appropriate arrow to the correlated participants in the corresponding Vensim model. However, the Vensim model of figure 3 has obtained the whole information that is derived by the corresponding Service Network model. Thus, the created Vensim model has the proper information for simulation or further editing by user.

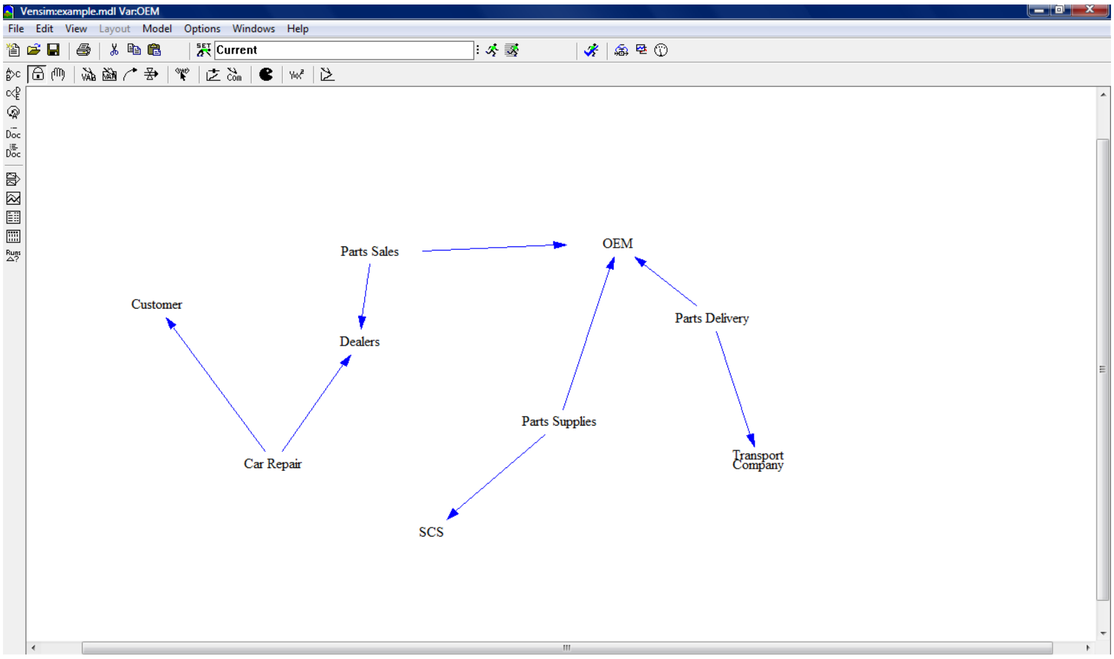


Figure 31. The Vensim model that was created from Service Network model of figure 1 using SNAPT2Vensim

5.3 Implementation of SNAPT2Vensim plugin

Service Network Analysis & Prediction Tool (SNAPT) is a tool developed on the Eclipse platform, based on extensibility of plugin. A plugin is independent component that provides additional function to the system. Each plugin may also include code, documentation or other data that can be used by other plugin. To integrate plugins with each other, the developer is enabled to define extension points in each plugin. An extensions point is a mechanism, which defines a type of contract that the interested plugins must agree with. Any interested plugin must implement this contract in order to achieve the integration, even if they know nothing for the plugin being extended beyond the scope of the extension point. Through mechanism of extension points we can implement anytime for example an additional option at menu of windows in SNAPT. We can also implement an additional export option from SNAPT or import option into

SNAPT. The logic of SNAPT2Vensim implementation is based on exporting a Service Network model of SNAPT and simultaneously creating a Vensim model that will have the whole information from the first one. We attempted to find a way of importing into Vensim tool any Service Network model that is exported from SNAPT, but we faced many difficulties. Firstly, Vensim tool is unable to import a file of type *sn_diagram*. Secondly, Vensim tool is not compatible with XML-based files and is unable to import XML-based files successfully. Thirdly, Vensim tool is able to import only *dat* files that include simulation results from previous simulations of current Vensim model. Thus, through importing any *dat* file into Vensim model automatically the current model retrieves the whole information. But *dat* files do not represent or describe Vensim models. Every file that describes and represents a Vensim model has a type named *mdl*. It is created through Vensim tool by user and is compatible with Vensim tool. Thus, we reached an impasse. The solution was given through the implementation of SNAPT. We used methods that are implemented in *ServiceNetwork* class and are related to each Service Network model that is modeled in SNAPT platform. The usage of these methods gives us the appropriate information in order to construct a Vensim model that corresponds to the specific Service Network model. We edited a file of a Vensim model and we studied the code that Vensim tool uses to describe the components of each model. So, we made an algorithm that retrieves information about a ServiceNetwork model and constructs a Vensim model according to specifications that we analyzed in previous paragraph. This construction of Vensim model will be done by creating a new file and writing into it the corresponding code of the specific ServiceNetwork information. Certainly, the creation and the construction of such a file is made by using java code. Thus, assuming that this algorithm works fine we take as a result of every SNAPT2Vensim execution is a created Vensim model that will be available to the user. So, we start the presentation of implementation of SNAPT2Vensim plugin with the part of visualization. We needed a way of visualizing an export option (from SNAPT to Vensim) to user. Thus, we created a new wizard that is named “Export Service Network to Vensim”.

To implement this wizard, we start by creating a java class named *ExportSnToVensimWizard* that extends the *Wizard* class and implements the *IExportWizard* interface. This java class is included in package that named “*org.eclipse.gmf.snapt.wizards*”. It is the package that comprises all java classes that are related implementation of a wizard. Eclipse framework defines the classes or interfaces that an Eclipse-based wizard should extend or implement. . This wizard is registered to SNAPT’s export wizards list using the “org.eclipse.ui.exportWizards” extension point, as we will see. Before the whole ExportSnToVensimWizard class implemented, we edited two significant files that are crucial in visualization of wizard. Both of them belong to *"org.eclipse.gmf.snapt.extensions"* java package of SNAPT. The first file is named *plugin.xml* and is used to define plugin extension points that we analyzed above. As is shown below in the black box, *extension* tag defines the extension point of export Wizards and that is common in all export extensions and it is defined once. The extension's definition includes *id*, *name* and *point* attributes. Afterwards, follows the *wizard* tag that correlates to our export wizard and its *class* attribute has the same value as the name of Java class that we created firstly. The *icon* attribute is used to display an icon next to wizard name (exportSnToVensim), as you can see in figure 4 below. The *category* attribute has a name that is related to System Dynamics and it is a link to the following *category* tag. The *name* tag is a variable that is related to the second file that we mentioned above. We refer to *plugin.properties* which is a file that used to define values to various properties that are used in *plugin.xml* like these at lines 10 and 14 of the XML code below. More precisely, the attribute at line 10 defines the name that will be displayed next to icon at page of SNAPT export wizards.

```
(plugin.xml)

1.<extension
2. id="org.eclipse.gmf.snapt.exportwizard"
3. name="Export Wizard Extension Point"
4. point="org.eclipse.ui.exportWizards">
5.   <wizard
6.      category="org.eclipse.gmf.snapt.extensions.
                                    categoryExportSystemDynamics"
7.      class="org.eclipse.gmf.snapt.wizards.ExportSnToVensimWizard"
8.      icon="icons/Vensim.bmp"
9.      id="org.eclipse.gmf.snapt.extensions.wizardSnToVensim"
10.     name="%wizards.sd.sntoVensim.name">
11.  </wizard>
12.  <category
13.     id="org.eclipse.gmf.snapt.extensions.
                                    categoryExportSystemDynamics"
14.     name="%wizards.sd.name">
15.  </category>
```

The attribute at line 14 defines the label of the category that includes our wizard. *Category* attribute of line 6 matches with the *id* attribute of category tag and then will be displayed the value of variable at line 14 as the name of category that our wizard belongs to. Below are shown the properties of plugin.properties that are correlated to our wizard.

```
(plugin.properties)

wizards.sd.name = System Dynamics
```

wizards.sd.sntoVensim.name = Export Service Network to Vensim

The results of editing these files are displayed in figure 4. It is the page of SNAPT export wizards that also includes ExportSnToVensimWizard. As you can see, there are two arrows that indicate the point that category and wizard are displayed. So, we complete the initialization of our export wizard and we continue with the java code of *ExportSnToVensimWizard* class.

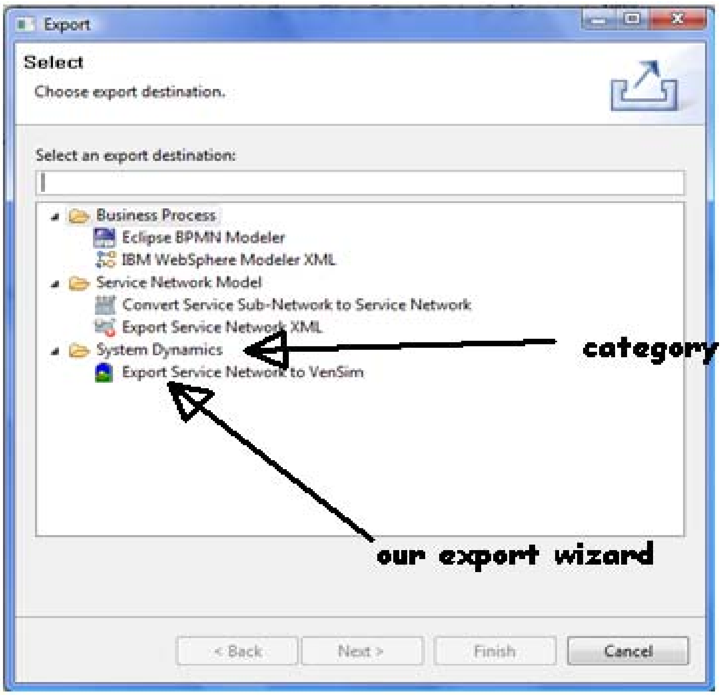


Figure 32. The page of SNAPT export wizards

In appendix, we exhibit the java code that represents our java class implementation. *ExportSnToVensimWizard* class has some methods but few of them are significant about its functionality. We start with method that is named *init* and initializes the page of *ExportSnToVensimWizard*. The initialization is about every feature of page such as: labels, icons, images and other information that are explicitly related to each wizard. This method is triggered when user selects exportSnToVensim wizard from page of export wizards. Another significant method named *overwriteIfFileExists*. This method is triggered when user press the "Browse…" button on the wizard dialog. Then user is able to choose a file path and the first extends it by adding a selection box that displays all available resources from where the user will choose. The last important method of this java class is named *performFinish* and is triggered when user press the "Finish" button. Then a new instance of another class is created. It is *VensimNetwork* class that we implemented in order to construct a Vensim model through current Service Network model of SNAPT. This java class is included in a package that named "*org.eclipse.gmf.snapt.Vensim*". The implementation of *VensimNetwork* class includes three methods that reflect the functionality of the whole class. We start with the constructor of the class that has the same name as class. The requirements of this method are an instance of the *ServiceNetwork* class and file name of current Service Network. It is a constructor method which means that it initializes each instance of *VensimNetwork* class. The next method to present is named *transform2Vensim*. Actually, it is the method that has the kernel of implementation of SNAPT2Vensim. At first lines of this java method three new *Elists* are created in java code. The first *EList* includes the instances of Business Entities that exist in Service Network model. The second *EList* includes the instances of Service Offerings that offered among the Business Entities. The third *EList* includes the Service Enablements that offered from an Enabler to another Service. Then, a java iterator gets the total cost and the name of each instance of Business Entities from first *EList*. Another java iterator gets the name, the revenues and the source of each instance of Enablements from second *EList*. A last java iterator gets the source Business Entity, the target Business Entity and the revenue of each Service Offering from third *EList*. Thus, essential information about the current Service Network model is assembled. The last part of the algorithm is the writing of Vensim model file in appropriate manner by using information that has been got above. In Appendix we exhibit the java code that implements the above algorithm. In addition, we exhibit a user manual about SNAPT2Vensim plugin.

6. Conclusion and future work

In this thesis we presented our approach of an e-business model that is based on fundamental concepts of value networks including the role of trust and risk and it is used to analyze and compute the values of its partners over time. The value computation and analysis also includes the additional value that is generated by the relationships which are developed through intangible exchanges among partners [7]. Apart from definition of the main formal structure of our e-business model we have proposed an application of our model in the repair service system as a part of the automotive industry of real world. The specific application of our model is an example of a car manufacturing value chain.

The primary contribution of this thesis is the proposition of our approach of the e-business model that is proposed in reference 18. The differences of our model with the model of reference 18 are based on value computation of specific partners. We also apply our business model to the repair service system as a part of automotive industry. The secondary contribution of this thesis is the designing, building, simulation and analysis of the specific application of our e-business model using Vensim tool. We used this simulation tool because is a visual modeling tool that allows us to conceptualize, document, simulate, analyze, and optimize the specific example of our model. We described the business objectives, difficulties, dependencies and the equations that compute the value of each partner as is defined by the formal structure of our model. We presented the view of each partner as it modeled in Vensim tool. A further discussion could be done about the tool that we used. We presented an alternative tool that has similar specifications and is named iThink and it is studied in master thesis [17].

We enhanced the example of our model with a second repair service system because the primary goal of this thesis is to examine how each OEM can increase its value under the conditions of competition. Towards accomplishing this goal contributes the application of Game Theory [30] in our model in Vensim tool. Assuming that the OEMs of our model are players in a game of Game Theory, we defined various scenarios where each OEM follows a specific strategy in order to increase its value in value network. Each strategy comprises specific actions that each OEM has to choose given that its competitor is already made a specific action. The main contribution of this thesis is the simulation each of these scenarios in order to examine if could be a steady state where no one of competitors has an incentive to deviate unilaterally from its choice and named Nash equilibrium. We found that there is a pair of values that leads the competitors to a Nash equilibrium [32] in every scenario that we defined. It is remarkable that at most scenarios the strategy of weaker OEM is more profitable than the strategy of its competitor. Further research is necessary towards examining the behavior of other partners in our model using Vensim or a similar tool. In this way, we can make a prediction of each partner behavior from more views.

The last contribution of this thesis is the implementation of SNAPT2Vensim plugin. It is created towards bridging the gap between SNAPT [33] [34] and Vensim tool. Due to the fact that SNAPT is implemented on the concept of plugins, we implemented a SNAPT extension which transforms service network models to systems dynamic models which are built in Vensim tool. Any file that is exported from SNAPT using this plugin is able to be visualized into Vensim tool keeping the whole information from service network which is modeled in SNAPT. In this way, using the specific plugin we are able to simulate and predict the value of service networks taking into account factors like business competition.

Bibliography

1. **C. Y. Baldwin and K. B. Clark,** "Managing in an Age of Modularity," *in Harvard Business Review on Managing the Value Chain,* Harvard Business Press, Boston, (2000)

2. **Brandenburg A. and Nalebuff B., J.**, *Co-opetition: A Revolution Mindset That Combines Competition and Cooperation : The Game Theory Strategy That's Changing the Game of Business*, Doubleday Book, NY, 1997

3. **Normann, R. and Ramirez, R.** "From Value Chain to Value Constellation: Designing Interactive Strategy", *Harvard Business Review*, Jul-Aug 1993, pp. 65-77

4. **Parolini, C.**, *The Value Net: A Tool for Competitive Strategy*, John Wiley & Sons Ltd, England, 1999

5. **Gordijn, J., Akkermans J.M. and Vliet J.C. Van**, "Business Modeling is not Process Modeling", In Conceptual Modeling for E-Business and the Web, LNCS 1921, Springer -Verlag, 2000.40—51

6. **Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., Llayperuma, T.**, "Strategic Analysis using Value Modelling – a c3 approach", In *Prooceedings of the 40th Hawaii InternationaI Conference on System Sciences* -2007

7. **Allee, V.**, *Understanding Value Networks*. [Online], 1998-2000. http://www.vernaallee.com/value_networks/Understanding_Value_Networks.html

8. **Hakanson and Johanson**, "A model of Industrial Network: A Review", In *Industrial Networks: A New View of Reality*, Axelsson, B., and Easton, G., eds., Routledge, London. 1992, pp. 28-34

9. **V. Allee**, "Reconfiguring the Value Network," *Journal of Business Strategy* **21**, No. 4 (2000), http://www.vernaallee.com/value_networks/

Reconfiguring_the_Value_Network.pdf

10. **Barney J.B.**, "Firm resources and sustained competitive advantage", *Journal of Management*, 17, 1991, 99-120

11. **G. D. Koutras.** *Model transformations to Leverage service networks and implementation of value network tool.* Computer Science Department. s.l. : University of Crete, 2009

12. *Estimating value in service systems: A case study of a repair service system.* **N. S. Caswell, C. Nikolaou, J. Sairamesh, M. Bitsaki, G. D. Koutras and G. Iacovidis.** 1, s.l. : IBM, 2008, IBM Systems Journal, Vol. 47

13. ‘‘Estimating Value in Value Networks,’’ **N. Caswell, S. Feldman, C. Nikolaou, J. Sairamesh, and M. Bitsaki**, http://www.tsl.csd.uoc.gr/media/workingpaper_value_nets.pdf

14. **Roger B. Myerson**, “Game Theory: Analysis of Conflict”, 1991

15. **Faruk Gul**, 2008. "Behavioural economics and game theory," The New Palgrave Dictionary of Economics, 2nd Edition

16. **Camerer, Colin** (2003), *Behavioral game theory: experiments in strategic interaction*, Russell Sage Foundation

17. **M. Voskakis.** *Simulation and optimization of value in Service Networks.* Computer Science Department. s.l. : University of Crete, 2010

18. —. Vensim tool tutorial. [Online] http://www.Vensim.com/documentation.html

19. **Michael J. Radzicki and Robert A. Taylor**, "Origin of System Dynamics: Jay W. Forrester and the History of System Dynamics". In: *U.S. Department of Energy's Introduction to System Dynamics*. Retrieved 23 Oktober 2008

20. **Martin J. Osborne**, “An introduction to game theory”, Version: 2002/7/23. Martin.Osborne@utoronto.ca. [Online] http://www.economics.utoronto.ca/osborne

21. **Johnson, Michael D.; Anders Gustafssonb, Tor Wallin Andreassenc, Line Lervikc and Jaesung Cha** (2001). "The evolution and future of national customer satisfaction index models". *Journal of Economic Psychology*

22. **G. Box, G. M. Jenkins, and G. Reinsel**, Time Series Analysis: Forecasting and Control, Prentice Hall, Upper Saddle River, NJ (1994)

23. **Harsanyi, John C., and Reinhard Selten**. 1998. *A General Theory of Equilibrium Selection in Games*. Cambridge, MA: MIT Press

24. **Chaos, Cheating and Cooperation**: *Potential Solutions to the Prisoner's Dilemma*, Björn Bremb

25. **Kerin, Roger, P.R. Varadarajan, and Robert Peterson**. "*First-Mover Advantage: A Synthesis, Conceptual Framework, and Research Propositions.*" *Journal of Marketing* 56, no. 4 (1996): 33–52

26. **James W. Friedman**, *Reaction Function as Nash Equilibria*, University of Rochester

27. **Johnson, Michael D.; Anders Gustafssonb, Tor Wallin Andreassenc, Line Lervikc and Jaesung Cha** (2001). "The evolution and future of national customer satisfaction index models". Journal of Economic Psychology

28. **G. Box, G. M. Jenkins, and G. Reinsel**, Time Series Analysis: Forecasting and Control, Prentice Hall, Upper Saddle River, NJ (1994)

29. **Harsanyi, John C., and Reinhard Selten**. 1998. A General Theory of Equilibrium Selection in Games. Cambridge, MA: MIT Press

30. Chaos, Cheating and Cooperation: Potential Solutionfs to the Prisoner's Dilemma, **Björn Brembs**

31. **Kerin, Roger, P.R. Varadarajan, and Robert Peterson**. "First-Mover Advantage: A Synthesis, Conceptual Framework, and Research Propositions." Journal of Marketing 56, no. 4 (1996): 33–52

32. **James W. Friedman**, Reaction Function as Nash Equilibria, University of Rochester

33. **P.Petridis.** "Towards a universal Service Network-centric framework to design, implement and monitor Services in complex Service Ecosystems: The Service Network Analysis & Prediction Tool (SNAPT)", Computer Science Department, University of Crete, 2010

34. **G. Stratakis.** "Analyzing Service Networks from different perspectives using the Service Network Analysis & Prediction Tool (SNAPT)"

Appendix

SNAPT2Vensim plugin java code & user manual

1. The main algorithm of SNAPT2Vensim plugin

```
package org.eclipse.gmf.snapt.vensim;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.HashMap;
import java.util.Iterator;

import org.eclipse.emf.common.util.EList;
import org.eclipse.gmf.snapt.sn.*;

public class VenSimNetwork {

        private ServiceNetwork sn;
        private String file;

        public VenSimNetwork(ServiceNetwork sn, String file){

                this.sn = sn;
                this.file = file;

        }

        public void transform2Vensim(){

                String [] Names = new String[50];

                // Participants
                String [] participants = new String[50];
                Double [] part_total_cost = new Double[50];

                // Services
                String [] Services = new String[50];
                String [] serv_source = new String[50];
                String [] serv_target = new String[50];
                Double [] serv_revenue = new Double[50];

                // Enablements
                String [] Enablements = new String[50];
                String [] enable_target = new String[50];
                Enabler [] SourceEnabler = new Enabler[50];
                String [] Enablers = new String [50];
                Double [] enable_revenue = new Double[50];

                //---------------------------------------------------------------------------

                EList<BusinessEntity> entities  = sn.getBusinessEntities();
```

```
EList<Service> offerings = sn.getOfferings();
EList<ServiceEnablement> enablements = sn.getEnablements();

Iterator<BusinessEntity> biz_iterator = entities.iterator();
Iterator<Service> serv_iterator = offerings.iterator();
Iterator<ServiceEnablement> enable_iterator = enablements.iterator();

int index = 0;

while(biz_iterator.hasNext()){

        BusinessEntity ben = (BusinessEntity)biz_iterator.next();

        Names[index] = ben.getName();

        part_total_cost[index] = ben.getTotalCost();

        index = index + 1;
}

int enable_index = 0;

while(enable_iterator.hasNext())
{
        ServiceEnablement enable =
(ServiceEnablement)enable_iterator.next();
        enable_revenue[enable_index] = enable.getRevenue();

        Enablements[enable_index] = enable.getName();
        SourceEnabler[enable_index] = enable.getSourceEnabler();
        Enablers[enable_index] = enable.getSourceEnabler().getName();
        enable_target[enable_index] =
enable.getTargetService().getSourceBusinessEntity().getName();

        enable_index = enable_index + 1;
}

int s_index = 0;

while(serv_iterator.hasNext()){

        Service s = (Service)serv_iterator.next();

        BusinessEntity source = s.getSourceBusinessEntity();
        BusinessEntity target = s.getTargetBusinessEntity();
        serv_source[s_index] = source.getName();
        serv_target[s_index] = target.getName();
        serv_revenue[s_index]= s.getRevenue();
        Services[s_index] = s.getName();

        s_index = s_index + 1;

}

try{
  // Create file
  FileWriter fstream = new FileWriter(file);
BufferedWriter out = new BufferedWriter(fstream);
```

```
out.write("{UTF-8}\n\n");

for(int i = 0; i < index; i++)
{
        out.write(Names[i] +"  = A FUNCTION OF( ");

    for(int j = 0; j < s_index ; j++)
    {
        if(Names[i].equals(serv_target[j]) == true)
        {
            out.write(Services[j] + ", ");

        }
        else if(Names[i].equals(serv_source[j]) == true)
        {
            out.write(Services[j] + ", ");
        }
    }
    for(int j = 0; j < enable_index ; j++)
    {
        if(Names[i].equals(Enablers[j]) == true)
        {
            out.write(Enablements[j] + ", ");
        }
        else if(Names[i].equals(enable_target[j]) == true)
        {
            out.write(Enablements[j] + ", ");
        }
    }
    out.write(Names[i] + ") ~~|\n");
    out.write(Names[i]+ "=\n");
    out.write("\t");

    for(int j = 0; j < s_index ; j++)
    {
        if(Names[i].equals(serv_target[j]) == true)
        {
            out.write(" - " + Services[j]);
        }
        else if(Names[i].equals(serv_source[j]) == true)
        {
            out.write(" + " + Services[j]);
        }

    }
    for(int j = 0; j < enable_index ; j++)
    {
        if(Names[i].equals(enable_target[j]) == true)
        {
            out.write(" - " + Enablements[j]);
        }
        else if(Names[i].equals(Enablers[j]) == true)
        {
            out.write(" + " + Enablements[j]);

        }
    }
    out.write("\t~\n");
    out.write("\t~\t|\n\n");
```

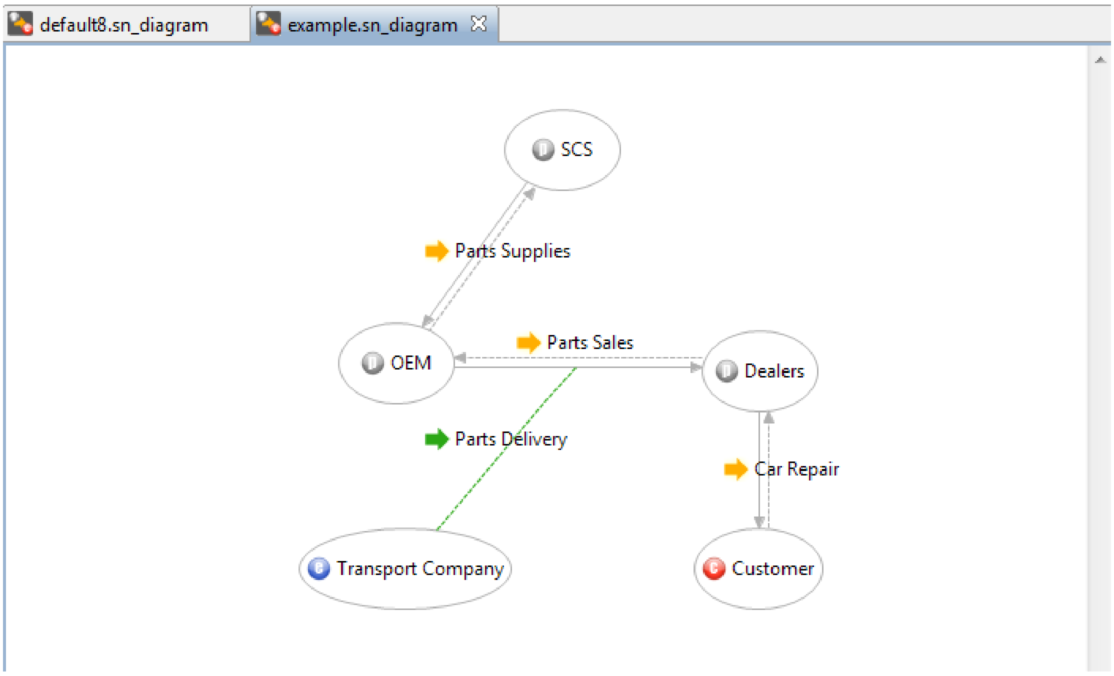
```
                }

                for(int s = 0; s < s_index; s++)
                    {
                        out.write(Services[s]+"= INTEG (\n\t 1,\n\t\t" + serv_revenue[s]+")\n\t~\n\t~\t\t|\n");
                    }
                for(int en = 0; en < enable_index; en++)
                    {
                        out.write(Enablements[en]+"= INTEG (\n\t 1,\n\t\t" + enable_revenue[en]+")\n\t~\n\t~\t\t|\n");
                    }

                out.write("*******************************************************\n");
                out.write("\t.Control\n");
                out.write("*******************************************************\n");
                out.write("\t|\n");
                out.write("FINAL TIME  = 100\n");
                out.write("\t~\tMonth\n");
                out.write("\t~\tThe final time for the simulation.\n");
                out.write("\t|\n\n");
                out.write("INITIAL TIME  = 0\n");
                out.write("\t~\tMonth\n");
                out.write("\t~\tThe initial time for the simulation.\n");
                out.write("\t|\n\n");
                out.write("SAVEPER  =\n");
                out.write("\tTIME STEP\n");
                out.write("\t~\tMonth [0,?]\n");
                out.write("\t~\tThe frequency with which output is stored.\n");
                out.write("\t|\n\n");
                out.write("TIME STEP  = 1\n");
                out.write("\t~\tMonth [0,?]\n");
                out.write("\t~\tThe time step for the simulation.\n");
                out.write("\t|\n\n");
                out.write("\\\\\\---/// Sketch information - do not modify anything except names\n");
                out.write("V300  Do not put anything below this section - it will be ignored\n");
                out.write("*View 1\n");
                out.write("$192-192-192,0,Times New Roman|12||0-0-0|0-0-0|0-0-255|-1--1--1|-1--1--1|96,96,100,0\n");


                //-------------------------------------------------------------------//
                HashMap<String, Integer> map = new HashMap<String,Integer>();

                int counter_id = 1;

                for(int p = 0; p < index; p++ )
                {
                    map.put(Names[p], counter_id);
                    counter_id++;
                }
                for(int p = 0; p < s_index; p++)
                {
                    map.put(Services[p], counter_id);
                    counter_id++;
                }
                for(int p = 0; p < enable_index; p++)
```

```
{
        map.put(Enablements[p], counter_id);
        counter_id++;
}

//---------------------------------------------------------------//

counter_id = 1;
int temp_xi = 100;
int temp_yi = 100;

for(int p = 0; p < index; p++ )
{
        temp_xi = temp_xi + 120;
        out.write("10,"+ counter_id +","+ Names[p] +","+ temp_xi +","+
temp_yi +",60,15,8,3,0,0,0,0,0,0\n");

        counter_id++;
}

temp_xi = 160;
temp_yi = 300;

for(int p = 0; p < s_index; p++ )
{
        temp_xi = temp_xi + 120;
        out.write("10,"+ counter_id +","+ Services[p] +","+ temp_xi +","+
temp_yi +",60,15,8,3,0,0,0,0,0,0\n");

        counter_id++;
}

for(int p = 0; p < enable_index; p++ )
{
        temp_xi = temp_xi + 120;
        out.write("10,"+ counter_id +","+ Enablements[p] +","+ temp_xi
+","+ temp_yi +",60,15,8,3,0,0,0,0,0,0\n");

        counter_id++;
}
//---------------------------------------------------------------//

for(int p = 0; p < index; p++ )
{
        for(int j = 0; j < s_index ; j++)
        {
                //out.write("10,"+ counter_id +","+ Names[p] +","+ temp_xi
+","+ temp_yi +",60,15,8,3,0,0,0,0,0,0\n");

                if(Names[p].equals(serv_source[j]) == true)
                {
                        int f1 = find(map, index, Services[j]);
                        int f2 = find(map, index, Names[p]);
                        out.write("1,"+ counter_id +","+ f1 +","+ f2
+",0,0,0,0,0,64,0,-1--1--1,,1|("+ temp_xi +","+ temp_yi +")|\n");
                        counter_id = counter_id + 1;
                }
                else if(Names[p].equals(serv_target[j]) == true)
                {
```

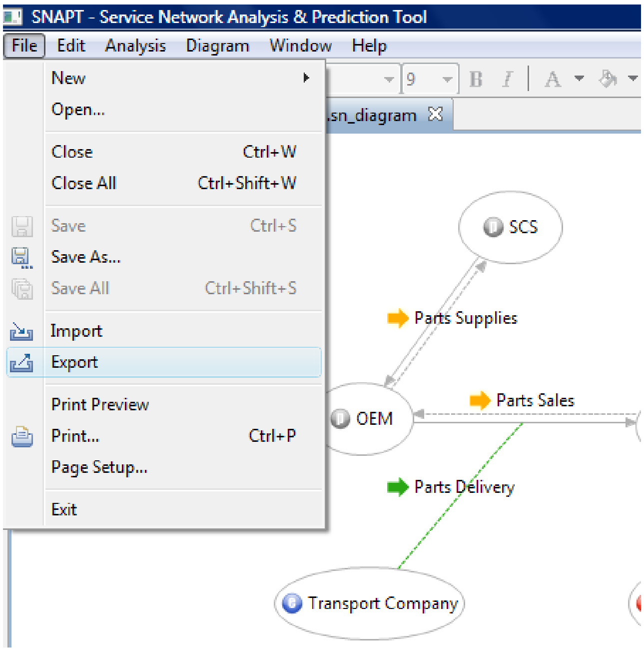
```
                                        int f1 = find(map, index, Services[j]);
                                        int f2 = find(map, index, Names[p]);

                                        out.write("1,"+ counter_id +","+ f1 +","+ f2
+",0,0,0,0,0,64,0,-1--1--1,,1|("+ temp_xi +","+ temp_yi +")|\n");
                                        counter_id = counter_id + 1;
                                }


                        }

                        for(int j = 0; j < enable_index ; j++)
                        {
                                if(Names[p].equals(enable_target[j]) == true)
                                {
                                        int f1 = find(map, index, Enablements[j]);
                                        int f2 = find(map, index, Names[p]);

                                        out.write("1,"+ counter_id +","+ f1 +","+ f2
+",0,0,0,0,0,64,0,-1--1--1,,1|("+ temp_xi +","+ temp_yi +")|\n");
                                        counter_id = counter_id + 1;

                                }
                                else if(Names[p].equals(Enablers[j]) == true)
                                {
                                        int f1 = find(map, index, Enablements[j]);
                                        int f2 = find(map, index, Names[p]);

                                        out.write("1,"+ counter_id +","+ f1 +","+ f2
+",0,0,0,0,0,64,0,-1--1--1,,1|("+ temp_xi +","+ temp_yi +")|\n");
                                        counter_id = counter_id + 1;
                                }
                        }
                }
                out.write("///---\\\\\\\\\\\\n");
                out.write(":L<%^E!@\n");
                out.write("1:Current.vdf\n");
                out.write("9:Current\n");

out.write("23:0\n15:0,0,0,0,0,0\n19:100,0\n27:2,\n34:0,\n4:Time\n24:0\n25:100\n26:1
00\n");

                //Close the output stream
                out.close();;
                }catch (Exception e){//Catch exception if any
                  e.printStackTrace();
                }
        }

        public int find(HashMap<String, Integer> map, int index, String x)
        {
                for(int n=0;n < index;n++)
                {
                        if(map.containsKey(x) == true)
                {
                        return map.get(x);
                }}
                return -1;
        }}
```

2. User manual of SNAPT2Vensim plugin

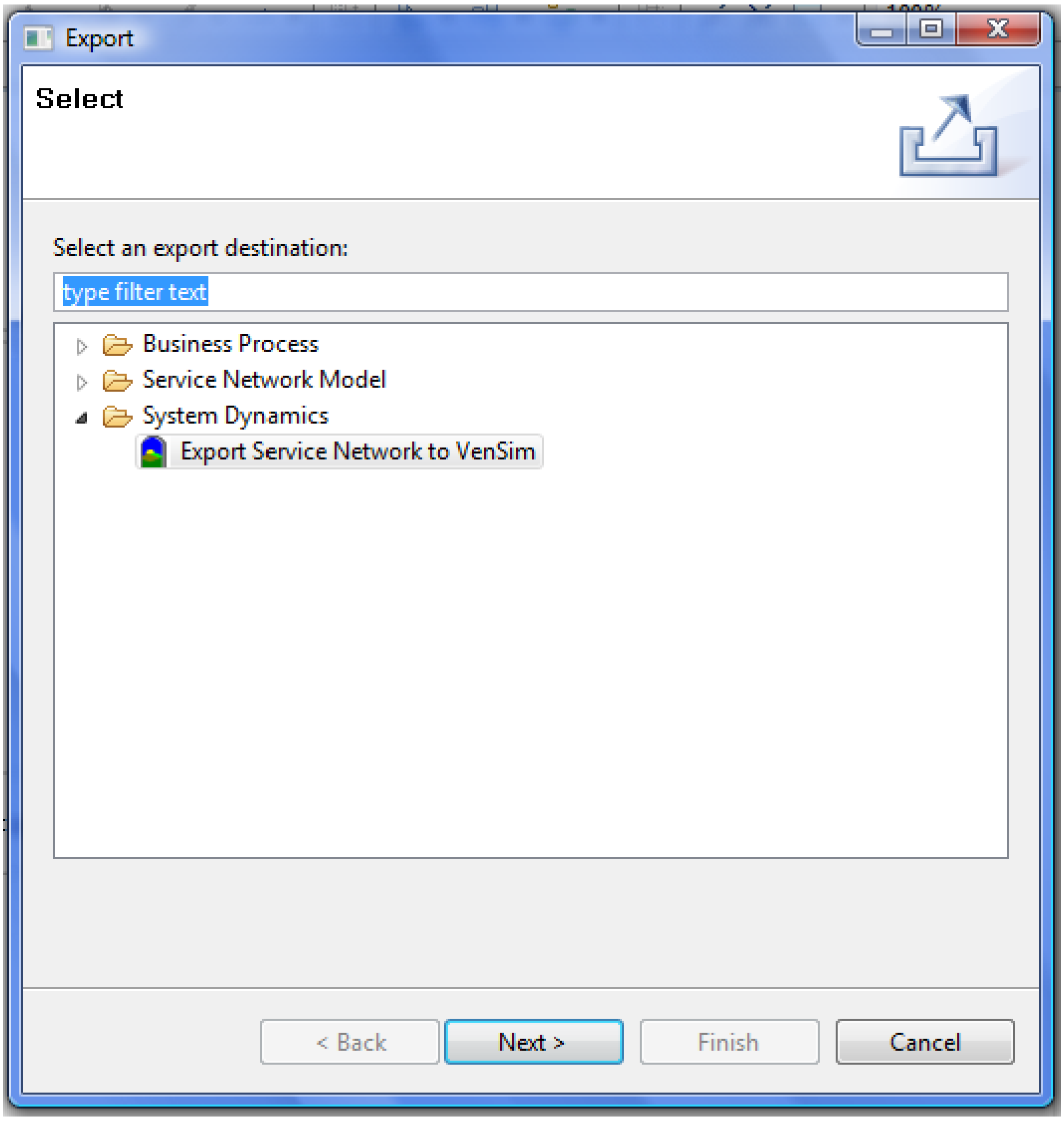
First of all, let us assume that user has already modeled the SN editors' diagram at SNAPT gmf editor that is shown at figure below.



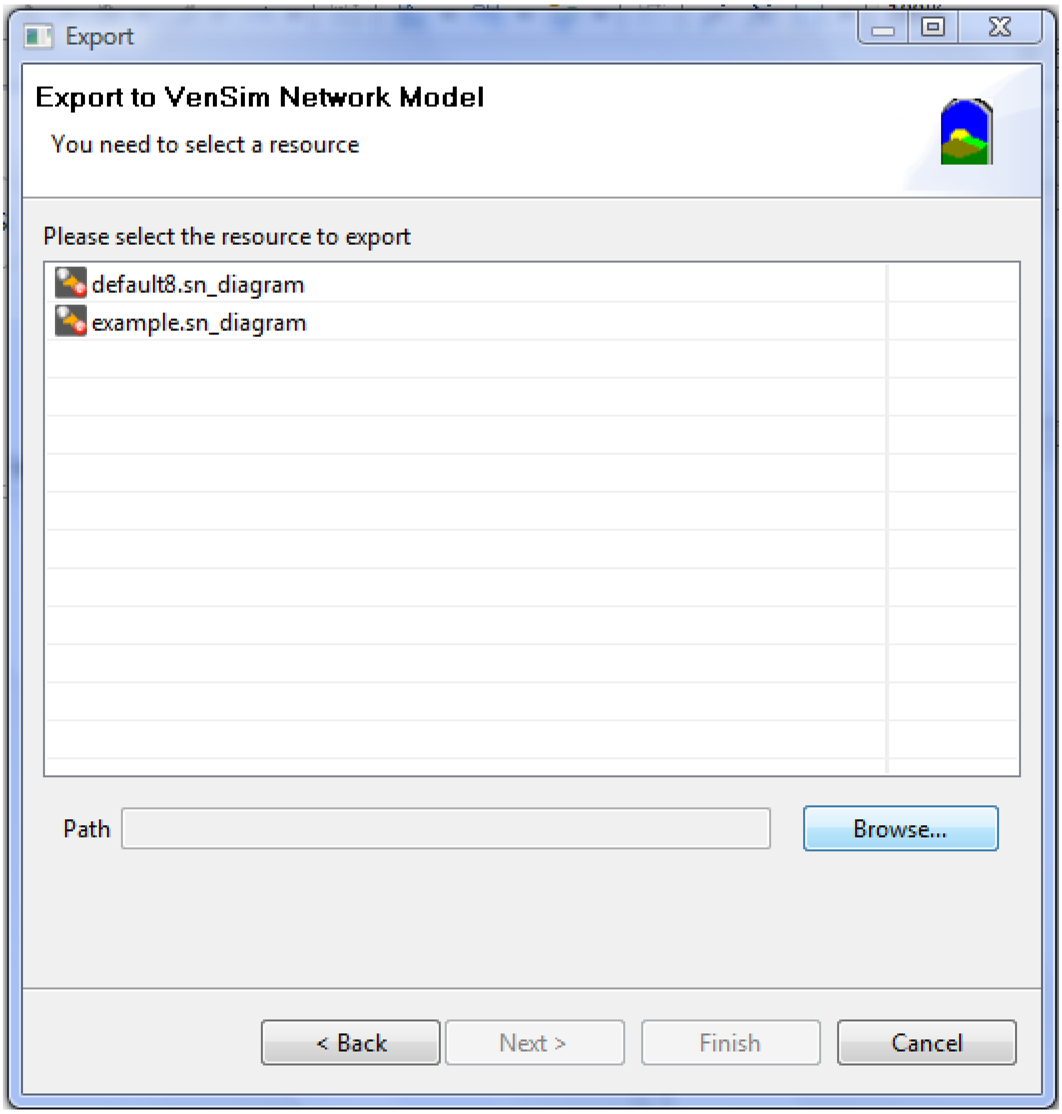
STEP 1: User is willing to export it, so selects File-> Export at top menu bar, as is shown at figure below.



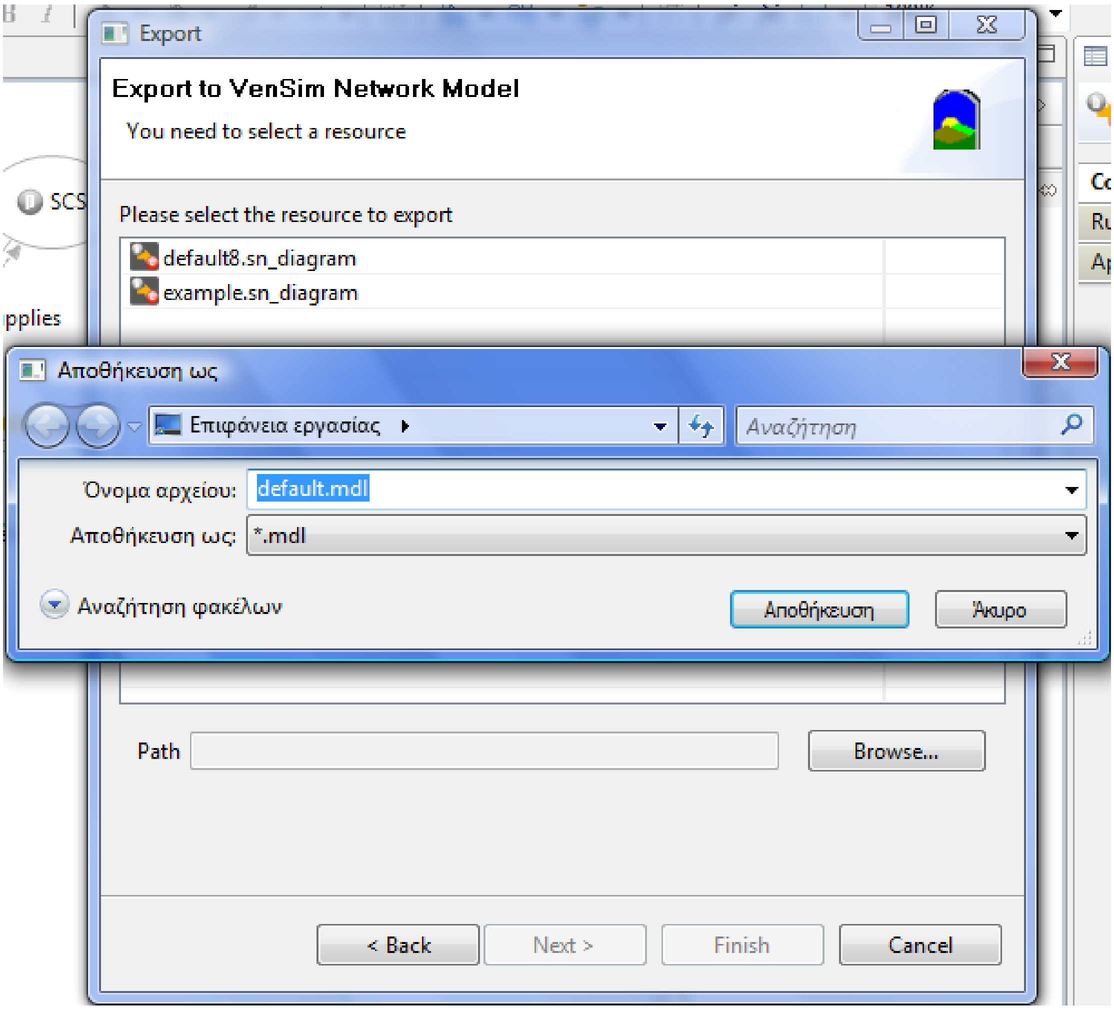
STEP 2: User has already chosen to export the Service Network model that has made and now has to choose the destination of exportation. We assume that user is willing to export his SNAPT model to Vensim tool. Thus, user has to choose the folder named "System Dynamics" that has a single option that corresponds to "Export Service Network to Vensim". Pressing button "Next >" user can navigate to another wizard window.



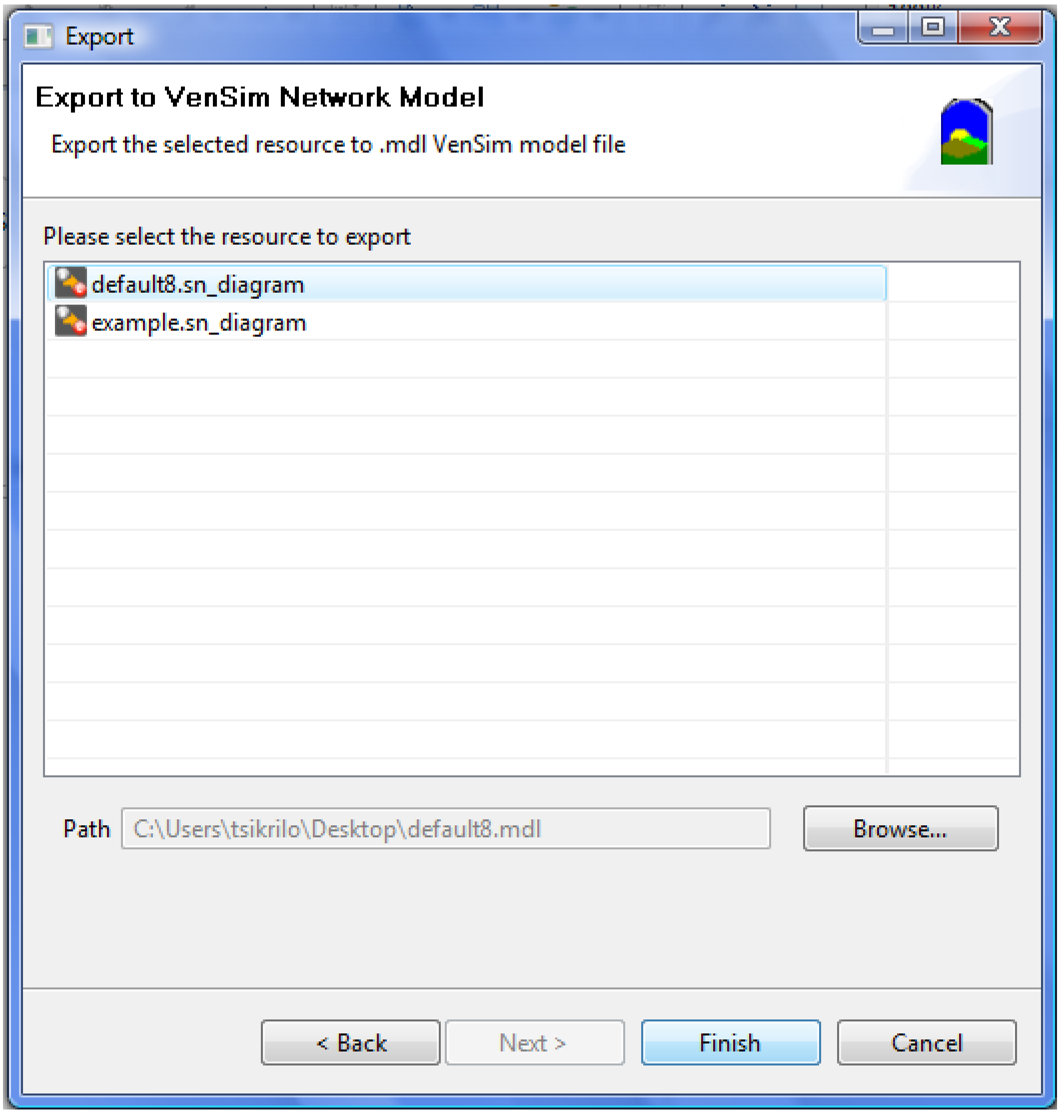
STEP 3: At the window wizard below is shown a list of currently open SN editors' diagram that are available for exportation. Below this list is a "Browse…" button. Pressing this button is triggered another window wizard that is shown at next figure.



STEP 4: A new window wizard is triggered and user can set the file path of exportation.



STEP 5: Finally, the user pressing the button "Finish" can complete the exportation and trigger the SNAPT2Vensim execution. As a result of these steps is a file creation that is absolutely compatible with Vensim tool and encapsulates information from SN editors' diagram that user has just selected.



STEP 6: Assuming that user did all steps successfully and then we get as a result the icon that is shown at the figure below. We also assume that user has selected the SN editors' diagram with the same name as the icon below (default8). If user make a double click at this icon then will be opened the Vensim environment including the specifications of new Vensim model file. Last figure shows how the corresponding Vensim model of the first SN editors' diagram is.