Fuzzy Networks Model, a Reliable Adoption in Corporations

John Velandia(✉), Gustavo Pérez, and Holman Bolivar

Engineering Faculty, Universidad Católica de Colombia, Bogotá, Colombia
{javelandia,gperezh,hdbolivar}@ucatolica.edu.co

Abstract. Computing huge amounts of information and performing complex operations in a unique fuzzy logic system is a challenge in the field of fuzzy logic. This paper presents a Knowledge engineering application whereby a Fuzzy Network (FN) is used to build a complex computing model to reproduce corporate dynamics and to implement a Model Reference Adaptive Control (MARC) strategy for Corporate Control [2]. This model is used as a What If? Environment to explore future consequences of actions planned within a strategic scenario context in terms of KPIs displayed in a Balanced ScoreCard (BSC) control board. Corporation's strategy map is required to plan the Knowledge Identification and Capture Activity (KICA) required to obtain the knowledge to be represented in the FN's nodes rule bases. KICA produces linguistic variables as well as the qualitative relationships amongst them. A FN appears as a natural solution to model the knowledge distributed within the members participating in all analysis and decision making tasks along the organization. Additionally, as proof of concept a prototype which capable of designing and simulating networks of fuzzy systems is presented based on the standard IEC 61131-7.

Keywords: Fuzzy logic · Fuzzy networks · Corporate model · Fuzzy system

1 Introduction

Fuzzy Logic (FL) is used when it is difficult or even impossible to construct precise mathematical models [1]. FL is a logical system which is an extension of multivalued logic to serve as a logic model of human mind [2]. Complex traffic and transportation problems are example of having subjective knowledge, commonly known as linguistic information, which is not solve using classical mathematical techniques, since they are hard to quantify [3].

Information systems with complexity in multivariable control are oriented to diagnose the knowledge management in companies, among many others. It is computationally impracticable to concentrate all heuristics in one fuzzy system, due to the number of variables and rules that are considered in a whole system [4]. Currently, exists a trend to build Networks of Fuzzy Systems (NFS), as it has been noted in the last World Congresses presided by the International Fuzzy Systems Association, IFSA and the North American Fuzzy Information Processing Society, NAFIPS [5]. Despite,

P. Perner (Ed.): MLDM 2018, LNAI 10935, pp. 28–41, 2018.
https://doi.org/10.1007/978-3-319-96133-0_3

commercial tools are available to build NFS, they are limited by parameters setting [6]. Moreover, commercial tools are expensive, and there is no official free tool available to build NFS.

Since BSC officially appeared [7] it has become an important corporate control tool; however the feedback it provides through the KPIs takes a good time after an action has been taken. This occurs because the time constants involved in corporate dynamics are rather long ranging from weeks to months depending of the strategic deep of decisions and the corresponding actions.

This paper presents an enhancement of the BSC control strategy by means of a knowledge based computing model, representing the corporate dynamics and implementing a Model Reference Adaptive Control (MRAC) strategy [8]. Since the strategy map [9] displays the inner causality in an organization's dynamic, it is used to conduct the KICA required to identify the involved linguistic variables and the qualitative relationships amongst them. The model is constructed using a FN which stores the knowledge gathered through a KICA. This model is used to provide the **What If?** Environment to test different strategic scenarios and to identify the best actions to be taken on order to obtain desired KPI's values in a given time horizon. FN have already been used and reported [4, 10] as a tool to build expert systems using distributed knowledge.

Additionally, as proof of concept of the proposed model a design of a prototype that builds NFS is presented. The prototype is named FuzzyNet, and its objective is to design and simulate networks of fuzzy systems. It is based on the standard IEC 61131-7, ensuring the basic components of the programming methodology, environment and functional characteristics of Fuzzy Logic Control (FLC) [11, 12]. This research is part of the project "RESIDIF", its aim is to deploy a free tool for simulating scenarios that require NFS for corporations.

The novelty of this research comprises:

- A model that encompasses a formal path to create fuzzy networks. This implies a business process definition, including its activities.
- A software prototype named FuzzyNet that allows materialize the proposed model. This prototype is presented by mean of software components and graphical interfaces. Additionally, JFuzzyLogic library is presented as an important software integration for achieving real simulations.
- A corporate model application is defined based on the fuzzy network model, in this way the software prototype and real business processes are integrated to demonstrate a real application of this model in the industry.

This paper is organized as follows: Sects. 1 and 2 present background of FLCs and corporate concepts. In Sect. 3 up to date related work is presented, in Sect. 4 the proposed model is defined, including a comparison of non-commercial fuzzy software is presented, in order to define a candidate framework that would be integrate to FuzzyNet. Section 5 presents the application of the proposed model for corporations. Finally, Sects. 6 and 7 presents conclusions and future work.

2 Fundamentals

2.1 Fuzzy Logic

The concept of fuzzy logic is deeply related to how people perceive the environment. People constantly make ambiguous statements that depend on how observer perceives the physical or chemical effect, e.g., if someone says "the bus station is going to be congested", it would be interested to assess if this statement is true, and when it would happen. The reasoning based on information that is not accurate is also a common procedure, and it is precisely what fuzzy logic does [13]. Observation of the environment, the formulation of logical rules and mechanisms of decision-making [14].

In order to automate and standardize processes arising from the theory of fuzzy logic, a basic model was defined for application development [11]. A typical FLC architecture is composed of four principal components: a fuzzifier, a fuzzy rule base, an inference engine and a defuzzifier [15, 16].

The first component of the model is known as fuzzifier, which is the gateway of the fuzzy model. This component performs a mathematical procedure that consists of converting an element of the universe of discourse in fuzzy values. The defuzzifier is based on a set of rules antecedents and consequences, which are known as linguistic expressions. "yes", is the antecedent and "then", the consequent.

2.2 Corporate Concepts

2.2.1 BSC Control Structure

The current BSC control loop is displayed in Fig. 1, no matter how good the measurement system is, but the values entered for the KPIs in the BSC Board, since they will reflect the consequences of the taken actions only after a good amount of time since the time constants involved in the corporate dynamics are sometimes weeks or even months long. The *What If?* environment introduced here allows to test groups of actions, in fact, whole strategies, to examine the future KPI's values and so to determine the best strategy in terms of future KPI's values.

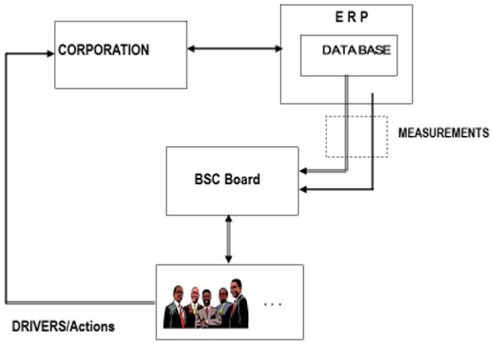

Fig. 1. Current BSC control loop structure.

2.2.2 Model Reference Adaptive Control Structure

The *What if?* environment is obtained using the mentioned MRAC control strategy and this is showed in Fig. 2. The Corporate Model allows to test actions showing future KPI's values in the BSC Model Board.

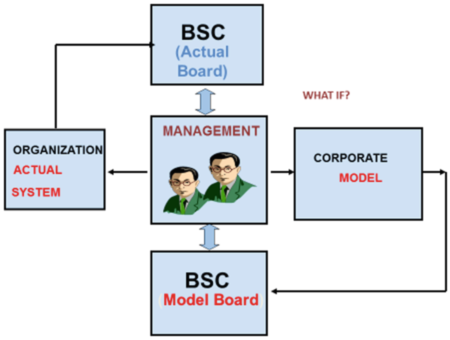

Fig. 2. MRAC structure with What If? Environment

Once the structure depicted in Fig. 2 is constructed, a concrete simulation is willing to start testing strategies and actions that would be the input of the proposed model. Afterwards, an observation is taking place over the BSC Model Board, in order to identify and analyze the resultant KPIs in the specified time horizon.

3 Related Work

Fuzzy networks have not been an ongoing discussion in fuzzy logic, despite this is a need for processing complex and parallel fuzzy systems. None of the indexed scientific databases provide research works, for instance Science Direct, Elsevier, ACM and Google Scholar. The closer research work to this topic are studies in fuzzy networks with aggregation of rule bases for decision-making problem solving, which is a model proposal [17].

There some additional studies are performed regarding cognitive and neural networks [18, 19], however they do not focus on fuzzy systems, which means that these sort of studies are not related to fuzzy networks, for example, these studies lack of information regarding inference engine, rules and some of the fundamental components of fuzzy systems. In conclusion, this research topic is novel and therefore further work is pending to develop in several scenarios.

4 Model for Building Networks of Fuzzy Systems

This section describes the architecture of FuzzyNet. The proposed architecture comprises: (A) Definition of the business process model. (B) The main components of the prototype to be developed. (C) Integration to fuzzy logic implementation.

4.1 Process Workflow to Design and Simulate Fuzzy Networks

Considering that any information system is created to support business processes, Fig. 3 describes the process to design and simulate NFS. Business Process Model Notation (BPMN) is used as convention. This process is accomplished in two phases:

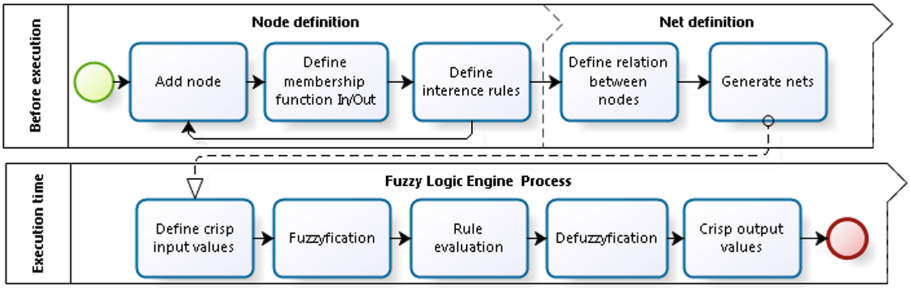

Fig. 3. Simulation process to create fuzzy networks.

(1) Before execution time: This phase consists of defining number of nodes needed for building a fuzzy network, and the network itself. The workflow consists of adding a node that contains a node's name, and linguistic variables to define fuzzy sets. Then, it is defined membership function to use with crisp input and output values. Followed by definition of inference rules that are provided by an expert. Once set of nodes (fuzzy systems) is defined, a relation among nodes is fixed, in order to build dependencies and prioritize nodes execution. The last activity in the workflow comprises generation of a fuzzy network. During this phase users only perform a network setting.

(2) During execution time: This phase encompasses the classic workflow of fuzzy logic systems, however in this process model acts as engine, since it is the core of the process. Once the network is designed, users provide crisp input values, which would be unique input for entire network, but for every node. Afterwards, the engine executes the matching of crisps values with the linguistic terms, this is named fuzzyfication. Then, the inference activity is performed, which comprises the assessment of rules, and a fuzzy set as result. Finally, the result of the inference process has to be converted into crisp numerical values, activity known as defuzzification, then information is presented throughout graphical interface.

4.2 Software Components Model

According to Fig. 4 *Net management component* is responsible for the definition of the fuzzy network. Network's name is set and relationships among nodes are stablished to accomplish the fuzzy network simulation. The setting is saved in a XML file to guarantee future interchange of information by mean of generic language.

Node management component is in charge of creating, updating and removing nodes from the fuzzy network; this implements a logic such that guarantee minimum

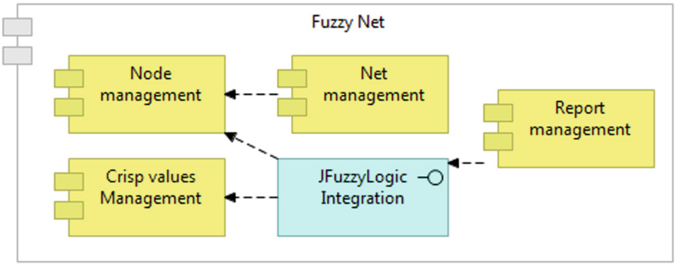

Fig. 4. Set of components for simulating fuzzy networks.

two nodes with their respective linguistic variables, linguistic qualifiers and membership function, i.e., triangular, Gauss, Cosine, Sigmoidal, Difference of sigmoidal, Trapezoidal and Generalized bell.

Crisp values management incorporates the variables and values that willing to use in the fuzzification process. These values are associated to the membership function automatically; these crisp values are transformed to fuzzy values after the simulation of the fuzzy network is performed.

JFuzzyLogic Integration is an external software package that allows using the membership functions automatically. This package acts as plugin, which means that the prototype may call any functionality natively. The following subsection specifies the reason of using this particular tool in this research.

Report management is an additional component that is not part of the model, however it is considered for obtaining general information regarding the simulations. For instance, this component provides an outline of the number of nodes that conform the network and their relationships, crisp values, linguistic qualifiers setting, inference engine method and rules that depend on the context, in this case corporate control model. Meanwhile Figure 6 presents a sampling interfaces of this model by mean of the prototype FuzzyNet.

4.3 Integration to JFuzzyLogic

IEC 61131 is a standard developed to integrate fuzzy control applications This standard is implemented among different providers to allow systems to connect with each other in a common way. The principal objective of this section is to find a tool that supports the IEC standard for being integrated into the software prototype that is used as proof of concept.

The main functionality of the tool that would be selected is the Fuzzy Controller (FC), because it allows to simulate and to run diffuse networks by mean of implemented libraries. FCL is a key functionality since it defines the parameters that are set in IEC for using adequately fuzzy controllers. The following criterion are considered to select the tool: FCL is provided, Programming Language - PL, functionality, and number of memberships functions - MF. Java language is the preference because the prototype is developed in Java, due to the portability and independency of operating system. Hence, Table 1 presents a comparison of noncommercial fuzzy logic tools:

Table 1. Comparison of open fuzzy logic software tools.

Name	FCL	PL	Functionality	MF
AwiFuzz	Yes	C++	No compiles	2
FLUtE	No	C#	Beta version	1
FOOL	No	C	No compiles	5
Funzy	No	Java	Run	2
FuzzyJToolkit	No	Java	No longer maintained	15
Jfuzzynator	No	Java	Run	2
JFCM	No	Java	Run	–
jFuzzyLogic	Yes	Java	Run	25
jFuzzyQt	Yes	C++		8
Libai	No	Java	Run	3
Nefclass	No	C++ Java	Run	1
Nxtfuzzylogic	No	Java	Run	1
UNFuzzy 2.0	Yes	C++	No compiles	8

According to the Table 1 a detailed analysis is performed for each criterion to select an adequate software tool:

FCL. The outcome after these tools were assessed is that only four of them are suitable for the required integration, specifically these tools are AwiFuzz, jFuzzyLogic, jFuzzyQt and Unfuzzy. It signifies that only these few tools remain in this analysis, because the surplus does not follow the FCL standard.

PL (Programming Language). As it was previously mentioned, the ideal language would be Java, given that the software prototype is developed in this language. Thereby, jFuzztLogic is currently the unique tool that meets this specific criterion.

Functionality. Before the aforementioned criteria were evaluated, a validation regarding whether each tool run on Windows properly was made, and the result shows that most of the tools work adequately in the operating system Windows. Another result to highlight is that four of them run properly on Windows, due to the lack of support and maintenance.

MF (Number of Memberships Functions). The membership functions are directly related with the inputs and outputs, having more memberships there more flexibility in its definition. The numbers with asterisks are membership functions which are built from the combination of membership functions. jFuzzyLogic is the most outstanding tool thanks to the 25 memberships functions implemented, which contributes to the enhancement of features in the proposed prototype.

Among the metrics established to choose a software tool, it is decided to opt for a tool that provide functionality for the development of fuzzy systems, also that implements Fuzzy Control Language specifications and that encompasses the highest number of membership functions, thus one could conclude that jFuzzyLogic is the most suitable tool for this research, specifically for the proposed prototype.

4.4 Principles of NFS

In the development phase they were established the principles for building networks fuzzy controllers, which describe the features that have the bindings between fuzzy controllers.

Binding. The binding between fuzzy nodes, it is established by means of linguistic variables, that play role of inputs or outputs in the nodes. Thus when a binding is established between nodes, the linguistic variable through which the operation is binding to the two nodes, in the origin node like an output and in the destination node like an input to the fuzzy controller.

Simulation. The simulation process of fuzzy networks consists in the individual simulations of nodes that compose the network, in a specific order defined by how they are configured the bindings, obeying the premise that to simulate a node should be considered if you have inputs function as outputs from another node, and if so the latter must first be simulated.

4.5 The Prototype

This phase is developed based on the architecture designed by diagrams obtained in the development phase, this architecture was validated through building prototyping, which involves the creation of a prototype software that will serve as the desired application archetype. This process is necessary to prove that the architecture designed to comply with the main objective, create networks of fuzzy logic.

Figure 5 shows the final FN designed architecture. Five layers were required in order to implement the network using FuzzyNet, a software tool to implement and simulate FN, developed at Universidad Católica de Colombia.

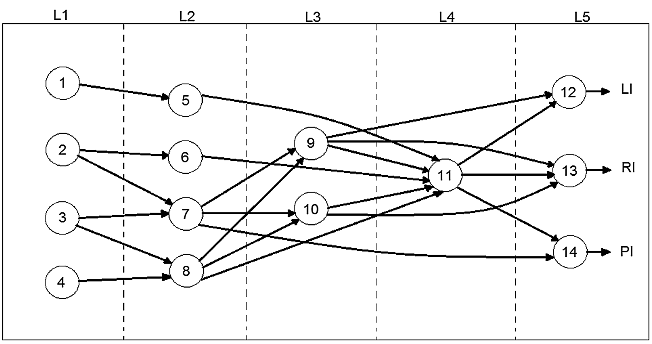

Fig. 5. FN architecture, a sample network.

In the previous figure, the simulation encompasses the following characteristics: all nodes employ Mandani (Minimum) for the fuzzy inference; membership functions implemented are the standard L, Lambda (Triangle) and Gamma Functions, available in majority fuzzy systems software tools; defuzzification is performed with center of gravity defuzzifier.

As result of testing the sample network, one could conclude that software components work correctly when the simulation is performed using the proposed model. In details, the process consists of definition of linguistic variable by typing the values by the user, including its possible results the risk that it is supposed to be assumed by the company.

Figure 6 presents a sampling interfaces of the prototype.

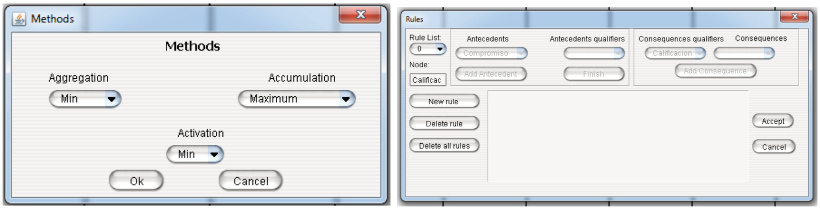

Fig. 6. Prototype interfaces sampling

5 Corporate Model Application

5.1 Corporate Model

Figure 7 shows a typical FN structure for the corporate model.

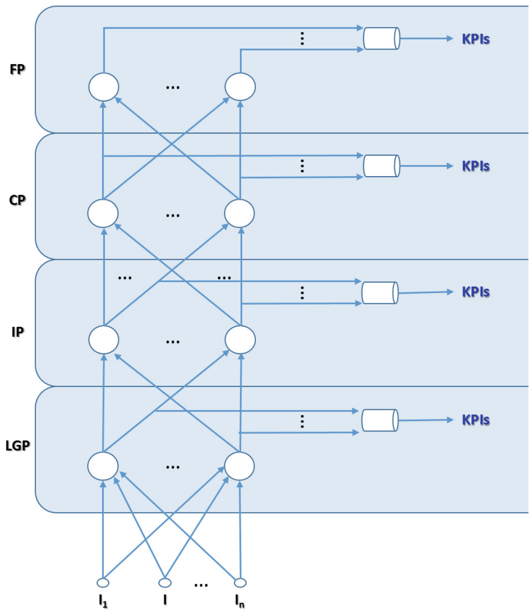

Fig. 7. Corporate model application

The FN structure is distributed along the four BSC Perspectives: Financial Perspective (FP), Customer Perspective (CP), Internal Perspective (IP) and Learning and Grow Perspective (LGP). Every Node in the FN is separately created and tuned using the MRAC interface included in the MANAGEMENT Block (Fig. 2). Design parameters for each node are determined through the tuning procedure using criteria obtained in the KICA. Figure 7 shows the structure of a node with the design parameters for every module. Depending on the particular dynamic associated to a company the FN could be either Feedforward or Recurrent. The Rule Base for every Node will contain the fuzzy rules obtained after analyzing the results obtained with the KICA. The nodes producing the KIP's values are explicitly shown. All the fuzzy rules for every node contain the time as a linguistic variable in the Antecedent. Rule k is then explicitly written as (1). Although the Inputs go into nodes in the Learning and Growing Perspective in Fig. 7, inputs can actually go into any node in any perspective.

Rk: IF X1 is LX_1 AND…AND Xm is LXm AND
Time is LT THEN Y1 is LY1 AND … AND Yn is LYn
Where:
X1…Xm are the inputs to the node and Y1…Yn are the outputs,
LXi $\mathcal{E}$ {L, M, H}, LYi $\mathcal{E}$ {L, M, H}
LT $\mathcal{E}$ {VS, S, M, L, VL}
With: L = Low, H = High, VS = Very Short, S = Short, M = Medium, L = Long, VL = Very Long.

Although only one layer is shown for the FN in each perspective, it is the particular dynamics for each corporation's value-creating processes which determines the FN's nodes structure for every perspective. The particular FN structure constructed for any particular organization will reflect the particular dynamics and the inner causality within the value-creating processes in that corporation.

5.2 The What If? Environment

The User Interface also contains the simulation environment so that managers and/or planners can perform the simulation tasks required to test any strategy. This environment keeps the record of all simulations to facilitate the supervision tasks required to identify the best strategies (Fig. 8).

When a corporation has a comprehensive and well maintained Data Base with a few years of data, rules can be identified using a mining procedure with the help of Adaptive (Trainable) Fuzzy Systems [8]. This capability is built in the UNFUZZY tool [6] used for the developments reported in [9, 10].

Implementing this MRAC strategy with *What If?* Environment requires the committed participation of all members in the organization. KEA in particular requires the open and patient collaboration since many times it is necessary to go over some

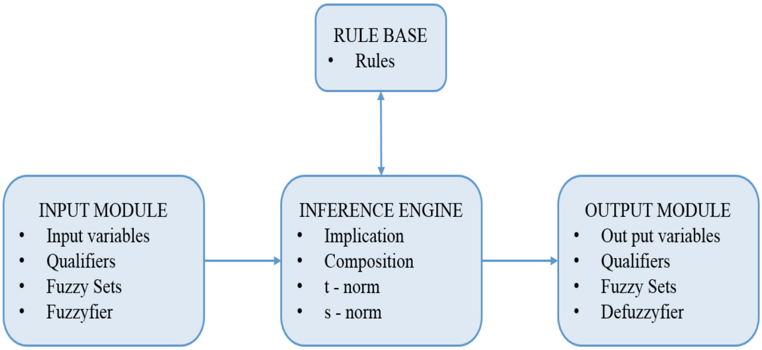

Fig. 8. Node's structure with design parameters.

particular subjects in order to clearly identify the linguistic variables as well as the nature of relationships.

MRAC with *What If?* Environment was implemented in a local utility. It is currently used in the planning tasks and it has been taken as a pilot experience to scout new possibilities. The task force organized for this accomplishment is now working on model refinement to include risks, using the experience gathered in a recent work [4]. This means identifying the associated risks to every strategic goal and to actually acquire the heuristics associated to its assessment. This approach intends to reach the point where we can modify the fuzzy rules for the KPI nodes in the corporate model to include risk values. The BSC Boards, Corporate Board as well as Model Board, will also be modified to include the column corresponding to risk values.

5.3 Utility Company Application

Figure 9 shows the strategy map for a utility company, where the inputs are also indicated. As a result of KICA the input and output linguistic variables for every process in the map were identified as well as the proper fuzzy relationship relating each input-output pair in order to elaborate the fuzzy rules for every node. Identifying the proper fuzzy rules includes detecting the adequate set of Qualifying Linguistic Terms (QLT) as well as the Trend relationships, i.e. whether a variable appearing in the consequent of a particular fuzzy rule is growing or decaying when a variable in the antecedent is growing.

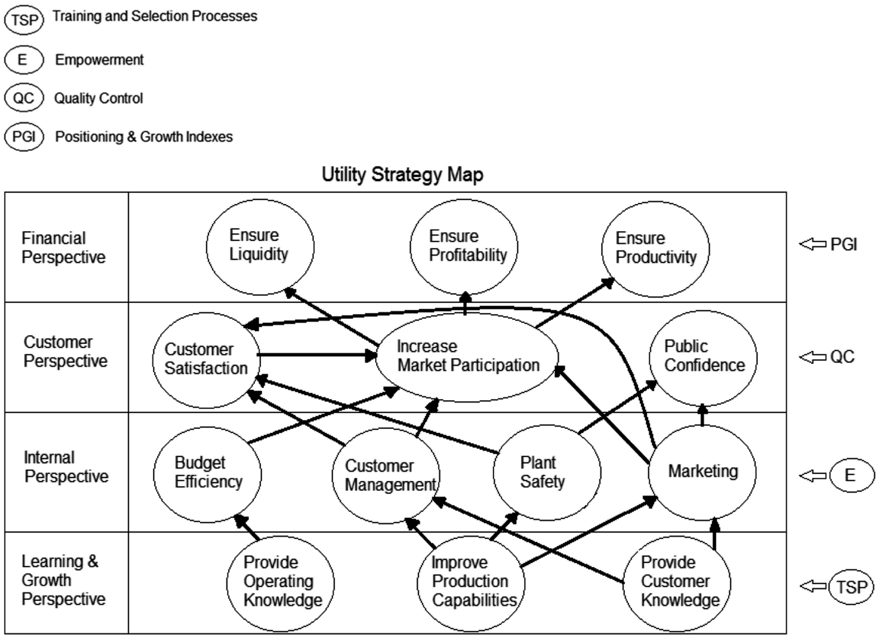

Fig. 9. Strategy map.

6 Conclusion

A deduction based on the proposed model is that network fuzzy systems attempt to bind individual fuzzy systems which are known in this research as nodes. Therefore, a fuzzy system is not seen a *silo* anymore, but a set of systems that form a fuzzy network. Thus, in a fuzzy network every node is able to send a receive values to simulate scenarios.

A limitation that arose during the corporate model application was the lack of performing stress and load tests. This would be a future work to measure the quality attributes in the hardware and software system, specially, when nodes require execute in parallel with complex operations.

The proposed model implies a scenario with at least two dependent fuzzy systems, hence, one fuzzy system must always execute first, then the second one is executed using the previous result from the prior system. In scenarios where there more than two systems, parallelism is applicable.

The validation of the proposed model concludes that the Corporate Model constructed after the KICA using a Fuzzy Network is an excellent example of a working asset obtained through capitalizing organization knowledge.

A Fuzzy Network is a versatile way to model corporate knowledge since this is distributed all over the people working in analysis and decision making activities within the organization’s value-creating processes.

A What If? Environment provides the BSC corporate control strategy a powerful tool to explore different strategic scenarios.

Identifying the risks associated to every strategic goal as well as the associated heuristic to its assessment would allow to add the Risk column to the BSC Board, adding Risk Management to the MRAC STRATEGY.

7 Future Work

This work is the starting point for ongoing fuzzy systems that are required to work in parallel or complex operations, even with different objectives. The proposed fuzzy model is willing to use in academic and industry implementations, because the model definition encompasses a general process and activities that any other fuzzy network would use it. Specifically, the proposed model is due to incorporate it into a real business process, thus in that way the model would be proven.

The prototype implemented for this research is able to future improvement, adding more components or activities regarding fuzzy sets or systems. Also, integrating some other plugins related to inference engines, in case more membership functions are required. hardware performance, software processing.

Disclosure

- **Funding:** This study was funded by Universidad Católica de Colombia (grant number is not public).
- **Conflict of Interest:** The authors declare that they have no conflict of interest.

References

1. Laskri, M.T., Beggas, M., Médini, L., Laforest, F.: Towards an ideal service QoS in fuzzy logic-based adaptation planning middleware. J. Syst. Softw. **92**, 71–81 (2014)
2. Zadeh, L.A.: Fuzzy logic: issues, contentions and perspectives. In: 1994 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1994, vol. 6, p. VI/183 (1994)
3. Sarkar, A.: Application of fuzzy logic in transport planning. Int. J. Soft Comput. **3**(2), 1 (2012)
4. Hoyos, G.P.: Pipeline risk assessment using a fuzzy systems network. In: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 1495–1498 (2013)
5. Seising, R., Trillas, E., Kacprzyk, J. (eds.): Towards the Future of Fuzzy Logic, vol. 325. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18750-1
6. Duarte, O.G., Pérez, G.: Unfuzzy: fuzzy logic system analysis, design, simulation and implementation software. In: Proceedings of the EUSFLAT-ESTYLF Joint Conference, Palma de Mallorca, Spain, 22–25 September 1999, pp. 251–254 (1999)
7. Norton, D.P., Kaplan, R.S.: Transforming the Balanced Scorecard from Performance Measurement to Strategic Management, 15th edn. Harvard Business School Publishing Corporation, Boston (2001)

8. Wang, L.-X.: Adaptive Fuzzy Systems and Control. PTR Prentice Hall, Upper Saddle River (1994)
9. Norton, D.P., Kaplan, R.S.: Strategy Maps, Converting Intangible Assets into Tangible Outcomes. Harvard Business School Publishing Corporation (2004)
10. Gustavo, P.: A fuzzy logic based expert system for short term energy negotiations. In: 18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No. 99TH8397), pp. 149–152 (1999)
11. Commission, International Electrotechnical technical committee industrial process measurement and control. Programmable Controllers, Part 7 - Fuzzy Control Programming. Commission, International Electrotechnical technical committee industrial process measurement and control (1997)
12. Cingolani, P., Alcalá-Fdez, J.: jFuzzyLogic: a Java library to design fuzzy logic controllers according to the standard for fuzzy control programming. Int. J. Comput. Intell. Syst. **6** (Suppl. 1), 61–75 (2013)
13. Zadeh, L.A.: Is there a need for fuzzy logic? Inf. Sci. **178**(13), 2751–2779 (2008)
14. Alfaro-Garcia, V.G., Gil-Lafuente, A.M., Klimova, A.: A fuzzy approach to competitive clusters using moore families. In: Rutkowski, L., et al. (eds.) ICAISC 2015. LNCS (LNAI), vol. 9119, pp. 137–148. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19324-3_13
15. Camastra, F., Ciaramella, A., Giovannelli, V., Lener, M., Rastelli, V., Staiano, A., Staiano, G., Starace, A.: A fuzzy decision system for genetically modified plant environmental risk assessment using Mamdani inference. Expert Syst. Appl. **42**(3), 1710–1716 (2015)
16. Meschino, G.J., Nabte, M., Gesualdo, S., Monjeau, A., Passoni, L.I.: Fuzzy tree studio: a tool for the design of the scorecard for the management of protected areas. In: Espin, R., Pérez, R.B., Cobo, A., Marx, J., Valdés, A.R. (eds.) Soft Computing for Business Intelligence. SCI, vol. 537, pp. 99–112. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-53737-0_6
17. Yaakob, A.M., Gegov, A., Rahman, S.F.A.: Decision making problem solving using fuzzy networks with rule base aggregation. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6 (2017)
18. Cruz-Vega, I., Garcia-Limon, M., Escalante, H.J.: Adaptive-surrogate based on a neuro-fuzzy network and granular computing. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 761–768 (2014)
19. Nápoles, G., Mosquera, C., Falcon, R., Grau, I., Bello, R., Vanhoof, K.: Fuzzy-rough cognitive networks. Neural Netw. **97**, 19–27 (2018)