

Type-2 Fuzzy Inference System Visual Components for Building Java Intelligent Applications

Manuel Castañón-Puga, *Member, IAENG*, Juan Ramón Castro, Josué Miguel Flores-Parra, Carelia Gaxiola-Pacheco and Luis Enrique Palafox-Maestre

Abstract—This paper introduces a JT2FISPanel and JT2FISClusteringPanel, a Java visual component for Interval Type-2 Fuzzy Inference Systems that can be used to build Java Intelligent Applications. The main features and functionalities are described. To do so we show user interfaces in order to compare the developed components with existing tools.

Index Terms—Type-2 Fuzzy Inference System, Java Visual Component, Java Intelligent Application.

I. INTRODUCTION

Fuzzy inference systems (FIS) have been broadly used for a wide range of engineering applications. FIS have been especially applied successfully in control and decision-making systems, where the main advantage is the way they deal with imprecise information related to some system variable allowing us to work with it.

Most of FIS used until now are FIS based on a Type-1 model [1], but lately, a Type-2 model has been developed and other applications are being extended with it [2]. This current technique led us to Type-2 General Fuzzy Inference Model [3] that has been developed as a next step to design and have Fuzzy Inference Systems with more capability to model real-world things [4][5].

The purpose of this paper is to introduce a Java visual component for Interval Type-2 Fuzzy Inference Systems based on JT2FIS class library that can be used to build Java intelligent applications.

A. Type-2 Fuzzy Inference System

A fuzzy inference system (FIS) is based on logical rules that can work with numeric values or fuzzy input, when rules are evaluated, the individual results form together what is known as the output fuzzy, then, a numerical value must be passed through a process of defuzzification if required. Figure 2 shows a block diagram of the classic structure of a FIS.

The idea of a type-2 fuzzy set was introduced by Zadeh [6] as an extension of the concept of fuzzy sets, usually type 1. A type-2 fuzzy set is characterized by a membership function whose membership value for each element of the universe is a membership function in the range [0, 1], unlike the type-1

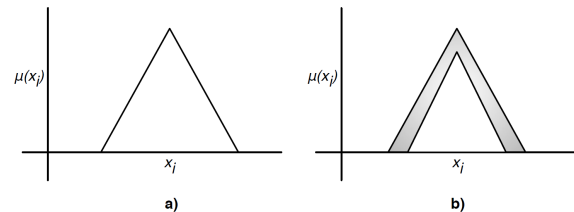


Fig. 1. Type-1 fuzzy set and type-2 fuzzy set with uncertainty.

fuzzy sets where the value of membership is a numeric value is within the range [0, 1]. The creation of a fuzzy set depends on two aspects: the identification of the universe with appropriate values and specifying a membership function properly. The choice of a membership function is a subjective process, meaning that different people can reach different conclusions on the same idea. This subjectivity derives from individual differences in the perception and expression of abstract concepts that have very little to do with randomness. Therefore, subjectivity and randomness of a fuzzy set are the main differences between the study of fuzzy sets and probability theory [7].

In type-1 fuzzy sets, once the membership function is defined for a concept, having based this on the subjective opinion of one or more individuals, it shows no more than one value for each element of the universe. In doing so, it loses some of the ambiguity on some discussed concepts, especially where people may have a slightly different opinion and all are considered valid. The type-2 fuzzy sets allow handling linguistic and numerical uncertainties. Figure 1 depicts two graphics of fuzzy sets, a) with type-1 fuzzy logic, and b) with type-2 fuzzy logic.

In a) the values set shown is $A = \{(x, \mu_A(x)) | x \in X\}$ where $A \subseteq X$, $X \subseteq X$, X is the universe of valid values and $\mu_A(x)$ is the membership function (MF) that contains a map of each value of X with its membership value corresponding to a value between 0 and 1. For b) the values set is $\tilde{A} = \{(x, u, \mu_{\tilde{A}}(x, u))\}$, where MF $\mu_{\tilde{A}}(x, u)$ has a membership value for each element of the universe as a function of membership in the range [0, 1], so the footprint can be seen around the curve of a).

B. Object-Oriented Fuzzy Inference Systems

There are code libraries and tool-kits available to build Fuzzy Inference Systems [8]. Some of these packages are object-oriented class libraries that are developed mainly to build Type-1 Fuzzy Logic with object-oriented programming

Manuscript received March 14, 2014; revised March 20, 2014. This work was supported in part by the Internal Fund for Research Projects (Grant No. 300.6.C.135.17) of the Autonomous University of Baja California, México.

M. Castañón-Puga, J. R. Castro, J. M. Flores-Parra, C. Gaxiola-Pacheco and L. E. Palafox-Maestre are with the Chemistry and Engineering Faculty, Autonomous University of Baja California, Tijuana, BC, 22390, México, e-mail: {puga, jrcastror, mlflores31, cgaxiola, lepalafox}@uabc.edu.mx

language [9]. jFuzzyLogic [10] and Juzzy [11] are examples of a class library written in Java. The advantage of a Fuzzy Inference System in Java is that we can build intelligent systems with Type-2 Fuzzy Logic capabilities using an object-oriented programming language. Java is a robust general use object-oriented programming language used in a wide range of applications.

C. Data Mining and Fuzzy C-Means Clustering

In recent years, the use of new information technologies that has come to help in the handling of a large amount of data, is the data mining extraction, an evolved technology that permits representing knowledge of data and the implicit storage. Data mining is a multidisciplinary field that combines the techniques from a machine learning, recognition pattern, statistics, database and visualization, to direct it to the extraction and interpretation of a huge database. The data mining focuses on filling the need to discover, predict and forecast the possible actions with some trustworthy factor for each prediction [12].

Moreover, it contributes in the making of tactical and strategic decisions provided by power users, can measure actions and results in the best way, it generates descriptive models to explore and understand the data and identifies patterns, relationships and dependencies that affect the final results. They create predictive models that allow undiscovered relationship through the data mining process, and these are expressed as possible business rules [13].

1) Fuzzy C-Means Clustering Algorithm for Data Mining:

The fuzzy proposals represent an important place in data mining providing intelligible results. One of these proposals is the Fuzzy C-Means clustering algorithm (FCM) [14] [15] that makes use of a membership function and centroid computation procedure iteratively to find the best centroid. The FCM is one of the most popular clustering algorithms; the effectiveness of the clustering method relies on the distance measure. An FCM is the resulting combination of the c-means approach with the handling of fuzzy data. The result of this combination is sufficient because it considers the uncertainty presented in the data avoiding incorrect results and creating crisp partitions in the correct way [14]. Additionally the FCM is used to acquire the adequate levels of the set clustering parameters [16].

D. JT2FIS Class Library

JT2FIS is a class library developed for Java. The main purpose is to deploy a library to build interval Type-2 fuzzy inference systems with an object-oriented programming language.

A fuzzy inference system (FIS) is based on logical rules that can work with numeric values or fuzzy inputs, these rules and individual results are evaluated together to form a fuzzy output, then, a numerical value must be passed through a process of defuzzification if necessary. Figure 2 shows a block diagram of the classic structure of a FIS.

JT2FIS architecture, design, implementation and Java programmers examples are described in detail in [17]. This paper is based on part, on the use of JT2FIS class library as a core Type-2 Fuzzy Inference System for the JT2FISPanel visual component implementation.

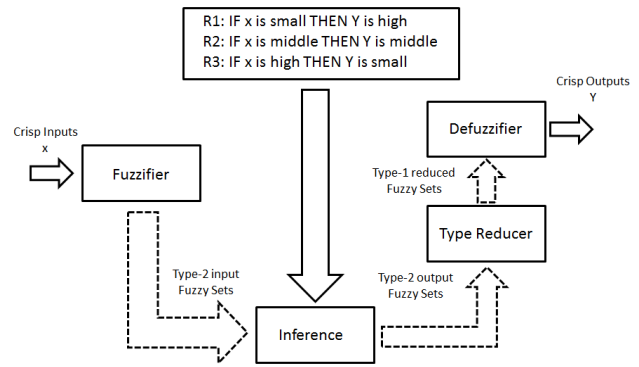


Fig. 2. Type-2 fuzzy inference system block diagram.

E. JT2FISClustering Class Library

JT2FISClustering is a class library developed for Java. Its main purpose is to deploy a library to building interval Type-2 fuzzy inference systems with an object-oriented programming language from data mining process. A clustering method is a data mining classic technique used to discover fuzzy sets and rules to configure the FIS from real data.

The library implements a fuzzy c-Means clustering algorithm for data mining. This paper is based on part, on the use of JT2FISClustering class library as a core clustering method for the JT2FISClusteringPanel visual component implementation.

II. THE JT2FIS PANEL

JT2FISPanel is a Java visual component that extends the functionality from the Java Swing JPanel component. This extended panel is part of the JT2FIS Library in order to facilitate the configuration and visualization of a Fuzzy Inference System. Visually, JT2FISPanel has four main functionalities to manage the FIS:

- 1) Inputs.
- 2) Outputs.
- 3) Members Functions.
- 4) Rules.

Figure 3 shows different options of configuration, for example, logic operators used in FIS evaluation methods and de-fuzzy type selected to depict a problem.

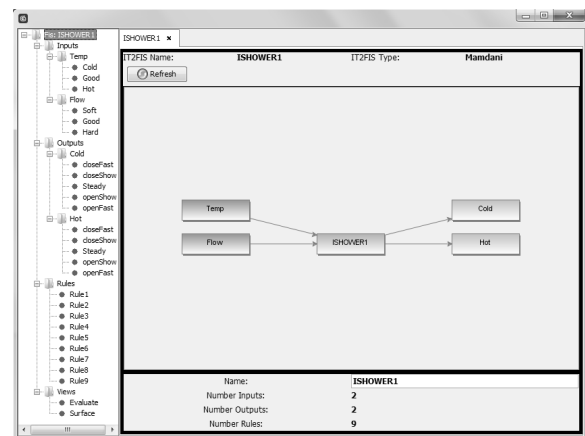


Fig. 3. Main JT2FISPanel for "ISHOWER" Example.

A. JT2FISPanel Inputs

In this section, we can manage the parameters on an input (input linguistic variable values) for a Fuzzy Inference System. In figure 4 we show the configuration of the “ISHOWER” FIS example “Temp” input. We can see different member functions that belong to “Temp” input and we can setup the attributes (Name, upper and lower range, etc.).

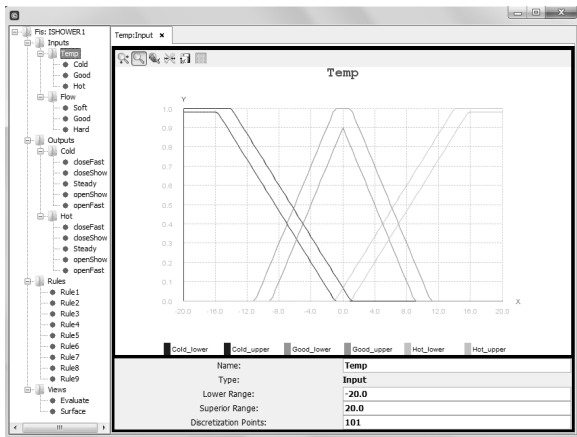


Fig. 4. Configuration of “Temp” Input in “ISHOWER” Example.

B. JT2FISPanel Outputs

In this section, we can manage the parameters on an output (output linguistic variable values) for a Fuzzy Inference System. Figure 5 shows the configuration of “ISHOWER” FIS example “Cold” output. We can see different member functions that belong to “Cold” output and we can setup the attributes (Name, upper and lower range, etc.).

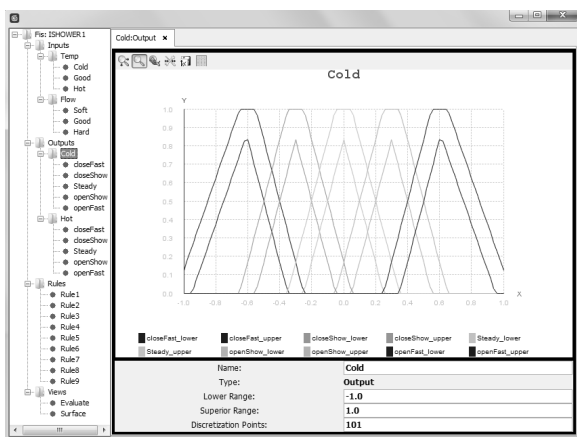


Fig. 5. Configuration of “Cold” Output in “ISHOWER” Example.

C. JT2FISPanel Members Functions

In this section, we can manage the parameters of a member function of an input or output. The figure 6 shows the “Good” member function for “Temp” input for the “ISHOWER” FIS example. We can see the linguistic selected value for “Temp” linguistic variable and the selected function member type. A graph depicts the linguistic value and shows visually the configuration.

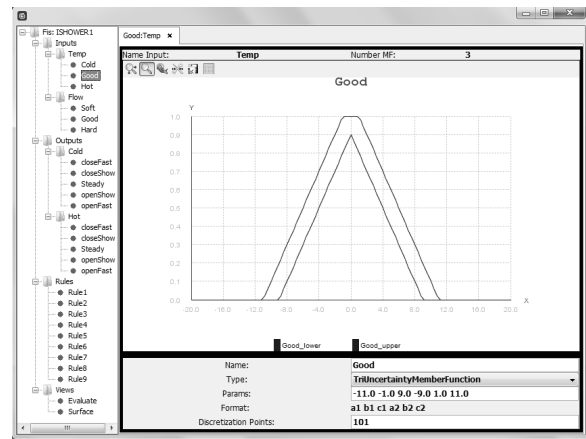


Fig. 6. Configuration of “Good” Member Function for “Temp” Input in “ISHOWER” Example.

D. JT2FISPanel Rules

In this section, we can manage the rules. The figure 7 shows “Rule 1” for “ISHOWER” FIS example. We can see the available inputs and outputs for building the rule. In the lower-left of the panel, we can see the inputs with its related member functions that can be selected as an antecedent of the rule. In the lower-right of the panel, we can see the outputs with its related member functions that can be selected as a consequent of the rule. Inputs can be connected by logical operators AND and OR. The added rule is shown in the center component.

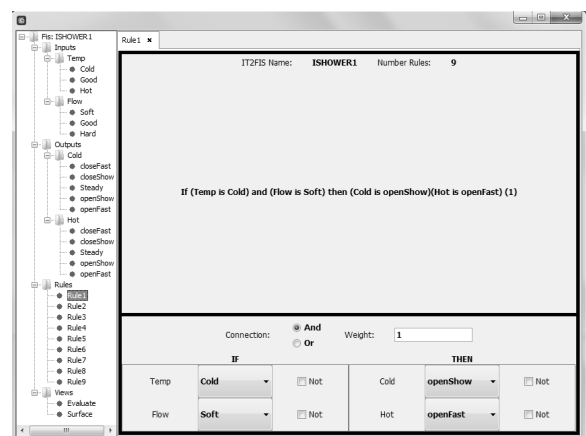


Fig. 7. Configuration of Rule 1 in “ISHOWER” Example.

E. Data evaluation on JT2FISPanel

The figure 8 shows the configuration for evaluating a fis. In these, you can choose different methods for defuzzification, points to discretize, methods and, or, implication and aggregation for inference.

JT2FISPanel has 2 different options to select points to be evaluated. You can add points to be evaluated manually as shown in figure 9, or you can import a csv file to evaluate a set of points automatically. JT2FISPanel can export the results in CSV format.

III. USING THE JT2FISPanel

The JT2FISPanel extends the functionality of the Java Swing JPanel class. In listing 1, we show how to add a

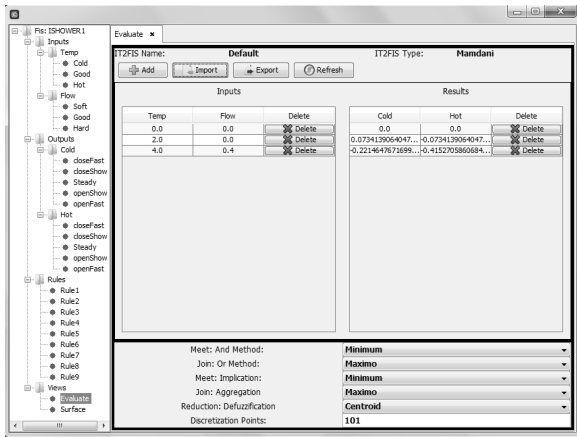


Fig. 8. Evaluate "ISHOWER" Example.

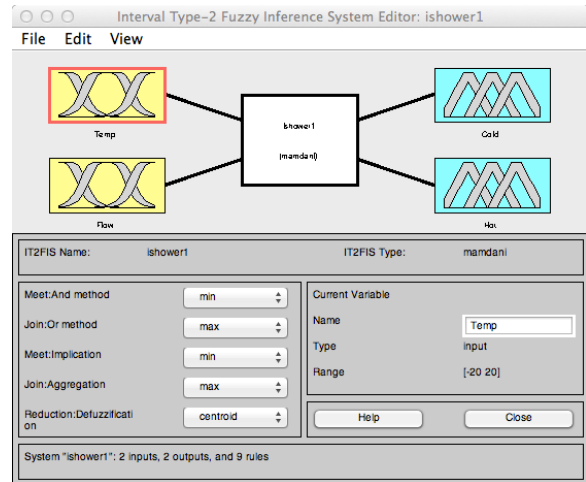


Fig. 10. Matlab IT2FUZZY tool Graphic User Interface.

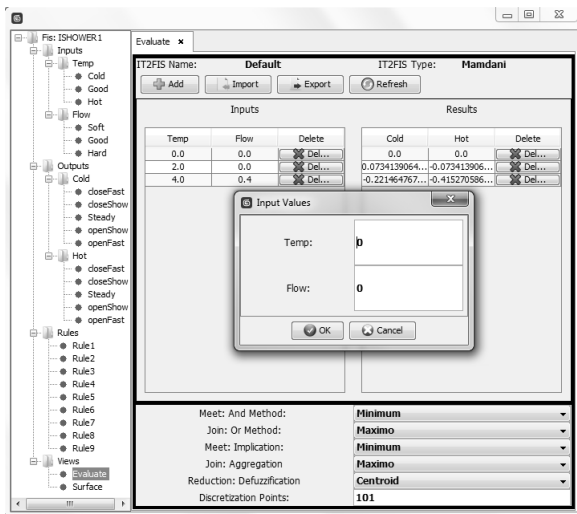


Fig. 9. Add Point Evaluate "ISHOWER" Example.

We can basically use it in the same way and obtain the same results, but with the convenience of a Java API. Java programmers can customize the use and purpose of JT2FISPanel.

IV. THE JT2FISCLUSTERINGPANEL

The figure 11 shows the graphic user interface of the JT2FISClusteringPanel. In this way, the user can set-up generation member functions and clustering method in order to apply the desired generation process.

Fuzzy c-Means is the default clustering method. In Table I list Type-2 member function available in JT2FISClustering.

TABLE I
JT2FISCLUSTERING MEMBER FUNCTIONS.

Type Clustering	Type-2 Member Functions	Description
Fuzzy c-Means	GaussCutMemberFunction	Params=[inputs outputs uncertainty]
Fuzzy c-Means	GaussUncertaintyMeanMemberFunction	Params=[inputs outputs] Params=[inputs outputs uncertainty]
Fuzzy c-Means	GaussUncertaintyStandardDesviationMemberFunction	Params=[inputs outputs] Params=[inputs outputs uncertainty]

JT2FISPanel instance to a Java Swing JFrame:

```
Listing 1. Calling a JT2FISPanel
//Creating a new JFrame instance.
JFrame frame = new JFrame("Example_
JT2FISPanel");
//Optional: What happens when the frame
closes?
frame.setDefaultCloseOperation(JFrame.
EXIT_ON_CLOSE);
//Creating a new JT2FISPanel instance.
JT2FISPanel panelJT2FIS = new
JT2FISPanel();
//Adding JT2FISPanel instance in the
frame.
frame.getContentPane().add(panelJT2FIS,
BorderLayout.CENTER);
//Size the frame.
frame.pack();
//Show it.
frame.setVisible(true);
```

The Figure 10 shows the classic Matlab Type-2 Fuzzy Toolbox visual interface of IT2FUZZY tool.

The JT2FISPanel design is visually close to IT2FUZZY tool, the Matlab Type-2 Fuzzy Toolbox visual application.

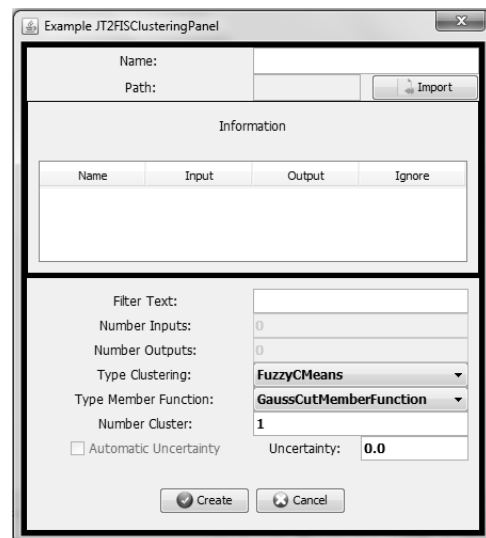


Fig. 11. JT2FISClusteringPanel Graphic User Interface.

JT2FISClusteringPanel generates an FIS from a csv file. The user select inputs and outputs from data columns in order

to configure the generation process as shown in the Figure 12.

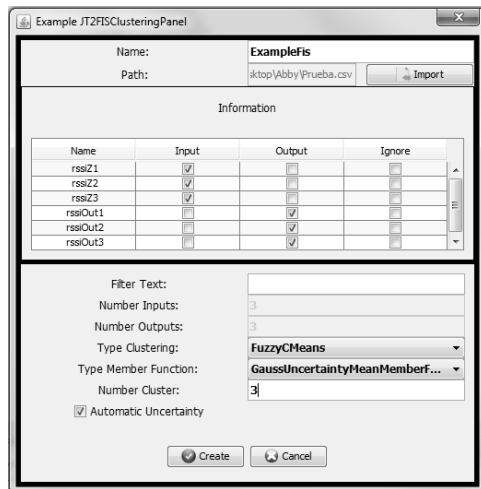


Fig. 12. Configuring JT2FISClusteringPanel for create Fis.

V. USING THE JT2FISCLUSTERINGPANEL

The JT2FISClusteringPanel extends the functionality of the Java Swing JPanel class. In listing 2, we show how to add a JT2FISPanel instance to a Java Swing JDialog:

Listing 2. Calling a JT2FISClusteringPanel

```
//Creating a new JDialog instance.
final JDialog dialog=new JDialog();
dialog.setTitle("Example_
    JT2FISClusteringPanel");
dialog.setBounds(100,100,450,500);
//Creating a new JT2FISClusteringPanel
instance.
final JT2FISClusteringPanel clusterPanel
    = new JT2FISClusteringPanel();
//Adding JT2FISClusteringPanell instance
in the dialog.
dialog.add(clusterPanel);
//Show it
dialog.setVisible(true);
//Adding actions to
    JT2FISClusteringPanel
clusterPanel.addActionListener(new
    OnActionsJT2FISClusteringListener() {
//Cancel action.
public void onCancel(int selectOption)
    {
        dialog.setVisible(false);
        dialog.dispose();
    }
//Create action.
public void OnApprove(int selectOption
    , Fis fis) {
        //Check if fis was create
        correctly
        if(selectOption==
            JT2FISClusteringPanel.
            APPROVE_OPTION){
            //Print information to fis.
```

```
System.out.println(
    clusterPanel.getFis().
    toString());
dialog.setVisible(false);
dialog.dispose();
}
}
});
```

VI. JT2FISPANEL AND JT2FISCLUSTERINGPANEL APPLICATIONS

Using JT2FIS visual components into visual applications is useful for facilitating the FIS user interaction. JT2FISPanel could be used to show the FIS structure and help to configure the system. The components could be embedded on every Java Swing desktop application. Figure 13 show a data mining process using JT2FISClusteringPanel and JT2FISPanel on visual applications.

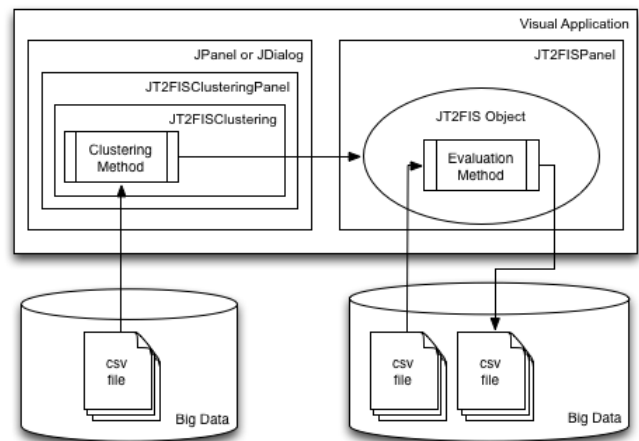


Fig. 13. Data mining process using JT2FISClusteringPanel and JT2FISPanel on visual applications.

The JT2FISClusteringPanel is beneficial for facilitating the FIS creation from data mining process. Using clustering methods, the JT2FISClusteringPanel uses real data to build a FIS configuration. The generated FIS can be sent to JT2FISPanel for visualization. Linking both components, we can produce and analyze a FIS before utilizing it on concrete intelligent software solutions.

VII. CONCLUSIONS AND FUTURE WORK

JT2FISPanel and JT2FISClusteringPanel are Java visual components to Build Java Intelligent Applications using Java Interval Type-2 Fuzzy Inference System and clustering method for fuzzy sets discovered from data mining process. We present an object oriented design of a JT2FISPanel and JT2FISClusteringPanel visual components based on SWING JPanel. We provide an example of how to use the JT2FISPanel and JT2FISClustering Panel for FIS easy configurations.

The JT2FISClusteringPanel is beneficial for facilitating the FIS creation and visualization of an FIS before using them it on concrete intelligent software solutions.

As a future work, first, we will continue adding features and re-factoring the code in order to improve performance and usability. Finally, we are planning to extend

the JT2FISPanel and JT2FISClusteringPanel tests in order to compare usability with other similar Type-2 Fuzzy Inference Systems Tools.

ACKNOWLEDGMENT

This work was supported in part by the Internal Fund for Research Projects (Grant No. 300.6.C.135.17) of the Autonomous University of Baja California, México.

REFERENCES

- [1] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [2] —, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.
- [3] L. Lucas, "General Type-2 Fuzzy Inference Systems: Analysis, Design and Computational Aspects," *Fuzzy Systems . . .*, no. 5, 2007.
- [4] O. Castillo, L. Aguilar, N. Cázarez-Castro, and S. Cardenas, "Systematic design of a stable type-2 fuzzy logic controller," *Applications Soft Computing*, 2008.
- [5] O. Castillo, P. Melin, and J. R. Castro, "Computational intelligence software for interval type-2 fuzzy logic," *Journal Computer Applications in Engineering Education*, 2010.
- [6] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Science*, vol. 8, no. 199249, pp. 301–357, 1975.
- [7] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, ser. MATLAB Curriculum Series. Upper Saddle River, NJ: Prentice Hall, 1997.
- [8] J. R. Castro, O. Castillo, and L. G. Martínez, "Interval type-2 fuzzy logic toolbox," *Engineering Letters*, vol. 15, no. 1, 2007.
- [9] M. García-Valdez, G. Licea-Sandoval, A. Alaníz-Garza, and O. Castillo, "Object oriented design and implementation of an inference engine for fuzzy systems," *Engineering Notes*, vol. 15, no. 1, August 2007.
- [10] P. Cingolani and J. Alcalá-Fdez, "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation," *2012 IEEE International Conference on Fuzzy Systems*, pp. 1–8, Jun. 2012.
- [11] C. Wagner, "Juzzy a java based toolkit for type-2 fuzzy logic," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, Singapore, 2013.
- [12] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann, 2006.
- [13] Two-Crows, "Introduction to Data Mining and Knowledge Discovery," *Two Crows Corporation*, 1999.
- [14] S. Bozkir and E. Sezer, "FUAT - A fuzzy clustering analysis tool," *Expert Systems with Applications, FUZZYSS11: 2nd International Fuzzy Systems Symposium*, vol. 40, p. 403: 842849, 2013.
- [15] J. Bezdek, "FCM: The fuzzy c-means clustering algorithm," *Computers and Geosciences*, p. 1023 and 191203, 1984.
- [16] X. Yin, L. Khoo, and Y. Chong, "A fuzzy c-means based hybrid evolutionary approach to the clustering of supply chain," *Computers and Industrial Engineering*, p. 664 and 768780, 2013.
- [17] M. Castañón-Puga and J. Castro and M. Flores-Parra and C. Gaxiola-Pacheco and L. Martínez-Méndez and L. Palafox-Maestre, "JT2FIS A Java Type-2 Fuzzy Inference Systems Class Library for Building Object-Oriented Intelligent Applications," in *Advances in Soft Computing and Its Applications*, ser. Lecture Notes in Computer Science, vol. 8266. Berlin Heidelberg: Springer, 2013, pp. 204–215.