**An Open-Source Software for Low-Code Implementation of Fuzzy Logic for Process Engineering**

Raphael A. Santanaa, Vitor A. O. Silvaa, Tiago F. Souzaa, Vitor M. V. Cruza,Ana M. F. Filetia, Flávio Vasconcelos da Silvaa,\*

*aUniversity of Campinas (UNICAMP), School of Chemical Engineering, Chemical Systems Engineering Department – Albert Einstein Ave., 500, Campinas – SP, 13083-852, Brazil*

*\*flaviovs@unicamp.br*

**Abstract**

Fuzzy logic is suitable and widely applied to modeling and controlling multivariable processes with non-linearities and susceptible to uncertainties. Furthermore, fuzzy logic has been utilized in risk analysis and process safety. The main reason for widely applying fuzzy logic in modeling, control, risk analysis, and process safety is using human experience to evaluate the processes through linguistic variables, which are naturally diffuse and imprecise. Implementing fuzzy logic for the previously mentioned situations requires using some programming languages. However, a lack of programming skills is a barrier to the engagement of engineers, educators, and students in applying fuzzy logic in several engineering areas. In this context, this paper aims to evidence the development and characteristics of open-source software for low-code implementation of fuzzy logic to process control, modeling, risk analysis, and process safety. This software, named FuzzyWise, is entirely based on the Python language. It provides a comprehensive library that offers a range of tools for creating various fuzzy systems through the code editor or using an intuitive and versatile interface, allowing users to download the source code for use in other applications. The FuzzyWise will empower engineers, educators, and students in process engineering to implement fuzzy logic effectively.

**Keywords**: fuzzy logic, process engineering, open-source software, low-code software, engineering education.

1. **Introduction**

Fuzzy logic has emerged as a very powerful tool in dealing with complex problems and uncertainty, particularly in engineering application in problems that lays on lack of data and dependence on specialists knowledge (Simão and Shawn, 2007).

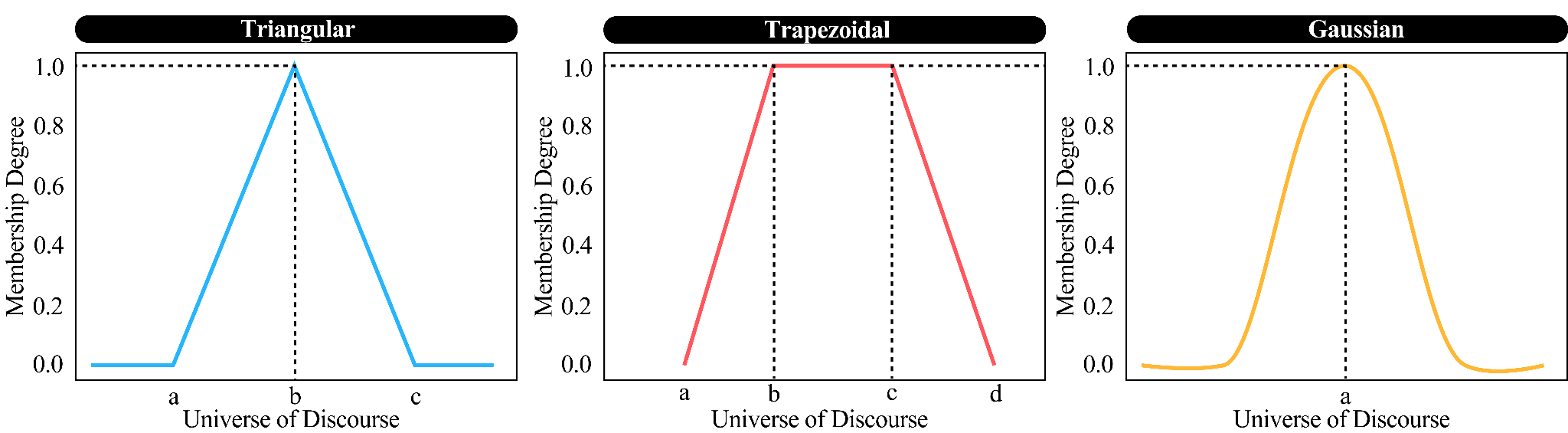
The theory of fuzzy logic introduces the notion of fuzzy sets, referred to as membership functions, which determine the degree of membership of a specific element within a set. This concept serves as a gateway to explore and apply human reasoning. There are a lot of real-world problems where engineers and scientists do not have mathematical models to describe, but it has specialists observations: "when the temperature of a turbine is cool and the pressure is weak, the action of the throttle is positively large". Through that kind of description, the fuzzy logic allows to simulate human decision making, mathematically translating the human reasoning through various methods.

There are several Python libraries available for implementing fuzzy logic, but a solid understanding of the Python language is essential. It wouldn't be sufficient to merely comprehend the concept of fuzzy logic; acquiring proficiency in utilizing the Python language would be equally crucial. In this regard, there are graphical UI, such as the Fuzzy Logic Toolbox in Matlab, that facilitate the application of fuzzy logic. However, it's important to note that a paid subscription is required to access this toolbox.

In this context, the current study seeks to create a completely free, intuitive, and low-code application for implementing fuzzy inference systems across diverse scenarios in process engineering. Also, aims to establish a library that consolidates various methods in fuzzy logic, thereby facilitating the dissemination of the concept.

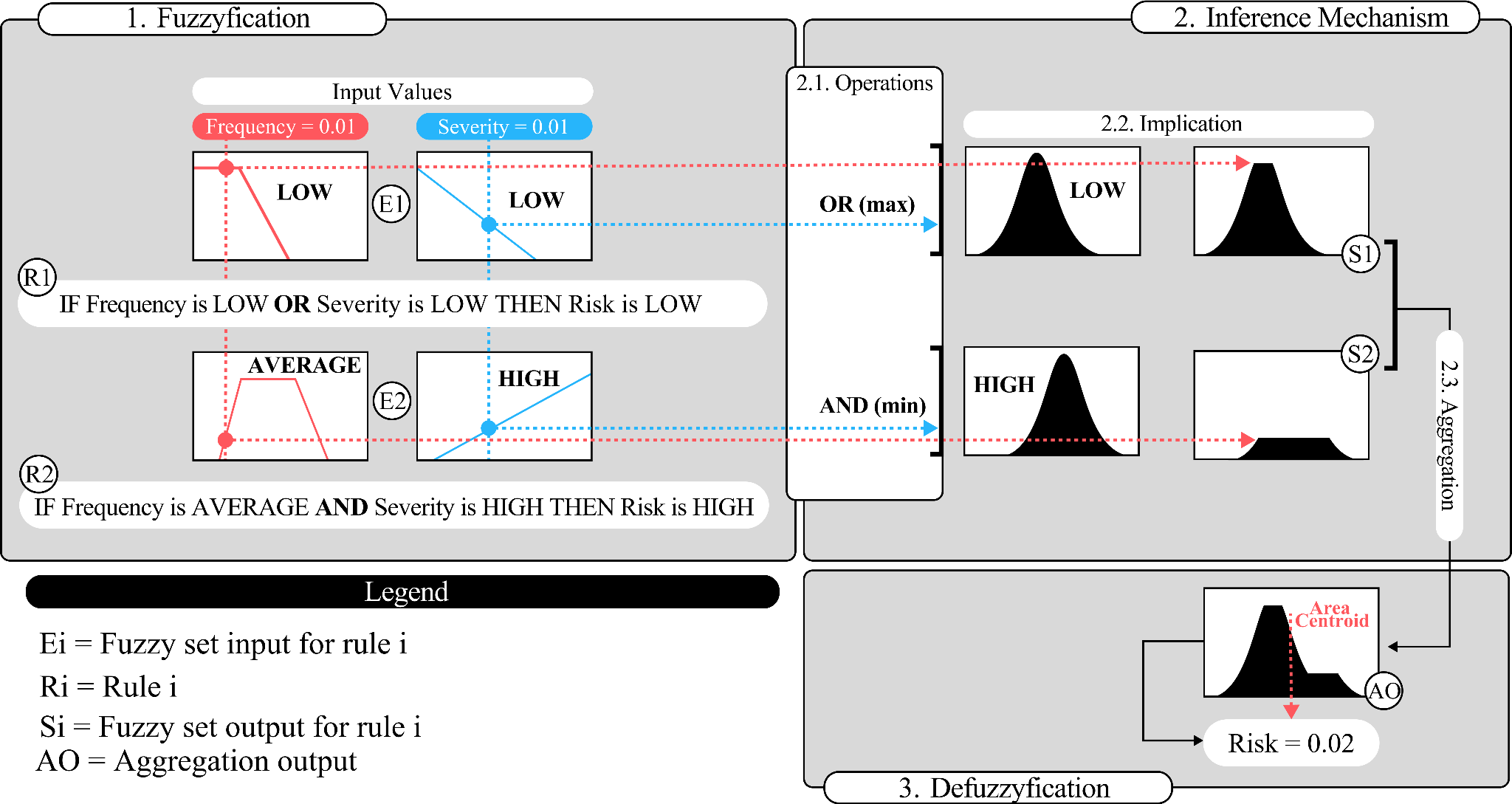
1. **Fundamentals and Methodology**
   1. *Fuzzy Logic*

The fuzzy logic was introduced by Zadeh (1965) as a mechanism for modelling human conceptualizations that involve uncertainties not representable by the traditional probability theory. Unlike boolean logic, where an observation either belongs or does not belong to a set (1 or 0), the fuzzy logic extends this characterization using degrees of membership. In this concept, observations partially belong to one or more sets defined by linguistic variables (Vilela, Oluyemi and Petrovski, 2020). For instance, a temperature of 40 ºC might be 80% a part of the “hot” set and 20% of the “warm” set. Consequently, for each value in a universe of discourse, the membership function assigns a value between 0 and 1, describing various types of mathematical functions, such as triangular, trapezoidal, and gaussian. Figure 1 illustrates these previously mentioned membership functions.



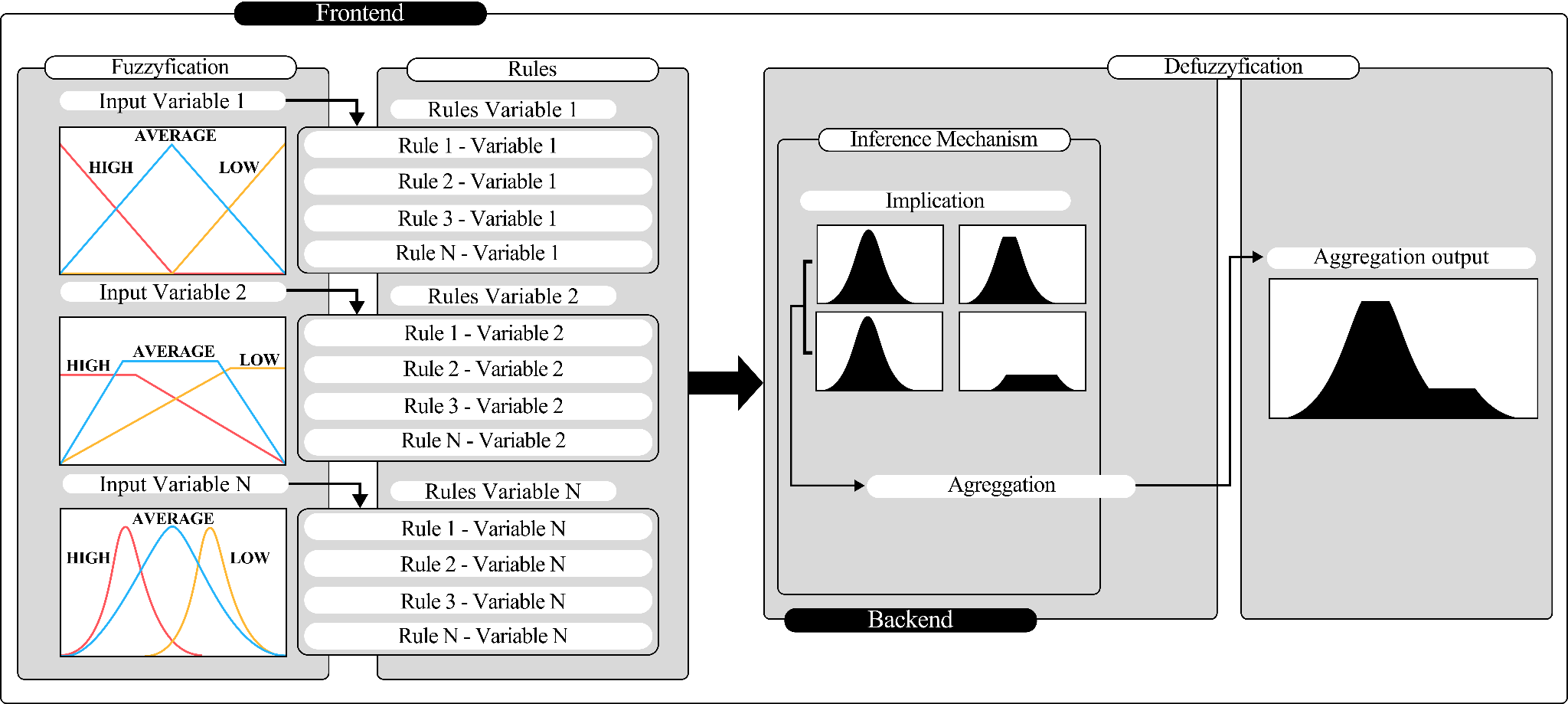
**Figure 1:** Triangular, trapezoidal, and gaussian membership functions.

Given that the entire universe of discourse for input and output variables is represented by a set of membership functions (fuzzy sets) through the capture of human knowledge, the fuzzy inference system is the process that uses fuzzy logic to map input variables to output variables once a fuzzy set. This process is inherently computational and comprises three fundamental blocks: Fuzzification, Inference Mechanism (including operation, implication and aggregation), and Defuzzification (Figure 2).

**Figure 2:** Illustration of the fuzzy logic system and fuzzy inference mechanism.

1. **Fuzzification:** This initial stepinvolves mapping the universe of discourse, establishing the correspondence between each numerical value of the variables through appropriate membership functions.
2. **Inference Mechanism:** Based on the fuzzified variables, the mechanism evaluates the fuzzy rules activated from a predefined rule base. Each rule defines an implication between conditions using logical operators such as intersection (AND), union (OR), and complement (NOT). Then, all activated rule implications are aggregated. The rules structure depends on the inference method employed. The most commonly used methods are the Mamdani method and the Takagi-Sugeno method (Santana *et al*., 2022). Mamdani uses linguistic variables for both inputs and outputs (“**IF input 1 is A AND/OR input 2 is B THEN output is C”**), while Takagi-Sugeno defines the output of a rule directly as a crisp value through the weighted average of inputs [**“IF input 1 is A AND/OR input 2 is B THEN output is a1(input 1) + b1(input 2)”**].
3. **Defuzzification:** For the Mamdani inference method it is necessary to transform the fuzzy output obtained by implication into a crisp value. The most common method for this conversion is the centroid.
   1. *FuzzyWise Architecture*

This section elaborates on the methodology employed to develop FuzzyWise, providing a detailed insight into the systematic approach adopted for the software implementation, as can be seen in Figure 3.

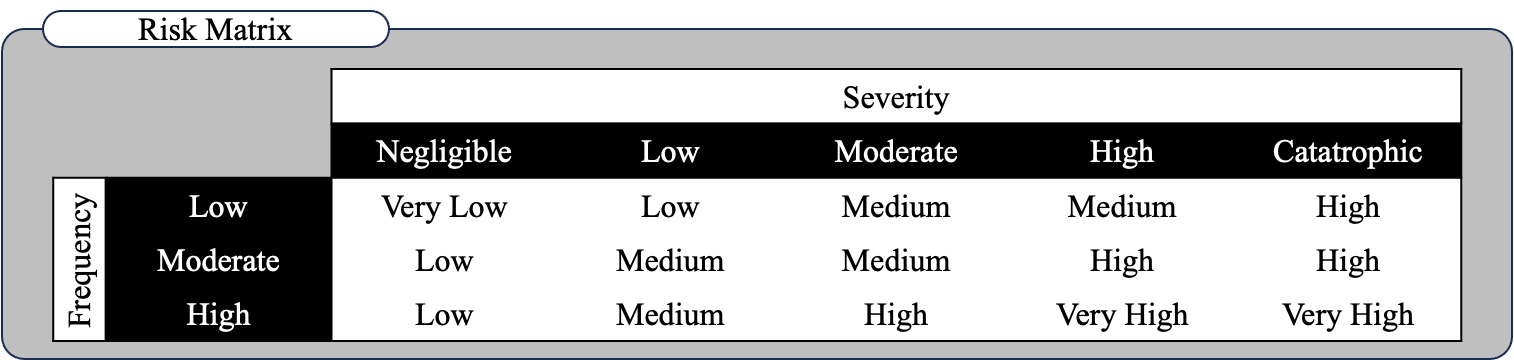


**Figure 3:** FuzzyWise architecture diagram.

The backend architecture of FuzzyWise serves as a robust computational framework, adept at overseeing intricate fuzzy logic processes through meticulously optimized algorithms developed for a newly created Python library. Its scalability guarantees steadfast decision-making across different scenarios, maintaining computational efficiency. On the frontend, developed using the PyQt5 library, FuzzyWise offers users an intuitive interface to input linguistic variables and witness real-time evaluations of fuzzy logic. The incorporation of responsive design and graphical representations enhances user comprehension, rendering complex fuzzy logic concepts easily accessible.

1. **Results and Discussion**

To showcase the capabilities of FuzzyWise, we analyzed a common case study widely used in risk analysis and process safety, specifically focusing on the risk matrix depicted in Figure 4.

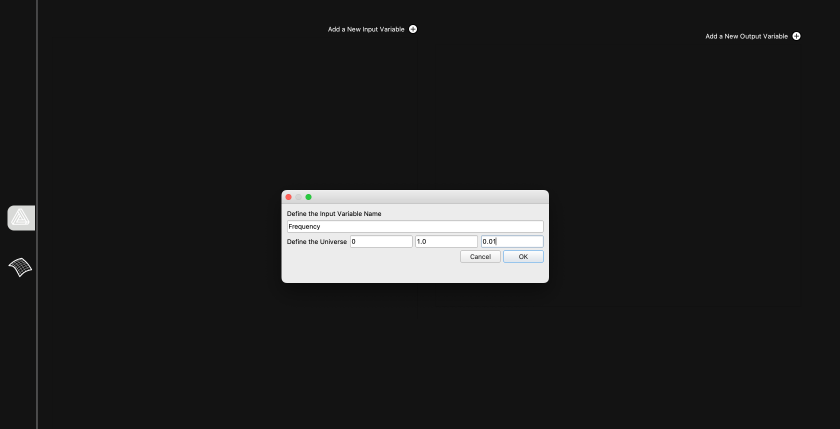


**Figure 4:** Representation of the case study risk matrix.

The illustrated risk matrix relates the frequency and severity of a general event to determine the consequence risk associated with the event. Only relying on the risk matrix has the following limitation: the determined risk value is merely qualitative. In order to overcome this limitation, fuzzy logic can be applied to determine the quantitative risk values (Almeida et al., 2023). Based on the risk matrix illustrated in Figure 4, a fuzzy inference system can be developed to model the relation between input variables (frequency and severity) and output (consequence risk).

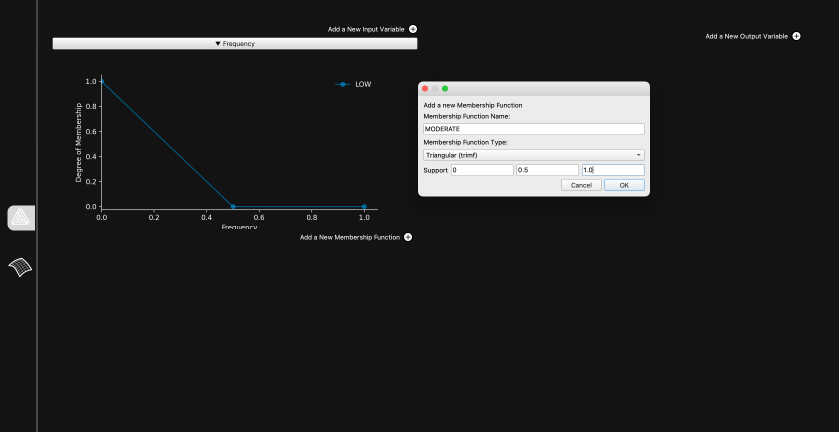
The results will be presented as the sequential steps necessary for implementing the ongoing case study under investigation, with each step thoroughly elucidated. It is worth noting that identical results can be obtained by employing the FuzzyWise library in any code editor through the terminal.

***Step 1: Creating the Input Variables - Fuzzification***

******

**Figure 5:** FuzzyWise variable creation.

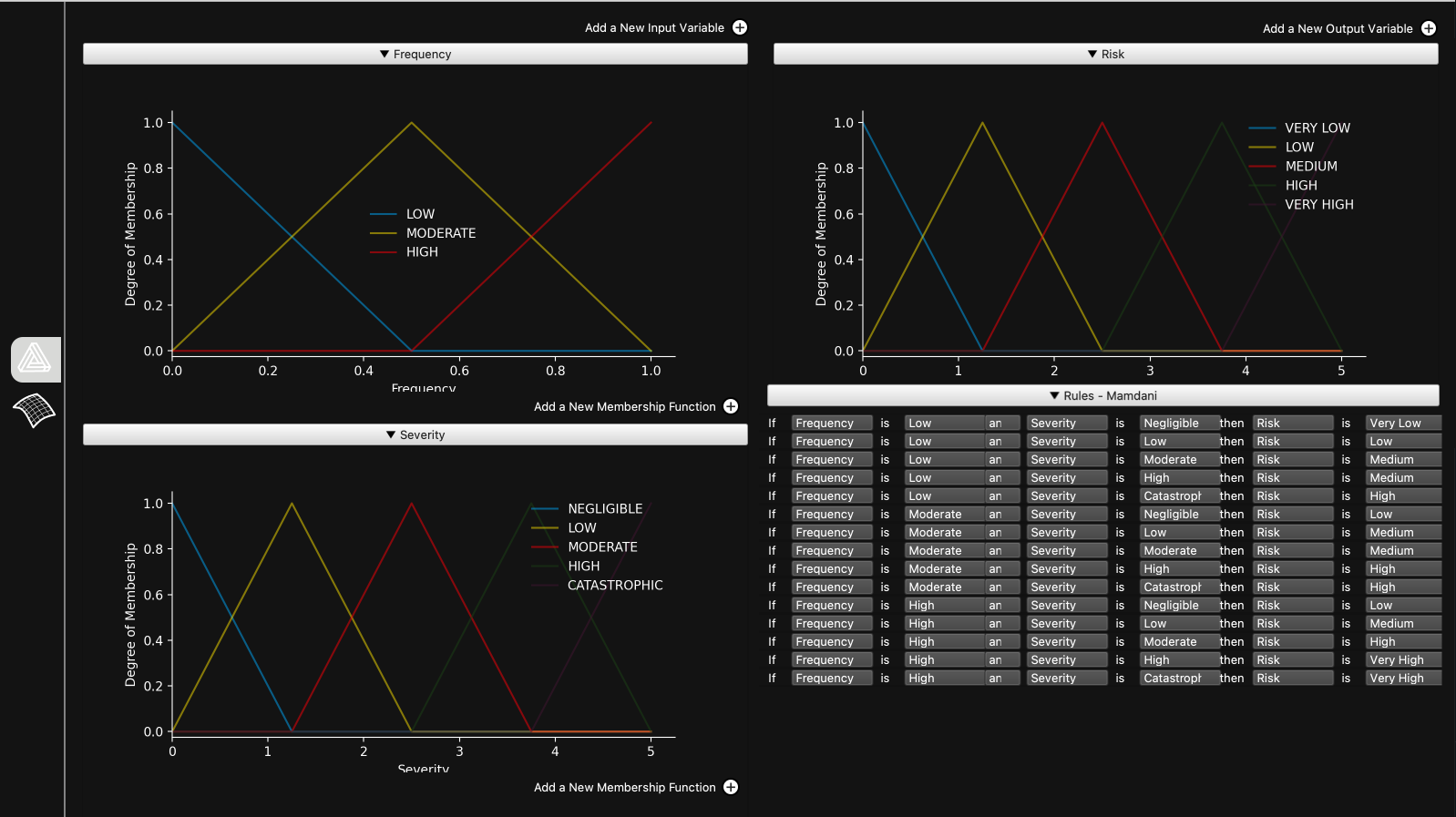
***Step 2: Adding Membership Functions***

******

**Figure 6:** FuzzyWise membership functions creation.

***Step 3: Rules and Inference Mechanism***

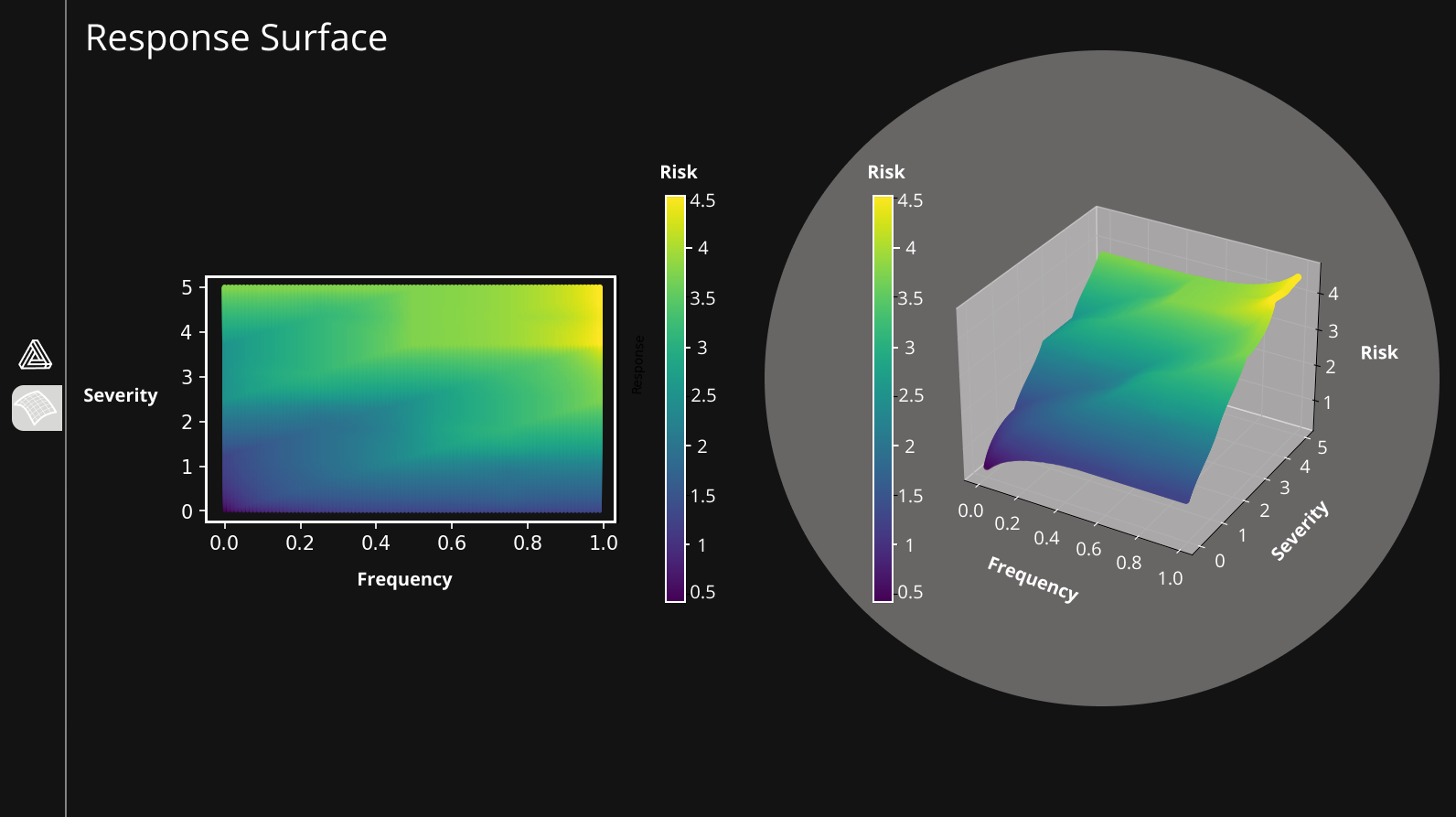
Figure 7 shows a comprehensive overview of the input variables, universe of discourse, associated membership functions support, output variable, and inference method.



**Figure 7:** Interface representation of the case study implementation using FuzzyWise.

***Step 4: Defuzzification and Response Surface***

Tuning the fuzzy system is a critical step, and retrieving the real-time response surface as soon as it is updated can significantly enhance the specialist's ability to quickly achieve the desired configuration. Therefore, Figure 8 shows the response surface for the case study under discussion.

****

**Figure 8:** Response surface for the implemented case study.

The response surface is presented in two dimensions for improved user understanding, with an additional interactive 3D representation for a more engaging and insightful exploration of the results. This approach enhances the visual appeal and facilitates a comprehensive data analysis, promoting intuitive interpretation. The implementation of the case study is detailed in a tutorial video available on [FuzzyWise Online Library](https://drive.google.com/drive/folders/1aVc-jYhv0WjulosRfWsD9pdM2-nmPCYD?usp=sharing).

1. **Conclusion**

FuzzyWise is an open-source software that enables implementing fuzzy logic systems without prior knowledge of any programming languages. The user-friendly graphical interface of this software makes it possible to implement fuzzy logic systems by directly defining the following parameters: fuzzy system input and output variables and their membership functions, inference method, defuzzification method, and rule base. The characteristics previously mentioned about FuzzyWise are the main contributions related to developing this software since there is a lack of open-source softwares for low-code implementation of fuzzy logic. The available version of FuzzyWise has limitations, such as the possibility of using only the Mandani or Takagi-Sugeno inference method. Therefore, software improvements will enable the development of the most diverse systems based on fuzzy logic.

**References**

Simões, M. G., & Shaw, I. S. (2007). Controle e modelagem fuzzy. Editora Blucher.

Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8, 338-353.

Vilela, M., Poluem, G. F., & Petrovski, A. (2020). Sensitivity analysis applied to fuzzy inference on the value of information in the oil and gas industry. International journal of applied decision sciences, 13, 344-362.

Santana, R., Vianna, S. S. V., & Silva, F. V. (2022). A novel approach in fuzzy bowtie analysis applying Takagi–Sugeno inference for risk assessment in chemical industry. Journal of Loss Prevention in the Process Industries, 80, 104892.

Almeida, R. S., da Silva, F. V., & Vianna, S. S. (2023). Combining the bow-tie method and fuzzy logic using Mamdani inference model. Process Safety and Environmental Protection, 169, 159-168.