The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

ID Number: 16522057

Supervisor: Professor Christian Wagner

Module Code: G53IDS

2020/05

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

School of Computer Science

Towards Data Driven Environmental Management

Submitted [May 2020], in partial fulfilment of
the conditions for the award of the degree **BSc Hons Computer Science**

ID Number: 16522057
School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature _____Yixin Qi_____

Date ___03__/__05___/__2020___

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive.*

Public access to this dissertation is restricted until:04/05/2020

Abstract

How to judiciously manage the environment to ensure human well-being has long been an important but complex problem. A values-driven environmental management framework has been recognized as a feasible solution to this problem which uses the nature's contributions to human well-being to guide the management process. But due to the lack of tools for quantifying the natural elements' contributions to human well-being, it has been difficult to use the framework in real-world scenarios.

Thus, in this project, a tool for supporting environment management will be developed and examined to quantify natures' contributions to people and facilitate the operationalizing of a values-driven framework.

Acknowledgements

I would like to express my sincere thanks to my supervisor Professor Christian Wagner for his encouragement and guidance throughout this project. I would also like to thank Chao Chen for helping me with the implementation of the software and Dr Michael Smith for providing me the data from Atlantis Management case. Additionally, I would like to express my gratitude to the anonymous potential user for his/her contributions.

A special thanks to my friends and family for their continuous support.

Table of Contents

1 Introduction and Motivation ---------- *8*
1.1 Overview ---------- 8
1.2 Aim and Objectives ---------- 8
1.3 Innovation and Previous work ---------- 9
2 Background ---------- *9*
2.1 Values-driven Framework ---------- 9
2.2 Apply Concept of Properties to Quantify Elements' Contributions ---------- 10
2.3 Fuzzy Inference System ---------- 13
2.4 Generate Uncertain Fuzzy Logic Rules from Surveys ---------- 16
2. 5 Interval-valued Assessment of Value Importance ---------- 20
2.6 Interval Agreement Approach (IAA) ---------- 22
3 Development Methodology ---------- *23*
4 Requirements ---------- *23*
4.1 Functional Requirements ---------- 24
4.1.1Frontend Requirements ---------- 24
4.1.2 Backend Requirements ---------- 24
4.2 Non-Functional Requirements ---------- 25
5 Design ---------- *26*
5.1 System Overview ---------- 26
5.2 Rule Generation ---------- 28
5.3 FIS Generation ---------- 33
5.4 Rating Information Extraction and Utility Calculation ---------- 34
5.4.1 Rating Information Extraction ---------- 34
5.4.2 Utility Calculation ---------- 36
5.5 Data Visualization ---------- 36
5.6 Rule Customization and Rule Exhibition ---------- 37
5.6.1 Rule Customization ---------- 37
5.6.2 Rule Exhibition ---------- 38
5.7 Frontend Design ---------- 38
5.7.1 Set Up Page Design ---------- 39
5.7.2 Utility Page Design ---------- 42
5.7.3 Customization Page Design ---------- 43
5.7.4 Rule Page Design ---------- 44
6 Implementation ---------- *46*
6.1 Language ---------- 46
6.2 Implementation of Rule Generation ---------- 46
6.3 Implementation of FIS Generation ---------- 57
6.4 Implementation of Rating Information Extraction ---------- 58

6.5 Implementation of Utility Calculation ---------- **63 -**

6. 6 Implementation of Data Visualization ---------- **64 -**

6.7 Implementation of Rule Customization ---------- **66 -**

6.8 Implementation of Rule Exhibition ---------- **67 -**

6.9 Frontend Implementation ---------- **68 -**

7 Case Study and Evaluation ---------- ***- 69 -***

7.1 Background ---------- **69 -**

7.2 Procedures ---------- **70 -**

7.3 Observations ---------- **70 -**

7.3.1 Property-Value Relationship Elicitation ---------- - 70 -

7.3.2 Rating Elements and Values ---------- - 72 -

7.3.3 Elements' Contribution Calculation ---------- - 75 -

7.4 Evaluation Summary ---------- **79 -**

8 User Review ---------- ***- 80 -***

8.1 Functionality Improvement ---------- **80 -**

8.1.1 User Defined Property Score Range ---------- - 80 -

8.1.2 Visualize the Membership Function Change ---------- - 80 -

8.2 User's Review on Usability and Portability ---------- **81 -**

8.3 Conclusion ---------- **81 -**

9 Summary and Further work ---------- ***- 81 -***

9.1 Project Progress ---------- **81 -**

9.2 Further Work ---------- **83 -**

9.3 Conclusion and Reflection ---------- **84 -**

Bibliography ---------- ***- 85 -***

1 Introduction and Motivation

1.1 Overview

The pivotal relationship between human well-being and environment has been recognized in previous research. As it is emphasized in MEA (2005) [1], environment has contributions to all aspects of human well-being. In recent papers, the necessity for better personnel management over natural resource distribution has been highlighted [2] due to the increasing destructive conflicts among natural resources [3][4]. However, how to reconcile resources' conflicts and judiciously manage environment to ensure human well-being has long been a vital problem.

One valid approach is to utilize a values-driven environmental management framework that is proposed in [7] which utilizes the natural elements' contributions to human well-being as drivers for the decision-making process in environment management (the elements here indicate the natural biotic and abiotic elements in an environment such as fish and tree in a lake conservation). Yet, as noted by Mace (2014) [5], one major obstacle for using this framework is the dearth of scientific tools and technology for linking natural elements and human well-being.

Thus, in this project, a software is developed to quantify natural elements' contributions to human well-being (i.e. utility) using fuzzy inference system and facilitate the operationalization of values-driven framework. This software's functionalities will be further examined in section 9 using a data from a case study.

1.2 Aim and Objectives

The overall goal of this project is to develop a software to facilitate the values-driven framework for natural resource management. It will be able to utilize fuzzy inference system to model the contributions of natural elements to people. As mentioned above, such knowledge is pivotal for the managers during decision-making debates over resource management. In values-driven framework, the environment managers can utilize this information to make judicious management decisions towards natural elements to further ensure human well-being.

Achieving this aim entailed:

- Creating interfaces using ShinyR to allow users to set up the software using their own data. (The ShinyR is a library for frontend implementation. This will be clarified in Section 6)

- Linking the interfaces to the backend library FuzzyR (which will be introduced in more details in Section 6) to construct a fuzzy inference system based on users' input elements, properties, values and property-value relationships. And this FIS system will be later used by users to calculate the utilities using elements' property scores as inputs. (detailed definitions for value, property and property-value relationships can be found in in Section 2.1, Section 2.2).
- Examining the functionalities of the software using data from a case study. (Details will be demonstrated in Section 7)
- Inviting potential user (e.g. expert in environment management) to test the software (Section 8).
- Improving the software based on user's suggestions (e.g. change the design of the interface or add more functions so that the software can be easier to use).

1.3 Innovation and Previous work

A feasible fuzzy logic-based system that can be used to facilitate the values-driven framework has been proposed by Pourabdollah et al. (2014) [13]. This system can support decision-making in environment management by quantifying the nature elements' contributions to human well-being through their properties (definition in Section 2.2). However, a drawback of the system is that although it is fully functional, it can only be used limitedly by specific projects (i.e. the managers (users) cannot customize the system to suit the context of their own cases). Therefore, in this project, a software that can be adapted and used by environment management practitioners and researchers for their own cases will be introduced to support the environment management by providing quantified links between elements and human well-being.

2 Background

2.1 Values-driven Framework

This software is designed and implemented to facilitate a values-driven framework [7] (detailed steps for utilizing the framework will be demonstrated in Section 2.1) which uses "human values" to drive the decision-making process in nature resource management.

As defined by Wallace (2007) [8] the term "human value" here stands for the "The preferred end-states of existence, including those required for human survival and reproductive success". In other words, it means the states of environment preferred by one or more stakeholder groups (the stakeholders here indicate people who affect or are affected willingly by the environment management [12]). For example, in a lake conservation management case, if the stakeholders such as local residents, government or investors expect this conservation to be more suitable for leisure activity as well as to be more aesthetically pleasing after management then the human values in this scenario entail "recreation (The

importance of biological elements for leisure activities) [9]" and "aesthetic pleasure (Scenic and other aesthetic values of natural biological elements) [9]".

In values-driven framework, in order to achieve human values, all the decisions and policies are designed and conducted to manage the natural elements that can deliver human values. In an idea state, all management actions will increase these elements' contributions to human values. Since, according to Gregory et al. [10], human well-being is composed by the satisfaction of a combination of human values, this framework can be a reasonable and feasible approach to link natural elements and human well-being through human values. That is, the framework can ensure human well-being by managing natural elements. The relationship among element, human value and human well-being is summarized using the example exhibited in *Figure1.*

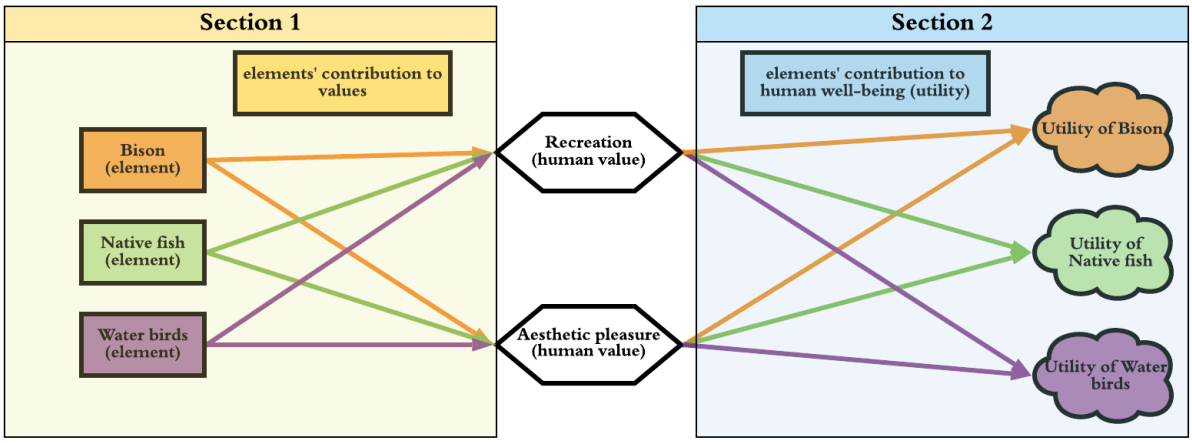


Figure 1. This graph shows three elements (*Bison, Native fish* and *Water birds*) and their contributions to two human values (*Recreation* and *Aesthetic pleasure*) and human well-being. It is assumed in this example that the satisfaction of human well-being in this context solely depends on the two values mentioned in the graph. The arrow lines in Section 1 of the figure (the yellow block) represent the corresponding elements' contributions to the two human values. And in Section 2 (the blue block), the elements' contributions to human values are combined into their contributions to human well-being (i.e. utility). Through this relationship, the manager can ensure human well-being by managing the elements to increase their contributions to human values and human well-being.

2.2 Apply Concept of Properties to Quantify Elements' Contributions

While utilizing the values-driven framework, it is inevitable to encounter two difficulties: 1) How to calculate the elements' contributions to human values (i.e. utilities). 2) How to manage these elements to effectively increase human values. To support this framework, this software solved these two problems by employing the method proposed by Smith et al. (2016) [6] which quantifies the elements' contributions to human well-being based on elements' "states".

To solve the first problem (i.e. quantify utility), three key steps of the method are elaborated below:

1) *Describe the "state" using property scores.*

According to [6], the "states" of elements are described by its properties (or "criteria" as referred by Margules and Usher (1981) [11]) such as the *richness* (*number of species* [6]) or *population*. To utilize this method, properties need to be quantified using property scores. For example, if the state of the element *oak tree* is described by two properties: *richness* and *population*, then the two properties can be quantified using scores ranging from 0-100 with 0 being the smallest in *richness* and *population* and 100 being the largest in these two properties. Since the "states" are composed of properties, it can be said that the "states" are quantified when they are described by property scores. In reality, one major challenge of using this method is defining properties' maximum property scores appropriately. For example, for a property whose property scores are within the range {1,2, 3…, *N*}, it is always difficult to define the '*N*' based on the given context of the property. One valid approach is to consult experts about the proper value for '*N*'. After the score ranges are defined, the score of a property in a given situation can be decided through, literature and departmental data as well as experts' opinions [6].

2) *Quantify property-value relationships*

As demonstrated in [6], elements' contributions to human values are functions of their properties (e.g. *If the Population of an Element is Large then the element's contribution to Recreation Value is High or If the Richness (number of species) of an Element is Low then the element's contribution to the Aesthetic pleasure is Low*). To quantify the elements' utility, these elements' property-value relationships need be collected and quantified to connect the elements' quantified states (i.e. property scores) to human well-being. (The details about the quantification of the relationships will be clarified in Section 2.3)

3) *Generate utility using property scores and quantified property-value relationship*

Once the property-value relationships are quantified, the elements' contributions to human values at certain states can be obtained based on these relationships using the elements' property scores as inputs. And these contributions to human values can be further combined to obtain the elements' utilities (contributions to human well-being).

As for the second problem mentioned above, since the elements' contributions can be obtained by using properties as inputs, the managers or decision makers can further explore the input/ output relationship (i.e. the properties/ contributions relationship) to effectively manage the element by adjusting its properties. For example, in a lake reservation case, if the managers acknowledge that the *ducks* (element) have larger contributions to *recreation* (value) when they are larger in *size* (property), then to effectively fulfil the *recreation* value, they can consider increasing the *size* by increasing the population of the *ducks*. (The *size* here stands for "The total area of occupancy of the species that characterise the element [6]".)

This method has been summarized in *Figure 2*. It should be noticed that each arrow line between a property and a value in the yellow highlighted part represents all the quantified

relationships between this property and this value. In the green block of the figure, the elements' contributions to human values are further combined to obtain their utilities.

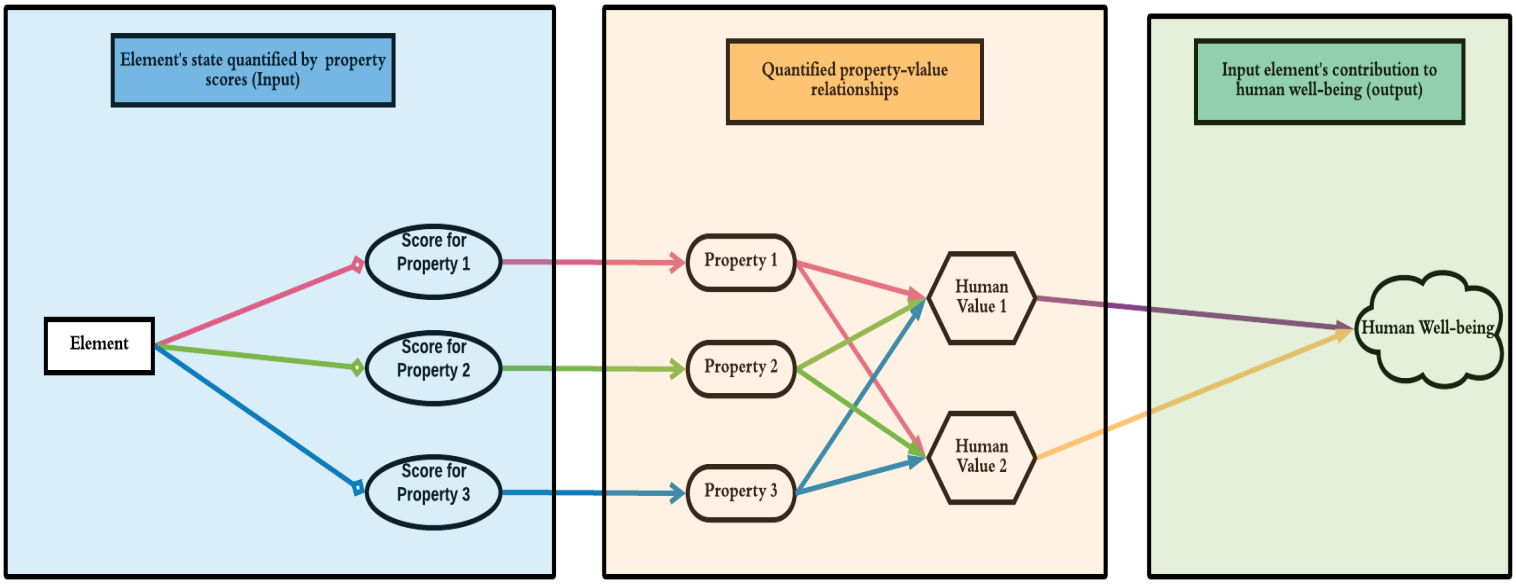


Figure 2. This graph summarizes the method described in Section 2.2 which uses the elements' property scores as inputs to estimate the elements' contributions to human well-being based on the quantified property-value relationships.

To cooperate this method with values-driven framework, the following steps of conducting values-driven framework [31] are formalized by expanding the first two steps mentioned in [7]. As exhibited in Section 2.1, this framework will use the outcomes of the last step (the elements' contributions to values) as indicators for future decision-makings. The design and implementation of this software will focus on facilitating the steps 3 to 5.

1) ***Elicit key values desired by the different stakeholders.***
2) ***Identify elements that can deliver the key values.***
3) ***Rate the importance of the chosen values based on stakeholders' views.***
4) ***Collect and quantify the property-value linkage.***
5) ***Obtain the elements' contributions to values and prioritize them based on the contributions.***

To operationalize this framework by following these steps, the property-value relationships and ratings of values need to be explored. However, as stated in [6], there is rarely any absolute definition for them. As a result, the value ratings will be elicited from groups of stakeholders while the property-value relationship collection and quantification will be based on the opinions collected from experts. Since these stakeholders and experts are likely to have different stands and perspectives on relative value importance and property-value relationships, it is reasonable to take the uncertainties within the collected data and the potential imprecisions caused by the uncertainties into consideration while conducting step

3, step 4 and step 5 [31]. Thus, in this project, specific relationship collection/quantification (details in Section 2.5) and value rating manners (details in Section 2.4) will be utilized in cooperation with a Fuzzy Inference System (FIS) (details in Section 2.3) due to their recognized competence for dealing with uncertainties [6][13].

2.3 Fuzzy Inference System

The fundamental principle behind Fuzzy inference system (FIS) is to employ a set of "fuzzy rules" as guidelines to construct a mapping mechanism between inputs and outputs based on fuzzy logic. The overall structure of FIS is summarized in *Figure 3* and the highlighted part is where the system maps the input to output based on the fuzzy rules in rule base.

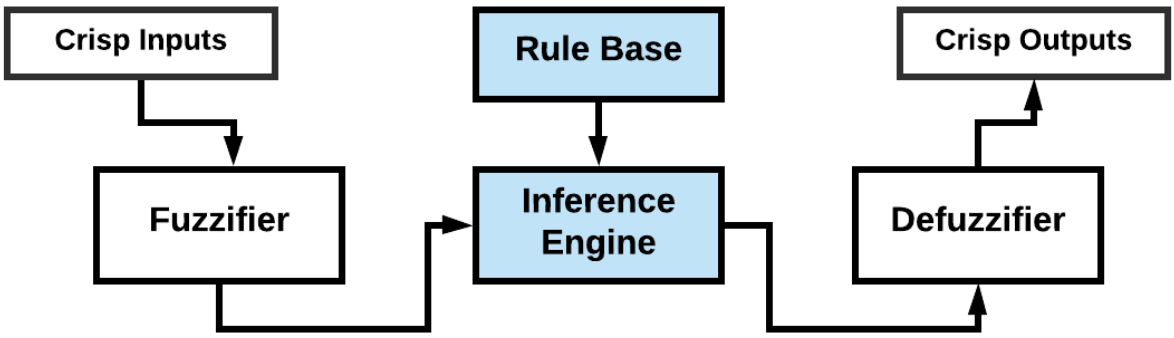


Figure 3. This figure summarizes the structure of Fuzzy Inference System. The highlighted part utilizes the stored rules to map a fuzzified input to a fuzzified output.

Specifically, each rule will generate a fuzzified output based on its fuzzified input(s) and these fuzzified outputs generated by different rules will be combined and defuzzified to generate the final crisp output. This is demonstrated in *Figure 4* using an example of tipping in a restaurant. One thing needs to be noticed is that, in this graph only one output variable (i.e. "Tip") is involved in the rules, thus, there is only one crisp output. The output variables in the FIS can be understood as the term after the word "then" in rules.

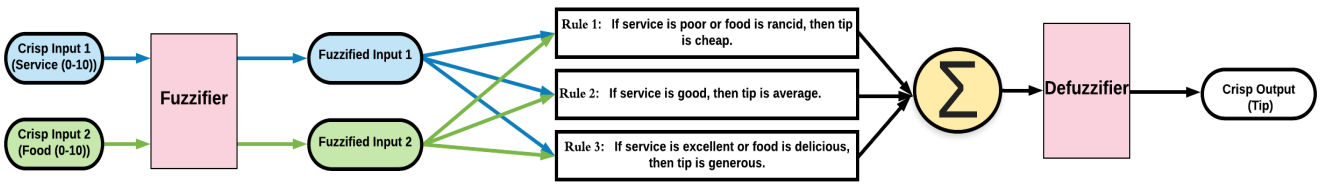


Figure 4. This figure uses an example of tipping in restaurant to show the process of using rules to map inputs to output in FIS. The crisp inputs (property scores) from Service and Food are first fuzzified though fuzzifier. Then the fuzzified outputs generated by rules based on the fuzzified inputs will be combined and defuzzified into the crisp output (tip).

In the context of this project, the FIS will be utilized to implement the process demonstrated in *Figure 2* with the "Crisp Inputs" in *Figure 3* being the property scores, the rules in "Rule Base" being the quantified property-value relationships and "Crisp Outputs" being the elements' deliveries for values. Since multiple human values will be included in the rules, there will be more than one crisp output. The utility then will be calculated as a weighted average of the crisp outputs (the weights are assigned based on the values' importance which will be clarified in Section 2.5). In other words, this software will employ the FIS to support the step 5 of the values-driven framework mentioned in Section 2.2. It will output the elements' contributions to values based on the collected and quantified property-value relationships by using the elements' property scores as input. The relationships among these parts are again clarified in *Figure 5.* In this figure, the crisp outputs (elements' contributions to values) are further combined to obtain the utility of the input element. It should be noticed that in the "*Fuzzified Output*" block (i.e. the purple block), the fuzzified outputs generated by the rules can also be aggregated using weights (details of the weights will be mentioned in Section 2.4).

The use of FIS here addresses the shortcomings (e.g. imprecisions) that may be introduced by uncertainties for this software.

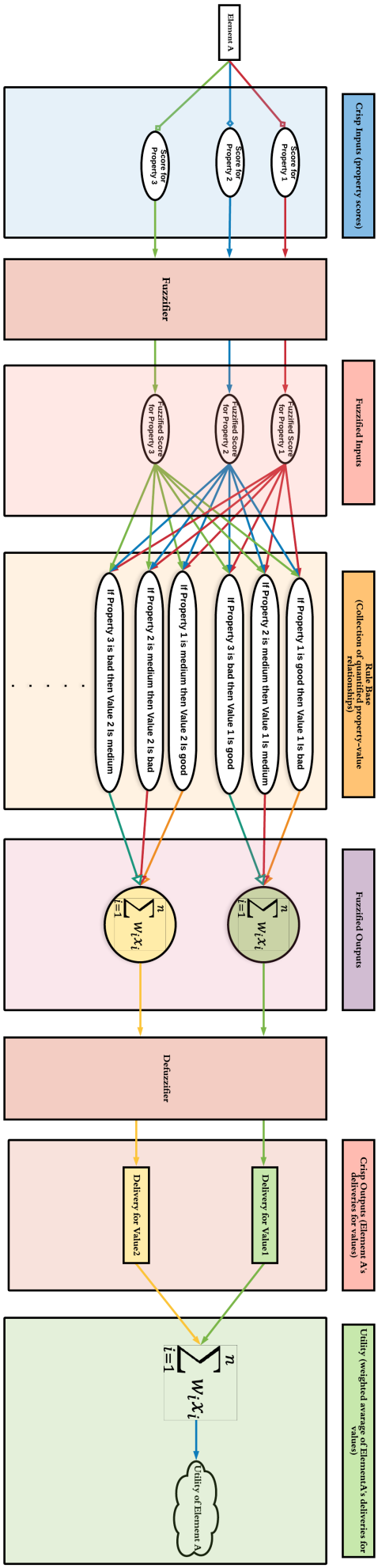

Crisp Inputs (property scores)
Element A
Score for Property 1
Score for Property 2
Score for Property 3
Fuzzifier
Fuzzified Inputs
Fuzzified Score for Property 1
Fuzzified Score for Property 2
Fuzzified Score for Property 3
Rule Base (Collection of quantified property–value relationships)
If Property 1 is good then Value 1 is bad
If Property 2 is medium then Value 1 is medium
If Property 3 is bad then Value 1 is good
If Property 1 is medium then Value 2 is good
If Property 2 is medium then Value 2 is bad
If Property 3 is bad then Value 2 is medium
. . . . . .
Fuzzified Outputs
$\sum_{i=1}^{n} w_i x_i$
$\sum_{i=1}^{n} w_i x_i$
Defuzzifier
Crisp Outputs (Element A's deliveries for values)
Delivery for Value1
Delivery for Value2
Utility (weighted average of ElementA's deliveries for values)
$\sum_{i=1}^{n} w_i x_i$
Utility of Element A

Figure 5. This graph shows the process of FIS calculating *Element A's* utility based on its property scores.

2.4 Generate Uncertain Fuzzy Logic Rules from Surveys

As mentioned in the Section 2.3, the rule base of the FIS will be composed of quantified property-value relationships that are collected from experts in step 4 of the values-driven framework. Yet, extracting the quantified property-value relationships from experts will encounter two problems. First, the environment management cases normally involve environment experts who do not have prior understanding about fuzzy rules nor FIS. Although most of the existing methods for extracting fuzzy rules such as [18] and [19] have been effective and successful, they require the target audiences to have adequate fuzzy logic knowledge. Thus, extracting fuzzy rules from non-fuzzy-logic experts can be challenging. Second, due to the heterogeneous nature of expert groups, the inevitable uncertainties introduced by different experts' various opinions should be taken into consideration (the inter-expert uncertainty [17]) as well as the uncertainties generated from the conflicting rules expressed by one expert (intra-expert uncertainty [17]). To address these two problems, one novel approach introduced by Wagner et al. 2015 [17] will be employed to collect and quantify these property-value relationships from experts.

In this approach, the rules are elicited from surveys. The experts are initially provided with a blank graph same as the one shown in *Figure 6*. The values on y-axis and x-axis of these graphs respectively represent values' and properties' different levels. Thus, each cell of these graphs represents a property-value relationship (i.e. rule). For example, for the cell highlighted in *Figure 7*, it represents the relationship: "*if food quality is level 10 then the tip is level 30*". The "levels" are the linguistic labels. They are used to describe the "degree" of properties and values. They can also be replaced with other linguistic terms such as "*good*" or "*poor*". For example, "level 1" to "level 5" of a value can be replaced with "extremely poor", "poor", "medium", "good", "extremely good". For convenience, the numeric levels will be used in this survey.

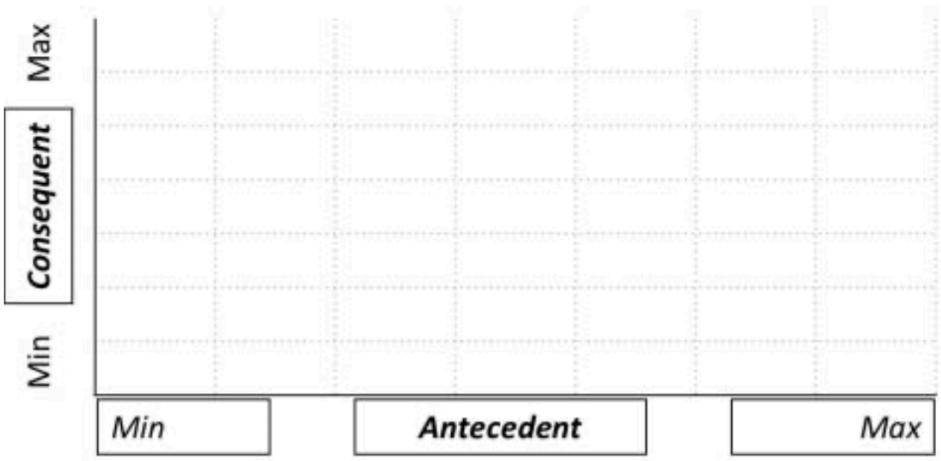


Figure 6 (cited from [17]). Example of the graphs that are given to experts during surveys.

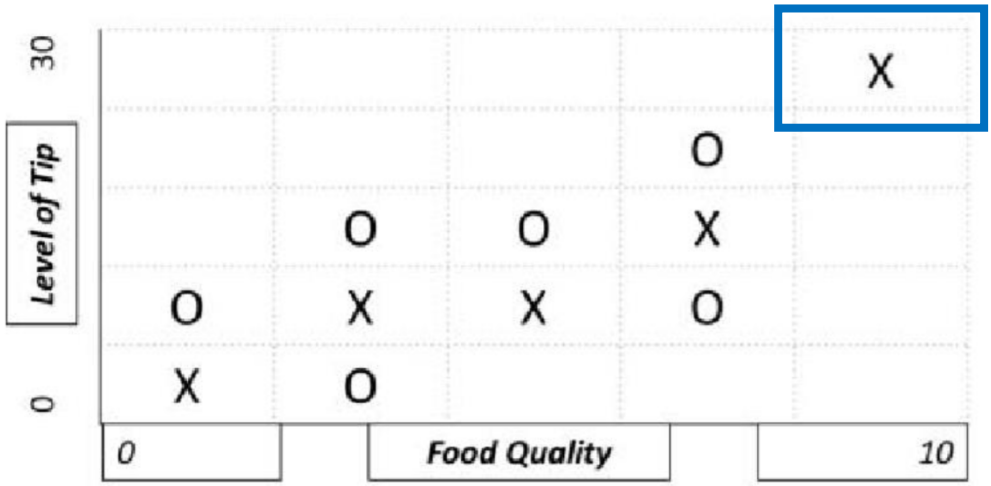


Figure 7(cited from [17]). An example of experts' responses collected during survey.

During these surveys, the experts are required to answer the question "For a given degree of X, what level of Y can be considered appropriate/ possible?" for each column. The rule of the survey is that the experts can only mark one cell each column as best estimate (i.e. the relationship that they think is most appropriate/possible) and can fill the remaining cells to represent their estimates of uncertainty (i.e. the relationships that are possible but less likely than the best estimates) under the restriction that in each column no blank cells can appear between two marked cells. The experts will use "X" to represent their best estimates and "O" to represent their estimates of uncertainty (*Figure 7*). [17]

After all the experts have finished the survey, the results are combined into two summary matrixes: M_{best} and M_{poss} to represent the Best Estimate Relationships and the Possible Relationships between properties and values (*Figure 8*). The number in each cell of the M_{best} / M_{poss} is the number of experts who agree that this cell's corresponding relationship is best estimates/estimates of uncertainty. For example, the highlighted cell in *Figure 8* contains the

number "3" which means that during this survey there are three experts agree that this relationship represented by the cell (i.e. *if the food quality is level 0 then the level of tip is level 0*) is the best estimate of the property-value relationship for the given property level.

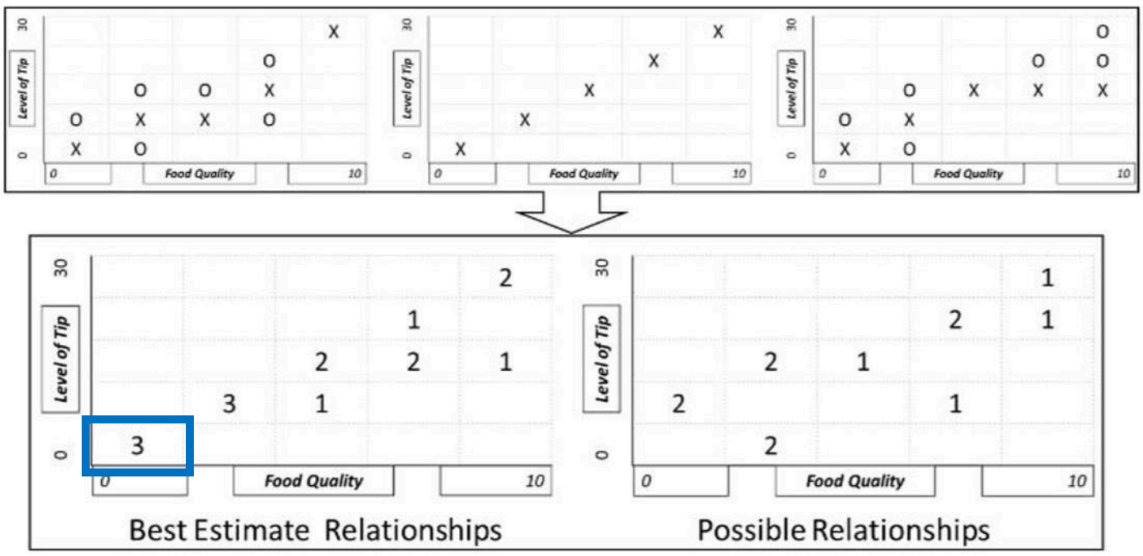


Figure 8 (cited from [17]). The process of combining the survey results collected from experts into Best Estimate Relationships and Possible Relationships

This special form of graph used in this survey conveys all the possible combinations of the properties and values (i.e. all the possible relationships/rules). Since it is meaningless for FIS to own a rule base that contains all the possible rules, different weights that will be generated from the best estimate relationships and possible relationships need to be assigned to the rules so that the FIS can only utilize the rules with non-zero weights.

To extract the rule weights, the best estimate relationships and the possible relationships (i.e. M_{best} and M_{poss}) are assigned with the weights ω_{best} and ω_{poss} ranging from 0-1 respectively. Based on these, the weight for each rule can be calculated as:

$$W_{i,j} = \frac{M_{Best_{i,j}} \times \omega_{Best} + M_{Poss_{i,j}} \times \omega_{Poss}}{\omega_{Best} + \omega_{Poss}},$$

Figure 9 (cited from [17]). The formula for calculating rule weights based on best estimate relationships and the possible relationships.

Where $W_{i,j}$ means the weight for rule $R_{i,j}$ which is the property-value relationship represented by the cell at row i column j. And the $M_{Best_{i,j}}$ and $M_{Poss_{i,j}}$ represent the numbers stored in row i and column j of M_{best} and $M_{poss.}$

One example of this is demonstrated in *Figure 10*. The M_{best} and M_{poss} here are represented by "Certain Ratings" and "Uncertain Ratings" and the weights assigned for them (i.e. ω_{best} and ω_{poss}) are 1 and 0.5 respectively. These two relationships are then combined using the equation in *Figure 9* to obtain the third summary matrix: "Merged Ratings" which is a summary matrix of weighted rules. Take the rule that is highlighted in red square as an example, its weight is calculated as $\frac{6\times1+2\times0.5}{1+0.5} \approx 4.7$ based on the two numbers highlighted in blue squares and their corresponding weights.

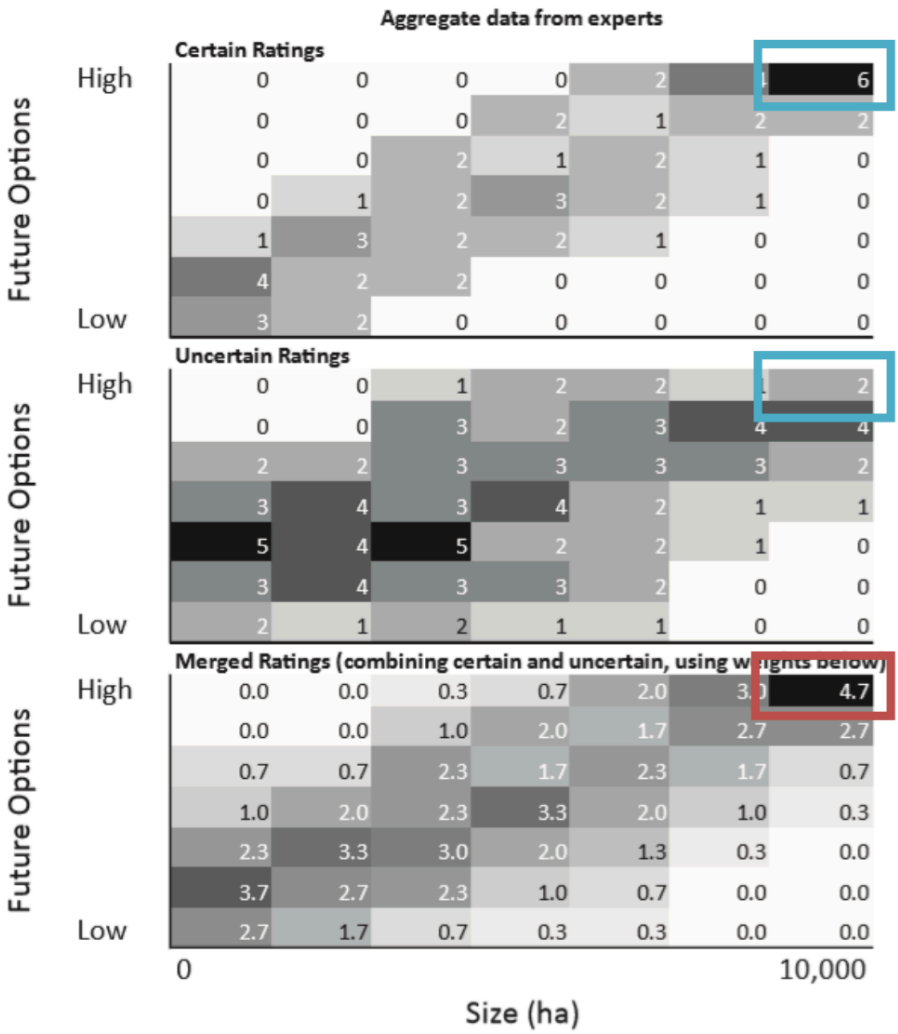


Figure 10 (cited from [6]). The best estimate relationship and possible relationship between value *Future Option* and property *Size* and the summary matrix of rule weights generated from these two relationships.

As stated in Section 2.3, part of FIS's workflow is to aggregate the outputs generated by rules to obtain the final output of the system (demonstrated in *Figure 4 and 5*). To cooperate the FIS process with the method introduced in this section, the results generated by rules will be multiplied by their corresponding rule weights before going into the aggregation (the weighted aggregation will occur in the "*Fuzzified Output*" block (i.e. the purple block) in *Figure 5*). The formalized equation is exhibited in *Figure 11.*

$$f'_{i,j} = f_{i,j} * W_{i,j} \ ,$$

where $*$ is any t-Norm, commonly multiplication.

Figure 11 (cited from [17]). The equation of combing the rule weights with the fuzzified output generated from corresponding rules.

The $f'_{i,j}$ is the result that will be aggregated to generate system output, the $f_{i,j}$ is the original result generated by the rule $R_{i,j}$ and $W_{i,j}$ is the corresponding weight of the rule elicited by using the method introduced above.

Overall, this method simplifies the fuzzy rule elicitation process by using surveys that can be comprehended and utilized by non-fuzzy-logic experts. It also introduces the concept of "rule weight" to quantify the property-value relationships as well as to capture the uncertainties generated during the process. Thus, this is an effective and feasible method for conducting the step 4 of the values-driven framework to collect and quantify rules for the FIS that will be utilized in step 5. In the software, the managers (i.e. users) can customize the software by uploading M_{best} and M_{poss} to generate the rules for their own cases.

2. 5 Interval-valued Assessment of Value Importance

In the step 3 of values-driven framework, the values will be rated based on their importance in stakeholders' views. These resulting ratings can be used to prioritize the management actions that focuses on different human values to support informing decision-making when the management resources are limited. To utilize this idea in the software, the importance ratings of values are quantified as weights that are later assigned to the corresponding values. As mentioned in the Section 2.3, the values' weights will be employed while combining the elements' contributions to values into utility.

Since the values' importance ratings need to be elicited from different groups of stakeholders, the uncertainties caused by the various information sources are inevitable. Thus, specific methods need to be employed to capture the uncertainties in the collected data for later analysis and interpretations. In this section, a feasible method for completing the value-driven framework's step 3 will be clarified.

One interval-based value importance rating approach that is conducted on a continuous scale proposed by Wagner et al. (2015) [14] is appropriate for this project due to its competence

for capturing rich information including uncertainties. The method's capability for eliciting value importance ratings will be emphasized in this section while in Section 2.6 a further application of this method will be explained.

In this method, each stakeholder is provided with a score sheet (*Figure 12*) and they are asked to rate the importance of each value by placing an ellipse on the corresponding position of the provided linear scale with width of the ellipse presenting the uncertainty about this rating (*Figure 13*). For example, on *Figure 14*, since (a) has a shorter ellipse width then (b), the response captured in (a) is more certain than the one in (b).

As indicated in [15], to accomplish the step 3 of values-driven framework, the ratings of values can be elicited from these score sheets as the mean of the maximum importance of each value (i.e. the mean of each value's collected ellipses' right endpoints.)

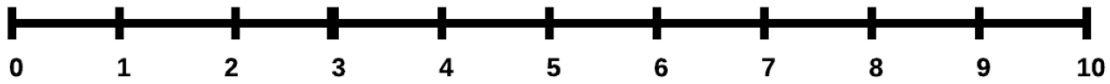


Figure 12. An example of the score sheet that is provided to stakeholders.

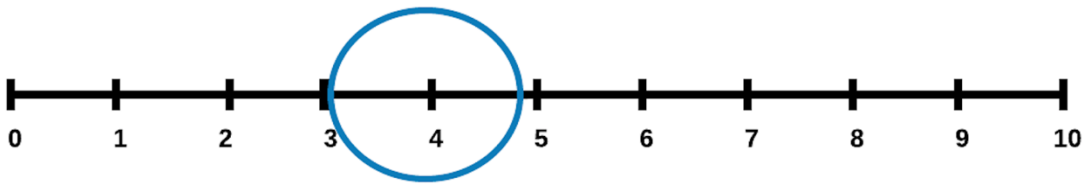


Figure 13. An example of stakeholders' response.

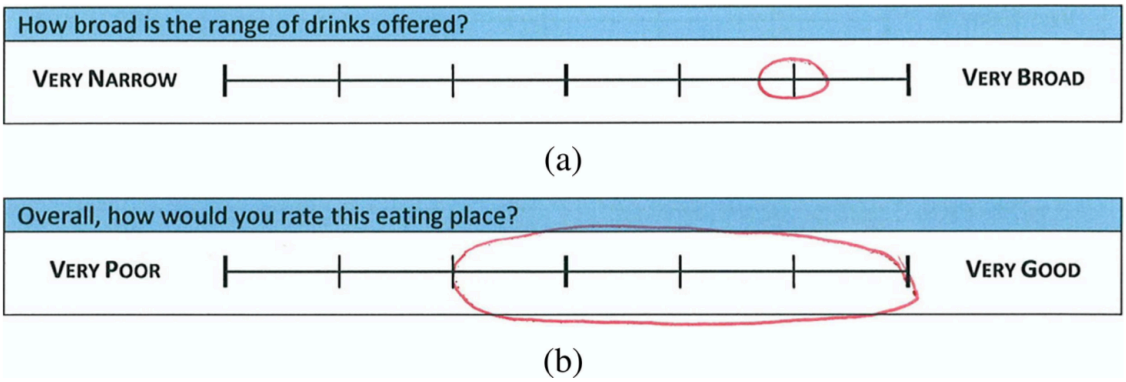


Figure 14 (cited from [14]). Responses from stakeholders for the questions "How broad the range of drinks coffee" and "Overall, how would you rate this eating place".

In this project, the users can upload the left and right end points of each ellipse from the score sheets that are collected from stakeholders to customize the software to meet the needs of their own cases.

2.6 Interval Agreement Approach (IAA)

As mentioned in Section 2.5, the interval-based assessment can provide considerable amount of information. To facilitate further information elicitation, the collected score sheets can be aggregated by values using the Interval Agreement Approach (IAA) [14]. The results obtained from applying IAA can be visualized into graphs like the one in *Figure 15*. These obtained graphs visualize rich set of information for later analysis and interpretation. They graphically demonstrate the uncertainties in stakeholders' opinions of value importance ratings and the relationship between stakeholders' agreement (y-axis) and the importance ratings of the values (x-axis). Thus, to further support the environment managers' informing decision-making process, this software will be designed and implemented to present these graphs to users. (The details will be elaborated in Section 5.5 and Section 6.6)

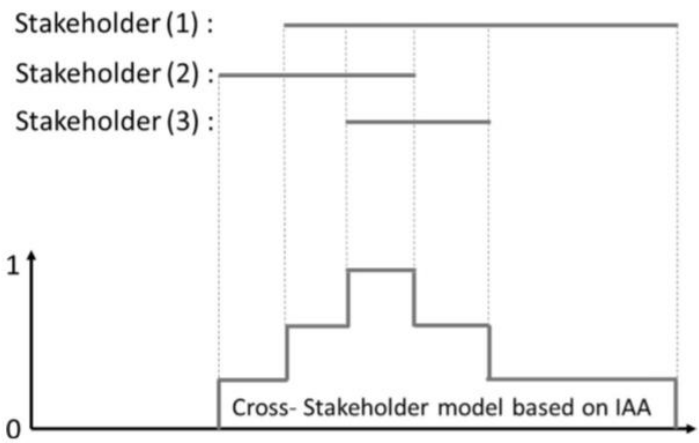


Figure 15 (Cited from [9]). An example of applying IAA method to aggregate stakeholders' responses collected from score sheets (mentioned in Section 2.5) into a graph.

Apart from human values, in this software, these two methods mentioned in Section 2.5 and 2.6 will also be used for elements' importance ratings. This information can provide a prior understanding for users/managers about element' importance in stakeholders' views which will allow for a comparison with the elements' importance elicited from the property-value relationships [15]. As mentioned in the Section 2.3, the FIS can use elements' property scores as inputs to estimate the elements' utilities based on the quantified property-value relationships collected from experts. Since the utility represents one element's contribution to human well-beings, it can also be considered as a measure for the element's importance in environment management cases. As a result, the importance of elements that is based on the utilities can be compared to the one collected from the stakeholders to provide rich information concerning the uncertainties entailed in the differences between stakeholders' and experts' opinions which will later benefit the informing decision-making process in environment management.

3 Development Methodology

In this project, the requirements (Section 4) are first specified based on the aim of the project, the innovative adaptivity expected from the software mentioned in Section 1.3 and the workflow of the values-driven framework mentioned in Section 2.2. The design and implementations are then constructed and conducted following the requirements. The functionalities of the system and its competence for assisting the value-driven framework are later assessed using a synthetic case study referred as "Atlantis Management case" (Section 7). Last, a potential user (an expert in environment management) is invited to test the software. Based on the user experience reviews, further improvements will be made.

4 Requirements

In this section, the requirements for the whole software will be clarified. They are classified into functional and non-functional requirements following the most common categorization pattern.[20]

The functional requirements will state the behaviours and the functions the software should be able to provide. The non-functional requirements will exhibit basic constrains for the software to ensure that the final product will perform well and meet the users' needs.

Both of them will be used as guidelines through the design and implementations of the software. The functionalities specified in functional requirements and the *flexibility* in non-functional requirements will later be tested using data from Atlantis Management case (section 7). The non-functional requirements *usability* and *portability* will be assessed by a

potential user (an expert in environment management). The feedback from the user will be shown in Section 8.

4.1 Functional Requirements

In order to deliver clear requirements and facilitate future design and implementations, the functional requirements will be divided into two parts focusing on frontend and backend of the software respectively. These requirements are the minimal requirements for the software.

4.1.1Frontend Requirements

1.a **The system must have an interface for users to upload the best estimate relationships and the possible relationships (mentioned in Section 2.4) collected from experts.**

2.a **The interfaces of the system must be able to visualize the quantified property-value relationships.**

3.a **The system mush have an interface for users to upload the value-rating and element-rating data (i.e. the left and right endpoints of ellipses from collected score sheets that are mentioned in Section 2.5).**

4.a **The system must be able to visualize the value-rating and element-rating results on the interfaces.**

5.a **The interfaces of the system must allow users to choose element and input its property scores.**

6.a **The utility of the chosen element based on its input property scores must be shown on interface.**

7.a **The interface must show the chosen element's deliveries for human values based on its input property scores.**

8.a **The interface must enable users to change the names of linguistic labels (mentioned in Section 2.4) and their associated membership functions of chosen element's values and properties.**

9.a **The software must provide interface for users to show the useful information such as centroid, spread, maximum agreement and average maximum importance (details in Section 5.4.1) elicited from elements/values rating data uploaded by users.**

10.a **The software must have an interface that shows users all the rules (i.e. property-value relationships) and weights associated with the chosen element.**

4.1.2 Backend Requirements

1.b **The backend of the software must be able to extract input/output variables, number of linguistic labels, rules and rule weights for constructing FIS from the best estimate relationships and the possible relationships uploaded by users.**

2.b **The backend must have default settings for properties' and values' linguistic label names and membership functions (i.e. the software must be able to function before users customize the linguistic labels and membership function based on their needs).**

3.b **The backend must support users to customize the linguistic labels and their associated membership functions of properties and values.**

4.b **The backend must be able to construct FIS based on the data uploaded by users.**

5.b **Since different elements may have different property-value relationships (i.e. rules), linguistic labels and membership functions, the backend must be able to appropriately adjust the FIS when users choose different elements.**

6.b **The backend must support the generation of the chosen element's utility based on the input property scores.**

7.b **The backend must support the generation of the chosen element's deliveries for values based on the input property scores.**

8.b **The backend must support the visualization of the quantified property-value relationships.**

9.b **The backend of the software must be able to extract elements'/ values' ratings from the elements/values rating data uploaded by users.**

10.b **The backend of the software must be able to elicit useful information such as centroid, spread, maximum agreement and average maximum importance from elements/values rating data uploaded by users.**

11.b **The backend must support the visualization of the elements'/values' ratings.**

12.b **The backend of the software must support users to view rules and rule weights of different elements**.

4.2 Non-Functional Requirements

1. **Usability:** since the target users of the software are people who work in environment management, it is reasonable to assume that most of them do not have deep understanding about computer science. Thus, the software needs to be designed and implemented to be friendly to non-computer science experts.

2. **Portability:** as mentioned above, the users of the software are likely to be non-computer science experts. Thus, the launch of the software should not be too complicated. Since different users are likely to have different settings and operating systems on their computers, the software must be able to run in most of the environments disregarding the different settings on users' computers.

3. **Flexibility:** it has been stated in the innovation section that the software needs to be able to work with users' uploaded files and it should allow users to customize its settings. Thus, this software needs to be flexible to users' customization.

5 Design

5.1 System Overview

To fulfil the functional and non-functional requirements, the software is designed to include the following components:

- **Rule Generation**
- **FIS Generation**
- **Rating Information Extraction**
- **Utility Calculation**
- **Rule Customization**
- **Data Visualization**
- **Rule Exhibition.**

In this overview section, a high-level overview of the whole system that shows the basic relationships among these components will be clarified.

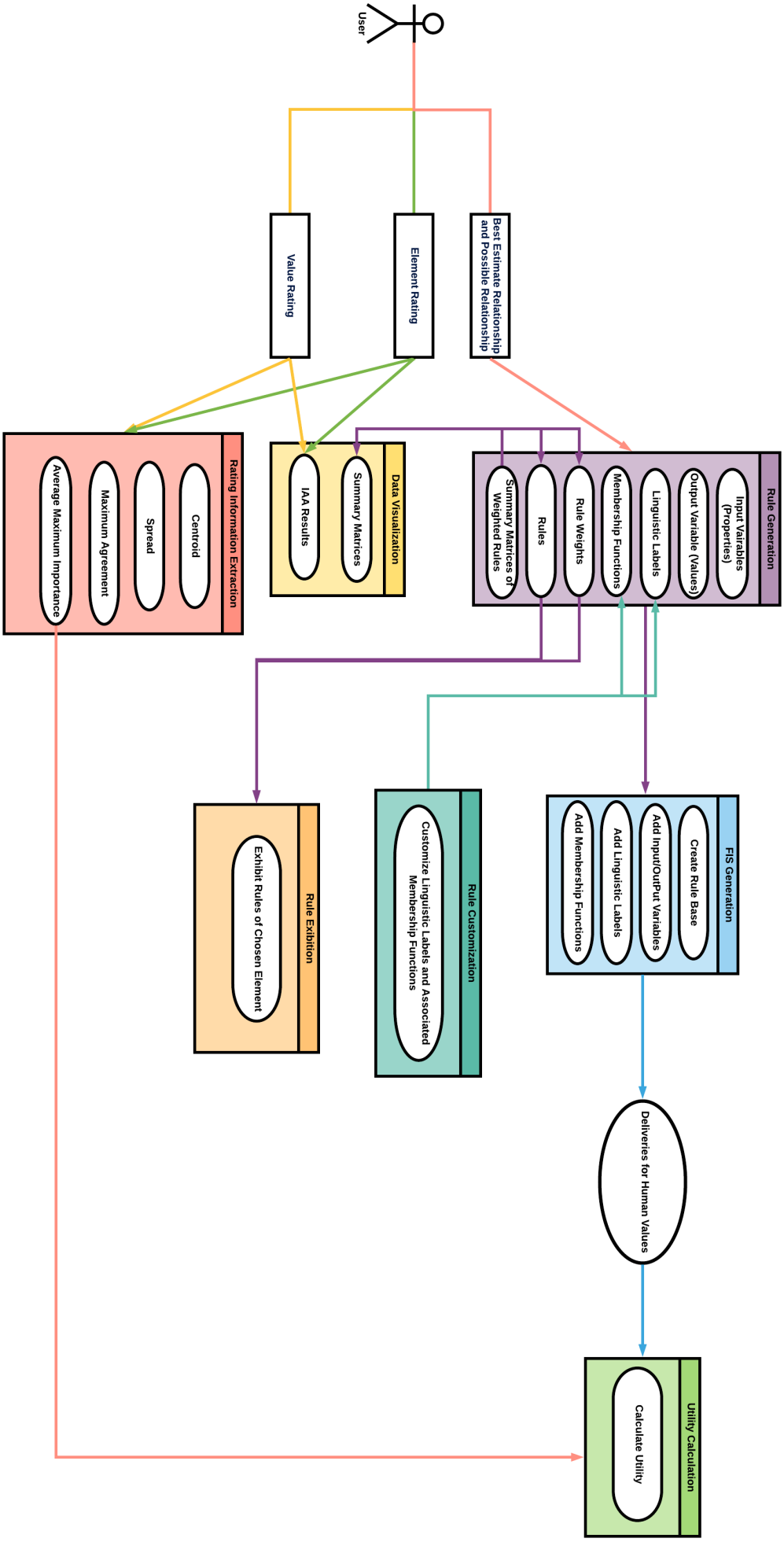

User
Best Estimate Relationship and Possible Relationship
Element Rating
Value Rating
Rule Generation
Input Vairables (Properties)
Output Variable (Values)
Linguistic Labels
Membership Functions
Rule Weights
Rules
Summary Matrices of Weighted Rules
Data Visualization
Summary Matrices
IAA Results
Rating Information Extraction
Centroid
Spread
Maximum Agreement
Average Maximum Importance
FIS Generation
Create Rule Base
Add Input/OutPut Variables
Add Linguistic Labels
Add Membership Functions
Rule Customization
Customize Linguistic Labels and Associated Membership Functions
Rule Exibition
Exhibit Rules of Chosen Element
Deliveries for Human Values
Utility Calculation
Calculate Utility

Figure 16. The relationships among software's components.

The connections among the components of the software have been graphically summarized in *Figure 16*. The followings are the detailed introductions about these linkages.

Rule Generation, Rule Exhibition, Rule Customization and FIS Generation: The *Rule Generation* part is responsible for preparing all the essential elements for building a FIS. It will elicit summary matrices of weighted rules (mentioned in Section 2.4), inputs (properties), outputs(values) and linguistic labels from the file that contains the best estimate relationships and possible relationships. The rules and rule weights needed by FIS will be extracted from the summary matrices. It will also assign default membership functions and names for each extracted linguistic label. These membership functions and their associated linguistic labels can later be modified in *Rule Customization* part. The rules and rule weights extracted in this part can be shown to users in the *Rule Exhibition* part. In the *FIS Generation* part, a FIS will be generated based on the results from *Rule Generation* and it can generate element's deliveries for all the values in the FIS.

Utility Calculation, FIS Generation and Rating Information Extraction: In the *Utility Calculation* part, the utility of the element will be calculated as a weighted average of the element's deliveries for values produced by *FIS Generation* part. The weights will be based on the average maximum importance values (more details in Section 5.4) that are obtained from Element Rating and Value Rating files in *Rating Information Extraction* part.

***Rating Information Extraction and Uploaded Files*:** In *Rating Information Extraction* part, information such as centroid, spread, maximum agreement and average maximum importance (more details in Section 5.4) will be elicited from Element Rating and Value Rating files.

Data Visualization, Rule Generation, Uploaded Files: The summary matrices of weighted rules generated in *Rule Generation* will be visualized in *Data Visualization* part. And the IAA resulting figures (mentioned in Section 2.6) generated based on the data entailed in Element and Value Rating files will be visualized.

In the following part, Section 5.2 to Section 5.6 will mainly focus on the design for the backend of the software. It will elaborate on a high-level about *what* tasks that each part is responsible for. And the frontend design will be clarified in Section 5.7.

5.2 Rule Generation

The *Rule Generation* part is designed to satisfy the requirement **1.b ,2.b and the non-functional requirement *flexibility***. Thus, it will be responsible for ensuring the software's capability for dealing with user-uploaded best estimate relationship and possible relationship files and preparing rules, rule weights, inputs (properties), outputs (values), linguistic labels and membership functions for the construction of FIS.

Value	Value_Name	Property	Property_Name	Element	Element_Name	Best_Estimate	1	2	3	4	5	6	7	Possible	1	2	3	4	5	6	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	1	0	0	0	0	0	0	9	1	0	0	0	0	1	13	4
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	2	0	0	0	0	0	13	4	2	0	0	0	3	13	0	9
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	3	0	0	0	1	13	0	0	3	1	1	6	12	0	13	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	4	0	0	0	12	0	0	0	4	3	10	13	1	13	7	3
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	5	0	0	13	0	0	0	0	5	11	13	0	13	8	1	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	6	5	13	0	0	0	0	0	6	8	0	13	9	1	0	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	7	8	0	0	0	0	0	0	7	5	13	6	2	0	0	0

Figure 17. A fragment of uploaded file that contains the element *Forest of tranquillity*'s best estimate and possible relationships between human value *Health value (arising from the physical environment)* and property *Species richness.*

Summary Matrices of Weighted Rules, Rules and Rule Weights: The user-uploaded file will contain the best estimate relationships and possible relationships of all the property-value combinations of elements. A fragment of user-uploaded file is shown in *Figure 17*. In this graph, the relationships between property *Species richness* and value *Health value (arising from physical environment)* of element *Forest of tranquillity* are demonstrated. The best estimate relationships and possible relationships are highlighted by red and blue squares respectively.

In Section 2.3 it has been demonstrated that the rules and rule weights in FIS are composed by quantified property-value relationships and their associated weights. **Thus, what *Rule Generation* part needs to do is to extract the quantified property-value relationships and their weights from these best estimate relationships and possible relationships.** To obtain the quantified property-value relationships, as explained in Section 2.4 the *Rule Generation* will first need to **combine elements' best estimate relationships and possible relationships between each property-value combination to obtain summary matrices that contains weighted property-value relationships (rules).**

An example of the resulting summary matrix generated from *Figure 17* is shown in *Figure 18.* By definition, each cell of the resulting summary matrices represents a rule (a property-value relationship) and its associated weight. Thus, ***Rule Generation* needs to extract the rules and rule weights from the cells of the resulting summary matrices.** These summary matrices will later be visualized in *Data Visualization* part.

Health value (arising from the physical environment)	Status zero			current state			Double current state
100	0.00	0.00	0.00	0.00	0.33	4.33	7.33
83.3	0.00	0.00	0.00	1.00	4.33	8.67	5.67
66.7	0.33	0.33	2.00	4.67	8.67	4.33	2.33
50	1.00	3.33	4.33	8.33	4.33	2.33	1.00
33.3	3.67	4.33	8.67	4.33	2.67	0.33	0.00
16.7	6.00	8.67	4.33	3.00	0.33	0.00	0.00
0	7.00	4.33	2.00	0.67	0.00	0.00	0.00

Species richness

Figure 18. The summary matrix that contains the weighted property-value relationships between value *Health value (arising from the physical environment)* and property *Species richness.*

Inputs, Outputs Linguistic Labels and Membership Functions: As mentioned above, to construct a FIS, the input and output variables, linguistic labels and their associated membership functions need to be defined. Since in this project, the properties are the inputs of the FIS and the values are the outputs**, the *Rule Generation* part needs to elicit the properties, values and their associated linguistic labels from the user-uploaded files to support the construction of FIS**. However, the user-uploaded file only contains the number of linguistic labels. Thus, as stated in requirement **2.b**, **the *Rule Generation* part also needs to assign default names and membership functions for the linguistic labels.** They can later be customized to fit users' needs in the *Rule Customization* part.

Store Linguistic Labels, Membership Functions Rules and Rule Weights by Element: It should be noticed that all linguistic labels and membership functions are associated with their corresponding properties and values. Normally, in one environment management case, **different elements will have the same properties and values associated with them but, due to the different natures of elements, they are likely to have different linguistic labels and membership functions for these properties and values.**

Notice that the different linguistic labels here stand for different linguistic label names since normally, in one environmental case, different elements' properties and values will have the same number of linguistic labels.

For example, for two elements: *Forest of tranquillity* and *Lake danger reeds*, they all have the same property: *Species richness* associated with them, but they have assigned different

linguistic label names and membership functions for the property. The linguistic labels for *Species richness* of *Forest tranquillity* are named “level 1” to “level 5” with gaussian functions as membership functions for each label while the linguistic labels for *Species richness* of *Lake danger reeds* can be named “extremely poor”, “poor”, “medium”, “good ” and “extremely good” and using trapezoid functions as membership functions for each label.

Although, in *Rule Generation* part, the linguistic labels’ names and membership functions of properties and values of different elements will initially be assigned with the same default names and membership functions, the users will change them later to fit the scenarios of their own cases.

Also, due to the distinction among elements, the relationships between properties and values will be different for each element. In other words, different elements will have different rules and rule weights associated with them.

Thus, **it is reasonable for *Rule Generation* to store different elements’ linguistic labels, membership functions, rules and rule weights separately**.

Based on this information**, the inputs, outputs, linguistic labels, membership functions, rules and rule weights generated in *Rule Generation* part can be stored by element in three modules: *Inputs, Outputs and Rules*.** For each element, the Inputs/Outputs module will entail properties/values and their associated linguistic labels and membership functions for this element. And the Rules module will store all the rules and rule weights of the corresponding element. This is summarized in *Figure 19*. By doing so, all the information extracted in this part can be nicely combined.

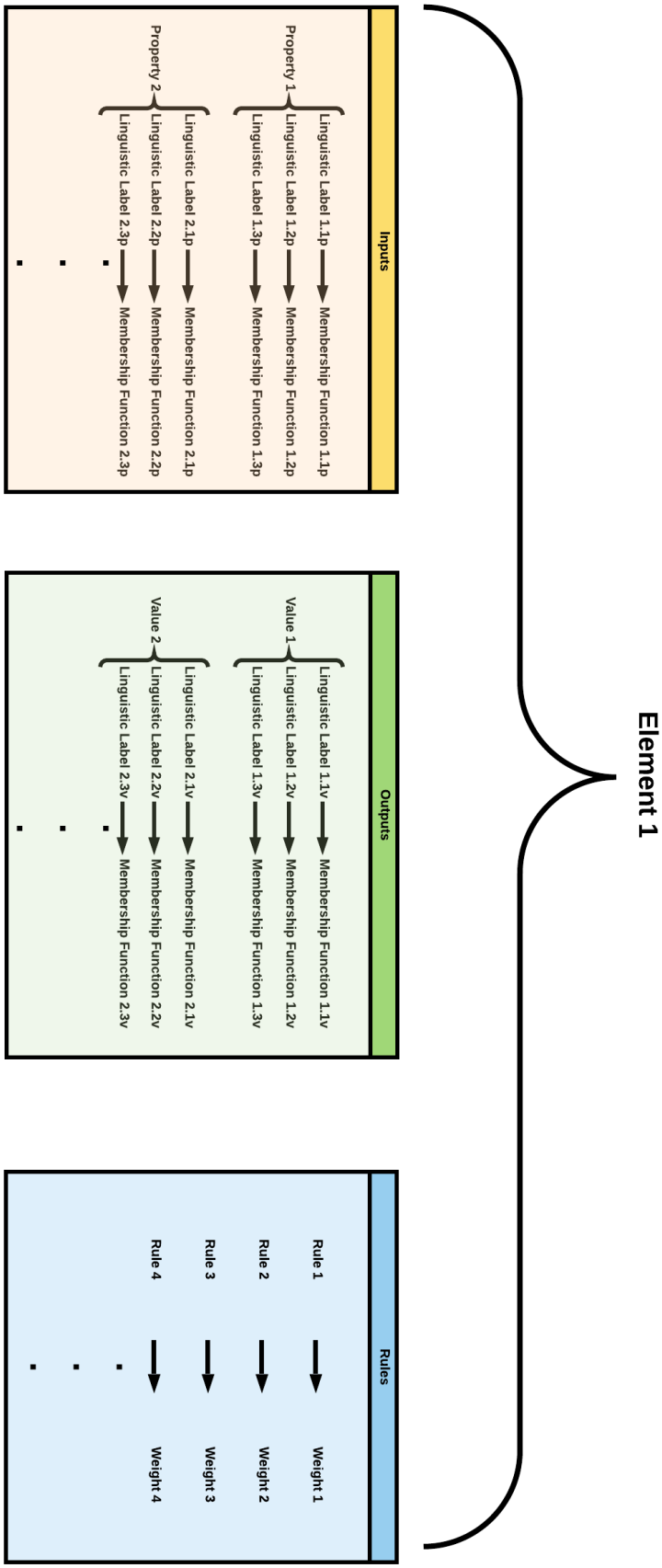

Element 1
Inputs
Property 1
Linguistic Label 1.1p
Membership Function 1.1p
Linguistic Label 1.2p
Membership Function 1.2p
Linguistic Label 1.3p
Membership Function 1.3p
Property 2
Linguistic Label 2.1p
Membership Function 2.1p
Linguistic Label 2.2p
Membership Function 2.2p
Linguistic Label 2.3p
Membership Function 2.3p
Outputs
Value 1
Linguistic Label 1.1v
Membership Function 1.1v
Linguistic Label 1.2v
Membership Function 1.2v
Linguistic Label 1.3v
Membership Function 1.3v
Value 2
Linguistic Label 2.1v
Membership Function 2.1v
Linguistic Label 2.2v
Membership Function 2.2v
Linguistic Label 2.3v
Membership Function 2.3v
Rules
Rule 1
Weight 1
Rule 2
Weight 2
Rule 3
Weight 3
Rule 4
Weight 4

Figure 19. The Inputs, Outputs and Rules modules of Element 1. The Inputs/Outputs module stores this element's associated properties/values and their corresponding linguistic labels and membership functions. The Rules module stores this element's rules and rule weights.

It should be noticed that a FIS can only calculate one element's utility which means **it can only store the properties, values, linguistic labels, membership functions, rules and rule weights of one element**. Thus, when users change the chosen element, the element and its associated information stored in FIS should also be changed. To facilitate the element change process, **the *Rule Generation Part* will be responsible for providing the properties, values, linguistic labels, membership functions, rules and rule weights to *FIS Generation* part based on users' choice of element.** This information modulization design introduced in this section can facilitate the construction of FIS for different elements (details in Section 5.3).

5.3 FIS Generation

This part is responsible for constructing a FIS to calculate the chosen element's deliveries for human values. It should satisfy the requirements **4.b, 5.b** and **7.b**.

FIS Construction: To construct a FIS, input variables, outputs variables, linguistic labels, membership functions, rules and rule weights are all essential factors. These factors will be provided by *Rule Generation* part. What this part needs to do is to utilize these factors to construct a FIS.

As mentioned above, one FIS can only store the information of one element. As a result, when users want to see the utility of different elements, the properties, values, linguistic labels, membership functions, rules and rule weights stored in FIS should change accordingly. Thus, the construction of the FIS should enable it to be capable of **agilely adapting to different elements and changes**. To achieve this, the FIS should be "**modularized**". In other words, the essential factors of FIS should be constructed as different modules that are relatively independent from each other. In this way, when one of these factors changes (e.g. when users customize the linguistic labels or membership functions), it will not have significant effect on the whole FIS. Also, when the users change the chosen element, the FIS can just change the modules that store the old chosen element without worrying other parts of the system.

As explained in Section 5.2, the extracted essential factors in *Rule Generation* part are stored in three modules: ***Inputs, Outputs and Rules* by element**. **In order to work with the *Rule Generation* part and facilitate the element change, the FIS should be constructed using these three modules to store the factors that will be changed with users' different element choices**. By doing so, the *FIS Generation* part can simply swap corresponding modules when users change the element. The module-swapping that will occur when users change the chosen element from Element 2 to Element 1 is summarized *Figure 20.* The *Rule Generation* part will provide the Inputs, Outputs and Rules modules of Element 1 to swap the corresponding modules of Element 2 that are stored in the FIS.

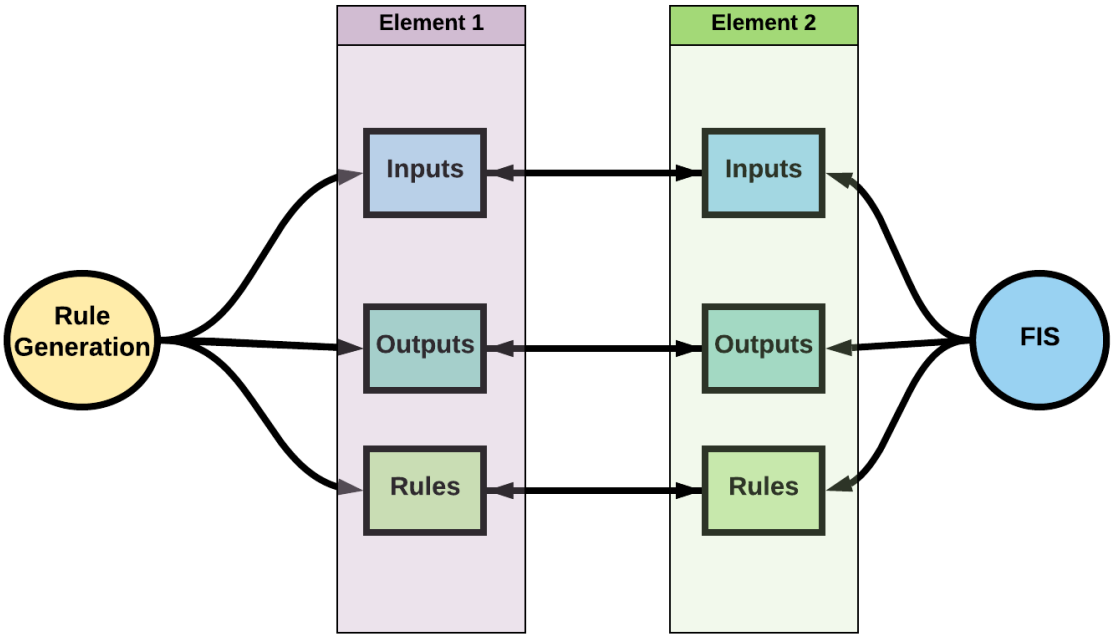


Figure 20. When the user changes the chosen element from Element 2 to Element 1, the three modules (Inputs, Outputs and Rules) of Element 1 should swap the corresponding modules in FIS that store the information about Element 2.

Generate Element's Deliveries for Values: Another key aspect of the *FIS generation* is to calculate the chosen element's deliveries for human values. It should **utilize the FIS constructed in this part to output the deliveries using chosen element's property scores as inputs**. These deliveries will be passed to *Utility Calculation* part to produce the utility of chosen element.

5.4 Rating Information Extraction and Utility Calculation

5.4.1 Rating Information Extraction

The Element Rating and Value Rating files contain each stakeholder's opinions about the importance ratings of all the elements and values. These opinions are collected using the methods introduced in Section 2.5. Thus, these files contain the left and right endpoints of the ellipses collected from score sheets mentioned in Section 2.5. Each pair of left and right endpoints represents one stakeholder's opinion about one element's/value's lowest possible rating and highest possible rating.

A fragment of the Value Rating file is shown in *Figure 21*. This figure shows 13 different experts' opinions about the possible ratings of value *Provision of adequate resources* on a

scale from 0 to 100. The *Low* column (highlighted in blue square) stores the left endpoints of the collected ellipses and the *High* column (highlighted in red square) stores the right endpoints.

Stakeholder	Value	Value_No	Low	High	Stakeholder_Des
1	Provision of adequate resources	1	20	100	Atlantis Bureau of Education
2	Provision of adequate resources	1	30	60	Atlantis Bureau of Recreation
3	Provision of adequate resources	1	90	100	Atlantis Indigenous Owners Council
4	Provision of adequate resources	1	75	95	Atlantis Landscape Strategy
5	Provision of adequate resources	1	10	30	Director Archaeology
6	Provision of adequate resources	1	10	100	Health Authority
7	Provision of adequate resources	1	25	60	Inland Waterways Association and Environment Agency
8	Provision of adequate resources	1	60	88	Lake Danger cross-country charioteering association
9	Provision of adequate resources	1	23	49	Lake Danger Partnership
10	Provision of adequate resources	1	5	15	Lake Danger Yachting Association
11	Provision of adequate resources	1	95	100	Port of Trouble Town Authority
12	Provision of adequate resources	1	79	98	Ramblers' Greater Trouble Town Forum
13	Provision of adequate resources	1	50	100	Water Authority

Figure 21. A fragment of uploaded Value Rating file that contains 13 stakeholders' opinions about the possible importance ratings of value *Provision of adequate resources.*

As mentioned in Section 2.5, since Element Rating and Value Rating files are all collected using Interval-valued assessment of Value Importance method, they can provide wealth information that is potentially useful for environment management. Thus, in *Rating Information Extraction* part, four key values: centroid, spread, maximum agreement and average maximum importance should be elicited from the uploaded files to support the environment managers' informing decision-making process [15] as well as to satisfy the requirement **10.b**. Also, since this part involves working with user-uploaded Element/Value Rating files, it will need to satisfy the **non-functional requirement *flexibility*** and ensure the software's competence for dealing with different uploaded files.

***Centroid and Spread*:** As mentioned above, **The *Rating Information Extraction* part needs to elicit the centroid (i.e. central tendency [15]) and spread values of elements'/ values' possible ratings from the user-uploaded Element Rating and Value Rating files**. In this project, the centroid and spread will be calculated as arithmetic mean and standard deviation of the elements'/values' possible ratings elicited from Element and Value Rating files uploaded by Users.

Maximum Agreement and Average Maximum Importance: The maximum agreement elicited from the files are values that can be used to present the uncertainties and agreement levels in stakeholders' opinions. And the average maximum importance values will be extracted from the possible ratings contained in the uploaded files as the importance ratings for the corresponding elements/values to satisfy the requirement **9.b**. In other words, these values will be the final importance ratings assigned to the elements/values based on the possible ratings collected from stakeholders.

The maximum agreement of an element/value will be elicited as the largest number of stakeholders that agree on a rating. For example, among 4 stakeholders, if their opinions about the rating for a certain element is 50,50,51,52 respectively, then the rating 50 has been considered as possible by most of the stakeholders. In other words, of all the possible ratings, this rating has the maximum agreement which is the number of stakeholders that supports this rating. Thus, the maximum agreement of this element is 2.

As for the average maximum importance of an element/value, **the *Rating Information Extraction* part will calculate it as the arithmetic mean of the highest possible ratings of this element/value collected from stakeholders.** For example, *Figure 21* shows the possible ratings for the value *Provision of adequate resources* and the *high* column contains the highest possible ratings of this value in stakeholders' views. Thus, the average maximum importance of the value will be calculated as an arithmetic mean of all the values in *high* column which will be viewed as the final importance rating for this value based on the stakeholders' opinions.

5.4.2 Utility Calculation

As stated in the Section 5.1, the utility of an element is the weighted average of the element's deliveries for human values generated from *FIS Generation* part. The weights, according to [15], depend on the human values' average maximum importance elicited in *Rating Information Extraction* part. **Thus, the task for *Utility Calculation* part is to use the average maximum importance as weights to calculate utility of elements**. This part is responsible for fulfilling the requirement **6.b**.

5.5 Data Visualization

It has been emphasized in the functional requirement that this software should be able to visualize the quantified property-value relationships and the ratings of elements and values. As a result, the *Data Visualization* part of the software will use summary matrices of weighted rules obtained in *Rule Generation* part to visualize the property-value relationships. And it will apply IAA method mentioned in Section 2.6 to the data contained in Element and Value Rating files to visualize the possible ratings of values and elements. This part should satisfy the requirements **8.b** and **11.b**.

Visualization of Summary Matrices: As introduced in Section 2.4, each cell of summary matrices can represent a rule (property-value relationship) and its associated weight. As a result, these matrices can be effective and concise ways to visualize the property-value relationships. **Thus, in Data Visualization part the summary matrices generated in in *Rule Generation* part will be visualized**.

In order to further improve the interpretability of the summary matrices and to enable the users to view the property-value relationships more directly, **the summary matrices will be visualized using a heat-map style**. In other words, each cell of the summary matrices will be coloured and the cells with higher values will have darker shades. An example for this is represented in *Figure 18.*

Visualization of IAA Results: Since the Element and Value Rating files contain the highest and lowest possible rating for the elements and values, the ranges of possible ratings agreed by different stakeholders can be elicited from the uploaded files. **In the Data Visualization part these ranges will be aggregated using the IAA method introduced in Section 2.6 to obtain the visualized possible ratings of elements and values**. A range aggregation example is shown in *Figure 15*. In this figure, three possible rating ranges of an element/value collected from three stakeholders are combined together to obtain the resulting graph. The values on the x-axis of the graph represent the possible ratings for the element/value and their corresponding values on y-axis manifest the stakeholders' agreement level of these ratings. In other words, for a rating on x-axis, it will have a higher corresponding value on y-axis if more stakeholders agree that this rating is possible for this element/value. Thus, this graph is an elegant way to visualized information about importance ratings of elements and values as well as their associated agreement levels among stakeholders.

5.6 Rule Customization and Rule Exhibition

5.6.1 Rule Customization

To satisfy the requirement **3.b and the non-functional requirement *flexibility*, the *Rule Customization* part should enable users to customize the linguistic labels and membership functions**.

It has been explained in Section 5.2 that the linguistic labels and membership functions of properties and values have been stored in corresponding elements' Inputs and Outputs modules. **Thus, what the *Rule Customization* part needs to do is to utilize the information provided by users to locate the Inputs or Outputs modules of the user-chosen element and find the stored linguistic labels or membership functions that will be changed by users.**

For example, if the user wants to change the name of the linguistic label *level 7* of element *Forest of tranquillity*'s property *Species richness* then this *Rule Customization* part needs to first locate the element *Forest of tranquillity* stored in the software and go into this element's Inputs module to find the property *Species richness* and its linguistic label *level 7*. Last, this part can apply changes to this linguistic label based on users' order.

5.6.2 Rule Exhibition

The *Rule Exhibition* part will be designed to fulfil the requirement **12.b**.

As mentioned in Section 5.2, since each element is different in nature, the rules and rule weights extracted from the uploaded files will be stored in corresponding elements' Rules Module. Based on this storing structure, ***Rule Exhibition* part can show different elements' rules by presenting the contents of their Rules modules**. An example is shown in *Figure 22*.

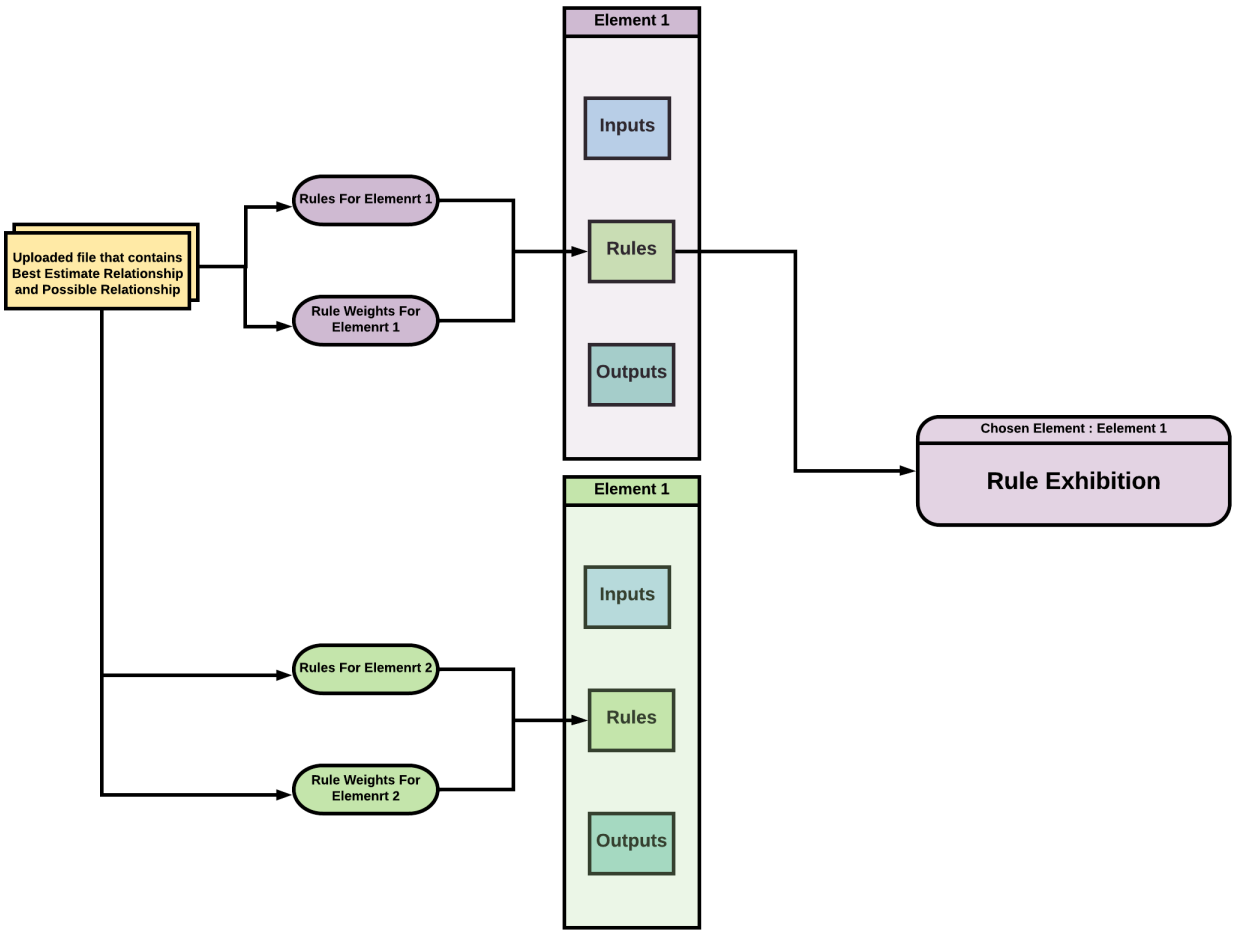


Figure 22. In this figure, the rules and rule weights of Element 1 and Element 2 elicited from the uploaded file are stored into the two elements' corresponding Rules modules. Then when users choose to show the rules and rule weights associated with Element 1, the Rules module of Element 1 will be passed to *Rule Exhibition* part to show the contents of this module (i.e. the rules and rule weights of Element 1).

5.7 Frontend Design

In order to fulfil the frontend requirements mentioned in Section 4.1.1, the interfaces of the software will be composed by the following four pages:

- **Set Up Page**
- **Utility Page**
- **Customization Page**
- **Rule Page**

Apart from the functional requirements, the front-end design also needs to consider the *Usability* of non-functional requirements. Thus, the interfaces should be constructed to be easily understood and used by non-computer science experts.

The detailed design of above interfaces will be elaborated in the following sections.

5.7.1 Set Up Page Design

The design of this page should satisfy the requirements **1.a, 2.a, 3.a, 4.a** and **9.a**.

To conclude these requirements, in the Set Up Page, the users should able to set up this software by uploading Best Estimate Relationship and Possible Relationship file, Element Rating file and Value Rating file. This page should also show the visualized summary matrices and IAA resulting graphs generated from uploaded files. It should also show the centroid, spread, maximum agreement and average maximum importance elicited from the Element and Value Rating files.

Design: A mock-up interface design for this Set Up Page is shown below in *Figure 23* and *Figure 24*.

Figure 23 shows the interface for uploading files that contain best estimate relationships and possible relationships. There are two options that the users can choose (highlighted in red square). If the users choose the File option, then the blue shaded area will present users the uploaded file. If choosing the graph option, then the summary matrices extracted from the uploaded file will be exhibited in the blue area.

The users can switch to interfaces for uploading the Element and Value Rating files by using the panels on the left side. (highlighted in green square)

The interfaces for uploading Element and Value Rating files have been shown in *Figure 24*. These two interfaces have three options (highlighted in red square) that users can choose. The File option is also for presenting the uploaded files. And the Graph option shows the IAA resulting graphs extracted from the uploaded files. As for the Values option, it presents users the centroid, spread, maximum agreement and average maximum importance for each element/value elicited from the files.

As explained before, the average maximum importance will be used as the final importance ratings of these elements and values. To manifest these ratings on the interface, for the Graph and Values option, the order of presenting IAA results and the key values will be based on the importance ratings of their associated elements/values. In other words, the first IAA graph and key values on the interface will belong to the element/value with the highest importance rating.

If the users want to navigate to other pages, they can use the tab on the top of the interface (highlighted in purple square.)

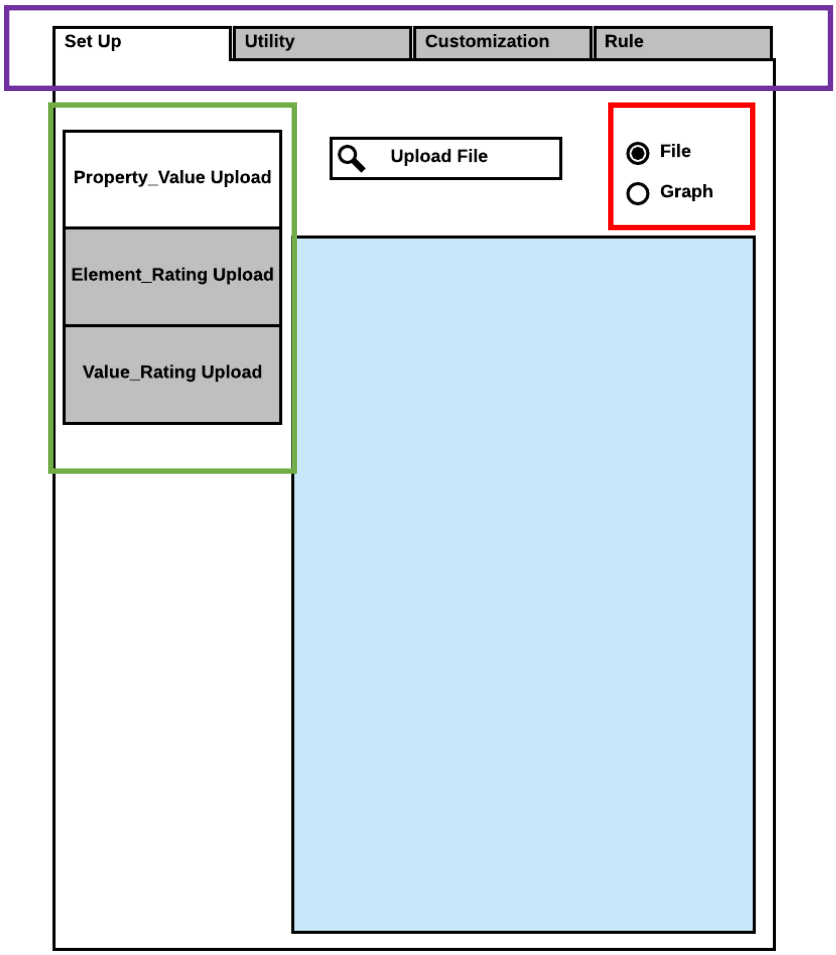


Figure 23. The Set Up Page interface design for uploading file that contains best estimate and possible relationships.

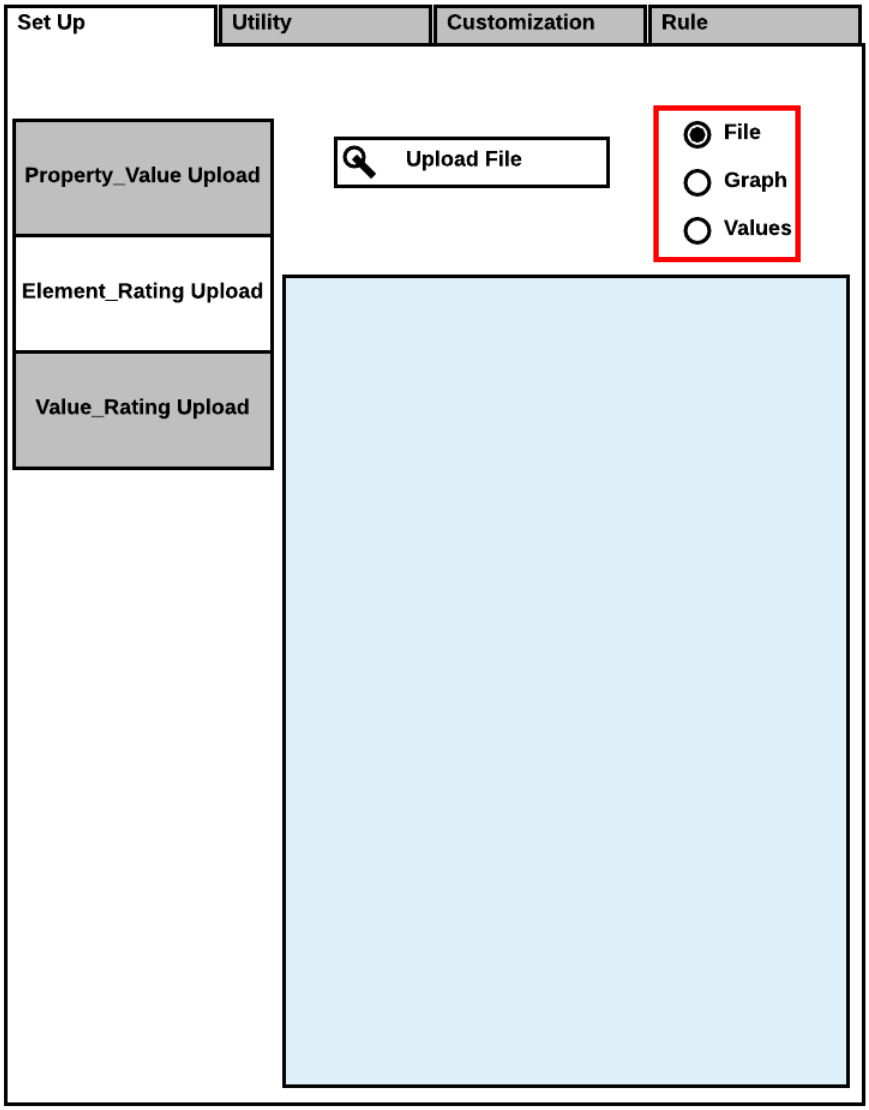


Figure 24. The Set Up Page interface design for uploading file that contains elements/values ratings.

5.7.2 Utility Page Design

As stated in requirements **5.a, 6.a** and **7.a**, this Utility Page needs to allow users to choose elements and input property scores. It also should present the chosen element's utility and deliveries for human values based on the input property scores.

Design: The design for this interface is shown in *Figure 25*. As presented in the figure, the users can use the drop-down menu (highlighted in green square) to choose elements and use the slide bars to input property scores. Each property will have a corresponding slide bar. The chosen element's deliveries for human values and utility will be shown in the blue shaded area on the right hand.

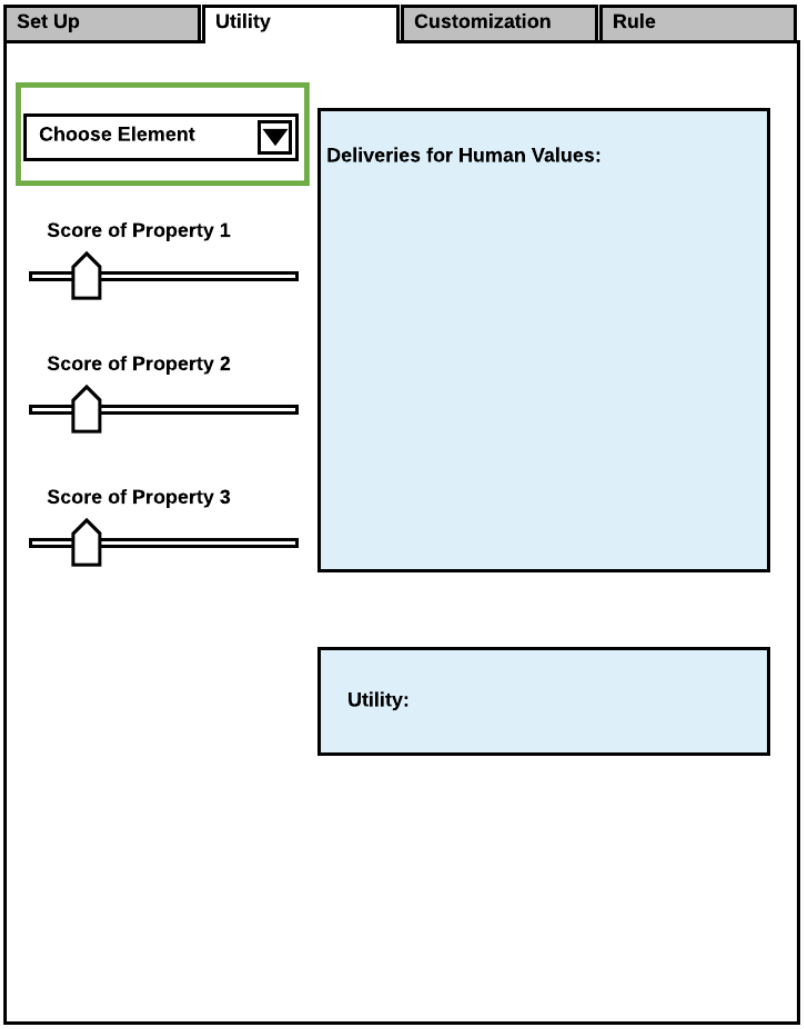


Figure 25. The interface design for Utility Page.

5.7.3 Customization Page Design

For the Customization Page, the interface should be able to allow users to customize elements' linguistic labels' names and membership functions. It should fulfil the requirement **8.a**.

Design: The design for this interface is shown in *Figure 26*. The users can choose if they want to change the linguistic labels' names and membership functions of properties or values using the radio button highlighted in the yellow square. After users choose a radio button, the "Choose Property/Value" drop-down menu will adjust its choices accordingly. The users then can choose element, properties/values, linguistic labels and membership functions using the drop-down menu on the left. The users can also choose the parameters for the chosen membership function. Since different types of membership functions will require different numbers of parameters, the number of numeric input fields for inputting parameter (highlighted in green square) will be adjusted based on the different membership functions chosen by users. Also, the name of the chosen linguistic label can be changed in the text input field highlighted in red square. The blue shaded area will show the linguistic labels and membership functions based on the chosen element and property/value.

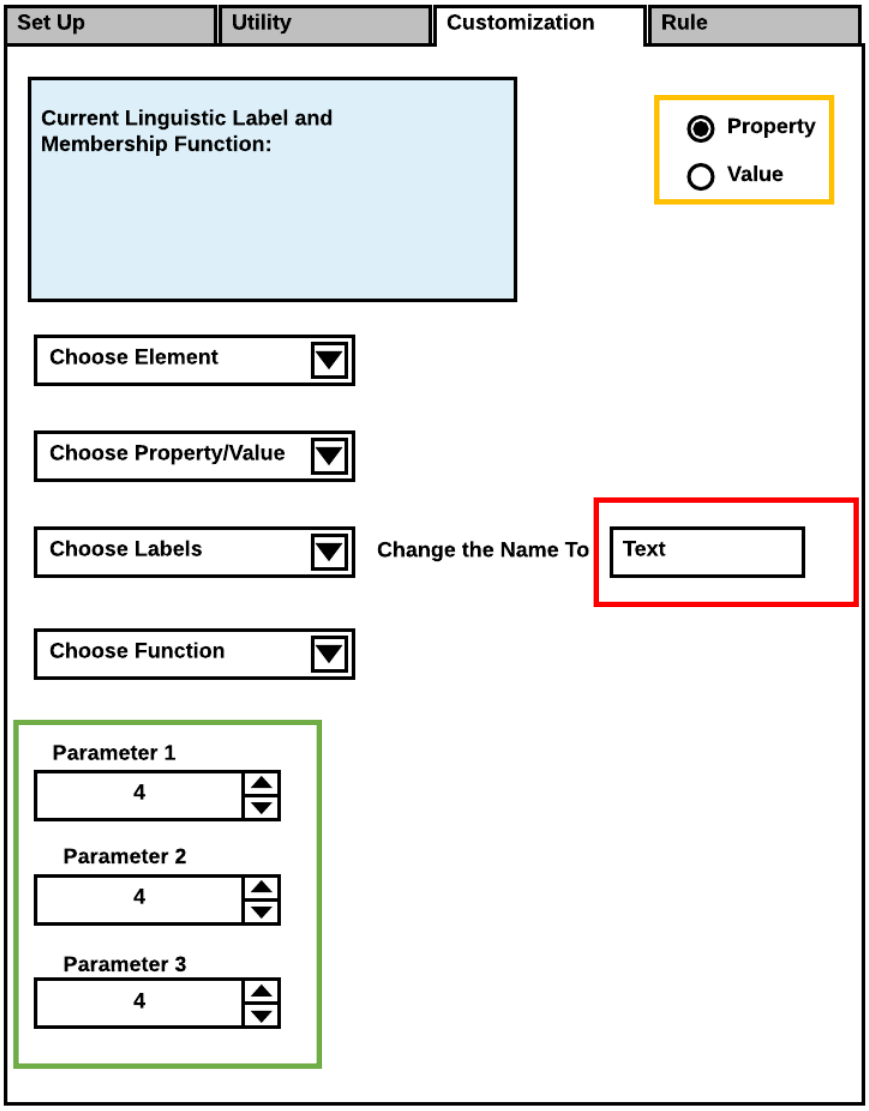


Figure 26. The interface design for Customization Page.

5.7.4 Rule Page Design

As stated in requirement **10.a**, this interface should show the rules and rule weights of the chosen element.

Design: The design for this interface has been shown in *Figure 27.* The users can choose elements in the drop-down menu. The blue-shaded area will show the rules and rule weights of the chosen element.

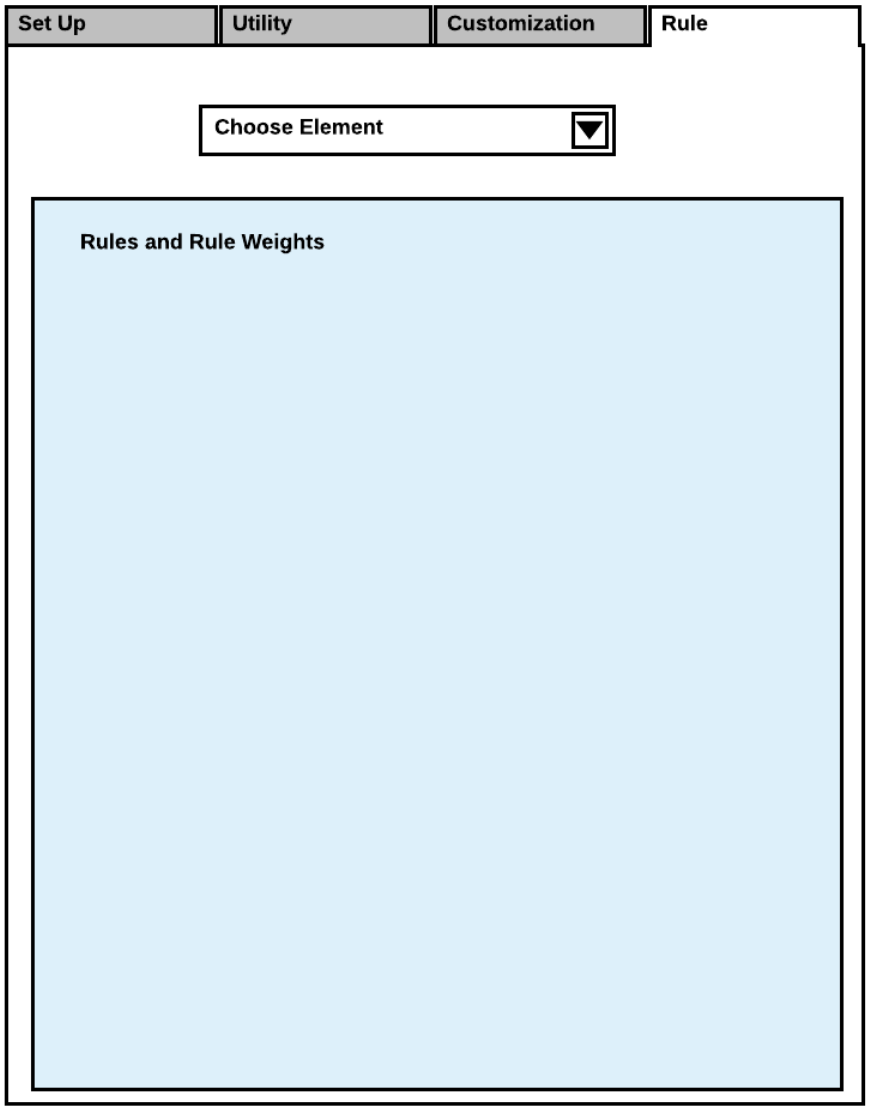


Figure 27. The interface design for Rule Page.

6 Implementation

6.1 Language

In this project, the language R will be employed to implement the software. It has two fully functional packages (ShinyR and FuzzyR) that can be used for the frontend and the backend of the system respectively. These two packages can largely facilitate the fulfilment of requirements and realization of the above design.

ShinyR: To fulfil the *Portability* non-functional requirement, this software should be easily launched and utilized by non-computer science experts disregarding the different software and hardware settings on their computers. One approach to achieve this is to implement the software as a web-based application since these applications will be able to launch and run on a web server instead of locally on users' personal computers, [21] the various local settings of users' computers do not interfere with the normal functioning of this software which renders it highly portable. To implement this idea, the ShinyR package will be employed to implement the frontend of this software due to its competence for constructing user-interactive web applications [22].

FuzzyR: It has been mentioned in previous sections that a FIS needs to be constructed at the backend of the software to support its functions. As stated in Section 5.3, in order to agilely work with different elements, the FIS should be constructed by three main modules: Inputs, Outputs and Rules. To implement this FIS, a fuzzy logic package *FuzzyR* will be employed. This package contains a set of comprehensive functions for implementing and operating FIS. More importantly, by using this package, the FIS can be implemented as an object which contains three main lists that correspond to the Input, Output and Rules modules mentioned above. Thus, using this package can largely realize the FIS design and facilitate the implementation of the software.

6.2 Implementation of Rule Generation

As mentioned in the design section, the *Rule Generation* part is responsible for extracting rules, rule weights, properties, values, linguistic labels from the uploaded Best Estimate and Possible relationship files and assigning default names and membership functions to the extracted linguistic labels. It stores these factors by element and provides the user-chosen element's factors to *FIS Generation* part for constructing FIS.

FIS Object: Since this *Rule Generation* is responsible for providing essential factors for constructing FIS in *FIS Generation* part, the structure of the FIS object generated using FuzzyR package needs to be scrutinized.

The FIS object in FuzzyR packages is implemented as a list which contains one element's input/output variables, linguistic label, membership functions, rules and rule weights. This list has three main sub-lists which correspond to the Input, Output, Rules modules mentioned in the design part. In the Input/Output sub-list, each input (property)/ output(value) of the element is stored as a list element. And these list elements are also sub-lists that are composed by three components: *name* (the name of the property/value), *range* (the range of the property/value) and *mf* (membership functions for the property's/value's linguistic labels). Among these three components, the *mf* is another list of sub-lists and each of these sub-lists contains one linguistic label's name and membership function. Specifically, each sub-list of *mf* has three components: *name* (the name of the linguistic label), *type* (the type of the linguistic label's membership function) and *params* (the parameter of the membership function). As for the Rules sub-list of the FIS object, it contains all the rules and rule weights of this element. The structures of a FIS object's Input and Rules sub-lists are presented in *Figure 28*.

In the FuzzyR package, a FIS object can be easily constructed using the built-in function *newfis()* [23] . The FIS object returned by this function is an "empty" FIS with empty Input, Output and Rules lists. To construct the Input and Output lists in FIS object (*Figure 28*), function *addvar()* and function *addmf()* can be utilized. The built-in function *addvar()* can add properties/values to the Input and Output lists while the *addmf()* can add the properties'/values' membership functions and linguistic labels. In other words, **for each Input/Output list element in FIS object (e.g. the input1 in *Figure 28*), the *addvar()* is responsible for constructing the *name* and *range* sub-lists (the yellow and purple blocks) while the *addmf()* constructs the whole *mf* sub-list.** As for constructing the Rules list of FIS object, the built-in function ***addrule()*** can be utilized to add rules and rule weights to the list.

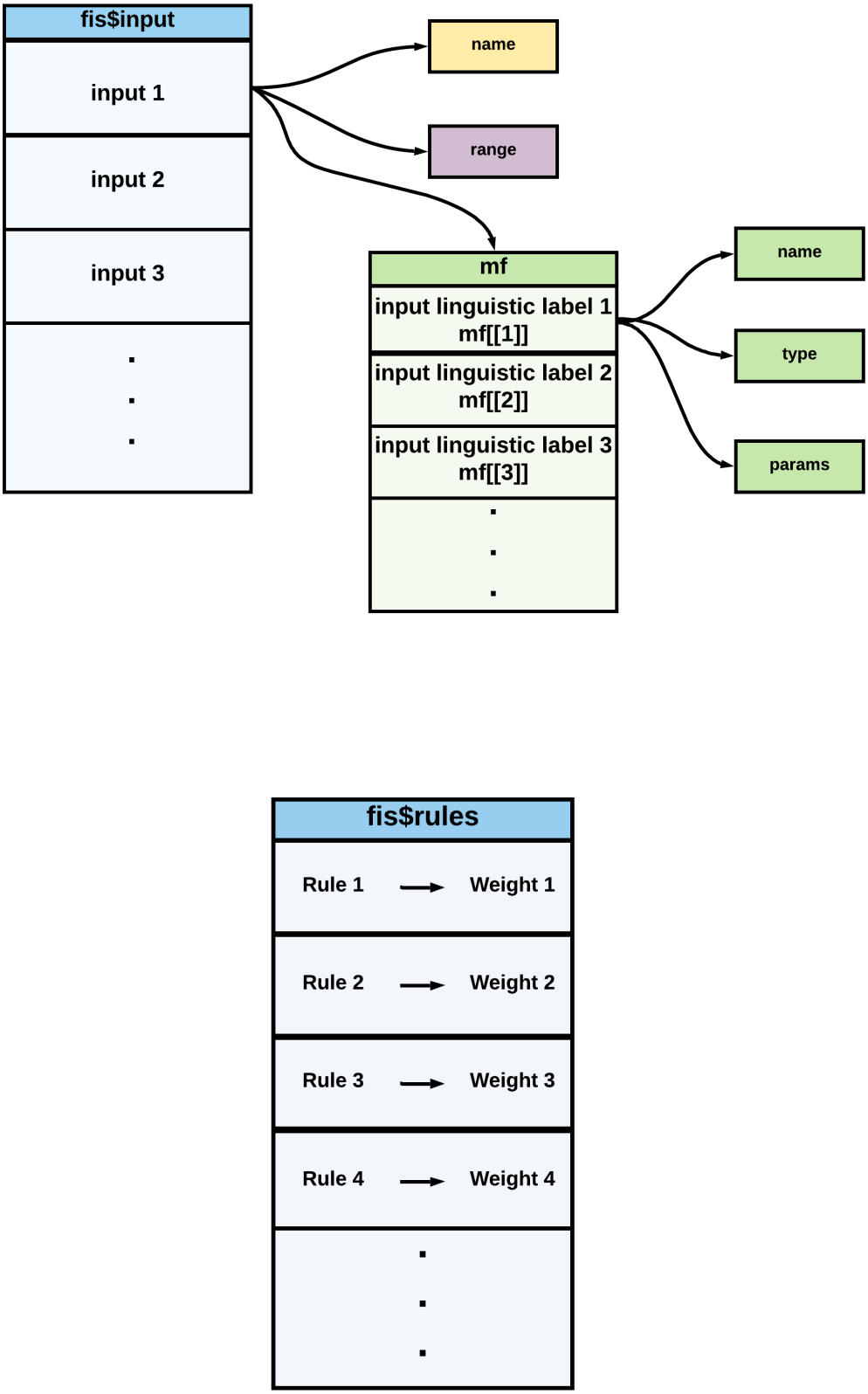


Figure 28. The structures of Input and Rules sub-lists of FIS object.

Inputs, Outputs, Linguistic Labels and Membership Functions: A part of the uploaded file is shown in *Figure 29*. As it can be seen from the figure, the names of properties, values and elements are contained in the Property_Name, Value_Name, and Element_Name columns. And their associated indexes are stored in Property, Value and Element columns.

Value	Value_Name	Property	Property_Name	Element	Element_Name	Best_Estimate	1	2	3	4	5	6	7	Possible	1	2	3	4	5	6	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	1	0	0	0	0	0	0	9	1	0	0	0	0	1	13	4
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	2	0	0	0	0	0	13	4	2	0	0	0	3	13	0	9
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	3	0	0	0	1	13	0	0	3	1	1	6	12	0	13	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	4	0	0	0	12	0	0	0	4	3	10	13	1	13	7	3
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	5	0	0	13	0	0	0	0	5	11	13	0	13	8	1	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	6	5	13	0	0	0	0	0	6	8	0	13	9	1	0	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	7	8	0	0	0	0	0	0	7	5	13	6	2	0	0	0

Figure 29. A fragment of uploaded file that contains the element *Forest of tranquillity*'s best estimate and possible relationships between human value *Health value (arising from the physical environment)* and property *Species richness.*

To extract these properties, values and elements, the uploaded file will first be passed to a function called *addVariables()*. **In this function, the names of properties, values and elements will be stored into three global lists: *property_list*, *value_list* and *element_list* respectively.** The position of the names in these three lists are the indexes of the corresponding properties/values/elements in the file. For example, the *addVariables()* function will store value *Health value (arising from physical environment)* as the first element of *value_list* since this value's index is 1 in the file.

Apart from properties, values and elements, the numbers of linguistic labels for properties and values should also be extracted. As introduced in Section 2.4, in best estimate relationships and possible relationships matrices, the numbers of columns and rows are also the numbers of linguistic labels for properties and values. For example, in *Figure 29*, as indicated in *Best_Estimate*, *Possible* columns (highlighted in blue squares) and the rows between and after them (highlighted in green squares), there are 7 columns and 7 rows in best estimate relationships and possible relationships, thus, there are 7 linguistic labels for properties and values in the file. **In the code, the numbers of properties' and values' linguistic labels will be extracted and stored into two global variables: *input_level* and *output_level* in function *addVariables()***.

In Section 5.2, it has been explained that the extracted input/output variables and their associated linguistic labels and membership functions should be stored by element in three different modules. To implement this idea in the coding level, three global lists: *total_functions_input*, *total_functions_output* and *total_rules* are defined. **The *total_functions_input* /*total_functions_output* is list of sub-lists with each of these sub-lists containing one element's linguistic labels, and membership functions of its properties/ values**. In other words, **each of the sub-list in these two variables correspond to one element's Input and Output modules.** And **the *total_rules* variable is also a list of sub-lists. Each of its sub-list stores the rules and rule weights for one element**. It corresponds to one element's Rules module. The structure of these three variables have been summarized in *Figure 30*.

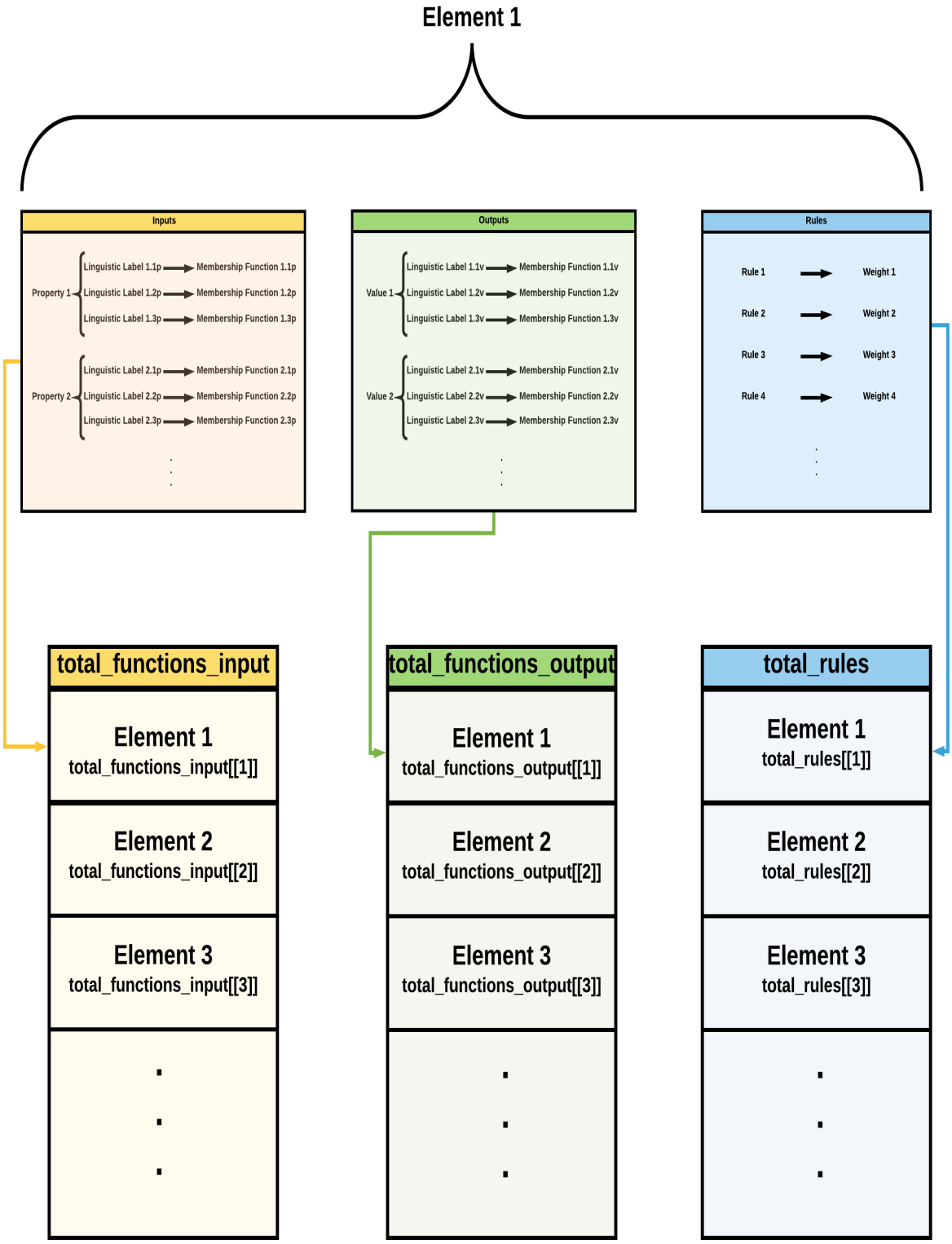

Element 1
Inputs
Property 1
Linguistic Label 1.1p
Membership Function 1.1p
Linguistic Label 1.2p
Membership Function 1.2p
Linguistic Label 1.3p
Membership Function 1.3p
Property 2
Linguistic Label 2.1p
Membership Function 2.1p
Linguistic Label 2.2p
Membership Function 2.2p
Linguistic Label 2.3p
Membership Function 2.3p
Outputs
Value 1
Linguistic Label 1.1v
Membership Function 1.1v
Linguistic Label 1.2v
Membership Function 1.2v
Linguistic Label 1.3v
Membership Function 1.3v
Value 2
Linguistic Label 2.1v
Membership Function 2.1v
Linguistic Label 2.2v
Membership Function 2.2v
Linguistic Label 2.3v
Membership Function 2.3v
Rules
Rule 1
Weight 1
Rule 2
Weight 2
Rule 3
Weight 3
Rule 4
Weight 4
total_functions_input
Element 1
total_functions_input[[1]]
Element 2
total_functions_input[[2]]
Element 3
total_functions_input[[3]]
total_functions_output
Element 1
total_functions_output[[1]]
Element 2
total_functions_output[[2]]
Element 3
total_functions_output[[3]]
total_rules
Element 1
total_rules[[1]]
Element 2
total_rules[[2]]
Element 3
total_rules[[3]]

Figure 30. Each sub-list of *total_functions_input* and *total_functions_output* corresponds to the Inputs and Outputs modules of an element. And each sub-list of *total_rules* corresponds to an element's Rules module.

As stated in Section 5.3, when users change the chosen element, the three modules of the newly chosen element should be able to directly replace the Input, Output and Rules lists in the FIS object. Thus, the structures of these three modules of one element should be the same as the ones in FIS object. **In other words, each of these sub-lists of *total_function_input, total_function_output* and *total_rules* should have the same structures as the Input, Output and Rules modules (lists) in FIS object**.

The structures of these three modules (lists) in FIS object have been exhibited above. As it can be seen, the structures for Input and Output modules are relatively complicated. To manually construct the structure for each of the sub-list in *total_function_input* and *total_function_output* is undoubtful a difficult and time-consuming task. Thus, another approach is taken.

Since the file does not contain any information about names and membership functions of linguistic labels, the software will initially assign the same default linguistic label names and membership functions to each element. Since apart from rules and rule weights, these two factors are all the differences among elements, thus, **before users customize them, the sub-lists stored in *total_function_input/total_function_output* are the same from one element to another. To construct these sub-lists, the *newfis()* function mentioned above can first be used to construct a new empty FIS object, then the *addvar()* and *addmf()* functions can be used to construct the Input and Output lists of this object. Since these two lists in FIS contain information about one element and different elements' Input and Output modules stored in *total_function_input* and *total_function_output* lists are the same at this stage, the Input and Output lists in the FIS object can be assigned to each element (sub-list) in these two lists.** By doing this, the structures of sub-lists in *total_function_input* and *total_function_output* are the same as the Input and Output lists in FIS object. This process has been summarized in *Figure 31*. To construct the Input/Output list of the FIS object, the *addvar()* will take the *property_list* /*value_list* generated in *addVariables()* function and the ranges of properties /values as inputs. And the *addmf()* will take the default membership functions, default linguistic label names and numbers of linguistic labels (*input_level* and *output_level*) as inputs. It should be noticed that the orders of elements in *total_function_input* and *total_function_output* are the same as the ones in *element_list*. If one element's index in *element_list* is 1, then its Input and Output modules are stored as the first element in *total_function_input* and *total_function_output* lists. This process of constructing *total_function_input* and *total_function_output* lists will occur when users upload the file that contains best estimate relationships and possible relationships.

This software will use triangular functions as the default membership functions for properties' linguistic labels and use triangular and trapezoid functions for values' linguistic labels. And the default names of the linguistic labels will be *level 1* to *level n* with n being the number of linguistic labels. The default setting for the property *Species richness* is shown in *Figure 32.* It should be noticed that in this software, the range for property scores will be 0-100 for now.

In future development, the software will be improved to allow users to define the range (mentioned in Section 8).

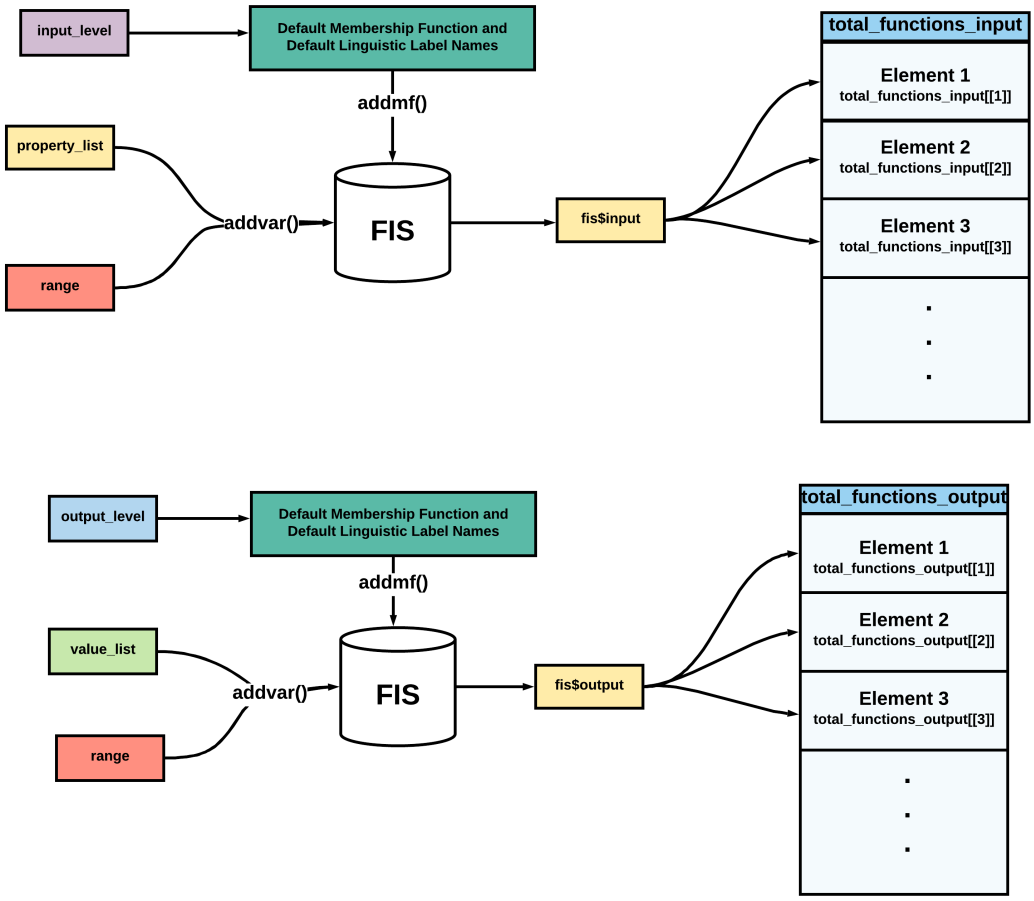


Figure 31. The process of constructing *total_function_input* and *total_function_output* with default linguistic label names and default membership functions when users upload best estimate relationship and possible relationship file.

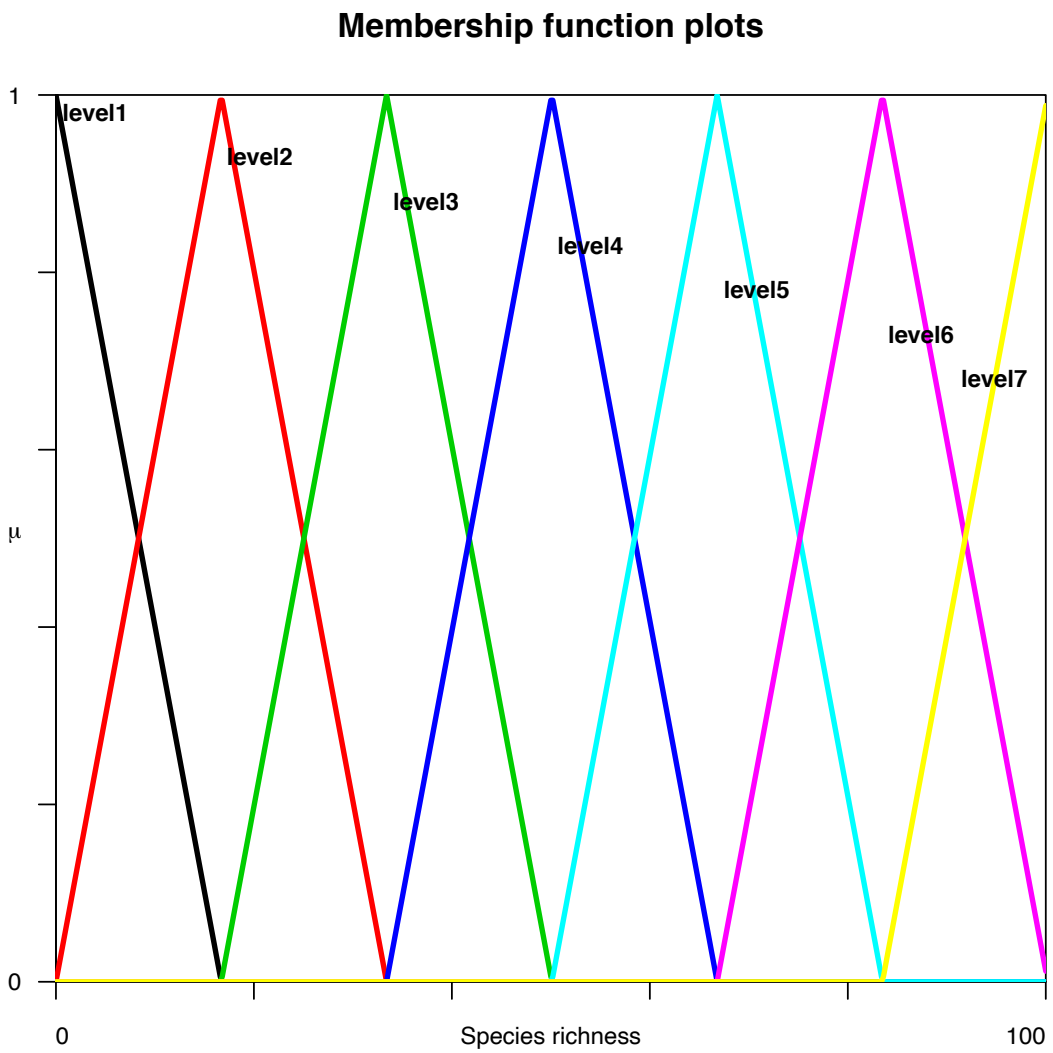


Figure 32. The default setting for the property *Species richness.* The default linguistic label names are level 1 to level 7 and the default membership function type is triangular function.

Extracting Rules and Rule Weights: The file uploaded by users contains different elements' best estimate relationships and possible relationships of all the property-value pairs. A fragment of the file is presented in *Figure 33*. As shown in the figure, the file can be divided into different sections and each of them contains the best estimate relationships and possible relationships between a property-value pair of an element. For example, the section highlighted by green colour shows the best estimate relationships (highlighted by red square) and possible relationships (highlighted in purple square) between property *Species richness* and value *Health value (arising from the physical environment)* of element *Forest of tranquillity.*

To extract rules and rule weights from these uploaded files, the best estimate relationships and possible relationships of each section need to be combined using the method introduced in Section 2.4 to generate summary matrices that contains rules and rule weights of the corresponding elements. As introduced in Section 2.4, while combining these two relationships, each of them should be given a weight. In this software, the weights for best estimate relationships and possible relationships will be 1 and 0.5 as suggested by [17]. After these summary matrices are obtained, the software will go through each cell of these matrices to extract the rules ad rule weights stored in them.

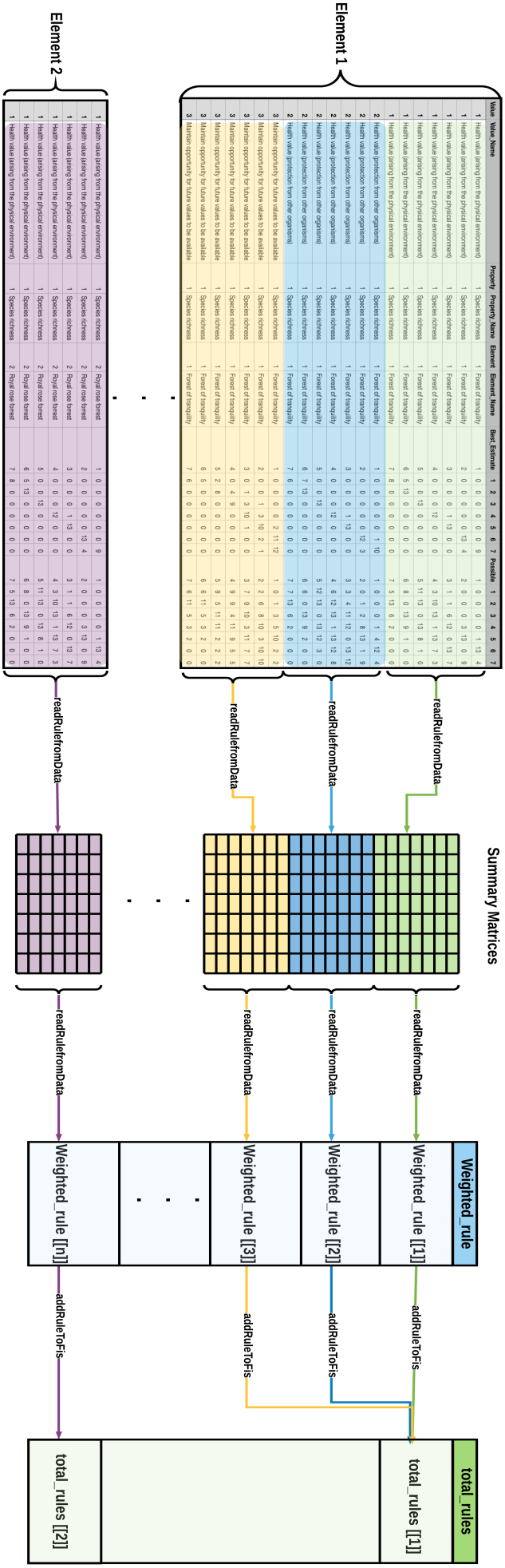
To implement this process at coding level, two global variables: *weighted_rule* and *total_rules* are employed. **The *weighted_rule* variable should be a list that contains the summary matrices generated from each section of the file and, as mentioned above, *total_rules* should store all the rules and rule weights generated from the file by element**.

The uploaded file will first be passed to a function called *readRulefromData()*. This function will be going through the file section by section and combining the best estimate relationships and possible relationships of each section into summary matrices. Then these matrices will be stored into *weighted_list* variable where the index for each matrix is the order of the corresponding section in the file. For example, in *Figure 33*, the green highlighted section is the first section of the file and the blue one is the second. Then the summary matrices obtained from the green and blue sections will be stored as the first and second element in *weighted_list* (*weighted_list [[1]]* and *weighted_list [[2]]*).

After obtained the *weighted_list*, **each summary matrix stored in it will be passed to a function called *addRuleToFis()*** to extract the rules and rule weights stored in these matrices. Apart from the matrices, the element names associated with each matrix will also be passed to the function. The function will go through these matrices cell by cell and extract the rules and rule weights. And the element names that are passed to this function will be compared to the *element_list* to obtain the elements' indexes in the list which will be used as the indexes of elements' corresponding sub-lists in the *total_rules* variable. For example, if an element's index in the *element_list* is 2, then the extracted rules and rule weights of this element will be stored in the second sub-list of the *total_rules* variable (*total_rules[[2]]*). The process described here is summarized in *Figure 34*.

Value	Value_Name	Property	Property_Name	Element	Element_Name	Best_Estimate	1	2	3	4	5	6	7	Possible	1	2	3	4	5	6	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	1	0	0	0	0	0	0	9	1	0	0	0	0	1	13	4
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	2	0	0	0	0	0	13	4	2	0	0	0	3	13	0	9
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	3	0	0	0	1	13	0	0	3	1	1	6	12	0	13	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	4	0	0	0	12	0	0	0	4	3	10	13	1	13	7	3
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	5	0	0	13	0	0	0	0	5	11	13	0	13	8	1	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	6	5	13	0	0	0	0	0	6	8	0	13	9	1	0	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	7	8	0	0	0	0	0	0	7	5	13	6	2	0	0	0
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	1	0	0	0	0	0	1	10	1	0	0	0	1	4	12	4
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	2	0	0	0	0	0	12	3	2	0	1	2	8	13	1	9
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	3	0	0	0	1	13	0	0	3	3	4	11	12	0	13	12
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	4	0	0	0	12	0	0	0	4	6	12	13	1	13	12	8
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	5	0	0	13	0	0	0	0	5	12	13	0	13	12	3	0
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	6	7	13	0	0	0	0	0	6	6	0	13	9	2	0	0
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	7	6	0	0	0	0	0	0	7	7	13	6	2	0	0	0
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	1	0	0	0	0	2	11	12	1	0	1	3	5	10	2	2
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	2	0	0	1	3	10	2	1	2	2	6	8	10	3	10	10
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	3	0	1	3	10	1	0	0	3	7	9	10	3	11	7	7
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	4	0	4	9	0	0	0	0	4	9	9	4	11	9	5	5
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	5	2	8	0	0	0	0	0	5	9	5	11	11	2	2	2
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	6	5	0	0	0	0	0	0	6	6	11	5	3	2	0	0
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	7	6	0	0	0	0	0	0	7	6	11	5	3	2	0	0
⋮																					
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	1	0	0	0	0	0	0	9	1	0	0	0	0	1	13	4
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	2	0	0	0	0	0	13	4	2	0	0	0	3	13	0	9
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	3	0	0	0	1	13	0	0	3	1	1	6	12	0	13	7
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	4	0	0	0	12	0	0	0	4	3	10	13	1	13	7	3
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	5	0	0	13	0	0	0	0	5	11	13	0	13	8	1	0
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	6	5	13	0	0	0	0	0	6	8	0	13	9	1	0	0
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	7	8	0	0	0	0	0	0	7	5	13	6	2	0	0	0

Figure33. The fragment of uploaded file that contains element *Forest of tranquillity*'s best estimate relationships and possible relationships between value *Health value (arising from the physical environment), Health value (protection from other organisms), Maintain opportunity for future values to be available* and property *Species Richness.* It also contains element *Royal rose forrest*'s best estimate relationship and possible relationship between value *Health value (arising from the physical environment)* and property *Species Richness.*



Value	Value_Name	Property	Property_Name	Element	Element_Name	Beat_Estimate	1	2	3	4	5	6	7	Possible	1	2	3	4	5	6	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	1	0	0	0	0	0	9		1	0	0	0	0	1	13	4
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	2	0	0	0	0	13	4		2	0	0	0	3	13	0	9
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	3	0	0	0	1	13	0	0	3	1	1	6	12	0	13	7
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	4	0	0	0	12	0	0	0	4	3	10	13	1	13	7	3
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	5	0	0	13	0	0	0	0	5	11	13	0	13	8	1	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	6	5	13	0	0	0	0	0	6	8	0	13	9	1	0	0
1	Health value (arising from the physical environment)	1	Species richness	1	Forest of tranquility	7	8	0	0	0	0	0	0	7	5	13	6	2	0	0	0
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	1	0	0	0	0	0	1	10	1	0	0	0	1	4	12	4
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	2	0	0	0	0	0	12	3	2	0	1	2	8	13	1	9
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	3	0	0	0	0	1	13	0	0	3	3	4	11	12	0	13
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	4	0	0	0	0	12	0	0	0	4	6	12	13	1	13	12
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	5	0	0	13	0	0	0	0	0	5	12	13	0	13	12	3
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	6	7	13	0	0	0	0	0	0	6	6	0	13	9	2	0
2	Health value (protection from other organisms)	1	Species richness	1	Forest of tranquility	7	6	0	0	0	0	0	0	0	7	7	13	6	2	0	0
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	1	0	0	0	0	2	11	12	1	0	1	3	5	10	2	2
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	2	0	0	1	3	10	2	1	2	2	6	8	10	3	10	10
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	3	0	1	3	10	1	0	0	3	7	9	10	3	11	7	7
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	4	0	4	9	0	0	0	0	4	9	9	4	11	9	5	5
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	5	2	8	0	0	0	0	0	5	9	5	11	11	2	2	2
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	6	5	0	0	0	0	0	0	6	6	11	5	3	2	0	0
3	Maintain opportunity for future values to be available	1	Species richness	1	Forest of tranquility	7	6	0	0	0	0	0	0	7	6	11	5	3	2	0	0
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	1	0	0	0	0	0	9		1	0	0	0	0	1	13	4
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	2	0	0	0	0	13	4		2	0	0	0	3	13	0	9
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	3	0	0	0	1	13	0	0	3	1	1	6	12	0	13	7
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	4	0	0	0	12	0	0	0	4	3	10	13	1	13	7	3
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	5	0	0	13	0	0	0	0	5	11	13	0	13	8	1	0
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	6	5	13	0	0	0	0	0	6	8	0	13	9	1	0	0
1	Health value (arising from the physical environment)	1	Species richness	2	Royal rose forrest	7	8	0	0	0	0	0	0	7	5	13	6	2	0	0	0

Figure 34. The process of extracting rules and rule weights from uploaded file.

Providing Element for FIS Construction: When users change the chosen element, the FIS object should also be updated accordingly. As mentioned in the design section, the *Rule Generation* part should provide the chosen element's Input, Output and Rules modules for the FIS object. Specifically, **this part will compare the element name that is chosen by user on the Utility Page with the *element_list* to provide the chosen element's index on this list**. This index will be further used in the *FIS Generation* part to extract the chosen element's Input, Output and Rules modules from the *total_function_input* and *total_function_output* and *total_rules.*

6.3 Implementation of FIS Generation

As mentioned in Section 5.3, the *FIS Generation* part is responsible for generating a FIS based on users' chosen element and calculating the chosen element's deliveries for human values based on the property scores.

Construct FIS: As stated in Section 6.2, a FIS object has already been generated during the *Rule Generation* part to construct *total_function_input* and *total_function_output (Figure 31)*. In the *FIS Generation* part, this object will again be used and updated to calculate the chosen element's deliveries for human values.

To utilize this FIS object, the Input, Output and Rules lists of this object should be first updated based on the chosen element. Since the indexes of one element's Input, Output and Rules modules on *total_function_input, total_function_output* and *total_rules* are the same as its index on the *element_list*, the user-chosen element's index in *element_list* that is provided by *Rule Generation* part can be used to access the chosen element's Input, Output and Rules modules. **Then the chosen element's three modules can directly swap the corresponding lists in FIS to update this object.** A snippet of code for updating the FIS based on chosen element is shown in *Figure* 35. The *chosen_index* is the element's index on *element_list* provided by *Rule Generation* and fis$input, fis$output and fis$rule are the Input, Output and Rules lists in FIS object.

```
fis$rule<<-total_rules[[chosen_index]]
fis$input<<-total_functions_input[[chosen_index]]
fis$output<<-total_functions_output[[chosen_index]]
```

Figure 35. The code for updating the FIS object when users change the chosen element.

Calculate Chosen Element's Deliveries: In the FuzzyR package, the built-in function: *evalfis()* can be used to calculate the chosen element's deliveries for all the values stored in the Output list based on the property scores. To utilize the function, the software will first need to extract the user-input property scores from the slide bars on the Utility Page, then these scores will be put into a list and passed to the *evalfis()* function. This function will then return the chosen element's deliveries for human values. The code snippet for this function is demonstrated in *Figure 36.* The *input_data* is the list that stores the user-input property scores and the *fis* is the FIS object. The element's deliveries for values will be stored in the variable *value_uti*.

```
value_uti<<-evalfis(input_data,fis)
```

Figure 36. The code for calculating the chosen element's deliveries for human values using function *evalfis().*

6.4 Implementation of Rating Information Extraction

As introduced in the design section, the *Rating Information Extraction* part should extract the centroid, spread, maximum agreement and average maximum importance from the uploaded Element and Value Rating files.

Centroid and Spread: As introduced above, the centroid and spread values should be the arithmetic mean and standard deviation of each element 's/value's possible ratings. To obtain these two values, the possible ratings should first be extracted from the uploaded files.

A fragment of the user-uploaded Value Rating file is exhibited in *Figure 37.* As it can be seen from this figure, this uploaded file can be divided into sections. Each of these sections contains one element's/value's highest and lowest possible ratings in different stakeholders' views. Each row represents one stakeholder's opinion and through these highest and lowest possible ratings collected from different stakeholders, the rating ranges supported by these stakeholders can be obtained. For example, through the highest and lowest possible ratings stored in the blue section of *Figure 37* the possible rating ranges of value *Provision of adequate resources* in 13 stakeholders' opinions can be obtained. To extract elements'/values' possible ratings from these files, these possible rating ranges of one section should first be aggregated.

Stakeholder	Value	Value_No	Low	High	Stakeholder_Des
1	Provision of adequate resources	1	20	100	Atlantis Bureau of Education
2	Provision of adequate resources	1	30	60	Atlantis Bureau of Recreation
3	Provision of adequate resources	1	90	100	Atlantis Indigenous Owners Council
4	Provision of adequate resources	1	75	95	Atlantis Landscape Strategy
5	Provision of adequate resources	1	10	30	Director Archaeology
6	Provision of adequate resources	1	10	100	Health Authority
7	Provision of adequate resources	1	25	60	Inland Waterways Association and Environment Agency
8	Provision of adequate resources	1	60	88	Lake Danger cross-country charioteering association
9	Provision of adequate resources	1	23	49	Lake Danger Partnership
10	Provision of adequate resources	1	5	15	Lake Danger Yachting Association
11	Provision of adequate resources	1	95	100	Port of Trouble Town Authority
12	Provision of adequate resources	1	79	98	Ramblers' Greater Trouble Town Forum
13	Provision of adequate resources	1	50	100	Water Authority
1	Provision of aesthetic pleasure	2	61	87	Atlantis Bureau of Education
2	Provision of aesthetic pleasure	2	71	100	Atlantis Bureau of Recreation
3	Provision of aesthetic pleasure	2	41	86	Atlantis Indigenous Owners Council
4	Provision of aesthetic pleasure	2	80	100	Atlantis Landscape Strategy
5	Provision of aesthetic pleasure	2	33	71	Director Archaeology
6	Provision of aesthetic pleasure	2	64	89	Health Authority
7	Provision of aesthetic pleasure	2	71	100	Inland Waterways Association and Environment Agency
8	Provision of aesthetic pleasure	2	74	100	Lake Danger cross-country charioteering association
9	Provision of aesthetic pleasure	2	43	74	Lake Danger Partnership
10	Provision of aesthetic pleasure	2	16	81	Lake Danger Yachting Association
11	Provision of aesthetic pleasure	2	25	64	Port of Trouble Town Authority
12	Provision of aesthetic pleasure	2	48	96	Ramblers' Greater Trouble Town Forum
13	Provision of aesthetic pleasure	2	53	100	Water Authority

Figure 37. The fragment of Value Rating file that contains 13 stakeholders' opinions about the importance ratings of value *Provision of adequate resources* and *Provision of aesthetic pleasure.*

The most direct way to aggregate these ranges is to combine all the ratings within them. For example, if, for element *Forest of tranquillity,* the possible ratings ranges that are generated from the lowest and highest possible ratings collected from two stakeholders are 50 to 52 and 48 to 53 respectively. Then, the possible ratings resulting from aggregating these two ranges are 48,49,50,51,52,53.

However, in this method, the levels of stakeholders' agreement of the possible ratings are not taken into consideration. In other words, if the *Rating Information Extraction* part combines all the ranges like above, the centroid and spread values generated from the resulting possible ratings do not contain information about stakeholders' agreement levels. Since these agreement levels can provide rich information for later analysis and supporting environment manager's informing decision-making [9], other approaches need to be considered to include this information into the possible ratings. One feasible approach that is just a simple variation of the above method is to **combine these ranges without eliminating the duplicated possible ratings of elements/values collected from different stakeholders**

For example, in the case of element *Forest of tranquillity* mentioned above, if the *Rating Information Extraction* part does not eliminate duplicated ratings when combining these two

possible rating ranges, the result will be 48,49,50,50,51,51,52,52,53. This shows that both of the stakeholders agree that the importance rating for this element can be 50,51 or 52 but only one of them thinks it can also be 48, 49 or 53. In other words, the stakeholders' agreement levels are nicely manifested in the result by the numbers of ratings' appearances. As a result, the centroid and spread values should be elicited from the possible ratings obtained using this method.

To implement this method, the uploaded file will be passed to a function called *val_rank()*. This function will go through the file section by section to elicit each element's/value's centroid, spread, maximum agreement and average maximum importance. **This function will be called every time an Element/Value Rating file is uploaded. If an Element Rating file is uploaded, these extracted centroids, spread, maximum agreement and average maximum importance belong to elements otherwise they belong to values**. Also, it should be noticed that in this software, the ratings of elements and values are on the scale from 0-100.

To represent the rating ranges in coding level, the range contained in each row will be presented in a list with 101 elements and each of these elements corresponds to a rating from 0-100. The offset between the actual rating and the index of its corresponding list element is 1 since in R the first index of a list is 1 not 0. For example, for rating n, its corresponding list element will be at the index n+1. The list elements correspond to possible ratings will be assigned to 1 and the rest will be 0. For example, if a possible rating range for an element is from 46-78, then the list elements on indexes 1 to 46 and 80 to 101 will be 0 and the elements on positions 47 to 79 will be 1.

Each row will generate a list containing the possible rating ranges and these lists obtained from one section will be summed together to obtain a resulting list with 101 list elements. The indexes on the resulting list also represent the ratings from 0-100 and the value stored at each index stands for the number of stakeholders that agree this corresponding rating is possible for the element/value of the section. For example, if a number "8" is stored in the list at index 50 then there are 8 stakeholders agree that the element/value's rating is 49 (the offset between index and actual rating is 1). **These resulting lists elicited from each section will then be stored in a global list called *interval_value*.** The indexes of these lists in *interval_value* are the orders of their associated sections in the file.

After obtained the *interval_value*, the next step of the function *val_rank()* is to **use each list stored in *interval_value* to obtain possible ratings for each element/value.** As mentioned above, in the possible ratings, the numbers of ratings' appearances should be stakeholders' levels of agreement. **Thus, for each list stored in *interval_*value, the value at each list index will be the number of appearances of the index's corresponding rating.** For example, if the scale of the ratings is 0 to 5, then a list in *interval_value* should contain 6 elements such as: "0 0 2 2 1 2". If this list is turned into possible ratings, then the resulting ratings will be "2 2 3 3 4 5 5" since this list indicates that two stakeholders agree that the rating can be 2,3 and 5, one stakeholder thinks it can be 4 and no stakeholders think it can be 0 or 1. **Then the centroid and spread values will be calculated based on these possible ratings for each element/value. The resulting centroids and spreads will be stored into *centroid_all* and *spread_all***

respectively by element/value. The process of calculating the centroid and spread is manifested in *Figure 38.*

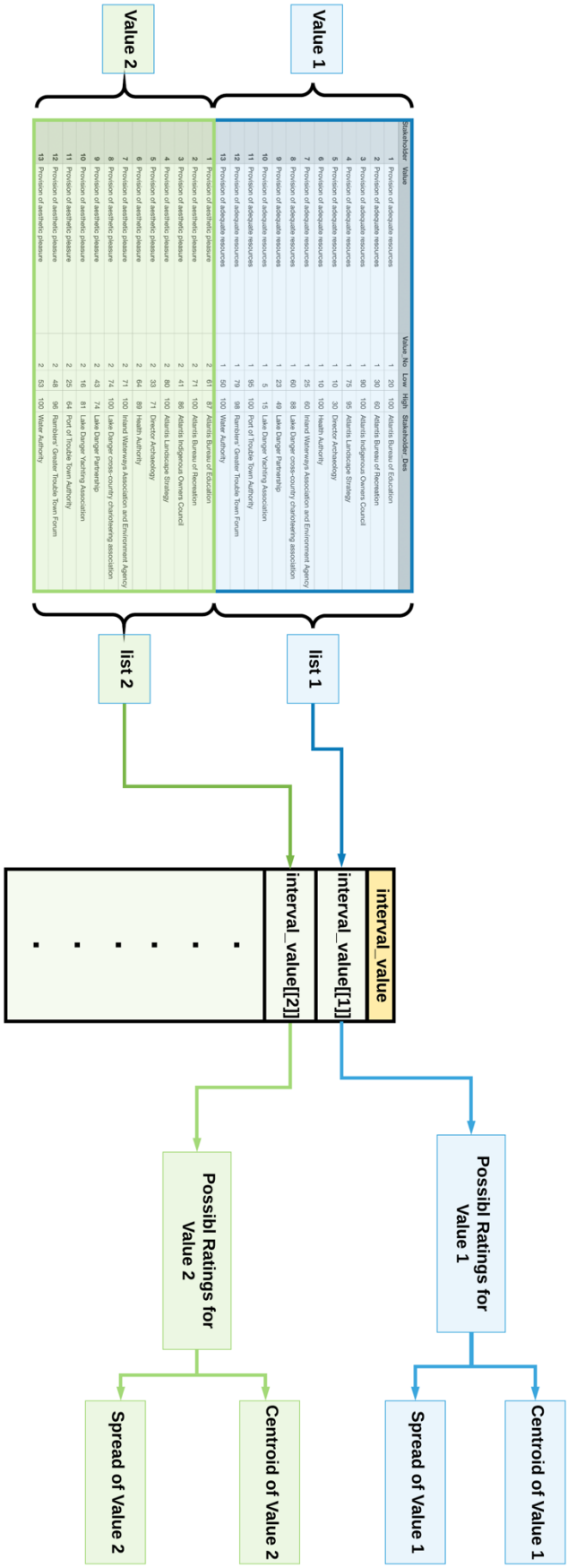

Value 1
Value 2
Stakeholder Value Value_No Low High Stakeholder_Des
1 Provision of adequate resources 1 20 100 Atlantis Bureau of Education
2 Provision of adequate resources 1 30 60 Atlantis Bureau of Recreation
3 Provision of adequate resources 1 80 100 Atlantis Indigenous Owners Council
4 Provision of adequate resources 1 75 95 Atlantis Landscape Strategy
5 Provision of adequate resources 1 10 30 Director Archaeology
6 Provision of adequate resources 1 10 100 Health Authority
7 Provision of adequate resources 1 25 60 Inland Waterways Association and Environment Agency
8 Provision of adequate resources 1 60 88 Lake Danger cross-country chandsteering association
9 Provision of adequate resources 1 23 49 Lake Danger Partnership
10 Provision of adequate resources 1 5 15 Lake Danger Yachting Association
11 Provision of adequate resources 1 95 100 Port of Trouble Town Authority
12 Provision of adequate resources 1 79 98 Ramblers Greater Trouble Town Forum
13 Provision of adequate resources 1 50 100 Water Authority
1 Provision of aesthetic pleasure 2 61 87 Atlantis Bureau of Education
2 Provision of aesthetic pleasure 2 71 100 Atlantis Bureau of Recreation
3 Provision of aesthetic pleasure 2 41 86 Atlantis Indigenous Owners Council
4 Provision of aesthetic pleasure 2 80 100 Atlantis Landscape Strategy
5 Provision of aesthetic pleasure 2 33 71 Director Archaeology
6 Provision of aesthetic pleasure 2 64 89 Health Authority
7 Provision of aesthetic pleasure 2 71 100 Inland Waterways Association and Environmet Agency
8 Provision of aesthetic pleasure 2 74 100 Lake Danger cross-country chandsteering association
9 Provision of aesthetic pleasure 2 43 74 Lake Danger Partnership
10 Provision of aesthetic pleasure 2 16 81 Lake Danger Yachting Association
11 Provision of aesthetic pleasure 2 25 64 Port of Trouble Town Authority
12 Provision of aesthetic pleasure 2 48 96 Ramblers Greater Trouble Town Forum
13 Provision of aesthetic pleasure 2 63 100 Water Authority
list 1
list 2
interval_value
interval_value[[1]]
interval_value[[2]]
Possibl Ratings for Value 1
Possibl Ratings for Value 2
Centroid of Value 1
Spread of Value 1
Centroid of Value 2
Spread of Value 2

Figure 38. The process of eliciting the centroids and spreads from the uploaded Element/Value Rating files.

Maximum Agreement: As introduced in the design section, the maximum agreement of an element/value is the largest number of stakeholders that agree on a single rating.

To elicit the maximum agreement from the uploaded files, the list *interval_value* that is generated while extracting the centroid and spread will be utilized in the *val_rank()* function. As mentioned above, each sub-list of *interval_value* is generated from possible rating ranges of one element/value and inside these sub-lists, each index relates a rating and the value at that index represents the stakeholders' agreement level for the corresponding rating.

Thus, the *val_rank()* function will extract the maximum agreement of elements/values as the largest number in their corresponding sub-lists in *interval_value*. For example, when the scale of the rating is 0-5, one element's corresponding sub-list in *interval_value* is: "0 2 3 7 3 2". Since the largest number of the list is 7 then the maximum agreement of this element will be 7. This is because, among all these ratings, the rating 3 is supported by most stakeholders (7 stakeholders). As a result, the number of stakeholders support this rating will be the maximum agreement for this element. **The maximum agreements elicited from elements/values will be stored in variable *maxi_agree by element/value.***

Average Maximum Importance: The average maximum importance of one element/value is the arithmetic mean of the element's /value's highest possible ratings in stakeholders' views. **To implement this, for each section of the uploaded file, the *val_rank()* function will extract the values stored in the *High* column and calculate the arithmetic mean of these values as the section's corresponding element's/value's average maximum importance. The resulting average maximum importance will be stored in variable *maxi_avg* by element/value.**

6.5 Implementation of Utility Calculation

The element's utility should be calculated as a weighted average of the element's deliveries for human values. **Thus, to calculate the chosen element's utility, the *value_uti* variable mentioned in Section 6.3 that is returned by the function *evalfis()* will be used to provide the chosen element's deliveries for human values**.

As elaborated in design section, the weights for the chosen element's deliveries for human values will depend on the average maximum importance of the corresponding values. To turn the average maximum importance stored in *maxi_avg* into weights within the range of 0 to

1, each of these values will be divided by the sum of all the average maximum importance stored in the variable. For example, if the average maximum importance stored in *maxi_avg* are 23 45 26 78 13 then the first list element which belongs to value *Provision of adequate resources* can be turned into weight: $\frac{23}{23+45+26+78+13)} \approx 0.12$. This weight will then be applied to the chosen element's delivery for value *Provision of adequate resources.* **The weights generated from the *maxi_avg* are then stored in variable *weight_sum_val*. It should be noticed that the weights generation will occur when users upload the Value Rating file since as introduced in Section 6.4, the *maxi_avg* will only contain values' average maximum importance if the uploaded file is Value Rating file**.

After the weights are generated, the chosen element's utility can be calculated by combining the deliveries in *value_uti* and weights in *weight_sum_val.*

6. 6 Implementation of Data Visualization

As mentioned in the design section, the software should be able to visualize the summary matrices extracted from Best Estimate and Possible files and the IAA graphs generated from Element and Value Rating files.

Visualize Summary Matrices: The summary matrices generated from the uploaded files have been stored in the *wighted_rule* variable as elaborated in Section 6.2. To visualize these summary matrices using the heat-map style that is mentioned in Section 5.5, a frontend table: DataTables from the DT package will be employed since it is equipped with the ability for filtering and colouring the data stored in it [24]. This visualization process will be completed in *draw_plots()* function in which each of the summary matrices in *weighted_rule* will be presented to users in heat-map style by DataTables. A snippet of code for visualizing the summary matrices using DataTables is shown in *Figure 39*.

```
output[[paste0("plot", i) ]] <- DT::renderDataTable(
  datatable(weighted_rule[[i]],
            options = list(searching = FALSE,paging = FALSE,ordering=F), caption = paste0(rule_set[[(i*7)+1,"Value_Name"],," and ",rule_set[[(i*7)+1,"Property_Name"],," of element ", rule_set[[(i*7)+1,"Element_Name"])
  ) %>%
    formatStyle( names(weighted_rule[[i]]),backgroundColor = styleInterval(brks, clrs))
)
```

Figure 39. The code for visualizing the summary matrices using DataTables.

Visualize IAA Result: The IAA graphs are obtained by aggregating the possible rating ranges of elements and values with the x-axis being the ratings of elements/values and the y-axis being stakeholders' agreement levels. **These graphs can be generated based on the sub-lists stored in *interval_value.***

As introduced above, these sub-lists contain all of the ratings and their associated levels of stakeholders' agreement. Thus, to plot these IAA graphs, for each of the sub-lists, the ratings represented by indexes will be the values on the x-axis and their corresponding values on y-axis will be the values stored in these indexes. Each of these sub-lists will generate one element's /value's IAA graph. As mentioned in the Section 5.7.1, the orders of IAA graphs on

the interface should be based on the final importance ratings (average maximum importance) of their corresponding values/elements. For example, if one element has the second highest importance rating, then its IAA graph should be presented as the second IAA graph on the interface. **This plotting process will be accomplished by the function *draw_plots()* in which ShinyR's frontend plotting module *plotOutput [25]* will present users the plots generated by R's function *plot() [26].*** Also, to further support users to understand the information contained in these graphs, **the centroid and average maximum importance of the elements/values are highlighted on the graphs using red and blue dotted lines** [15]. The code for the IAA graph plotting is shown in *Figure 40*.

```
output[[paste0("plot", i) ]] <- renderPlot({
  plot(x,y,type = "l",xlab = "Importance",ylab = "Agreement",main=value_name[i])
  abline(v=mean(dataset[[i]]),col="red",lty=2, lwd=3)
  abline(v=interval_value[,value_name[i]][length(interval_value[,value_name[i]])],col="blue",lty=2, lwd=3)
})
```

Figure 40. The code for visualizing the IAA graphs.

6.7 Implementation of Rule Customization

As introduced in the design section, the software should be able to customize the linguistic labels and membership functions of chosen element's chosen property/value.

As mentioned above, the linguistic labels and membership functions of properties and values are all stored in variable *total_function_input* and *total_function_output* by element. **Thus, any changes that users make will be applied to the corresponding element in these two variables.**

To implement this, first thing needs to know is which one of these two variables will be changes. **In other words, the software needs to figure out if users want to change the linguistic labels and membership functions of properties or values**. This information can be extracted from the radio button on the Customization Page as introduced in the frontend design section. Then, the name of element that is chosen on the Customization Page will be compared to the *element_list* to obtain the chosen element's index on the list. **This index can then be used to access the chosen element's Input and Output lists in *total_function_input* and *total_function_output*.**

Then, the software needs to have information about which property/value stored in the Input/Output lists the user wants to change. **Since the user will choose the property/value name on the Customization Page, the chosen name will be compared to *property_list/value_list* to obtain the index of chosen property's/value's corresponding sub-list on the Input/Output list.**

After obtained the sub-list that contains information about the chosen property/value, the software needs to know which linguistic label and membership function stored in the sub-list's *mf* list will be changed by user. This information will be collected from the linguistic label name chosen by user on the Customization Page. **The chosen name will be compared to the *name* field of each sub-list in *mf* list to locate the sub-list that contains linguistic label and membership function the user wants to change.**

After this, the users can assign new linguistic label names, new membership function type and its associated parameters to the *name*, *type* and *params* components of the chosen sub-list in *mf*. The relationships among *total_function_input, total_function_output,* Input/Output list and *mf* list can be reviewed in Section 6.2.

6.8 Implementation of Rule Exhibition

The *Rule Exhibition* part should be able to show the rules and rule weights of user's chosen element. As introduced above, all the extracted rules and rule weights are stored in the variable *total_rules* by element. **Thus, the software should first extract the name of the chosen element from the Rule Page and compare it with *element_list* to obtain the chosen element's index which will be used to access the element's associated rules and rule weights stored in *total_rules.***

One thing needs to notice is that, the rules stored in the software are represented by lists of numbers. For example, for a rule whose length is m+n+2. Then the number of input variables (properties) is m and the number of output variables (values) is n. Each column's order in m and n represents the corresponding property's/value's index on elements' Input/Output modules and the value stored in the column is the index of the *mf*'s sub-list that contains the corresponding property's/value's linguistic label and membership function. Then the last two digits stand for the weight of the rule and the fuzzy operators (1 = AND, 2 = OR) between the input variables. Since in this case, each rule only associates with one property and one value, the fuzzy operator is not needed thus it can be randomly assigned to 1 or 2. In this software, all the rules are assigned with 2 for the fuzzy operator digit. An example of the rule is shown in *Figure 41*.

For example, if one has a fis with 2 input variables, and 1 output variable, each of which have 3 membership functions (the amount of membership functions need not be the same). The following rule: 1 3 2 1 2 will mean m = 2 (for 2 input variables), n = 1 (for 1 output variable), and the last 2 columns represent weight and fuzzy operator for the rule's antecedent respectively.

The first column refers to the first input variable's membership function at index 1

The second column refers to the second input variable's membership function at index 3

The third column refers to the first output variable's membership function at index 2

The fourth column refers to the weight to be applied to the rule.

The fifth column refers to the fuzzy operator for the rule's antecedent (in this case it represents 'OR').

Figure 41 (cited from [23]). An example of the rules stored in the software. In this example, the "first" and "second" used to describe the input/output variables are these variables' indexes on the Input/Output modules. The indexes of membership functions are the indexes of the sub-lists in *mf* list that contain the membership functions and their associated linguistic labels.

Since these rules will be shown to users, they should be turned into understandable linguistic terms. For example, the rule 1 3 2 1 2 shown in *Figure 41* can be turned into the readable sentence: *"if the first input variable is level 1 or the second input variable is level 3 then the output variable is level 2. weight: (1)"* (the level 1,2,3 here represent the associated linguistic labels). **Thus, the rules and rule weights of chosen element will first be passed to *demorules()* function to turn the rules into readable terms before presenting them to the users.**

6.9 Frontend Implementation

Local Implementation: **The frontend of the software is implemented using ShinyR library**. As it can be seen from the design section, the frontend will be using navigation bars to navigate through different pages. To achieve this, the *navbarPage* [27] in ShinyR library is used to contain all the pages in the software. Inside the navbarPage, the pages mentioned in the design section are implemented using *tabPanel* [28] and a unique id is assigned to each of these pages to support the implementation of page switch.

On-line Implementation: As mentioned above, the software that is implemented using ShinyR can be published online as web-applications [29]. In this project, the software is published on 'shinyapps.io' server by using package *rsconnect* [30]. This package is programmatic deployment interface for 'shinyapps.io' and 'RStudio Connect'. It enables the developers to publish and maintain the online software.

7 Case Study and Evaluation

In this section, the functional requirements and the non-functional requirement: ***flexibility*** of this software will be tested using data collected in the **Atlantis Management** case.

7.1 Background

This case study focuses on the ***Phoenix Catchment*** that is located approximately 80km west of Atlantis City in the western agriculture zone of Atlantis. The location of this catchment in Atlantis island is shown in *Figure 42.* The Phoenix Catchment is highlighted in the figure using white dotted line (highlighted in blue square). The management of this area will be planned and conducted by the Royal Atlantis Department of Conservation and Land Management (hereafter referred to as CALM).

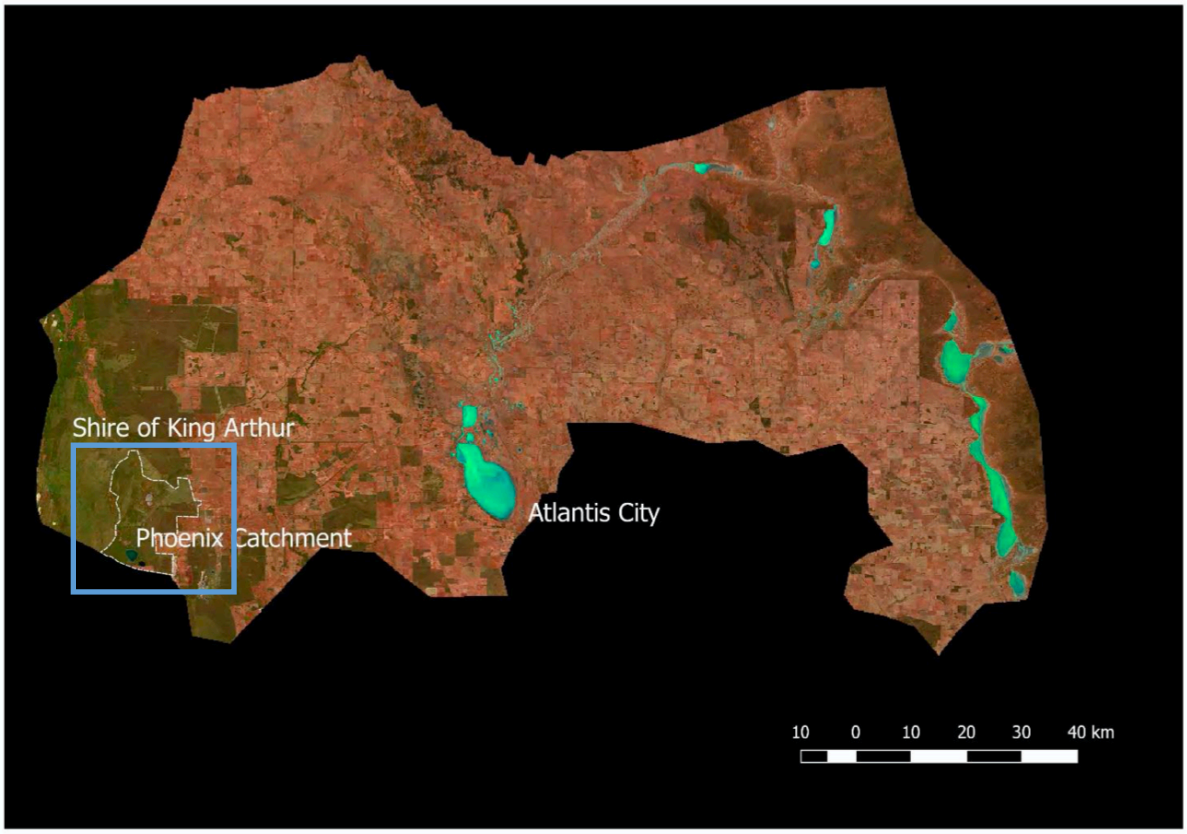


Figure 42 (cited from [15]). The locations of Phoenix Catchment, Shire of King Arthur and Atlantis City.

According to [15], with the increasing salinization in Atlantis, the abiotic and wildlife elements in this catchment are facing great risks. To counteract the negative effect, informing and wise decisions need to be made to manage the elements in this area. In this case, the CALM will employ the five-steps values-driven framework introduced in Section 2 to drive the decision-

making process with elements' utilities that are generated based on the elements' property scores. They will run a series of workshops for stakeholders and experts to 1) elicit key values and elements for this catchment, 2) rate the importance of the elicited key values and elements, 3) identify key element properties and quantify the property-value relationships. The importance rating and property-value relationship quantification will be carried out in workshops using the methods introduced in Section 2.5 and 2.4. As mentioned in Section 2.2, the software will facilitate the step3 to step 5 of the framework.

7.2 Procedures

This case study focuses on testing the software's functionalities. Since these functionalities are for supporting step 3 to step 5 in values-driven framework, the procedures of the evaluation will correspond to these three steps. Each of these procedures will be using data collected from the Atlantis Management case

1. ***Elicit property-value relationships from uploaded file.***
2. ***Rate key values and elements based on the stakeholder's opinions.***
3. ***Calculate the elements' contributions to human well-being.***

7.3 Observations

7.3.1 Property-Value Relationship Elicitation

To utilize values-driven framework, the managers need to elicit the quantified property-value relationships from the data collected from experts. The software should be able to facilitate this process by quantifying the property-value relationships, enabling users to understand these relationships more directly as well as turning these collected relationships into readable terms.

In this evaluation step, the file that contains the property-value relationships is uploaded to the software. Then, it has been observed that the software can successfully generate the quantified property-value relationships using method introduced in Section 2.4 and visualize them in summary matrices (*Figure 43*). These generated summary matrices largely help users view and understand these quantified relationships more directly.

Health value (protection from other organisms) and Species richness of element Forest of tranquility

	State 1	State 2	State 3	State 4	State 5	State 6	State 7
100	0	0	0	0	0.5	6.5	11
83.5	0	0	0	1.5	6.5	13	8.5
66.8	0.5	0.5	3	7	13	6.5	3.5
50.1	1.5	5	6.5	12.5	6.5	3.5	1.5
33.4	5.5	6.5	13	6.5	4	0.5	0
16.7	9	13	6.5	4.5	0.5	0	0
0	10.5	6.5	3	1	0	0	0

Figure 43. The visualized summary matrix that contains element *Forest of tranquillity*'s weighted property-value relationships between value *Health value (protection from other organisms)* and property *Species richness* elicited from the best estimate relationships and possible relationship file in Atlantis Management case

As it can be observed, the software generates the summary matrices in a heat-map style which can directly show the general relationships between the properties and values. For example, in *Figure 43*, based on the colours of these cells, it can be observed that the value *Health value (protection from other organisms)* and property *Species richness* of element *Forest of tranquillity* are positively related in most experts' views.

Also, the observation shows that the software can successfully turn these quantified relationships into linguistic terms that can be easily understood by users. It will also normalize these relationships' weights into the range 0 to 1. *Figure 44*

Element

Forest of tranquility

if (Species richness is level 1) then (Health value (arising from the physical environment) is level7). weight: (0)
if (Species richness is level2) then (Health value (arising from the physical environment) is level7). weight: (0)
if (Species richness is level3) then (Health value (arising from the physical environment) is level7). weight: (0)
if (Species richness is level4) then (Health value (arising from the physical environment) is level7). weight: (0)
if (Species richness is level5) then (Health value (arising from the physical environment) is level7). weight: (0.03)
if (Species richness is level6) then (Health value (arising from the physical environment) is level7). weight: (0.45)
if (Species richness is level7) then (Health value (arising from the physical environment) is level7). weight: (0.76)
if (Species richness is level 1) then (Health value (arising from the physical environment) is level6). weight: (0)
if (Species richness is level2) then (Health value (arising from the physical environment) is level6). weight: (0)
if (Species richness is level3) then (Health value (arising from the physical environment) is level6). weight: (0)
if (Species richness is level4) then (Health value (arising from the physical environment) is level6). weight: (0.1)
if (Species richness is level5) then (Health value (arising from the physical environment) is level6). weight: (0.45)
if (Species richness is level6) then (Health value (arising from the physical environment) is level6). weight: (0.9)
if (Species richness is level7) then (Health value (arising from the physical environment) is level6). weight: (0.59)
if (Species richness is level 1) then (Health value (arising from the physical environment) is level5). weight: (0.03)
if (Species richness is level2) then (Health value (arising from the physical environment) is level5). weight: (0.03)
if (Species richness is level3) then (Health value (arising from the physical environment) is level5). weight: (0.21)
if (Species richness is level4) then (Health value (arising from the physical environment) is level5). weight: (0.48)
if (Species richness is level5) then (Health value (arising from the physical environment) is level5). weight: (0.9)
if (Species richness is level6) then (Health value (arising from the physical environment) is level5). weight: (0.45)
if (Species richness is level7) then (Health value (arising from the physical environment) is level5). weight: (0.24)
if (Species richness is level 1) then (Health value (arising from the physical environment) is level4). weight: (0.1)
if (Species richness is level2) then (Health value (arising from the physical environment) is level4). weight: (0.34)
if (Species richness is level3) then (Health value (arising from the physical environment) is level4). weight: (0.45)
if (Species richness is level4) then (Health value (arising from the physical environment) is level4). weight: (0.86)
if (Species richness is level5) then (Health value (arising from the physical environment) is level4). weight: (0.45)
if (Species richness is level6) then (Health value (arising from the physical environment) is level4). weight: (0.24)
if (Species richness is level7) then (Health value (arising from the physical environment) is level4). weight: (0.1)
if (Species richness is level 1) then (Health value (arising from the physical environment) is level3). weight: (0.38)

Figure 44. Part of the rules and rule weights of element *Forest of tranquillity.*

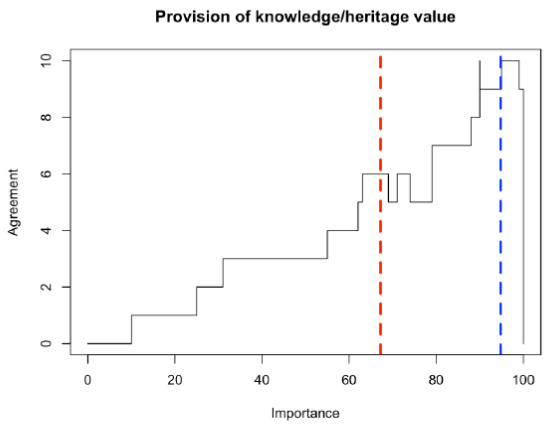
7.3.2 Rating Elements and Values

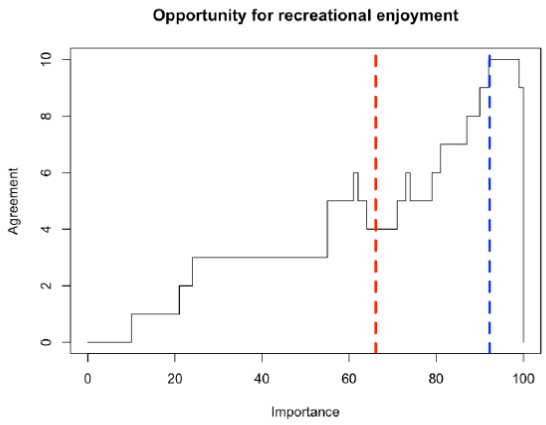
Since the values-driven framework requires rating the elements and values based on their importance, this software should be able to support this process and enable users to obtain rich set of information from the rating files for later analysis and interpretation.

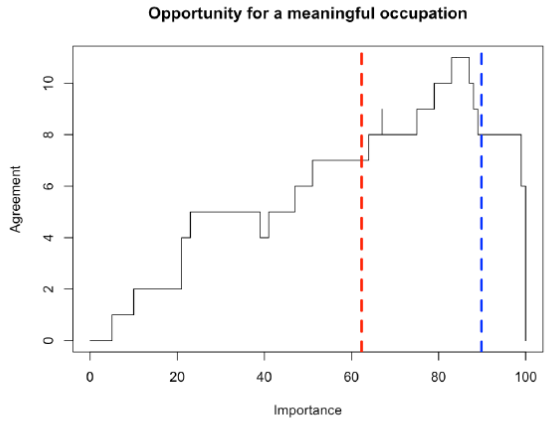
In this evaluation, it has been observed that after uploading the element/value rating files, the software is able to rate these elements and values using the average maximum importance elicited from the uploaded file. It can also be seen in the observation that the relationships between the collected possible ratings and their associated agreement levels of stakeholders are visualized by the software using IAA method to help users understand the uncertainties among stakeholders' responses. Additionally, four key values: centroid, spread,

maximum agreement and average maximum importance are elicited by the software from the possible ratings of elements/values to further help users understand the information contained in the ratings. The orders of these IAA graphs and key value tables on the interface successfully manifest the ratings of their corresponding elements/values.

Example of IAA graphs generated in the evaluation using Value Rating file are shown in *Figure 45.* And the examples of the tables that contain the key values are shown in *Figure 46*.


Provision of knowledge/heritage value
Agreement
Importance
0
2
4
6
8
10
0
20
40
60
80
100


Opportunity for recreational enjoyment
Agreement
Importance
0
2
4
6
8
10
0
20
40
60
80
100


Opportunity for a meaningful occupation
Agreement
Importance
0
2
4
6
8
10
0
20
40
60
80
100

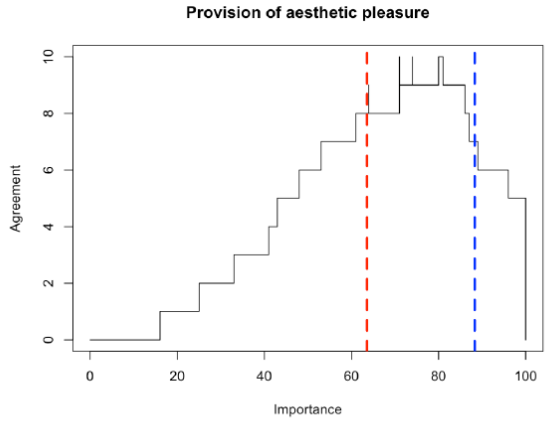

Provision of aesthetic pleasure
Agreement
Importance
0
2
4
6
8
10
0
20
40
60
80
100

Figure 45. Examples of visualized IAA graphs generated from the Value Rating file in Atlantis Management case.

	Provision of knowledge/heritage value
Centroid	68.20
Spread	26.30
Maximum agreement	10.00
Average maximum importance	94.80

	Opportunity for recreational enjoyment
Centroid	67.10
Spread	27.00
Maximum agreement	10.00
Average maximum importance	92.20

	Opportunity for a meaningful occupation
Centroid	63.30
Spread	25.10
Maximum agreement	11.00
Average maximum importance	89.80

	Provision of aesthetic pleasure
Centroid	64.50
Spread	23.70
Maximum agreement	10.00
Average maximum importance	88.30

Figure 46. Example of the tables that contain centroid, spread, maximum agreement and average maximum importance elicited from Value Rating file in Atlantis Management case.

7.3.3 Elements' Contribution Calculation

The final step of the values-driven framework is to calculate the elements contributions (utilities) based on the property-value relationships and element/value ratings collected in

the previous steps. This software should be able to support this utility calculation process using Fuzzy Inference System and allow users to customize the system by changing the linguistic labels and membership functions.

The observation of the evaluation shows that by uploading the collected files the software can use the default settings to successfully generate the user-chosen element's deliveries for human values as well as its overall contribution to human well-being (utility) based on the property scores input by users. An example of this is shown in *Figure 47.*

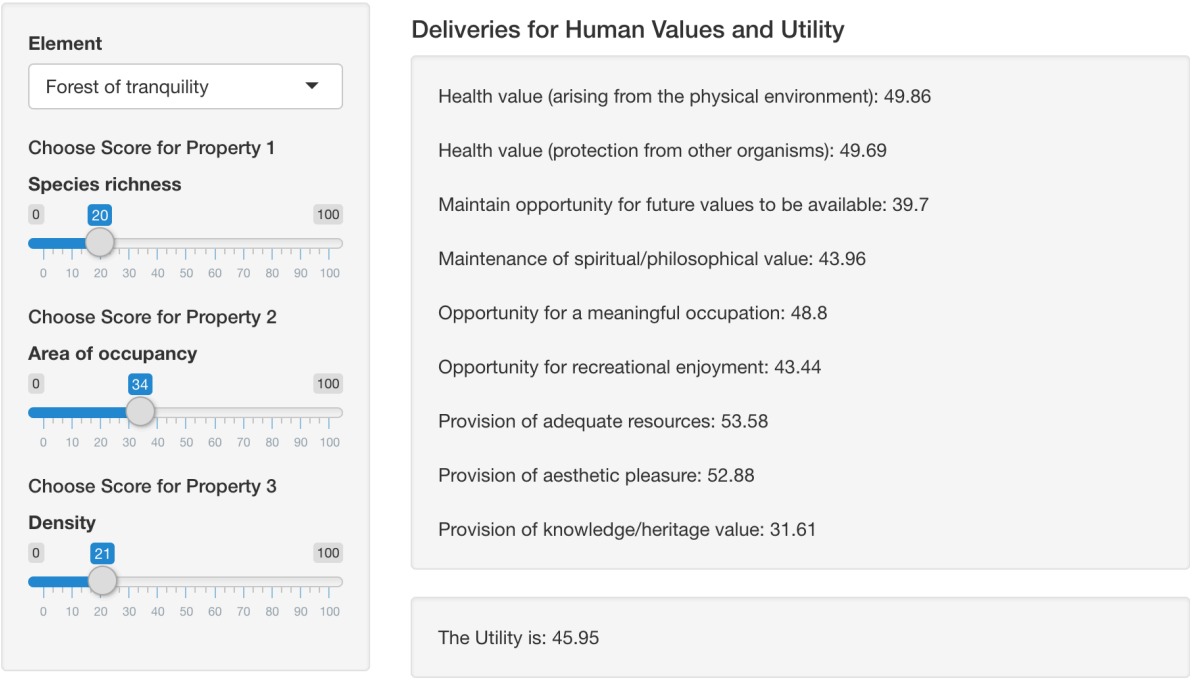


Figure 47. The software's interface that shows users the chosen element's utility and deliveries to human values based on the property scores inputted on this interface.

It has also been shown in the observation that the software is able to provide users the liberty to change the membership functions and the names of linguistic labels. In *Figure 48,* the default linguistic label and membership function of element *Forest of tranquillity*'s property *Species Richness* are inside the green square. Then, in the *Figure 49,* the customized linguistic label and membership function of element *Forest of tranquillity*'s property *Species Richness* are shown in the green square (the linguistic label's name has been changed to low and the membership function has been changed to gaussian function with parameters: 8 and 9). This change of linguistic label name will also be manifested in the readable property-value relationships (rules) (highlighted in green square in *Figure 50.*). It has also been shown in the observation that the software is still capable of calculating the chosen element's utility and deliveries for human values after its linguistic label and membership function are customized (*Figure 51)*.

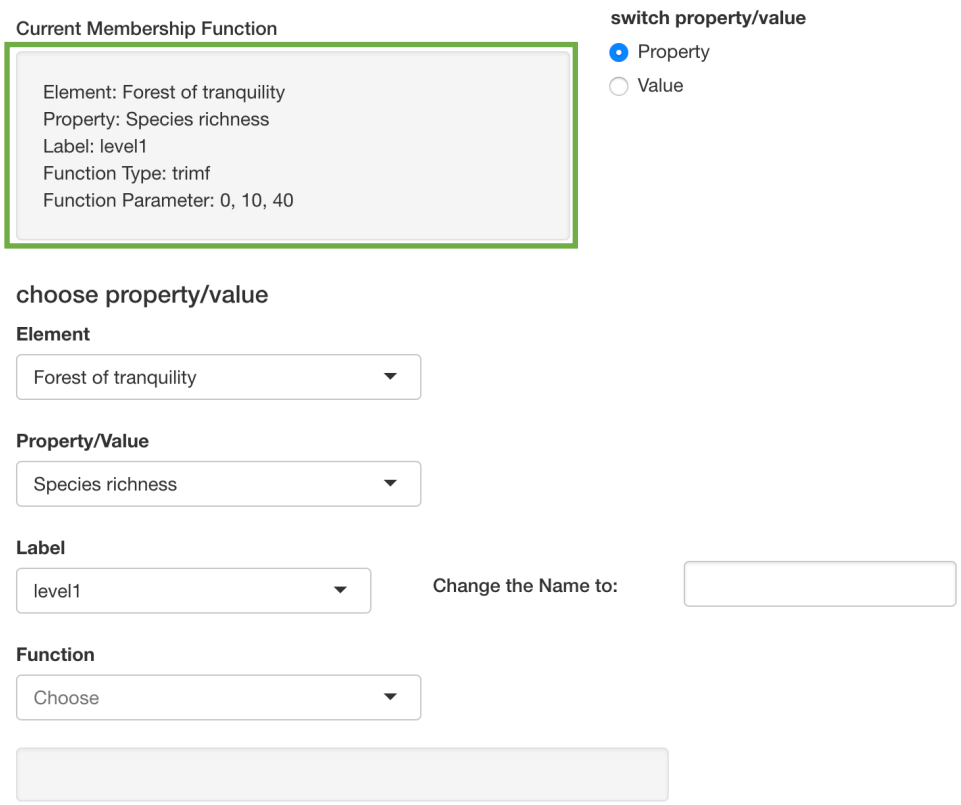


Figure 48. The interface that shows the element *Forest of tranquillity*'s property *Species richness's* one linguistic label's default name and membership function.

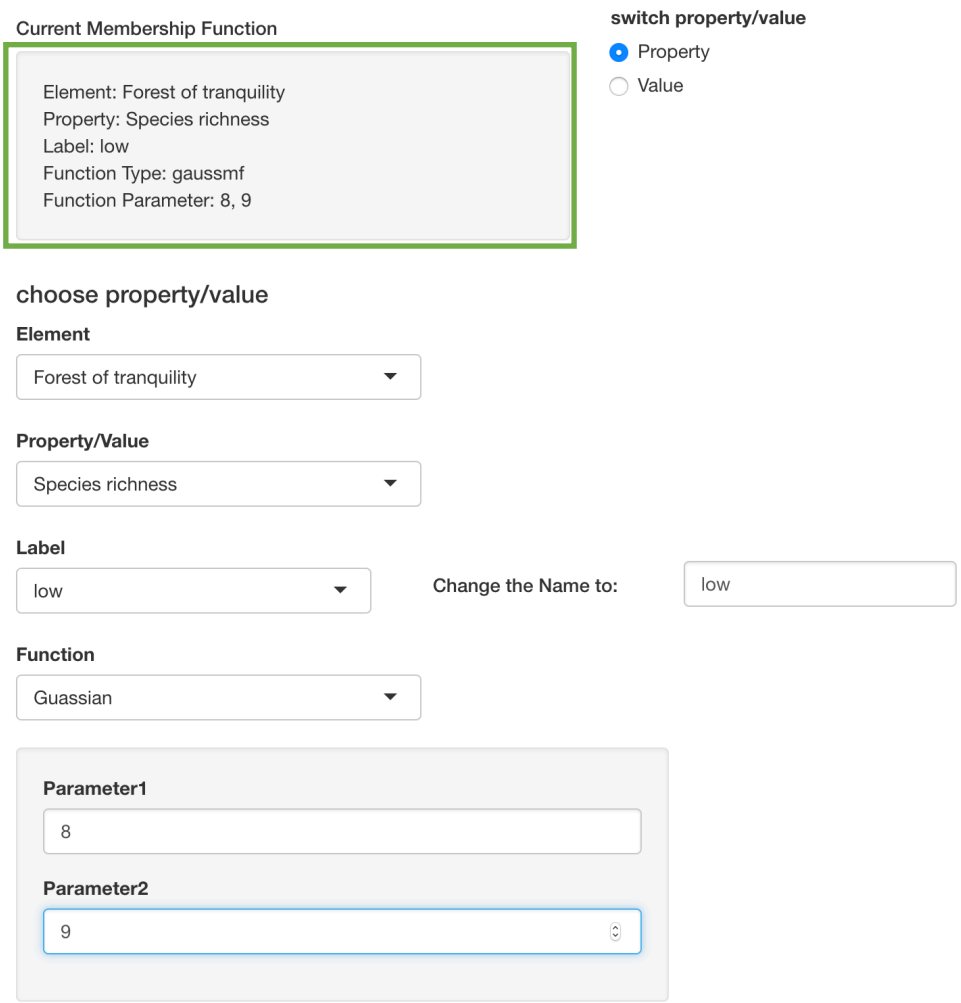


Figure 49. The interface that shows the element *Forest of tranquillity*'s property *Species richness's* one linguistic label's customized name and membership function.

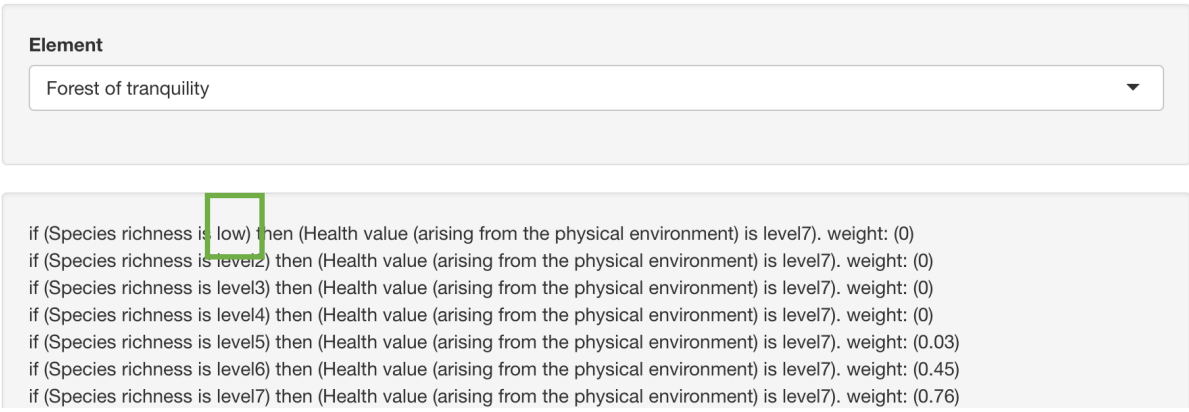


Figure 50. The change of linguistic label's name is manifested in the exhibited rules.

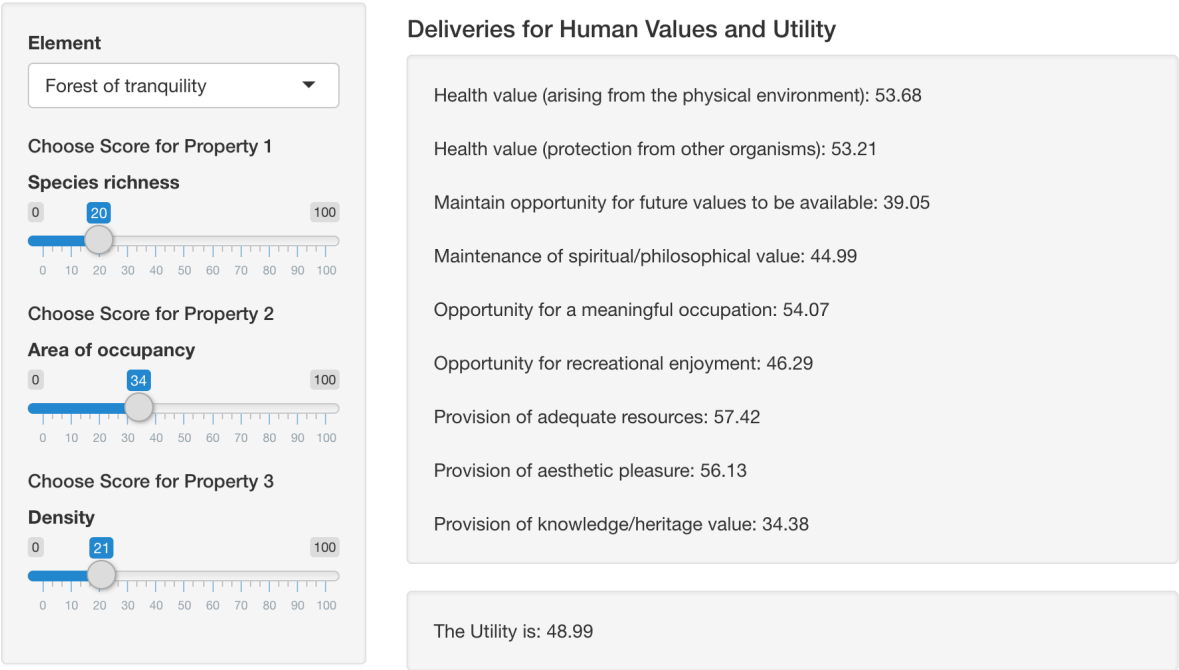


Figure 51. The interface that shows the chosen element's utility and deliveries for human values after the element's linguistic label and membership function are customized.

7.4 Evaluation Summary

The evaluations above have tested the software's capability for supporting step 3 to step 5 of value-drive framework using data from Atlantis Management case.

For the step 4, it can quantify the collected property-value relationships (rules) and present them in summary matrices and linguistic terms which further help users understand these relationships. Based on the observation of this step, it can be said that this software can satisfy the requirements: **1.a, 2.a, 10.a, 1.b, 8.b** and **12.b.**

And for the step 3, it successfully rates the elements and values based on their average maximum importance. Additionally, it provides rich information about rating data using IAA graphs and key values extracted from the uploaded file to further support analysis and interpretations that can benefit the informing decision-making process in environment management. These demonstrate that the software has fulfilled the requirements: **3.a, 4.a, 9.a, 9.b, 10.b** and **11.b.**

Finally, for step 5, the user-chosen element's deliveries for human values and utility can be calculated based on the uploaded files and property scores. Since this is achieved using FIS, it also allows users to customize linguistic labels and membership functions of properties and values to fit the situation of their own cases. These indicate that the software has satisfied the requirements: **5.a, 6.a, 7.a, 2.b, 3.b, 4.b, 5.b, 6.b** and **7.b.**

Since the observations above have exhibited that the software is competent to work with user-uploaded data and user-customized linguistic labels and membership functions, it can be said that the software is equipped with ***flexibility***.

In conclusion, the software has fulfilled all the functional requirements and the non-functional requirement *flexibility.* Its competence for supporting the decision-making process in values-driven framework has been proved.

8 User Review

The reviews are collected from an anonymous potential user who is an expert in environment management. In this section, two aspects of the reviews will be focused: 1) the potential improvements for this software's functionalities. 2) The user's opinions about the software's *usability* and *portability*.

8.1 Functionality Improvement

8.1.1 User Defined Property Score Range

In the current software, the property score ranges for different properties are uniformly defined as 0-100. As mentioned above, the maximum property score for each property is normally defined based on experts' opinions, thus, for each element, the property score ranges of different properties are likely to be different. At the current stage, these different property score ranges are all normalized into the range 0-100 when using the software which will inevitably cause inefficiency for the users. Thus, the potential user who assessed this software suggested in the review that **the software should allow users to define the property score ranges.**

8.1.2 Visualize the Membership Function Change

In the Customization Page, the users can customize the membership functions by changing their function types and associated parameters. As suggested by the invited potential user, since it is difficult to know what is happening when the parameters and function types are changed, **the membership functions should be visualized on the interface when users customize them.**

8.2 User's Review on Usability and Portability

As mentioned in Section 4, the potential user will assess if the software has fulfilled two non-functional requirements: *usability* and *portability*. The reviews from the user will be elaborated in this section.

1) *Usability*: The user mentioned that the non-computer science experts can easily use the software except for the Customization Page. Since this interface will inevitably involve basic knowledge about Fuzzy Inference System and Fuzzy Logic, it has been suggested in the review that **it would be helpful if some useful resources about the basic knowledge needed for this Customization Page can be provided in the User Manual**.
2) *Portability*: In the review, the user stated that this software is easy to launch and access since it is on a server.

8.3 Conclusion

Overall, this invited potential user has provided insightful suggestions about the potential improvements of the software. Based on the user's suggestion that is mentioned in Section 8.2, links of basic Fuzzy Logic and Fuzzy Inference System online tutorials have been added to the User Manual. However, the improvements mentioned in Section 8.1 can only be realized in the future due to the limited time for the project.

9 Summary and Further work

9.1 Project Progress

As stated in the initial proposal, this project can be divided into the following steps:

A. **Complete project proposal and ethics checklist.**
B. **Investigate related researches and literatures to obtain adequate understanding for Fuzzy logic, FuzzyR, Shiny R and value-driven framework.**
C. **Get familiar with the FuzzyR and ShinyR packages.**
D. **Combine these two parts into a functional integrated system.**
E. **Find appropriate case that can be used to run on the integrated system.**
F. **Collect case-related data.**

G. **Run the case on the system and obtain results.**
H. **Evaluate and discuss the results.**
I. **Review the project and produced results to fix problems and deficiencies.**
J. **Write final report**
K. **Prepare demonstration.**

The timeline for completing these tasks are listed below as well as the results obtained from each stage:

1. **Step A: 8/10/2019 – 23/10/2019**

Went as Planned

***Result*:** During this time, I had multiple meetings with my supervisor to discuss the plans for the project. Based on the results of these meetings, I finished the project proposal and ethic check list for this project.

2. **Step B to Step C: 9/10/2019 – 21/12/2019**

Went as Planned

***Result*:** Since before this project, I barely had any understanding about Fuzzy Logic or the language R, I invested a relatively long time on investigating and learning these two areas through researches and literatures. I also familiarized myself with the FuzzyR and ShinyR packages and used them to produce a software prototype. This prototype could only work with one element with two properties. But it could calculate the utility of this element based on users' uploaded rules and It also allowed users to customize the membership functions and linguistic labels. The future software development was based on this prototype.

3. **Step D: 27/01/2020 – 20/03/2020**

Went as Planned

***Result*:** At the beginning of the second semester, I started to develop the software based on the prototype that I finished in the first semester. This process took about two months. At the end, a functional software was generated. But in this version of the software, the users could not change the names of the linguistic labels and there were still some errors in the generated IAA graphs. Also, at this stage, the software was still a local application without being published on server.

4. **Step E to Step H: 27/01/2020 – 20/03/2020**

Change of Plan

***Original Plan*:** In the original plan, these four steps would be finished during **28/10/2019 - 20/03/2020.**

***Result*:** These four steps were carried out parallelly with step D to evaluate the software during its development. In the initial plan, the collection of stakeholders' and experts' opinions would be carried out by myself through workshops due to the fact that I did not have any data that could be utilized to evaluate the software. Fortunately, I was later informed that Dr Michael Smith from Australian Wildlife Conservancy could provide data from Atlantis Management case for me to evaluate the software. Thus, the time for these four steps have significantly decreased.

5. **Step I: 21/03/2020 – 10/04/2020**

Delay in Time

Delay: The deadline for this step is delayed to **30/04/2020.**

***Result*:** In this period, a potential user of this software was invited to give suggestions about this software. Before sending the software to this potential user, I reviewed and improved it to provide users the liberty to change the names of linguistic labels. I also fixed the errors in IAA graphs and published the software on the 'shinyapps.io' server. The time I spent on these modifications was longer than I expected. Thus, the deadline for this step was delayed for 20 days.

6. **Step J to Step K: 01/04/2020 – 20/04/2020**

Delay in Time

Delay: The deadline for this step is delayed to **04/05/2020** due to covid-19 situation.

***Result*:** During this time, I focused on finishing the final report and preparing the video for demonstration.

Conclusion: As demonstrated above, a couple changes have been made to the original plan due to multiple factors. But, overall, the project went very well as planned.

9.2 Further Work

The evaluation in Section 7 shows that this software has fulfilled all the functional requirements and it is competent to support the values-driven framework for natural resource management. It also achieved the non-functional requirement *flexibility* and realized the innovation stated in Section 1.3 by allowing users to customize the software through uploading files and changing linguistic labels and membership functions. Its *usability*

and *portability* are also approved by the invited potential user in Section 8. Yet, further improvements can still be made to the software.

Apart from the improvements suggested by the potential user in Section 8, the software can also be improved to work with more than one environment management case without having to be rebooted. At this moment, the software can only process the information from one environment management case. For example, if the users have uploaded the files from environment management case A, then they need to reboot this software if they want to use it for case B. One possible way for realizing this improvement is to update the *total_functions_input, total_functions_output* and *total_rules* variables every time the users upload a file.

These improvements are undoubtfully important and insightful. However, due to the limited time for this project, these modifications can only be achieved in the future development.

9.3 Conclusion and Reflection

Conclusion: In this project, a software that can be used to facilitate the value-driven framework in environment management has been developed. As mentioned above, the design and implementation of the software have strictly followed its functional and non-functional requirements. Although there are still some potential improvements that can be applied to the software in the future development, at current stage it has attained the aim of the project. Thus, it can be concluded that this project is overall a success.

Personal Reflection: Over the course of the project, there are two main challenges:

1) Understand the theoretic knowledge related to this project.

This project involves the knowledge about Fuzzy Logic, environment management framework and R which are three fields that I was not familiar with at the beginning of the project. Gaining knowledge about these three fields has been a challenging but rewarding task throughout the project. Even though it was not easy for me to learn all this information in such limited time, I still managed to gain adequate amount of knowledge for this project. Personally, this experience has largely improved my self-learning skills.

2) Develop a software in R.

This project is my first opportunity to develop a software using the language R. The coding process was not easy, and I have faced and overcome many difficulties along the way. But I took this as a great chance for me to familiarize myself with a new language and practice my software development skills.

Overall, during this project, I have had my fair share of obstacles and victories. As my first individual dissertation, I think this whole experience has largely benefited and prepared me for future academic study and industry career.

Bibliography

[1] MEA (Millennium Ecosystem Assessment), 2005, MEA (Millennium Ecosystem Assessment) Ecosystems and Human Well-being: Biodiversity Synthesis World Resources Institute, Washington, DC (2005)

[2] S. M. Redpath, J. Young, A. Evely, W. M. Adams, W. J. Sutherland, A. Whitehouse, A. Amar, R. A. Lambert, J. D.C. Linnell, A. Watt, and R.J. Gutierrez, “Understanding and managing conservation conflicts”, Trends in Ecology & Evolution, Volume 28, Issue 2, Pages 100-109, 2013

[3] R. Woodroffe, S. Thirgood, A. Rabinowitz, “People and Wildlife, Conflict or Co-existence?”, Cambridge University Press, 2005

[4] F. Madden, B. McQuinn, “Conservation’s blind spot: The case for conflict transformation in wildlife conservation”, Biological Conservation, vol 178, pp. 97-106, 2014

[5] G. M. Mace, “Whose conservation? Changes in the perception and goals of nature conservation require a solid scientific basis”, Science， Vol. 345, Issue 6204, pp. 1558-1560， 2014

[6] M. J. Smith, C. Wagner, K. J. Wallace, A. Pourabdollah and L. Lewis, “The contribution of nature to people: Applying concepts of values and properties to rate the management importance of natural element”, Journal of Environmental Management, vol. 175, pp. 76-86, 2016

[7] Wallace, 2012, K. J. Wallace, "Values: drivers for planning biodiversity management," Environmental Science & Policy, vol. 17, pp. 1-11, 2012.

[8] K. J. Wallace, “Classification of ecosystem services: Problems and solutions”, Biological conservation, vol. 139, pp. 235–246, 2007

[9] K. J. Wallace, C. Wagner and M. J. Smith, “Eliciting human values for conservation planning and decisions: A global issue”, Journal of Environmental Management, vol. 170, pp. 160-168, 2016

[10] R. Gregory, L. Failing, M. Harstone, G. Long, T. McDaniels, D. Ohlson, “Structured Decision Making: A Practical Guide to Environmental Management Choices”, Wiley-Blackwell, 2012

[11] C. Margules, M. B. Usher, “CRITERIA USED IN ASSESSING WILDLIFE CONSERVATION POTENTIAL: A REVIEW”, Biological Conservation, vol. 21, pp. 79-109,1981

[12] M. S. Reed, “Stakeholder participation for environmental management: A literature review”, Biological Conservation, vol. 141, pp. 2417–2431, 2008

[13] A. Pourabdollah, C. Wagner, M. Smith and K. Wallace, "Real-World Utility of Non-Singleton Fuzzy Logic Systems: A Case of Environmental Management", IEEE International Conference on, Fuzzy Systems (FUZZ-IEEE) (2015), pp. 1-8, 2015

[14] C. Wagner, S. Miller, J. M. Garibaldi, D.T. Anderson, T. C. Havens, "From Interval-Valued Data to General Type-2 Fuzzy Sets", IEEE TRANSACTIONS ON FUZZY SYSTEMS, vol. 23, 2015

[15] M. Smith, "Managing Atlantis A Natural Resource Management Plan for the Phoenix Catchment"

[16] "Fuzzy Inference Process", MathWorks, The MathWorks, Inc. [online]. Available: https://uk.mathworks.com/help/fuzzy/fuzzy-inference-process.html

[17] C. Wagner, M.J. Smith, K.J. Wallace, "Generating uncertain Fuzzy Logic Rules from surveys. Capturing subjective relationships between variables from human experts", International Conference on Fuzzy Systems, Institute of Electrical and Electronics Engineers, Hong Kong (2015)

[18] S-M. Chen, C-M. Huang, "Generating weighted fuzzy rules from relational database s2ystems for estimating values using genetic algorithms", IEEE Trans. Fuzzy Syst., vol. 4, no. 4, pp. 495-506, 2003.

[19] F. Herrera, M. Lozano and J. L. Verdegay, "Generating Fuzzy Rules from Examples using Genetic Algorithms," Fuzzy Logic and Soft Computing, World Scientific, pp. 11-20, 1995.

[20] I. Sommerville, "Software Engineering", Eighth ed., Glasgow: Pearson, 2015.

[21] "Advantages of Web Applications", Paul Stanley Software [online]. Available: https://www.pssuk.com/AdvantagesWebApplications.aspx

[22] "Shiny from R Studio", RStudio [online]. Available: https://shiny.rstudio.com

[23] C. Chen, J. Garibaldi, T. Razak, "FuzzyR: Fuzzy Logic Toolkit for R", CRAN, 2019

[24] GitHub [online]. Available: https://rstudio.github.io/DT/

[25] "Create an plot or image output element", Shiny from R studio, RStudio [online] Available: https://shiny.rstudio.com/reference/shiny/latest/plotOutput.html

[26] "plot", RDocumentation [online]. Available: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot

[27] "Create a page with a top level navigation bar", Shiny from R studio, RStudio [online]. Available: https://shiny.rstudio.com/reference/shiny/1.0.5/navbarPage.html

[28] "Create a tab panel", Shiny from R studio, RStudio [online]. Available: https://shiny.rstudio.com/reference/shiny/0.12.1/tabPanel.html

[29] "Deploying Shiny apps to the web", Shiny from R studio, RStudio [online]. Available: https://shiny.rstudio.com/articles/deployment-web.html

[30] J. J. Allaire, "Deployment Interface for R Markdown Documents and Shiny Applications", CRAN, 2019

[31] A. Pourabdollah, C. Wagner, S. Miller, M. Smith and K. Wallace, "Towards Data-Driven Environmental Planning and Policy Design – Leveraging Fuzzy Logic to Operationalize a Planning Framework", 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 2230-2237