The Development of a Bilingual Fuzzy Expert System Shell

Hassan Mathkour, Israa Al-Turaiki and Ameur Touir
Department of Computer Science
King Saud University
Riyadh, Saudi Arabia
mathkour@ccis.ksu.edu.sa, binmathkour@yahoo.com, touir@ksu.edu.sa

(Received 21/12/2008; accepted for publication 02/03/2009)

Abstract. Fuzzy logic has been incorporated in many expert systems to solve real world problems that are inherently ambiguous. With fuzzy logic it is possible to program human intuition through the development of fuzzy expert system shells. A fuzzy expert system shell is a tool that helps build expert systems to manage fuzzy problems. Commercial as well as non-commercial fuzzy expert system shells are available. These shells provide variety of functions to facilitate the development of fuzzy expert systems for real world problems in different application areas such as medicine, engineering, and finance. To the best of our knowledge, none of the available fuzzy shells is natively developed for the Arabic language. This paper describes the development and the experimentation of a bilingual fuzzy expert system shell. This shell is intended to be a research tool for fuzzy expert systems developers in bilingual environments similar to those in the Arab world where users and developers use multi-languages due to their educational backgrounds and working environments. The shell processes fuzzy terms of the Arabic language as well as the English language. The shell applies implication methods that bear resemblance to human intuition. In the process of the development, a comparison of various fuzzy expert system shells has been performed to identify strengths and weaknesses of available shells. Experiments with our shell are reported and its performance is compared to existing shells that use different implication methods.

Keywords: Expert system shells, knowledge based systems, Fuzzy Implication methods, Bilingual Systems.

1. Introduction

Expert system shells are versatile tools that are used to create expert systems. Fuzzy expert system shells have been developed to allow for reasoning that deals with crisp and fuzzy sets. These shells allow incorporation and manipulation of imprecise information using fuzzy set theory developed by Zadeh (Zadeh, 1965). They are used to create expert systems that can handle imprecise situations effectively. The ability to operate under imprecise environment makes expert systems closely behave like human being and provides a natural representation of people's daily terminologies. The ability of treating ambiguities, in a manner similar to human experts, makes expert systems versatile and adaptable to unforeseen circumstances which are difficult to avoid in real life applications. This has made fuzzy logic a suitable means to deal with the fuzziness of data and knowledge frequently encountered in the terminologies of human experts when developing knowledge based systems (Kelmet and Slany, 1993).

There have been attempts to design fuzzy expert system shells for large-scale general-purpose as well as domain specific applications (Philip, 1991; Aly and Vrana, 2006). Over the years, a large number of expert system shells have been developed and several of them are commercially available. JFK (López-Ortega, 2006), FuzzyShell (Pan, 1996), FuzzyJess (Orchard, 2001), FuzzyCLIPS (Orchard, 2004), FLINT (Shalfield, 2005), FLOPS (Siler and Buckley, 2005), Fuzzy Logic (Mathworks, 1999), and FuzzyJ toolkit (Council, 2001; Orchard, 2001) are examples of expert system shells. We have analyzed several of the existing shells in an attempt to indentify a shell having features that natively supports application development in Arabic language while allowing for application development in other languages. We searched for a shell that accommodates for Arabic fuzzy terms naturally and which employ intuitional inference methods. Our unsuccessful endeavor and realizing that making such a shell available will be useful for bilingual developers and users in research and educational environments motivated us to design and implement a fuzzy shell with Arabic/English support.
In the process, we have found it helpful to furnish a comparison for a set of the available fuzzy shells. These shells differ from one another in several aspects. For example, most of the shells implement inference methods that are mentioned in (Zadeh, 1975; Mamdani, 1977) while many (Fukami, 1980; Mizumoto, 1981; Mizumoto, 1982) have advocated that the methods that are based on the interpretation given in (Mizumoto et al., 1979; Mizumoto et al., 1979; Mizumoto et al., 1979) perform better as they induce human intuition. In this research, we have taken the interpretation that is supported in (Mizumoto et al., 1979; Mizumoto et al., 1979; Mizumoto et al., 1979; Mizumoto, 1981). Our work in (Mathkour et al., 2009) introduces an Arabized fuzzy expert system shell. In this paper, we present the development of a bilingual (Arabic/English) fuzzy expert system shell, which is an extension of our work in (Mathkour et al., 2009), to allow for both the Arabic and English languages. We also report on experiments with the shell using real life data to demonstrate and analyze its human-like behavior using the selected inference methods. To measure its effectiveness, we have compared its performance with some of the available shells. We report on the experiments and comparison of our shell with FuzzyClips (Orchard, 1996) and FuzzyJ (Council, 2001; Orchard, 2001).

The objective of our extended shell is to provide a comprehensive tool that is intended to be a research tool for fuzzy expert systems developers in multi-lingual environments similar to those in the Arab world where users and developers use multi-languages due to their educational backgrounds and working environments. It is a general purpose shell that is based on the implication methods: $R_s$, $R_g$, $R_{gg}$, $R_{sg}$ and $R_{ss}$ (Fukami, 1980; Siler and Buckley, 2005; Mizumoto et al., 1979; Mizumoto et al., 1979; Mizumoto, 1981).

It is also observed that many shells use dedicated programming languages for the expert system application development. Consequently, application developers are required to learn the programming languages that are supported by these shells. Learning a new programming language is not a desired requisite, especially for those who do not have a programming aptitude. Learning a new programming language distracts developers from their main objective of developing expert systems in their specific domains. In our shell, we have used a visual environment by adopting a simple graphical user interface. The interface supports both Arabic as well as English languages and it can be tailored for other languages by adding the user interface support for the required language.

In (Mathkour et al., 2009), we developed comparison criteria to evaluate aspects of available expert system shells. The criteria include evaluation of end-user interface, developer Interface and availability and installation of shell. In this paper, we further discuss these criteria and employ them to formulate comparison tables of a larger number of existing expert system shells.

The rest of the paper is organized as follows: Section 2 presents a comparison of twenty expert system shells along with brief description of the comparison criteria. Section 3 presents the developed fuzzy shell, describes the implication methods, and the implementation. Section 4 presents experimentation with the system. Section 5 presents a comparison of our shell to some existing ones. Section 6 concludes the paper.

2. Comparison of Existing Shells

We have endeavored to compare the features of twenty shells of those available commercially and otherwise. These include Fuzzy Logic (Mathworks, 1999), JFK (López-Ortega, 2006), FuzzyJess (Orchard, 2001), FuzzyCLIPS (Orchard, 2004), FuzzyShell (Pan, 1996), FLINT (Shalfield, 2005), FLOPS (Siler and Buckley, 2005), CLIPS (Giarratano, 1998), Jess (Friedmann-Hill, 1999), Flex (Vasey, 1996), PSS (Forgy, 1981), ESB (Kent and Denholm, 1990), ESBBuilder (Ishihara et al., 1995), and FuzzyJ toolkit (Council, 2001; Orchard, 2001). First we present a discussion of the comparison criteria, then present the results of our comparison in Table 8.

2.1 End-user interface

The user interface is an important component of any software development tool as it allows interaction between application developers and the tool. The user interface must be natural in the context applications that are being developed thereby releasing the developers from learning extraneous concepts and focusing on the development issues.
The quality of the user interface is judged by its ease of use and naturalness. The following features are indicators of the quality of an interface:
1. Explanation facilities: This is used to explain the process through which the system has arrived at a decision.
2. User friendliness: This is judged by the quality of graphical user interface components such as menus, buttons, and usage of a natural language.
3. The ability to change the earlier answers without having to repeat the session from the beginning.

2.2 Developer interface
The expert system developers enter their knowledge through the rule editor. The rule editor should support the rule type selection and creation, rule change and update process, mathematical operations to implement the inference engine strategies, built-in member functions, defuzzification methods, certainty factor handling, error correction, and fact refinement and documentation. In addition to these, the rule editor must have provisions to interact with external environments like DBMSs, Spread sheets and Programming in modern languages like Java and C#. Features related to the rule editor are shown in Table 1 to Table 5 with their respective weights.

2.3 Procurement and installation
The availability of these tools could be problematic in some linguistic regions of the world. Once available, their installation is not always straight forward. Hence we have used it as an evaluation factor. Table 6 and 7 shows the weight assigned to measure the ease of procurement and installation.

The proposed Bilingual Fuzzy Expert System Shell

The entry point to the system provides the users with the option of building expert systems using Arabic or English knowledge bases (Fig. 1). Upon selection, Arabic or an English screen portraying the main components of system is displayed (Fig. 2.a and 2.b). The main components of the shell are the variable editor, rule editor, and the inference engine.

3.1 The variable editor
The variable editor’s main purpose is to provide functions to create, edit, and delete fuzzy variables, their fuzzy values, membership functions, and universe of discourse. The layout of our variable editor is shown in Fig. 3.a and 3.b. The variable editor can be launched from the menu button of “Variable Editor” “محرر المتغيرات” in Fig. 3. Created variables and their properties can be seen from a dropdown menu.
Table 8. Shell companion

| End-user Interface | Natural/Shell | Fuzzy Facts | Fuzzy If/Then | Fuzzy Logic Module | Defuzzification | Rule Shell | Fuzzy Inference | Fuzzy Shell | FLIN | CLI | Java | Flex | F & B | F & E | ESE | ESE | CGI | ULB | DES |
|--------------------|---------------|-------------|---------------|-------------------|-----------------|-------------|----------------|-------------|------|-----|------|------|-------|-------|-----|-----|-----|-----|------|-----|
|                    | Natural Shell | 1           | 1             | 1                 | 1               | 1           | 1              | 1           | 1    | 1   | 1    | 1    | 1     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
|                    | Membership    | 1           | 1             | 1                 | 1               | 1           | 1              | 1           | 1    | 1   | 1    | 1    | 1     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
| Changing Answer    |               | 5           | 5             | 5                 | 5               | 5           | 1              | 1           | 1    | 1   | 1    | 1    | 1     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
| Rule Type          |               | 3           | 5             | 5                 | 5               | 5           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 1   | 1   | 1   | 1   | 1    | 1   |
| Rule Chaining      |               | F & B       | F             | F                 | F               | F           | F              | F           | F    | F   | F    | B    | F & B | F & B | F   | ESE | F   | ESE | NA   | NA  |
| Math possibilities |               | 5           | 5             | 5                 | 5               | 5           | 3              | 3           | 3    | 3   | 3    | 3    | 3     | 1     | 3   | 1   | 1   | 1   | 1    | 1   |
| Fuzzy Inference    |               | 3           | 3             | 3                 | 3               | 3           | 3              | 3           | 3    | 3   | 3    | 5    | 3     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
| Fuzzy membership   |               | 3           | 3             | 3                 | 3               | 3           | 3              | 3           | 1    | 1   | 1    | 1    | 1     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
| Membership         |               | 5           | 5             | 5                 | 5               | 5           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| Fuzzy             |               | 5           | 3             | 1                 | 5               | 5           | 3              | 3           | 1    | 1   | 1    | 1    | 1     | 1     | 3   | 1   | 1   | 1   | 1    | 1   |
| Defuzzification    |               | 5           | 5             | 1                 | 1               | 1           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| Fuzzy             |               | 5           | 5             | 1                 | 1               | 1           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| Decision Factors  |               | 3           | 3             | 3                 | 3               | 3           | 3              | 3           | 3    | 3   | 3    | 3    | 3     | 1     | 3   | 1   | 1   | 1   | 1    | 1   |
| Rating Tools       |               | 1           | 1             | 1                 | 1               | 1           | 1              | 1           | 1    | 1   | 1    | 1    | 1     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
| Debug Tools        |               | 1           | 1             | 1                 | 1               | 1           | 1              | 1           | 1    | 1   | 1    | 1    | 1     | 1     | 1   | 1   | 1   | 1   | 1    | 1   |
| Documentation      |               | 3           | 5             | 5                 | 5               | 5           | 5              | 5           | 1    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| External program   |               | 3           | 5             | 5                 | 5               | 5           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| Integration        |               | 3           | 5             | 5                 | 5               | 5           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| Document          |               | 3           | 5             | 5                 | 5               | 5           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |
| Installation       |               | 1           | 5             | 1                 | 1               | 1           | 5              | 5           | 5    | 5   | 5    | 5    | 5     | 5     | 5   | 5   | 5   | 5   | 5    | 5   |

Mallah et al.: The Development of a Bilingual Fuzzy Expert System Shell
3.2 Rules and Rule-base Editor

For the rule-base editor permits application developers to create, edit and delete rules. The rules are of the form IF antecedents Then consequents. A rule may have more than one antecedent and one consequent. Also, the editor allows the “Else part” in the rules. We use rules as our knowledge representation scheme because they are natural in representing expert knowledge, and they are easier to understand, modify, and maintain.

The rule editor can be launched from the menu entitled “KB Editor” on the menu bar of Fig. 2. Figure 4 shows the layout of the Knowledge base editor at the creation of a new knowledge base. A new knowledge base is created using “New KB” in the menu bar of Fig. 2.
3.3 Inference Engine

The inference engine uses implication, composition, aggregation and linkage as given in (Leung and Lam, 1988; Aly and Vrana, 2006; Bandler and Kohout, 1980), and briefly described in the following subsections. It is the part of the knowledge based system that is responsible for deriving conclusions from existing data, i.e., deriving new knowledge from existing ones.

3.3.1 The implication methods

Our shell has a backward inference engine and uses the implication methods discussed in (Fukami et al. 1980; Mizumoto et al., 1979; Mizumoto et al., 1979; Mizumoto et al., 1979, Mizumoto, 1981), namely, Rs, Rg, Rgs, Rgg, Rsg, and Rss. Details of the inference methods are found in (Mizumoto et al., 1979; Mizumoto et al., 1979). The choice of the inference methods is based on the observation that such methods closely mirror the human intuitions as compared to those in (Zadeh, 1999; Zadeh, 2006; Zadeh, 1975; Mamdani, 1977). This has been advocated in previous work (Mizumoto, 1981; Mizumoto, 1982). The shell allows the user to either use all the implication methods or select one of them. Figure 5 shows a screen shot of the working of the system when conclusion is obtained using the Rs implication method. The conflict resolution strategy used in our shell is the most specific strategy.

A fuzzy inference method needs to satisfy the criteria shown in Table 9, in order to resemble human intuition (Fukami et al, 1980; Mizumoto, 1981; Mizumoto, 1982). The inference methods presented in (Zadeh, 1994) do not satisfy the criteria in Table 9, except Criterion IV-1. The inference methods in (Mamdani, 1977) on the other hand satisfy Criterion I and II-2. Criterion II-2 is applicable when there is no strong relation between “x is A” and “y is B”. In criterion IV-1, information about y cannot be inferred from the conditional inference “if x is A then y is B” when “x is not A”. Details of related issues are found in (Bandler and Kohout, 1980; Willmott, 1980; Mizumoto, 1981, Mizumoto, 1982).

Following the criteria in Table 9, fuzzy inferences can be classified into the following four types. Illustration of criteria that are satisfied by the implication methods Rs, Rg, Rgs, Rgg, Rsg, and Rss is given in Table 10 (Fukami, 1980; Mizumoto, 1981; Mizumoto, 1982).

- **Type 1:** The binary relation between the antecedent A and the consequence B is translated into $R_s(A, B)$. In type 1 inference, Criteria I, II-1, III, IV-1 are satisfied.
Table 9. Fuzzy inference criteria

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Ant 1: if x is A then y is B</th>
<th>Ant 2: x is A</th>
<th>Cons: y is B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criterion II-1</td>
<td>Ant 1: if x is A then y is B</td>
<td>Ant 2: x is very A</td>
<td>Cons: y is very B</td>
</tr>
<tr>
<td>Criterion II-2</td>
<td>Ant 1: if x is A then y is B</td>
<td>Ant 2: x is very A</td>
<td>Cons: y is B</td>
</tr>
<tr>
<td>Criterion III</td>
<td>Ant 1: if x is A then y is B</td>
<td>Ant 2: x is more or less A</td>
<td>Cons: y is more or less B</td>
</tr>
<tr>
<td>Criterion IV-1</td>
<td>Ant 1: if x is A then y is B</td>
<td>Ant 2: x is not A</td>
<td>Cons: y is unknown</td>
</tr>
<tr>
<td>Criterion IV-2</td>
<td>Ant 1: if x is A then y is B</td>
<td>Ant 2: x is not A</td>
<td>Cons: y is not B</td>
</tr>
</tbody>
</table>

- **Type 2:** The binary relation between the antecedent A and the consequence B is translated into $R_g(A, B)$. In type 2 inference, Criteria I, II-2, III, IV-1 are satisfied.
- **Type 3:** The binary relation between the antecedent A and the consequence B is translated into $R_g(A, B)$. In type 3 inference, Criteria I, II-1, III, IV-2 are satisfied.
- **Type 4:** The binary relation between the antecedent A and the consequence B is translated into $R_g(A, B)$. In type 4 inference, Criteria I, II-2, III, IV-2 are satisfied.

An Rs implication example:
For the rule, If x is small then y is middle, where $U=V=0+1+2+3+4+5+7+8+9+10$, $A=small=0.2/0+0.8/1+0.6/2+0.4/3+0.2/4$, and $B=middle=0.2/2+0.4/3+0.8/4+1/5+0.8/6+0.4/7+0.2/8$, $R_d(A,B)$ is given in the Table 11.

3.3.2 Composition
A fuzzy composition relation $R(A,B)$ of R1 and R2 is simply the relation obtained by applying R1 and R2 one after the other. The most frequently used

Table 10. Criteria satisfied by each implication method

<table>
<thead>
<tr>
<th>Ant 2</th>
<th>Cons</th>
<th>$R_s$</th>
<th>$R_g$</th>
<th>$R_{sg}$</th>
<th>$R_{gg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Very A</td>
<td>Very B</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Very A</td>
<td>B</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>More or less A</td>
<td>More or less B</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Not A</td>
<td>Not B</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 11. $R_d(A,B)$

<table>
<thead>
<tr>
<th>V</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
composition operator in fuzzy logic is the Max-Min composition operator in (Zadeh, 1999; Zadeh, 1975) and it is the one we used in our shell.

**Max-Min Composition**

Let R be a fuzzy relation in \( X \times Y \), and S be a fuzzy relation in \( Y \times Z \). The Max-Min composition of R and S, \( R \circ S \), is a fuzzy relation in \( X \times Z \) such that

\[
R \circ S = \mu_{R \circ S}(x,z) = \cup \{ \mu_R(x,y) \land \mu_S(y,z) \}
\]

A Max-Min composition example:

Suppose we have the following two relations R and S:

<table>
<thead>
<tr>
<th></th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.4</td>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>x2</td>
<td>0.9</td>
<td>1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>z1</th>
<th>z2</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>y2</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>y3</td>
<td>0</td>
<td>0.6</td>
</tr>
</tbody>
</table>

\[
R \circ S = \begin{bmatrix}
0.4 & 0.6 & 0 \\
0.9 & 1 & 0.1 \\
\end{bmatrix} \circ \begin{bmatrix}
0.5 & 0.8 \\
0.1 & 1 \\
0 & 0.6 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.4 & 0.6 & 0 \\
0.9 & 1 & 0.1 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\text{max}(\text{min}(0.4,0.5), \text{min}(0.6,0.1), \text{min}(0,0)) = \text{max}(0.4,0.1,0) = 0.4 \\
\text{max}(\text{min}(0.4,0.8), \text{min}(0.6,1), \text{min}(0,0.6)) = \text{max}(0.4,0.6,0) = 0.6 \\
\text{max}(\text{min}(0.9,0.5), \text{min}(1,0.1), \text{min}(0.1,0.1)) = \text{max}(0.5,0.1,0) = 0.5 \\
\text{max}(\text{min}(0.9,0.8), \text{min}(1,1), \text{min}(0.1,0.6)) = \text{max}(0.8,1,0.1) = 1
\end{bmatrix}
\]

**3.3.3 The Aggregation and link operators**

An aggregation operator is needed when a rule has k conditions. The rule is decomposed into k implications. Each implication is used separately to infer a value by applying the fuzzy implication. The values are then aggregated using the aggregation operators used in the rule including OR and AND. The final result is obtained after a MAX operation over the corresponding values inferred by all the rules or fuzzy membership functions (Zadeh, 1975; Fukami, 1980; Mizumoto, 1981; Mizumoto1982; Zadeh, 1999).

**3.4 Implementation Issues**

Similar to that in (Mathkour et al., 2009), the main data structures used in the implementation of the shell are arrays. Since all the implication methods used here depend on the \( R_s \) and \( R_g \) operations, there was a need to implement \( R_s \) and \( R_g \) operations as separate methods. Both methods accept two parameters and return the result after performing \( R_s \) or \( R_g \) implication. Each of the six implications was implemented as methods that accept two matrices and return the result after performing the implication of the whole matrices.

**4. Experimental Results**

The purpose of this section is to illustrate that our fuzzy expert system shell produces correct results and the implication methods used satisfy the criteria mentioned in Table 9. The knowledge base used in this experiment is taken from (Ganoud, et al. 2005). In (Ganoud, et al. 2005), the authors study the influence of random factors on the planning of building works. Deciding the exact period of building projects is a very difficult job due to the fact that building projects are affected by different unpredictable factors. The random factors which are studied include three factors: the cessation of machines, the absence of some professionals, and the influence of weather condition. The rules are given in Table 12 and the membership functions are given in Figures 6 to 9 (Ganoud, et al. 2005).
The knowledge base consists of 27 fuzzy rules. All the rules have the same number of antecedents. The same fuzzy variables are used in all the 27 rules as well as the same conclusion (target). The shell was run several times, each time with different observation. The observations are:

- Observation 1: X is low, Y is high and Z is high.

It is observed (as illustrated in Figure 10) that using $R_g$, $R_{gs}$ and $R_{gg}$ has given the expected results according to the Table 9. On the other hand, using the implications $R_s$, $R_{sg}$, and $R_{ss}$ we found out upon examining the resulting matrices that the results were not exact as expected but were very close to what was expected.
Observation 2: X is not low, Y is not high and Z is not high.

It is observed (as illustrated in Figure 11) that all the implications have given the expected results according to the Table 9 except for $R_{gs}$ and $R_{ss}$. Examining the matrices that resulted from using $R_{gs}$ and $R_{ss}$ we have found that they were very much close to the matrix of the expected result which is "not high". The result in words should have been "more or less not high", but because the shell is not designed to handle composite hedges the result matrix was translated to "more or less low".

Observation 3: X is very low, Y is very high and Z is very high.

It is observed (as illustrated in Figure 12) that the implications $R_{g}$, $R_{gs}$ and $R_{gg}$ have given the expected results according to Table 9. On the other hand, the implications $R_{s}$, $R_{sg}$ and $R_{ss}$ have given different results than expected. Upon examining the matrices that resulted from using $R_{s}$, $R_{sg}$ and $R_{ss}$ we found that they were very much close to the matrix of the expected result which is "very high". The result in words should have been "more or less very high", but because the shell is not designed to handle composite hedges the result matrix was translated to "more or less high".

Observation 4: X is more or less low, Y is more or less high and Z is more or less high.

It is observed (as illustrated in Figure 13) that the implications $R_{s}$, $R_{sg}$ and $R_{ss}$ all have given the expected results according to Table 9. But for the rest of the implication methods the results are close to the expected.

5. Comparison with Existing Tools

We demonstrate that the proposed fuzzy shell performs in a natural manner that mirrors the human inferences of real world problems and yields the expected conclusions that conform to human possible conclusions as compared to other tools. The comparison is made with tools that use the inference methods in (Mamdani, 1977) which are commonly
used in commercial fuzzy expert system shells such as FuzzyClips, and with tools such as FuzzyJ Toolkit that allows for different inference methods including those discussed in (Aly and Vrana, 1977; Mamdani, 1977; Mizumoto, et al, 1979). We examine the performance of both FuzzyClips and FuzzyJ Toolkit shells and compare their results with that of our shell using the four observations detailed in Section 4. For this purpose, the English-translation of the data in Table 12 is used.

5.1 A comparison with fuzzyCLIPS

Mamdani’s inference methods are the most commonly used in commercial fuzzy expert system shells. It has been observed that inference methods in (Mamdani, 1977) do not satisfy human intuition (Fukami et al. 1980; Mizumoto, 1982). We demonstrate this observation by examining the behavior of FuzzyCLIPS with the rules and fuzzy variables of discussed in Section 4.

Table 12. The only criteria satisfied by mamdani's methods

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Ant1: IF x is A Then y is B</th>
<th>Ant2: x is A</th>
<th>Cons: y is B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criterion II-2</td>
<td>Ant1: IF x is A Then y is B</td>
<td>Ant2: x is very A</td>
<td>Cons: y is B</td>
</tr>
</tbody>
</table>

The definitions of the fuzzy variables and the fuzzy rule using FuzzyCLIPS are shown in Figures 14 to 18 below:

```
(deftemplate x ;definition of fuzzy variable 'Machine cessation'
  15 45 ;Universe of Discourse
  (low (16 0.1) (17 0.3) (18 0.5) (19 1) (20 0.6) (21 0.5) (22 0.3) (23 0.1))
  (medium (21 0.1) (22 0.2) (23 0.4) (24 0.5) (25 0.6) (26 0.7) (27 0.6) (28 0.5) (29 0.4) (30 0.2) (31 1))
  (high (31 0.1) (32 0.2) (33 0.3) (34 0.4) (35 0.5) (36 0.6) (37 0.7) (38 0.1) (39 0.7)(40 0.6) (41 0.5) (42 0.4) (43 0.3) (44 0.2) (45 0.1))
)
```

```
(defrule r1 ; a rule that matches and asserts fuzzy facts
  (x low) (y high) (z high)
  =>
  (assert (cons high))
)
```

Fig. 14. Variable x definition using fuzzyCLIPS

```
(deftemplate y ; definition of fuzzy variable ‘Absence of professionals’
  4 16 ;Universe of Discourse
  (low (5 0.2) (6 1) (7 0.2))
  (medium (7 0.4) (8 1) (9 1) (10 1) (11 1) (12 1))
  (high (11 0.2) (12 0.5) (13 1) (14 0.5) (15 0.2))
)
```

Fig. 15. Variable y definition using fuzzyCLIPS.

```
(deftemplate z ; definition of fuzzy variable ‘Weather changes’
  10 90 ;Universe of Discourse
  (low (10 0.2) (20 0.6) (30 1) (40 0.2))
  (medium (40 0.7) (50 1) (60 0.7))
  (high (70 0.7) (80 1))
)
```

Fig. 16. Variable z definition using fuzzyCLIPS.

```
(deftemplate cons ; definition of fuzzy variable ‘Conclusion’
  20 100 ;Universe of Discourse
  (low (25 0.2) (30 0.5) (35 0.7) (40 1) (45 0.7) (50 0.5) (55 0.2))
  (medium (55 0.5) (65 1) (70 0.6) (75 0.5))
  (high (80 0.2) (85 1) (90 0.5) (95 0.2))
)
```

Fig. 17. Conclusion definition using fuzzyCLIPS.

Fig. 18. Fuzzy rule definition using fuzzyCLIPS.
FuzzyCLIPS was run several times with the same observations in Section 4 and the results are as follows (Figures 19 to 22):

- **Observation 1**: X is Low, Y is High and Z is high.

FuzzyCLIPS gives the expected result according to Criteria I in Table 9. This is natural and expected as all observations match all antecedents.

- **Observation 2**: X is not low, Y is not high and Z is not high.

When the antecedents contain the NOT hedge, FuzzyCLIPS yields a fuzzy set that cannot be mapped to a linguistic expression. This is expected as Mamdani's methods do not satisfy Criterion IV-1 and Criterion IV-2 of Table 9.

- **Observation 3**: X is very low, Y is very high and Z is very high.

FuzzyCLIPS gives the expected result according to Criteria II-2 in Table 9.

- **Observation 4**: X is more or less low, Y is more or less high and Z is more or less high.

The resulting fuzzy set cannot be mapped to a linguistic expression. From the result shown in figure 9 it is clear that Mamdani's methods do not satisfy criterion III.

### 5.2 Comparison with FuzzyJ Toolkit

FuzzyJ Toolkit is a set of Java classes that provide the capability to handle fuzzy concepts and reasoning (Orchard, 2001). It allows for different inference methods including those in (Aly and Vrana, 2006; Mamdani, 1977). We examine the behavior of FuzzyJ Toolkit using the rules and fuzzy variables of Section 4. Figures 23 to 27 show the fuzzy variables and fuzzy rule definitions.
Fig. 23. Definition of fuzzy variable x using fuzzyJ toolkit.

```java
// definition of FuzzyVariable x - Machine Cessation
FuzzyVariable x = new FuzzyVariable("machine cessation", 15, 45);
// definition of FuzzyValues for concept x - Machine Cessation
double xLowL[] = {19, 17, 15, 20, 21, 22, 23, 24, 25};
double xLowM[] = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0};
x.addTerm("low", xLowL, xLowM, 10);
double xMedium[] = {21, 22, 23, 24, 25, 26, 27, 28, 29, 30};
double xHigh[] = {10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
x.addTerm("medium", xMedium, xHigh, 10);
```

Fig. 24. Definition of fuzzy variable y using fuzzyJ toolkit.

```java
// definition of FuzzyVariable y - Absence of professionals
FuzzyVariable y = new FuzzyVariable("Absence of professionals", 4, 16);
// definition of FuzzyValues for concept y - Absence of professionals
double yLowL[] = {4, 5, 6, 7, 8, 9};
double yLowM[] = {0, 0.2, 0.4, 0.6, 0.8, 1.0};
y.addTerm("low", yLowL, yLowM, 5);
double yMedium[] = {5, 6, 7, 8, 9, 10, 11, 12, 13};
double yHigh[] = {0, 0.4, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5};
y.addTerm("medium", yMedium, yHigh, 10);
```

Fig. 25. Definition of fuzzy variable z using fuzzyJ toolkit.

```java
// definition of FuzzyVariable z - Weather Changes
FuzzyVariable z = new FuzzyVariable("Weather changes", 10, 90);
// definition of FuzzyValues for concept z - Weather changes
double zLowL[] = {10, 20, 30, 40, 50};
double zLowM[] = {0, 0.2, 0.4, 0.6, 0.8, 1.0};
z.addTerm("low", zLowL, zLowM, 5);
double zMedium[] = {20, 30, 40, 50, 60, 70, 80};
double zHigh[] = {0, 0.5, 0.7, 1.0, 1.2, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5};
z.addTerm("medium", zMedium, zHigh, 10);
```
FuzzyJ was run several times with the same observations in Section 4 and with the inference method set to Larsen's inference method. The results of the inference are as follows:

- **Observation 1**: X is Low, Y is High and Z is High.
The resulting fuzzy set is \{0/70, 0.2/75, 0.5/80, 1/85, 0.5/90, 0.2/95, 0/100\}. Here the result given by FuzzyJ is "high" which is natural as all the observations match the all the antecedents of the fuzzy rule.

- **Observation 2**: X is not low, Y is not high and Z is not high.
The resulting fuzzy set is \{0/70, 0.1/75, 0.25/80, 0.5/85, 0.25/90, 0.1/95, 0/100\}. This result could not be mapped to a linguistic expression although it is rather close to the fuzzy set "high".

- **Observation 3 & Observation 4**: The resulting fuzzy set is \{0/70, 0.2/75, 0.5/80, 1/85, 0.5/90, 0.2/95, 0/100\}. Here the result given by FuzzyJ is "high". Notice that this is the same result when no hedges were used. It is obvious that the use of the hedge "very" and the "more or less" hedge had no effect on the result. In our shell the
hedges were recognized through the calculation of the implication criteria of Table 9.

6. Conclusion

In this paper, we discussed the development of our own bilingual fuzzy expert system shell. In the process, we have examined, evaluated, and compared various fuzzy expert system shells that adopt different inference methods for the sake of identifying desirable features and examining their performance. Our shell was developed using NetBeans 4.1 IDE. It has an Arabic user interface as well as an English user interface. The inference engine is a backward chaining inference engine. It uses the implication methods Rs, Rg, RSS, RSG, Rgs and Rsg. Several tests have been performed on this shell to ascertain its proper functionality. Some of the tests have given the expected results that reflect human intuitions. Few tests have given results which are very close to the expected outcome. We observe that when the membership function of fuzzy values covers a wide range from 0 to 1, the shell produces more accurate results. Experimental results for our shell have been reported and analyzed. A comparison of the performance of our shell with other shells such as FuzzyCLIPS and FuzzyJ has also been discussed.

We are in the process of extending the shell to allow for the processing of fuzzy terms in other natural languages.

References


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

تطوير صدافية ثنائية اللغة لبناء النظام الخبيزة الضبابية

حسن إسماعيل مذكور، إسراء محمد الطريفي وعيد الله طوير
قسم علوم الحاسب، كلية علوم الحاسب، جامعة الملك سعود، الرياض، المملكة العربية السعودية

mathkour@ksu.edu.sa

(قدم للنشر في 2/3/2004؛ وقبل النشر في 2/2/2008)

ملخص البحث: يستخدم المنطق الضبابي في بعض النظم الخبيزة خلف مشكلات من الحياة العملية والتي تتسم بالغموض. إذ من الممكن استخدام المنطق الضبابي لترجمة التطبيقات التي تتعتمد على الحدس البشري من خلال بناء صدفات لتطوير النظام الخبيزة الضبابية من أجل التعامل مع المشكلات الضبابية والغير واضحة. وتوفر هذه الصدفات تشكيلة من النوال يمكن استخدامها لتحسين تطوير خبرة ضبابية للمتعامل مع مشكلات في مختلف المجالات وتطبيقات مثل التطبيقات الطبية، والهندسية، والمالية. لا توفر حسب علماء صدفات ضبابية، لبناء النظام الخبيزة الضبابية، مطورة في الأصل للتعامل مع اللغة العربية. تصف ورقة العمل هذه تطوير ضبابي صدافي لبناء النظام الخبيزة الضبابية ثنائية اللغة. الهدف من هذه الصدافية البرمجية هو استخدامها كأداة بحثية لتطوير النظم الخبيزة الضبابية في البيئات ثنائية اللغة كما هو الحال في العالم العربي حيث تكون استخدامات المطورين متعددة اللغات بسبب نظام تعليمهم وبيئة عملهم. تسمح الصدافية بالتعامل مع الصدفات المتوفرة الضبابية في اللغتين العربية والإنجليزية. وتعتبر صدافية ضبابية في هذا الهدف توفر للمستخدمين القدرة على تطوير نظام خبرة ضبابية عربية وإنجليزية باستخدام واجهة مستخدم مبتكرة. وتشتمل تطبيقات نظرة الامتداد في مجموعة للحاصل البشري. ثم عند التطور، دراسة ومقارنة عدة صدفات لتطوير برامج النظم الخبيزة الضبابية لتحديد نقاط القوة والضعف للبرمجيات المتوفرة. كما تم إعداد تقرير عن تجربة استعرض الصدافية المتوفرة ومقارنة أدائها مع الصدفات المتوفرة والتي تستخدم طرقًا مختلفة.

J. King Saud University, Vol. 21, Comp. & Info. Sci, Riyadh (2009/1430H.)