J. King Saud Univ., Vol. 21, *Comp. & Info. Sci.*, pp. 27-42, Riyadh (2009/1430H.)

The Development of a Bilingual Fuzzy Expert System Shell

Hassan Mathkour, Israa Al-Turaiki and Ameur Touir
Department of Computer Science, College of Computer and Information Sciences,
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)

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

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 is a general purpose shell that provides users with the ability to develop Arabic/English fuzzy expert systems using a simple Graphical User Interface. It 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.

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 (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) and 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) and 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) and 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_{gs}, 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 20 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 20 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), ESBuilder (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, de-fuzzification 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 Tables 1-5 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. Tables 6 and 7 show the weight assigned to measure the ease of procurement and installation.

Table 1. Rule type weight

Rule Type	Weight
Complex IF-THEN-ELSE rule	5
Complex IF-THEN rule (multiple antecedents and/or multiple consequents)	3
Simple IF-THEN rule (one antecedent one consequent)	1

Table 2. Rule chaining weight

Method	Symbol
Forward	F
Backward	B
No built in chaining strategy (user defined)	NA

Table 3. Math capability weight

Supported Math Functions	Weight
Advanced math functions	5
Basic math functions	3
None	1

Table 4. Inference strategy weight

Supported Inference Strategies	Weight
None	1
One or two	3
Three	5

Table 5. Documentation weight

Documentation	Weight
Comprehensive & easy to read	5
Brief	1

Table 6. Procurement weight

Procurement Method	Weight
Download from the Internet	5
Order package CD	1

Table 7. Installation weight

Installation Method	Weight
Unpack (run) one file	5
Unpack source and compile	1

3. 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, an Arabic or an English screen portraying the main components of system is displayed (Fig. 2a and 2b). 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. 3a and 3b. 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.

Fig. 1. System entry point.

Table 8. Shell comparison

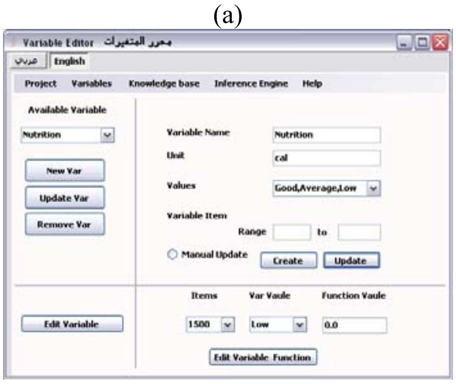
	Criteria/ Shell	Fuzzy Jess	Fuzzy Clips	FuzzyJ	JFK	Fuzzy Logic Toolbox	Cubi Calc	FLOPS	FOX	Fuzzy Shell	FLINT	CLIPS	Jess	Flex	ES Builder	OPSS	Exsys	PROTEST	ESIE	CGIShll	ESB
End–user Interface	Explanation Facility	1	1	1	1	1	1	1	1	1	1	5	1	5	1	1	5	5	1	5	5
	Unknown Answer	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	5	1	1	1
	GUI	1	1	1	1	5	5	1	5	5	5	1	1	1	5	1	5	1	1	5	5
	Changing Answer	5	5	5	5	5	1	1	1	1	1	1	1	1	5	5	1	1	1	1	1
Developer Interface	Rule Type	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1	1	5	1
	Rule Chaining	F&B	F	F&B	F	F	F	F	F	F	B	F&B	F&B	F&B	NA	F	F&B	B	B	NA	NA
	Math Capabilities	5	5	5	3	5	3	3	3	3	3	3	5	3	1	3	5	1	1	1	1
	Fuzzy Inference	3	3	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	1
	Fuzzy Membership Functions	5	5	5	5	5	3	5	5	3	5	1	1	1	1	1	3	1	1	1	1
	Fuzzy Defuzzification	5	3	3	1	5	5	3	5	3	3	1	1	1	1	1	3	1	1	1	1
	Certainty Factors	5	5	5	1	1	1	1	5	1	5	1	1	5	1	1	5	1	1	1	5
	Editing Tools	1	1	1	1	5	5	1	5	1	5	1	1	1	5	1	5	1	1	1	5
	Conflict Resolution	5	5	1	1	1	1		1	5	1	5	5	5	1	5	1	1	1	1	1
	Error Correction Facility	1	1	1	1	1	1	1	5	1	1	1	1	5	1	1	1	1	1	1	1
	Documentation	5	5	5	5	5	5	5	5	5	5	5	1	5	5	1	5	1	5	1	5
	External Program Integration	5	5	5	5	5	5	1	3	1	1	5	5	3	1	5	5	1	1	5	1
Procurement & Installation	Procurement	5	5	5	1	1	1	5	1	5	5	5	5	5	5	5	1	1	5	1	5
	Installation	1	5	1	1	5	5	5	5	5	5	5	1	5	5	5	5	5	5	5	5

(a)

(b)

Fig. 2. The Arabic and English components of the system.

(a)



(b)

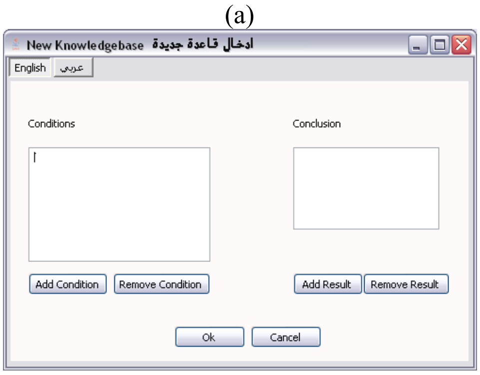
Fig. 3. Arabic/English variable editor layout.

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" "محرر قواعد المعرفة" 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" "إدخال قاعدة معرفة" in the menu bar of Fig. 2.

(a)



(b)

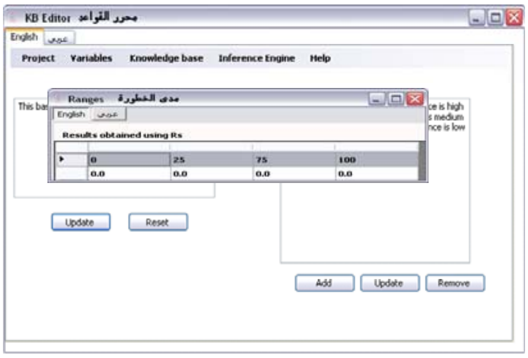
Fig. 4. The Arabic/English layout of the knowledge base editor.

3.3. Inference engine

The inference engine uses implication, composition, aggregation and linkage as given in Leung and Lam (1988), Aly and Vrana (2006) and 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) and Mizumoto (1981), namely, R_s, R_g, R_{gs}, R_{gg}, R_{sg} and R_{ss}. Details of the inference methods are found in Mizumoto *et al.* (1979) and 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) and 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 R_s Implication method. The conflict resolution strategy used in our shell is the most specific strategy.



(a)

(b)

Fig. 5. Conclusion obtained using R_s implication.

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) and Mizumoto (1982).

Table 9. Fuzzy inference criteria

Criterion I	Ant 1: if x is A then y is B An t2: x is A
	Cons: y is B
Criterion II-1	Ant 1: if x is A then y is B Ant 2: x is very A
	Cons: y is very B
Criterion II-2	Ant 1: if x is A then y is B Ant 2: x is very A
	Cons: y is B
Criterion III	Ant 1: if x is A then y is B Ant 2: x is more or less A
	Cons: y is more or less B
Criterion IV-1	Ant 1: if x is A then y is B Ant 2: x is not A
	Cons: y is unknown
Criterion IV-2	Ant 1: if x is A then y is B Ant 2: x is not A
	Cons: y is not B

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

Table 10. Criteria satisfied by each implication method

Ant 2	Cons	R_s	R_g	R_{gs}	R_{gg}	R_{sg}	R_{ss}
A	B	+	+	+	+	+	+
Very A	Very B	+	-	-	-	+	+
Very A	B	-	+	+	+	-	-
More or less A	More or less B	+	+	+	+	+	+
Not A	Not B	-	-	+	+	+	+

Table 11. R_s (A,B)

		V										
		0	1	2	3	4	5	6	7	8	9	10
U	0	0	0	0	0	0	1	0	0	0	0	0
	1	0	0	0	0	1	1	1	0	0	0	0
	2	0	0	0	0	1	1	1	0	0	0	0
	3	0	0	0	1	1	1	1	1	0	0	0
	4	0	0	1	1	1	1	1	1	1	0	0
	5	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	1	1	1	1	1

An R_s 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=1/0+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_s (A,B) is given in 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 composition operator in fuzzy logic is the Max-Min composition operator in Zadeh (1999) and Zadeh (1975), and it is the one we used in our shell.

> **Max-Min Composition**
> Let R be a fuzzy relation in X × Y, and S be a fuzzy relation in Y × Z. The Max-Min composition of R and S, RoS, is a fuzzy relation in X × Z such that
> RoS↔μRoS(x,z) = v {μR(x,y) ^ μS(y,z) }

A Max-Min composition example

Suppose we have the following two relations R and S:

R	y_1	y_2	y_3
x_1	0.4	0.6	0
x_2	0.9	1	0.1

S	z_1	z_2
y_1	0.5	0.8
y_2	0.1	1
Y_3	0	0.6

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

max{min(0.4,0.5), min(0.6, 0.1), min(0, 0)} = max{ 0.4, 0.1, 0} = 0.4
max{min(0.4,0.8), min(0.6, 1), min(0, 0.6)}= max{ 0.4, 0.6, 0} = 0.6
max{min(0.9,0.5), min(1, 0.1), min(0.1, 0)}= max{ 0.5, 0.1, 0} = 0.5
max{min(0.9,0.8), min(1, 1), min(0.1, 0.6)}= max{ 0.8, 1, 0.1} = 1

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 form 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 Figs. 6-9 (Ganoud *et al.*, 2005).

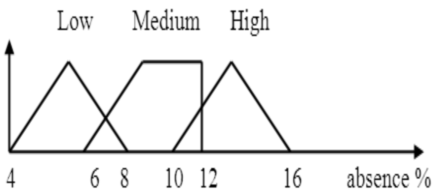


Fig. 6. The membership function of "absence of professionals".

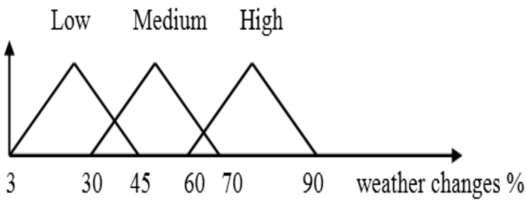


Fig. 7. The membership function of "weather changes".

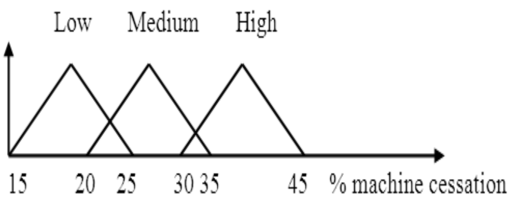


Fig. 8. The membership function of "machine cessation".

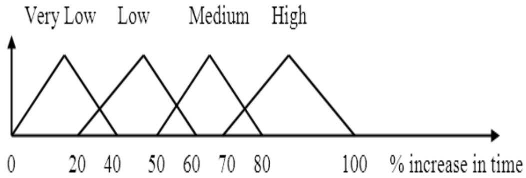


Fig. 9. The membership function of the conclusion "increase in the period of building".

Table 12. The rules data used in the experiment (Ganoud *et al.*, 2005)

النتيجة	نسبة التعطل الناجم عن الظروف الجوية	نسبة التعطل الناجم عن غياب العمال	نسبة التعطل الناجم عن تعطل الآليات	القاعدة
عالية	عالية	عالية	منخفضة	١
عالية	عالية	عالية	متوسطة	٢
عالية	عالية	عالية	عالية	٣
متوسطة	عالية	متوسطة	منخفضة	٤
متوسطة	متوسطة	متوسطة	منخفضة	٥
منخفضة	منخفضة	منخفضة	منخفضة	٦
متوسطة	متوسطة	متوسطة	متوسطة	٧
منخفضة	منخفضة	منخفضة	متوسطة	٨
متوسطة	عالية	منخفضة	متوسطة	٩
عالية	عالية	متوسطة	متوسطة	١٠
عالية	متوسطة	عالية	متوسطة	١١
متوسطة	متوسطة	منخفضة	متوسطة	١٢
متوسطة	منخفضة	عالية	متوسطة	١٣
منخفضة	منخفضة	متوسطة	متوسطة	١٤
منخفضة	متوسطة	منخفضة	منخفضة	١٥
متوسطة	عالية	منخفضة	عالية	١٦
منخفضة	منخفضة	متوسطة	منخفضة	١٧
منخفضة	متوسطة	منخفضة	منخفضة	١٨
عالية	متوسطة	عالية	عالية	١٩
متوسطة	متوسطة	عالية	منخفضة	٢٠
متوسطة	منخفضة	عالية	منخفضة	٢١
عالية	منخفضة	عالية	عالية	٢٢
عالية	عالية	متوسطة	عالية	٢٣
متوسطة	عالية	منخفضة	عالية	٢٤
متوسطة	منخفضة	متوسطة	عالية	٢٥
متوسطة	متوسطة	متوسطة	عالية	٢٦
منخفضة	منخفضة	منخفضة	عالية	٢٧

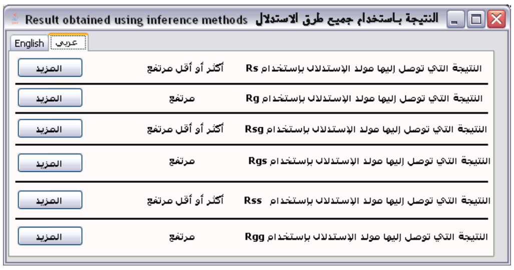
The knowledge base consists of 27 fuzzy rules. All the rules have the same number of antecedents. The same fuzzy variables are used in all the 27 rules as well as the same conclusion (target). The shell was run several times, each time with different observation. The observations are:

- **Observation 1:** X is low, Y is high and Z is high.
  It is observed (as illustrated in Fig. 10) that using R_g, R_{gs} and R_{gg} has given the expected results according to 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.

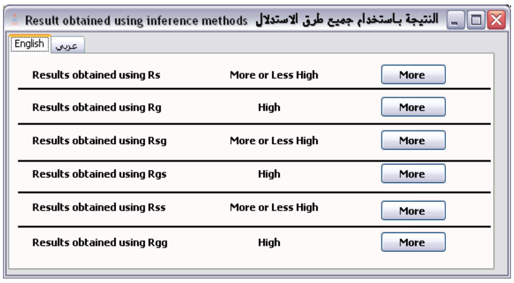
- **Observation 2:** X is not low, Y is not high and Z is not high.
  It is observed (as illustrated in Fig. 11) that all the implications have given the expected results according to Table 9 except for R_{gs} and R_{ss}. Examining the matrices that resulted from using R_{gs} and R_{ss}, we have found 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 Fig. 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 Fig. 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.

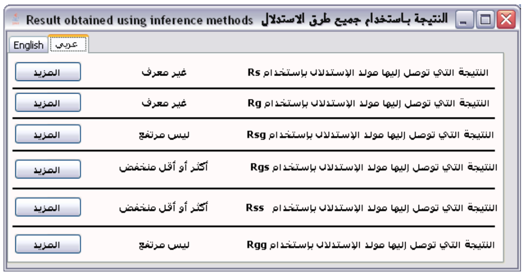


(a)

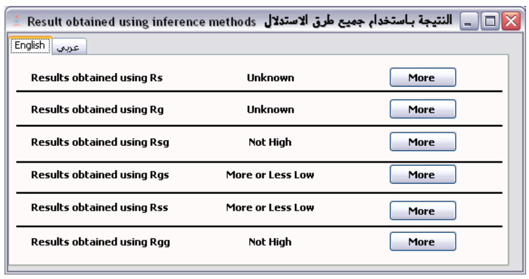


(b)

Fig. 10. Result of Observation 1.

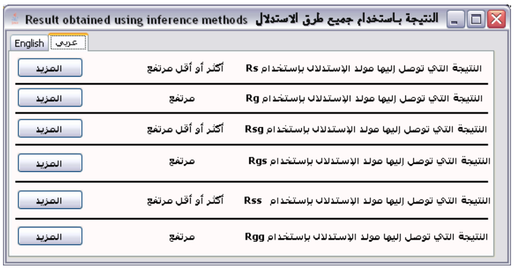


(a)

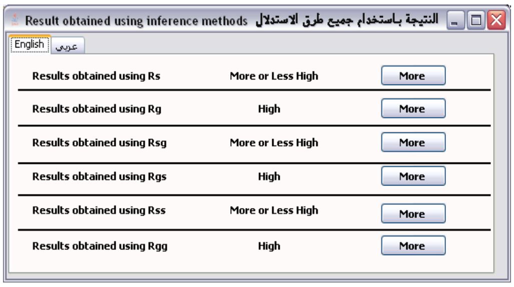


(b)

Fig. 11. Result of Observation 2.

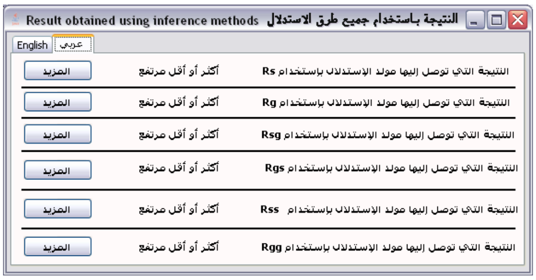


(a)

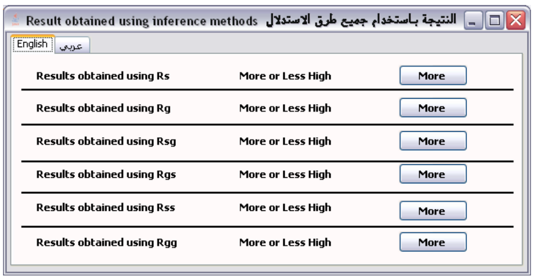


(b)

Fig. 12. Result of Observation 3.



(a)



(b)

Fig. 13. Result of Observation 4.

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) and 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 13 is used.

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

Criterion I	Ant1: IF x is A Then y is B Ant2: x is A
	Cons: y is B
Criterion II-2	Ant1: IF x is A Then y is B Ant2: x is very A
	Cons: y is B

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. The definitions of the fuzzy variables and the fuzzy rule using FuzzyClips are shown in Figs. 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))
  )
)
```

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.

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

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 (Figs. 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.

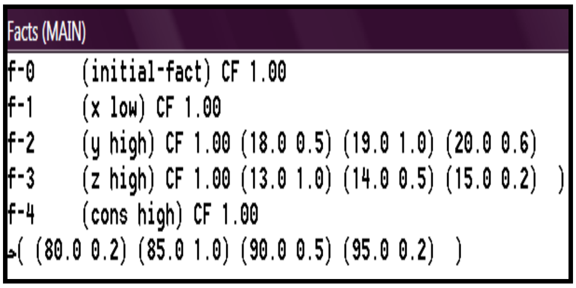


Fig. 19. FuzzyClips result for Observation 1.

- **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.

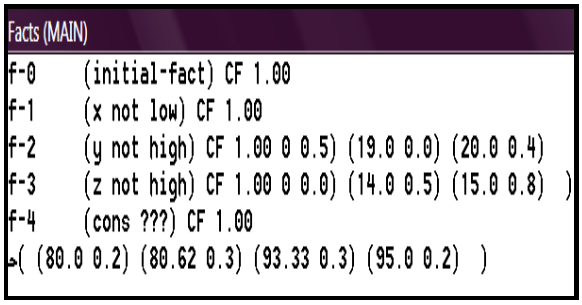


Fig. 20. FuzzyClips result for Observation 2.

- **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.

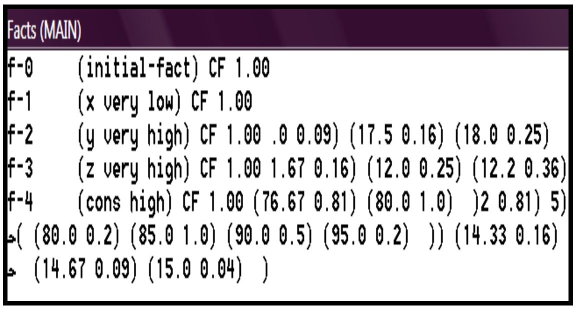


Fig. 21. FuzzyClips result for Observation 3.

- **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 Fig. 9, it is clear that Mamdani's methods do not satisfy Criterion III.

```
Facts (MAIN)
f-0    (initial-fact) CF 1.00
f-1    (x more-or-less low) CF 1.00
f-2    (y more-or-less high) CF 1.00 .6694) (17.5 0.7368) (18.0 0.7937)
f-3    (z more-or-less high) CF 1.00  0.7368) (12.0 0.7937) (12.2 0.8434)
f-4    (cons high) CF 1.00 83) (76.67 0.9655) (80.0 1.0)  )2 0.9655) 937)
( (80.0 0.2) (85.0 1.0) (90.0 0.5) (95.0 0.2)  )0 0.7937) (14.33 0.7368)
 (14.67 0.6694) (15.0 0.5848)  )
```

Fig. 22. FuzzyClips result for Observation 4.

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) and Mamdani (1977). We examine the behavior of FuzzyJ Toolkit using the rules and fuzzy variables of Section 4. Figures 23-27 show the fuzzy variables and fuzzy rule definitions.

```
// definition of FuzzyVariable x - Machine Cessation
  FuzzyVariable x = new FuzzyVariable("machine cessation",15,45);
// definition of FuzzyValues for concept x - Machine Cessation'
  double xLowx[] = {16, 17,18,19,20,21,22,23,24,25};
  double yLowx[] = {0.1,0.3,0.5,1,0.6,0.5,0.3,0.1,0.0,0.0};
  x.addTerm("low", xLowx, yLowx, 10);
  double xMediumx[] = {21,22,23,24,25,26,27,28,29,30,31};
  double yMediumx[] = {0.1,0.2,0.4,0.5,0.6,0.7,0.6,0.5,0.4,0.2,1};
  x.addTerm("medium", xMediumx, yMediumx, 11);
  double xHighx[] = {31,32,33,34,35,36,37,38,39,40,41,42,43,44,45};
  double yHighx[] = {0.1 ,0.2,  0.3, 0.4, 0.5,0.6, 0.7, 0.1, 0.7,0.6, 0.5, 0.4, 0.3,0.2,0.1};
  x.addTerm("high",xHighx,yHighx ,15);
```

Fig. 23. Definition of fuzzy variable x using FuzzyJ Toolkit.

```
// 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 xLowy[] = {4,5 ,6, 7,8 };
  double yLowy[] = {0,0.2,1,0.2,0};
  y.addTerm("low", xLowy, yLowy, 5);
  double xMediumy[] = {6,7,8,9,10,11,12,13};
  double yMediumy[] = {0,0.4,1,1,1,1,1,0};
  y.addTerm("medium", xMediumy, yMediumy, 8);
  double xHighy[] = {10,11,12,13,14,15,16};
  double yHighy[] = {0,0.2,0.5,1,0.5,0.2,0};
  y.addTerm("high",xHighy,yHighy ,7);
```

Fig. 24. Definition of fuzzy variable y using FuzzyJ Toolkit.

```
  // definition of FuzzyVariable z - Weather changes
  FuzzyVariable z = new FuzzyVariable("Weather changes'",10,90);
// definition of FuzzyValues for concept z - Weather changes
  double xLowz[] = {10,20,30,40,50};
  double yLowz[] = {0.2,0.6,1,0.2,0};
  z.addTerm("low", xLowz, yLowz, 5);
  double xMediumz[] = {20,30,40,50,60,70,80};
  double yMediumz[] = {0,0.5,0.7,1,0.7,0.5,0};
  z.addTerm("medium", xMediumz, yMediumz, 7);
  double xHighz[] = {50,60,70 ,80};
  double yHighz[] = {0,0.5,0.7 ,1};
  z.addTerm("high",xHighz,yHighz ,4);
```

Fig. 25. Definition of fuzzy variable z using FuzzyJ Toolkit.

```
 // definition of FuzzyVariable cons - Conclusion
  FuzzyVariable cons = new FuzzyVariable("Conclusion" ,20,100);
// definition of FuzzyValues for concept cons - Conclusion
  double xLow_cons[] = {20,25,30,35,40,45,50,55,60};
  double yLow_cons[] = {0,0.2,0.5,0.7,1,0.7,0.5,0.2,0};
  cons.addTerm("low", xLow_cons, yLow_cons, 9);
  double xMedium_cons[] = {45,50,55,65,70,75,80,85};
  double yMedium_cons[] = {0,0.2,0.5,1,0.6,0.5,0.2,0};
  cons.addTerm("medium", xMedium_cons, yMedium_cons, 8);
  double xHigh_cons[] = {70,75,80,85,90,95,100};
  double yHigh_cons[] = {0,0.2,0.5,1,0.5,0.2,0};
  cons.addTerm("high",xHigh_cons,yHigh_cons ,7);
```

Fig. 26. Definition of fuzzy variable conclusion using FuzzyJ Toolkit.

```
 // definition of FuzzyRule
 FuzzyRule rule1 = new FuzzyRule();
 fval1 = new FuzzyValue( x, "low");
 fval2 = new FuzzyValue( y, "high");
 fval3 = new FuzzyValue( z, "high");
 fval4 = new FuzzyValue( cons , "high");

 rule1.addAntecedent(fval1);
 rule1.addAntecedent(fval2);
 rule1.addAntecedent(fval3);
 rule1.addConclusion(fval4);
```

Fig. 27. Definition of the fuzzy rule using FuzzyJ.

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 and 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 R_s, R_g, R_{ss}, R_{gg}, R_{gs} and R_{sg}. 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

Aly, S. and Vrana, I. "Toward Efficient Modeling of Fuzzy Expert Systems: A Survey." *Agric. Econom. - Czech*, (2006), 456-460.

Bandler, W. and Kohout, L. "Fuzzy Power Sets and Fuzzy Implication Operators." *Fuzzy Sets and Systems*, Vol. 4, (1980), 13-30.

Council, C. and Orchard, R. "Fuzzy Reasoning in Jess: The FuzzyJ Toolkit and FuzzyJess." *The Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS 2001)*, Setubal, Portugal, (2001).

Forgy, C.L. "The OPSS User Manual." Technical Report, CMU-CS-81-135, Computer Science Department, Carnegie-Mellon University, Pittsburgh, (1981).

Friedmann-Hill, E.J. "Jess: The Java Expert System Shell." http://herzberg.ca.sandia.gov/jess/, (1999).

Fukami, S.; Mizumoto, M. and Tanaka, K. "Some Considerations on Fuzzy Conditional Inference." *Fuzzy Sets and Systems*, Vol. 4, (1980), 243-273.

Ganoud, A.; Ali, H. and Ibrahem, S. "Studying the Influence of Random Factors on the Planning of Building Work." *Tishreen University Journal for Studies and Scientific Research - Engineering Science Series*, Vol. 27, No. (3), (2005).

Giarratano, J.C. "The CLIPS User's Guide." http://www.twine.com/twine/11lslvj57-19d/clips-expert-system-shell, (1998).

Kent, J.R. and Denholm, P. "ESB-96 – A Graphical KBS Development Tool." *Proceedings of the First European Conference on the Practical Application of Lisp. Dorking (Europal '90)*, Cambridge, UK, (March 1990).

Ishihara, S.; Ishihara, K.; Nagamachi, M. and Matsubara, Y. "An Automatic Builder for a Kansei Engineering Expert System Using Self-organizing Neural Networks." *Int. Journal of Industrial Ergonomics*, Vol. 15, No. (1), (1995), 13-24.

Kelmet, E. and Slany, W. "Fuzzy Logic in Artificial Intelligence." *Proceedings of the 8th Austrian Artificial Intelligence Conference (FLAI '93)*, Linz, Austria, (1993).

Leung, K.S.; and Lam, W. "Fuzzy Concepts in Expert Systems." *IEEE Computer*, Vol. 21, No. (9), (1988), 43-56.

López-Ortega, O. "Java Fuzzy Kit (JFK): A Shell to Build Fuzzy Inference Systems According to the Generalized Principle of Extension." *Expert Systems with Applications*, Vol. 34, No. (1), (2006).

Mamdani, E.H. "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Systems." *IEEE Trans. Comp.*, Vol. 26, (1977), 1182-1191.

Mathkour, H.; Al-Turaiki, I. and Touir, A. "The Development of an Arabized Fuzzy Expert System Shell." *Int. Conf. on Information Management and Engineering (ICIME)*, Kuala Lumpur, Malaysia, (2009).

Mathworks, Inc. *Fuzzy Logic Toolbox User's Guide Ver. 2*. (1999).

Mizumoto, M.; Fukami, S. and Tanaka, K. "Fuzzy Conditional Inference and Fuzzy Inference with Fuzzy Quantifiers." *Proc. of the 6th Int. Conf. on Artificial Intelligence*, Tokyo, (1979), 589-591.

Mizumoto, M.; Fukami, S. and Tanaka, K. "Some Methods of Fuzzy Reasoning." In: M.M. Gupta, R.K. Ragade and R.R. Yager (Eds.), *Advances in Fuzzy Set: Theory and Application*. Amsterdam: North-Holland, (1979), pp. 117-136.

Mizumoto, M. "Note on the Arithmetic Rule by Zadeh for Fuzzy Conditional Inference." *Cybernetics and Systems*, Vol. 12, (1981), 247-306.

Mizumoto, M. "Fuzzy Inferences under Max-Min Composition in the Compositional Rule of Inference." In: M.M. Gupta and E. Sanchez (Eds.), *Fuzzy Information and Decision Processes*. Amsterdam: North-Holland, (1982); *Information Sciences*, Vol. 27, (1982), 183-209.

Mizumoto, M.; Fukami, S. and Tanaka, K. "Several Methods for Fuzzy Conditional Inference." *Proc. of IEEE Conf. on Decision & Control*, Florida, (1979), 777-782.

Orchard, R. *NRC FuzzyJ Toolkit for the Java™, Platform User's Guide*. [online] National Research Council of Canada. Available at: http://ai.iit.nrc.ca/IR_public/fuzzy/fuzzyJDocs/index.html, (2001).

Orchard, R. "Fuzzy Reasoning in Jess: The FuzzyJ Toolkit and FuzzyJess." *Proceedings of the ICEIS 2001, Third International Conference on Enterprise Information Systems*, Setubal, Portugal, (2001), 533-542.

Orchard, R. *FuzzyClips version 6.10d User's Guide*. National Research Council of Canada, (2004).

Pan, J. *FuzzyShell User's Manual*. (1996).

Philip, G. "A Case Study in Selecting an Expert System Shell." *Journal of Systems Management*, (1991).

Shalfield, R. *FLINT Reference*. London: Logic Programming Associates Ltd., (2005).

Siler, W. and Buckley, J. *Fuzzy Expert Systems and Fuzzy Reasoning*. New Jersey: John Wiley & Sons, Inc., (2005).

Vasey, P.; Westwood, D. and Johns, N. *Flex Expert System Toolkit*. London: Logic Programmers Associates Ltd., (1996).

Willmott, R. "Two Fuzzier Implication Operators in the Theory of Fuzzy Power Sets." *Fuzzy Sets and Systems*, Vol. 4, (1980), 31-37.

Zadeh, L.A. "Calculus of Fuzzy Restriction." In: L.A. Zadeh *et al.* (Eds.), *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. New York: Academic Press, (1975), pp. 1-39.

Zadeh, L.A. "From Computing with Numbers to Computing with Words: From Manipulation of Measurements to Manipulation of Perceptions." *IEE Transactions on Circuits and Systems, Fundamental Theory and Applications*, Vol. 45, No. (1), (January 1999), 105-119.

Zadeh L.A. "Fuzzy Sets." *Information and Control*, Vol. 8, (1965), 338-353.

Zadeh, L. A. "Why the Success of Fuzzy Logic is Not Paradoxial." *IEEE Expert*, Vol. 9, (1994), 43-45.

تطوير صدفية ثنائية اللغة لبناء النظم ا ـبـ ة الضبابية

حسن إ اعيل مذكور، وإسراء مد الطريفي، وعامر عبدا طوير

قسم علوم الحاسب، كلية علوم الحاسب والمعلومات، جامعة الملك سعود، الرياض، المملكة العربية السعودية

mathkour@ccis.ksu.edu.sa, binmathkour@yahoo.com, touir@ksu.edu.sa

(قدم للنشر في ٢٠٠٨/١٢/٢١م؛ وقبل للنشر في ٢٠٠٩/٣/٢م)

ملخص البحث. يستخدم المنطق الضبابي في بعض النظم الخبيرة لحل مشكلات من الحياة العملية والتي تتسم بالغموض، إذ من الممكن استخدام المنطق الضبابي لبرمجة التطبيقات التي تعتمد على الحدس البشري من خلال بناء صدفيات لتطوير النظم الخبيرة الضبابية من أجل التعامل مع المشكلات الضبابية وغير الواضحة. وتوفر هذه الصدفيات تشكيلة من الدوال يمكن استخدامها لتسهيل تطوير نظم خبيرة ضبابية للتعامل مع مشكلات في مختلف المجالات والتطبيقات مثل التطبيقات الطبية، والهندسية، والمالية. ولا تتوفر، حسب علمنا، صدفيات برمجة ضبابية لبناء النظم الخبيرة الضبابية، مطورة في الأصل للتعامل مع اللغة العربية. تصف ورقة العمل هذه تطوير وتجربة صدفية لبناء النظم الخبيرة الضبابية ثنائية اللغة. والهدف من هذه الصدفية البرمجية هو استخدامها كأداة بحثية لمطوري النظم الخبيرة الضبابية في البيئات ثنائية اللغة كما هو الحال في العالم العربي حيث تكون استخدامات المطورين متعددة اللغات بسبب نظام تعليمهم وبيئة عملهم. وتسمح الصدفية بالتعامل مع المصطلحات الضبابية في اللغتين العربية والإنجليزية. وتعتبر صدفية البرمجة عامة الأهداف توفر للمستخدمين القدرة على تطوير نظم خبيرة ضبابية عربية وإنجليزية باستخدام واجهة مستخدم مبسطة. وتقوم بتطبيق طرق الاقتضاء في محاكاة للحدس البشري. تم عند التطوير دراسة ومقارنة عدة صدفيات لتطوير وبرمجة النظم الخبيرة الضبابية لتحديد نقاط القوة والضعف للبرمجيات المتوفرة، كما تم إعداد تقرير عن تجربة واختبار الصدفية المطورة ومقارنة أدائها مع الصدفيات المتوفرة والتي تستخدم طرقاً اقتضائية مختلفة.