January 01, 2003

Second-hand market as an alternative in reverse logistics

Surendra M. Gupta
Northeastern University

Bibliographic Information

Pochampally, K. K. and, Gupta, S. M., "Second-Hand Market as an Alternative in Reverse Logistics", ***Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing III***, Providence, Rhode Island, pp. 30-39, October 29-30, 2003.

Copyright Information

Contact Information

Dr. Surendra M. Gupta, P.E.
Professor of Mechanical and Industrial Engineering and
Director of Laboratory for Responsible Manufacturing
334 SN, Department of MIE
Northeastern University
360 Huntington Avenue
Boston, MA 02115, U.S.A.

(617)-373-4846 ***Phone***
(617)-373-2921 ***Fax***
gupta@neu.edu ***e-mail address***

http://www.coe.neu.edu/~smgupta/ ***Home Page***

Second-Hand Market as an Alternative in Reverse Logistics

Kishore K. Pochampally and Surendra M. Gupta*
Laboratory for Responsible Manufacturing
334 SN, Department of MIME
Northeastern University
360 Huntington Avenue
Boston, MA 02115.

ABSTRACT

Collectors of discarded products seldom know when those products were bought and why they are discarded. Also, the products do not indicate their remaining life periods. So, it is difficult to decide if it is "sensible" to repair (if necessary) a particular product for subsequent sale on the second-hand market or to disassemble it partially or completely for subsequent remanufacture and/or recycle. To this end, we build an expert system using Bayesian updating process and fuzzy set theory, to aid such decision-making. A numerical example demonstrates the building approach.

Keywords: Second-hand Market, Expert System, Bayesian Updating, Fuzzy Sets, Uncertainty.

1. MOTIVATION

Reverse logistics encompasses the transfer of discarded products from consumers to producers within a distribution network. The most important driver for companies interested in collecting discarded products is recoverable value through reprocessing (remanufacture/recycle) [3], [5]. However, collectors of discarded products seldom know when those products were bought and why were they discarded. Also, the products do not indicate their remaining life periods. Hence, they often undergo partial or complete disassembly for subsequent remanufacture or recycle. We believe that for some discarded products, it might make more "sense" to make necessary repairs to the products and sell them on the second-hand market than to disassemble them for subsequent remanufacture and/or resale. To this end, we build an expert system using Bayesian updating process and fuzzy set theory, to decide if it is "sensible" to repair the product of interest for subsequent sale on the second-hand market.

In the next section, a brief introduction of expert systems is provided. In Section 3, some important formulae used in the Bayesian updating process are detailed. Section 4 introduces fuzzy set theory that we use to implement the Bayesian updating process. Finally, in Section 5, a numerical example demonstrates how an expert system can be built to decide if it is "sensible" to repair the product of interest for subsequent sale on the second-hand market. Section 6 gives some conclusions.

2. EXPERT SYSTEM

Expert systems are computer programs that can represent human expertise (knowledge) in a particular domain (area of expertise) and then use a reasoning mechanism (applying logical deduction and induction processes) to manipulate this knowledge in order to provide advice in this domain. Although conventional programs like C and C++ also contain knowledge, their main function is to retrieve information, and carry out statistical analysis and numerical calculations [10]. They do not reason with this knowledge or make inferences as to what actions to take or conclusions to reach. So, what mainly distinguishes expert systems from conventional programs is the capability to reason with knowledge. The main components of an expert system are the following:

---

*Correspondence: e-mail: gupta@neu.edu; URL: http://www.coe.neu.edu/~smgupta/
Phone: (617)-373-4846; Fax: (617)-373-2921

Environmentally Conscious Manufacturing III, edited by Surendra M. Gupta, Proc. of SPIE Vol. 5262
(SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.516226

- A *knowledge base*: This is where the knowledge is stored. Typically, this consists of a set of rules of the form: if EVIDENCE then HYPOTHESIS. The knowledge is written in the knowledge base using the syntax of what is termed the *knowledge representation language* (examples: Lisp and Prolog) of the system.
- *An inference engine:* This reasons with the knowledge resident in the knowledge base using certain mechanisms.
- *A reasoning mechanism:* This traces the path or the knowledge steps used to arrive at a conclusion and can relay it back to user as the justification for this conclusion. Examples of these mechanisms are deduction (cause + rule → effect), abduction (effect + rule → cause), and induction (cause + effect → rule).
- *An uncertainty modeling process:* This aids the inference engine when dealing with uncertainty. The uncertainty modeling process that we use in our approach to decide whether it is "sensible" or not to send a discarded product for any necessary repairs and subsequent sale on the second-hand market is called Bayesian updating.

A *shell* is an expert system that is complete except for the knowledge base [6]. Thus, a shell includes an inference engine, a user interface for programming, and a user interface for running the system. Typically, the programming interface comprises a specialized editor for creating rules in a pre-determined format, and some debugging tools. The user of the shell enters rules in a declarative fashion (If *X*, then *Y*), and ideally should not need to be concerned with the working of the inference engine. Expert system shells are easy to use and allow a simple expert system to be constructed quickly. In this paper, we use the FLEX expert system shell [8] whose inference engine is programmed in Prolog.

3. BAYESIAN UPDATING

Bayesian updating is an uncertainty modeling technique that assumes that it is possible for an expert in a domain to guess a probability to every hypothesis or assertion in that domain, and that this probability can be updated in light of evidence for or against the hypothesis or assertion. In our approach to decide if it is "sensible" to repair a particular discarded product for subsequent sale on the second-hand market, we use fuzzy set theory [11] to calculate certain probabilities that are difficult to guess (for example, it is difficult to guess how often a <u>component</u>, say *C*, is observed needing repair when it is decided that repairing the <u>product</u> is "sensible"). We introduce the fuzzy set theory to the reader in Section 4 and demonstrate its usage in Section 5.

Suppose the probability of a hypothesis is P(H). Then, the formula for the odds O(H) of the hypothesis H is given by:

$$O(H) = \frac{P(H)}{1\text{-}P(H)} \quad (1)$$

A hypothesis that is absolutely certain, i.e., has a probability of 1, has infinite odds. In practice, limits are often set on odds values so that, for example, if O(H) > 1000 then H is true, and if O(H) < 0.01 then H is false.

3.1. Updating probabilities with supporting evidence

The standard formula for updating the odds of hypothesis H, given that evidence E is observed, is:

$$O(H|E) = (A).O(H) \quad (2)$$

where O(H|E) is the odds of H, given the presence of evidence E, and A is the *affirms* weight of E. The definition of A is:

$$A = \frac{P(E|H)}{P(E|\sim H)} \quad (3)$$

where P(E|H) is the probability of E, given that H is true and P(E|~H) is the probability of E, not given that H is true.

3.2. Updating probabilities with opposing evidence

Bayesian updating assumes that the absence of supporting evidence is equivalent to the presence of opposing evidence. The standard formula for updating the odds of a hypothesis H, given that the evidence E is absent, is:

$$O(H|\sim E) = (D).O(H) \quad (4)$$

where O(H|~E) is the odds of H, given the absence of evidence E, and D is the *denies* weight of E. The definition of D is:

$$D = \frac{P(\sim E|H)}{P(\sim E|\sim H)} = \frac{1-P(E|H)}{1-P(E|\sim H)} \tag{5}$$

If a given piece of evidence E has an *affirms* weight A which is greater than 1, then its *denies* weight must be less than 1, and vice versa. Also, if A > 1 and D < 1, then the presence of evidence E is supportive of hypothesis H. Similarly, if A < 1 and D > 1, then the absence of E is supportive of H.

For example, while controlling a power station boiler, a rule - "IF (temperature is high) and NOT (water level is low) THEN (pressure is high)" can also be written as "IF (temperature is high - AFFIRMS A_1, DENIES D_1) AND (water level is low - AFFIRMS A_2, DENIES D_2) THEN (pressure is high)". Here,

$$A_1 = \frac{P(\text{Temperature is high | Pressure is high})}{P(\text{Temperature is high | \~Pressure is high})};\quad D_1 = \frac{P(\text{\~Temperature is high | Pressure is high})}{P(\text{\~Temperature is high | \~Pressure is high})};$$

$$A_2 = \frac{P(\text{Water level is low | Pressure is high})}{P(\text{Water level is low | \~Pressure is high})};\quad D_2 = \frac{P(\text{\~Temperature is high | Pressure is high})}{P(\text{\~Temperature is high | \~Pressure is high})};$$

3.3. Dealing with uncertain evidence

Sometimes, an evidence is neither definitely present nor definitely absent. For example, if one is diagnosing a TV set that is not functioning properly, it is not definite if this is due to a malfunctioning picture tube or not. In such a case, depending upon the value of the probability of the evidence P(E), the affirms and denies weights are modified using the following formulae [4]:

$$A' = [2.(A-1).P(E)]+2-A \tag{6}$$

$$D' = [2.(1-D).P(E)]+D \tag{7}$$

When P(E) is greater than 0.5, the *affirms* weight is used to calculate P(O|H), and when P(E) is less than 0.5, the *denies* weight is used.

3.4. Combining evidence

If, *n* statistically independent pieces of evidence are found that support or oppose a hypothesis H, then the updating equations are given by [4]:

$$O(H|E_1\&E_2\&E_3......E_n) = (A_1).(A_2).(A_3).......(A_n).O(H) \tag{8}$$

and

$$O(H|\sim E_1\&\sim E_2\&\sim E_3......\sim E_n) = (D_1).(D_2).(D_3).......(D_n).O(H) \tag{9}$$

A_i and D_i are given by Equations 10 and 11 respectively.

$$A_i = \frac{P(E_i|H)}{P(E_i|\sim H)} \tag{10}$$

$$D_i = \frac{P(\sim E_i | H)}{P(\sim E_i|\sim H)} \tag{11}$$

4. FUZZY SET THEORY

Expressions such as "not very clear", "probably so" and "very likely" can be heard very often in daily life. The commonality in such terms is that they are all tainted with imprecision. This imprecision or vagueness of human decision-making is called "fuzziness" in the literature. With different decision-making problems of diverse intensity, the results can be misleading if fuzziness is not taken into account. However, since Zadeh [11] first proposed fuzzy set theory, an increasing number of studies have dealt with imprecision (fuzziness) in problems by applying the fuzzy set theory. Our paper too makes use of this theory in building an expert system that can decide if it is "sensible" to repair (if necessary) a discarded product of interest and subsequently sell it on the second-hand market. The concepts of the fuzzy set theory, which we utilize in this paper, are as follows:

4.1. Linguistic values and fuzzy sets

When dealing with imprecision, decision-makers may be provided with information characterized by vague language such as: high risk, low profit and good customer service. By using *linguistic* values like "high", "low", "good", "medium", "cheap", etc., people are usually attempting to describe factors with uncertain or imprecise values. For example, "weight" of an object may be a factor with an uncertain or imprecise value and so, its linguistic value can be "very low", "low", "medium", "high", "very high", etc. The fuzzy set theory is primarily concerned with quantifying the vagueness in human thoughts and perceptions.

To deal with quantifying vagueness, Zadeh proposed a membership function which associates with each quantified linguistic value a grade of membership belonging to the interval [0, 1]. Thus, a fuzzy set is defined as:

$$\forall x \in X, \mu_A(x) \in [0,1]$$

where $\mu_A(x)$ is the degree of membership, ranging from 0 to 1, of a quantity x of the linguistic value, A, over the universe of quantified linguistic values, X. X is essentially a set of real numbers. The more x fits A, the larger the degree of membership of x. If a quantity has a degree of membership equal to 1, this reflects a complete fitness between the quantity and the vague description (linguistic value). Whereas, if the degree of membership of a quantity is 0, then that quantity does not belong to the vague description.

4.2. Triangular fuzzy numbers

A triangular fuzzy number (TFN) is a fuzzy set with three parameters, each representing a quantity of a linguistic value associated with a degree of membership of either 0 or 1. It is graphically depicted in Figure 1. The parameters a, b and c respectively denote the smallest possible quantity, the most promising quantity and the largest possible quantity that describe the linguistic value.

Each TFN, P, has linear representations on its left and right side such that its membership function can be defined as:

$$\mu_P = 0, \quad x < a \tag{12}$$
$$= (x-a) / (b-a) \quad a \leq x \leq b \tag{13}$$
$$= (c-x) / (c-b) \quad b \leq x \leq c \tag{14}$$
$$= 0, \quad x \geq c. \tag{15}$$

For each quantity x increasing from a to b, its corresponding degree of membership linearly increases from 0 to 1. While x increases from b to c, its corresponding degree of membership linearly decreases from 1 to 0. The membership function is a mapping from any given x to its corresponding degree of membership.

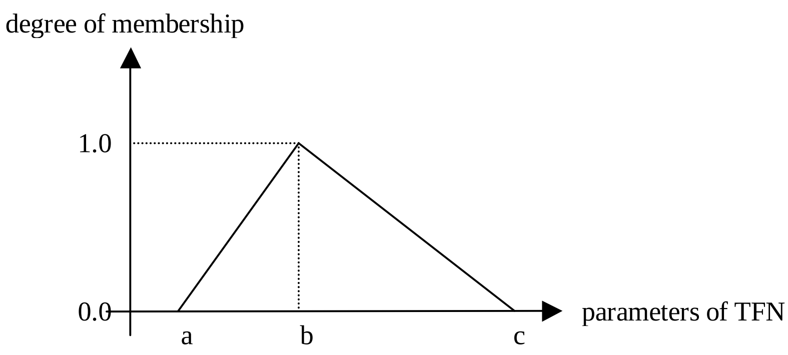


Figure 1. Triangular fuzzy number

The TFN is mathematically easy to implement, and more importantly, it represents the rational basis for quantifying the vague knowledge in most decision-making problems.

The basic operations on triangular fuzzy numbers are as follows [1], [2], [9]:
For example, $P_1 = (a, b, c)$ and $P_2 = (d, e, f)$.

$P_1 + P_2 = (a+d, b+e, c+f)$ addition; (16)
$P_1 - P_2 = (a-f, b-e, c-d)$ subtraction; (17)
$P_1 * P_2 = (a*d, b*e, c*f)$ where $a \geq 0$ and $d \geq 0$ multiplication; (18)
$P_1 / P_2 = (a/f, b/e, c/d)$ where $a \geq 0$ and $d > 0$ division. (19)

4.3. Defuzzification

Defuzzification is a technique to convert a fuzzy number into a crisp real number. There are several methods to serve this purpose [7]. For example, the Centre-of-Area method [12] converts a fuzzy number $P = (a, b, c)$ into a crisp real number Q where

$$Q = \frac{(c-a)+(b-a)}{3} + a \tag{20}$$

5. NUMERICAL EXAMPLE

Consider the discarded product shown in Figure 2. Given that this product is not functioning properly, we shall decide whether it is "sensible" to repair it for subsequent sale on the second-hand market or not. Table 1 shows the probability values we use to implement the Bayesian updating process.

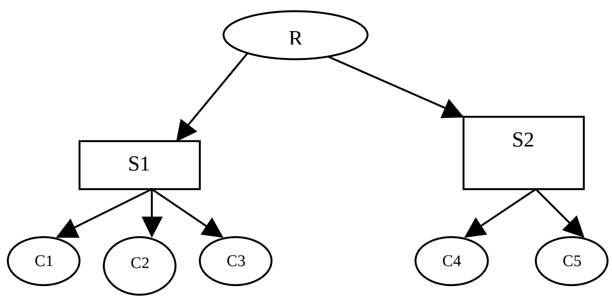


Figure 2. Discarded product

5.1. Usage of fuzzy set theory

Since it is difficult for an expert to guess the probabilities shown in bold (unlike the rest) in Table 1, we calculate them using the fuzzy set theory as follows:

I. Ask the expert to assign a linguistic rating to P(E|H) for each component with respect to each of the following factors (see Table 2):
   a. Is it economical to repair/replace the component? (more economical implies higher rating)
   b. If disposed, will the component be harmful to the environment? (more harmful implies lower rating)
   c. What is the remaining life period of the component? (longer life implies higher rating)
   d. Is the raw material used to make the component depleting fast? (faster depletion implies lower rating)
   e. Is it difficult to repair the component? (more difficult implies lower rating)
II. Use the data in Table 3 to convert the linguistic ratings into TFNs.
III. Calculate the average fuzzy P(E|H) value for each component.
IV. Defuzzify the average P(E|H) for each component.

Apply steps I, II, III, and IV to calculate P(E|~H) values for each component. In order to save ourselves from the tedious calculations, we assume here that the values shown in bold in Table 1 are the defuzzified average probabilities obtained *after* performing steps I, II, III and IV.

Table 1. Probability values used in Bayesian updating

H	E	P(H)	O(H)	P(E|H)	P(E|~H)	A	D
S1 needs repair	Product needs repair	0.60	1.50	1.00	0.60	1.67	0.00
S2 needs repair	Product needs repair	0.70	2.33	1.00	0.40	2.50	0.00
C1 needs repair	S1 needs repair	0.45	0.82	1.00	0.45	2.22	0.00
C2 needs repair	S1 needs repair	0.55	1.22	1.00	0.30	3.33	0.00
C3 needs repair	S1 needs repair	0.30	0.43	1.00	0.55	1.82	0.00
C4 needs repair	S2 needs repair	0.32	0.47	1.00	0.70	1.43	0.00
C5 needs repair	S2 needs repair	0.10	0.11	1.00	0.80	1.25	0.00
Sensible to repair product	C1 needs repair	0.60	1.50	**0.70**	**0.20**	3.50	0.38
Sensible to repair product	C2 needs repair	0.60	1.50	**0.60**	**0.30**	2.00	0.57
Sensible to repair product	C3 needs repair	0.60	1.50	**0.45**	**0.60**	0.75	1.38
Sensible to repair product	C4 needs repair	0.60	1.50	**0.10**	**0.75**	0.13	3.60
Sensible to repair product	C5 needs repair	0.60	1.50	**0.85**	**0.40**	2.13	0.25

5.2. Rules used in Bayesian updating

Rule 1: IF product needs repair (AFFIRMS: 1.67; DENIES: 0.00) THEN S1 needs repair.

Rule 2: IF product needs repair (AFFIRMS: 2.50; DENIES: 0.00) THEN S2 needs repair.

Rule 3: IF S1 needs repair (AFFIRMS: 2.22; DENIES: 0.00) THEN C1 needs repair.

Rule 4: IF S1 needs repair (AFFIRMS: 3.33; DENIES: 0.00) THEN C2 needs repair.

Rule 5: IF S1 needs repair (AFFIRMS: 1.82; DENIES: 0.00) THEN C3 needs repair.

Rule 6: IF S2 needs repair (AFFIRMS: 1.43; DENIES: 0.00) THEN C4 needs repair.

Rule 7: IF S2 needs repair (AFFIRMS: 1.25; DENIES: 0.00) THEN C5 needs repair.

Rule 8: IF C1 needs repair (AFFIRMS: 3.50; DENIES 0.38) AND C2 needs repair (AFFIRMS: 2.00; DENIES 0.57) AND C3 needs repair (AFFIRMS: 0.75; DENIES 1.38) AND C4 needs repair (AFFIRMS: 0.13; DENIES 3.60) AND C5 needs repair (AFFIRMS: 2.13; DENIES 0.25) THEN it is sensible to repair the product.

Table 2. Linguistic P(E|H) ratings

Component	a	b	c	d	e
C1	High	High	Medium	Medium	High
C2	Very High	High	Very High	Medium	Low
C3	Low	Low	Very Low	Very High	Medium
C4	High	High	High	High	Very Low
C5	Medium	High	High	High	Medium

Table 3. Linguistic value conversion table

Linguistic rating	Triangular fuzzy number
Very high (VH)	(0.7, 0.9, 1.0)
High (H)	(0.5, 0.7, 0.9)
Medium (M)	(0.3, 0.5, 0.7)
Low (L)	(0.1, 0.3, 0.5)
Very low (VL)	(0.0, 0.1, 0.3)

5.3. Bayesian updating

Rule 1: H = S1 needs repair; O(H) = 1.50; E = Product needs repair; A = 1.67; O(H|E) = O(H).(A) = 2.51.

Rule 2: H = S2 needs repair; O(H) = 2.33; E = Product needs repair; A = 2.50; O(H|E) = O(H).(A) = 5.83.

Rule 3: H = C1 needs repair; O(H) = 0.82; E = S1 needs repair; O(E) = 2.51; P(E) = 0.72; A = 2.22;
A' = [2(A-1)*P(E)] + 2 – A = 1.54; O(H|E) = O(H).(A') = (0.82).(1.54) = 1.26.

Rule 4: H = C2 needs repair; O(H) = 1.22; E = S1 needs repair; O(E) = 2.51; P(E) = 0.72; A = 3.33;
A' = [2(A-1)*P(E)] + 2 – A = 2.03; O(H|E) = O(H).(A') = (1.22).(2.03) = 2.48.

Rule 5: H = C3 needs repair; O(H) = 0.43; E = S1 needs repair; O(E) = 2.51; P(E) = 0.72; A = 1.82;
A' = [2(A-1)*P(E)] + 2 – A = 1.36; O(H|E) = O(H).(A') = (0.43).(1.36) = 0.58.

Rule 6: H = C4 needs repair; O(H) = 0.47; E = S2 needs repair; O(E) = 5.83; P(E) = 0.85; A = 1.43;
A' = [2(A-1)*P(E)] + 2 – A = 1.30; O(H|E) = O(H).(A') = (0.47).(1.30) = 0.61.

Rule 7: H = C5 needs repair; O(H) = 0.11; E = S2 needs repair; O(E) = 5.83; P(E) = 0.85; A = 1.25.
A' = [2(A-1)*P(E)] + 2 – A = 1.18; O(H|E) = O(H).(A') = (0.11).(1.18) = 0.13.

Rule 8: H = Sensible to repair product; O(H) = 1.50; E1 = C1 needs repair; O(E1) = 1.26; P(E1) = 0.56; A1 = 3.50;
A1' = [2.(A1-1).P(E1)] + 2 – A1 = 1.30;
E2 = C2 needs repair; O(E2) = 2.48; P(E2) = 0.71; A2 = 2.00; A2' = [2.(A2-1).P(E2)] + 2 – A2 = 1.42;
E3 = C3 needs repair; O(E3) = 0.58; P(E3) = 0.37; D3 = 1.38; D3' = [2.(1-D3).P(E3)] +D3 = 1.09;
E4 = C4 needs repair; O(E4) = 0.61; P(E4) = 0.38; D4 = 3.60; D4' = [2.(1-D4).P(E4)] + D4 = 1.88;
E5 = C5 needs repair; O(E5) = 0.13; P(E5) = 0.12; D5 = 0.25; D5' = [2.(1-D5).P(E5)] + D5 = 0.43;

O(H|E1&E2&E3&E4&E5) = O(H).(A1').(A2').(D3').(D4').(D5') = (1.50).(1.30).(1.42).(1.09).(1.88).(0.43) = 2.44;
P(H|E1&E2&E3&E4&E5) = (2.44)/(3.44) = 0.71.

P(sensible to repair the product) = 0.71.

If the cut-off value as decided the by the decision-maker is say, 0.55, he will decide to send the discarded product for repair and subsequent sale on the second-hand market.

5.4. FLEX-based expert system

We employ FLEX shell to build an expert system that can decide if it is "sensible" to repair a particular discarded product for subsequent sale on the second-hand market. Figure 3 shows the user-interface for building the expert system and Figure 4 shows the user-interface for running the expert system.

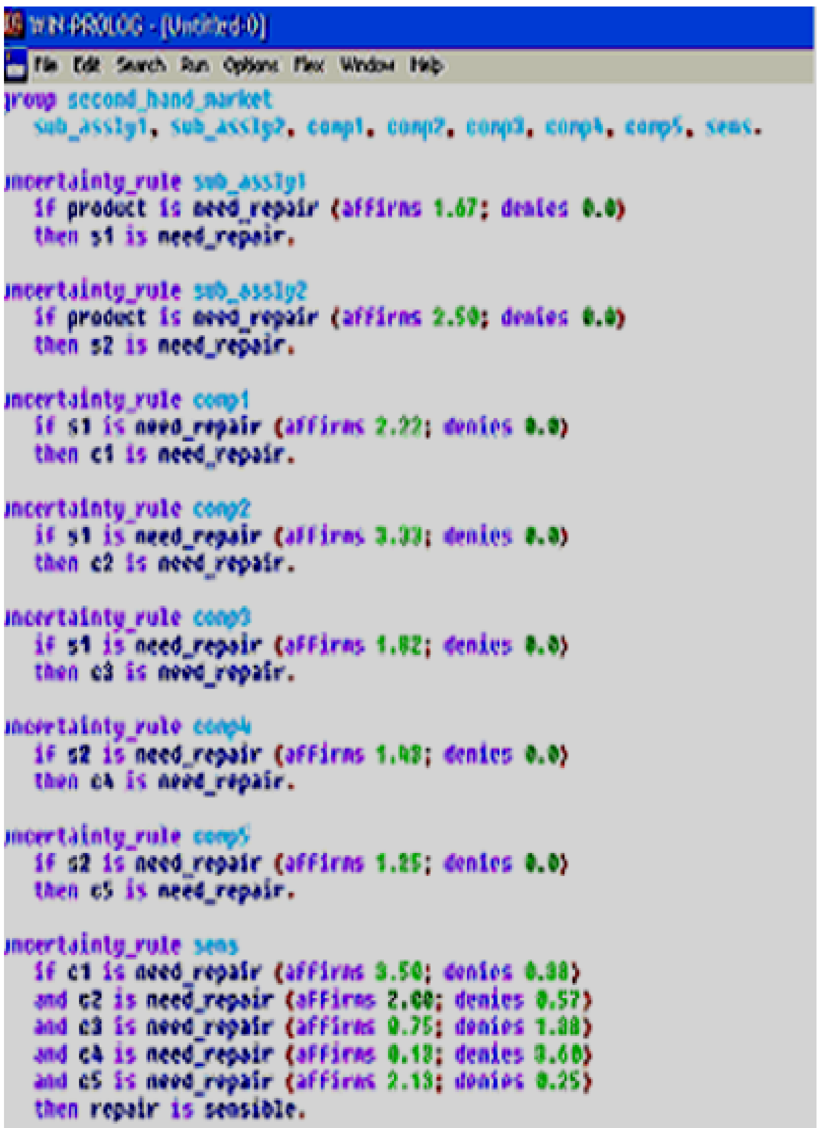
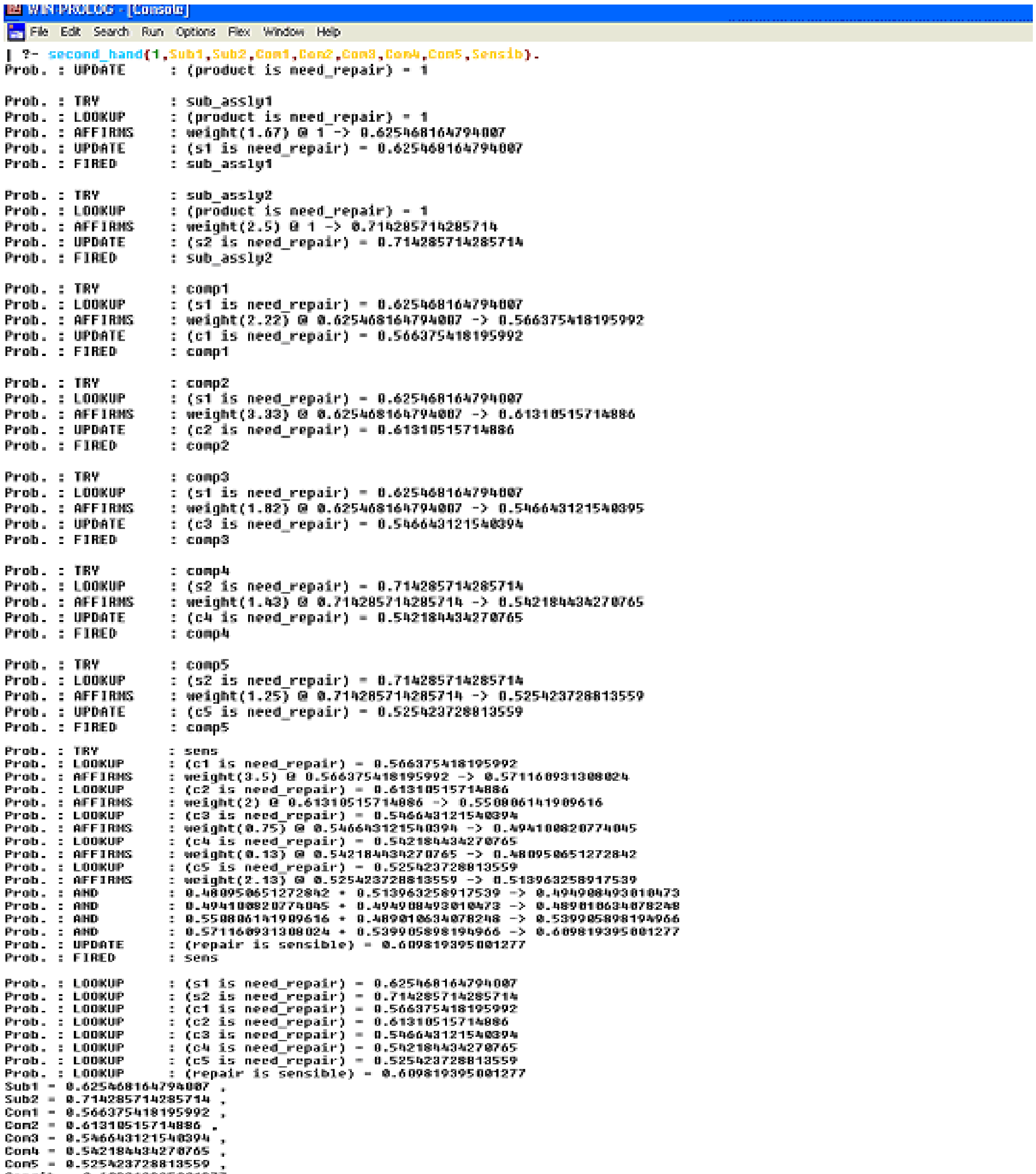


Figure 3. FLEX user interface for building the expert system

The probability that it is “sensible” to repair the product is calculated by the FLEX-based expert system as 0.61. The difference in the probability obtained manually in Section 4 and the one obtained by the expert system in this section is most likely due to the difference in the formulae used to calculate A’ and D’ (they are interpolated values). The user of an expert system shell cannot know how exactly the inference engine of the shell works. When there is a significant difference in the probability values, it is advisable to build the expert system using a knowledge representation language like Lisp or Prolog, rather than using an expert system shell.



```
WIN-PROLOG - [Console]
File  Edit  Search  Run  Options  Flex  Window  Help
| ?- second_hand(1,Sub1,Sub2,Com1,Com2,Com3,Com4,Com5,Sensib).
Prob. : UPDATE      : (product is need_repair) = 1

Prob. : TRY         : sub_assly1
Prob. : LOOKUP      : (product is need_repair) = 1
Prob. : AFFIRMS     : weight(1.67) @ 1 -> 0.625468164794007
Prob. : UPDATE      : (s1 is need_repair) = 0.625468164794007
Prob. : FIRED       : sub_assly1

Prob. : TRY         : sub_assly2
Prob. : LOOKUP      : (product is need_repair) = 1
Prob. : AFFIRMS     : weight(2.5) @ 1 -> 0.714285714285714
Prob. : UPDATE      : (s2 is need_repair) = 0.714285714285714
Prob. : FIRED       : sub_assly2

Prob. : TRY         : comp1
Prob. : LOOKUP      : (s1 is need_repair) = 0.625468164794007
Prob. : AFFIRMS     : weight(2.22) @ 0.625468164794007 -> 0.566375418195992
Prob. : UPDATE      : (c1 is need_repair) = 0.566375418195992
Prob. : FIRED       : comp1

Prob. : TRY         : comp2
Prob. : LOOKUP      : (s1 is need_repair) = 0.625468164794007
Prob. : AFFIRMS     : weight(3.33) @ 0.625468164794007 -> 0.61310515714886
Prob. : UPDATE      : (c2 is need_repair) = 0.61310515714886
Prob. : FIRED       : comp2

Prob. : TRY         : comp3
Prob. : LOOKUP      : (s1 is need_repair) = 0.625468164794007
Prob. : AFFIRMS     : weight(1.82) @ 0.625468164794007 -> 0.546643121540395
Prob. : UPDATE      : (c3 is need_repair) = 0.546643121540394
Prob. : FIRED       : comp3

Prob. : TRY         : comp4
Prob. : LOOKUP      : (s2 is need_repair) = 0.714285714285714
Prob. : AFFIRMS     : weight(1.43) @ 0.714285714285714 -> 0.542184434270765
Prob. : UPDATE      : (c4 is need_repair) = 0.542184434270765
Prob. : FIRED       : comp4

Prob. : TRY         : comp5
Prob. : LOOKUP      : (s2 is need_repair) = 0.714285714285714
Prob. : AFFIRMS     : weight(1.25) @ 0.714285714285714 -> 0.525423728813559
Prob. : UPDATE      : (c5 is need_repair) = 0.525423728813559
Prob. : FIRED       : comp5

Prob. : TRY         : sens
Prob. : LOOKUP      : (c1 is need_repair) = 0.566375418195992
Prob. : AFFIRMS     : weight(3.5) @ 0.566375418195992 -> 0.571160931308024
Prob. : LOOKUP      : (c2 is need_repair) = 0.61310515714886
Prob. : AFFIRMS     : weight(2) @ 0.61310515714886 -> 0.550806141909616
Prob. : LOOKUP      : (c3 is need_repair) = 0.546643121540394
Prob. : AFFIRMS     : weight(0.75) @ 0.546643121540394 -> 0.494100820774045
Prob. : LOOKUP      : (c4 is need_repair) = 0.542184434270765
Prob. : AFFIRMS     : weight(0.13) @ 0.542184434270765 -> 0.480950651272842
Prob. : LOOKUP      : (c5 is need_repair) = 0.525423728813559
Prob. : AFFIRMS     : weight(2.13) @ 0.525423728813559 -> 0.513963258917539
Prob. : AND         : 0.480950651272842 * 0.513963258917539 -> 0.494908493010473
Prob. : AND         : 0.494100820774045 * 0.494908493010473 -> 0.489010634078248
Prob. : AND         : 0.550806141909616 * 0.489010634078248 -> 0.539905898194966
Prob. : AND         : 0.571160931308024 * 0.539905898194966 -> 0.609819395001277
Prob. : UPDATE      : (repair is sensible) = 0.609819395001277
Prob. : FIRED       : sens

Prob. : LOOKUP      : (s1 is need_repair) = 0.625468164794007
Prob. : LOOKUP      : (s2 is need_repair) = 0.714285714285714
Prob. : LOOKUP      : (c1 is need_repair) = 0.566375418195992
Prob. : LOOKUP      : (c2 is need_repair) = 0.61310515714886
Prob. : LOOKUP      : (c3 is need_repair) = 0.546643121540394
Prob. : LOOKUP      : (c4 is need_repair) = 0.542184434270765
Prob. : LOOKUP      : (c5 is need_repair) = 0.525423728813559
Prob. : LOOKUP      : (repair is sensible) = 0.609819395001277
Sub1 = 0.625468164794007 ,
Sub2 = 0.714285714285714 ,
Com1 = 0.566375418195992 ,
Com2 = 0.61310515714886 ,
Com3 = 0.546643121540394 ,
Com4 = 0.542184434270765 ,
Com5 = 0.525423728813559 ,
Sensib = 0.609819395001277
```

Figure 4. FLEX user-interface for running the expert system

6. CONCLUSIONS

It is difficult to decide if it is "sensible" to repair (if necessary) a particular discarded product for subsequent sale on the second-hand market or to disassemble it partially or completely for subsequent remanufacture and/or recycle. To aid making such a decision, in this paper, we built an expert system using Bayesian updating process and fuzzy set theory. We illustrated the building approach using a numerical example.

REFERENCES

1. Chan, F. T., Chan, H. K. and Chan, M. H., "An integrated fuzzy decision support system for multi-criterion decision making problems," *Journal of Engineering Manufacture*, Vol. 217, 11-27, 2003.
2. Chiu, C. and Park, C. S., "Fuzzy cash flow analysis using present worth criterion," *The Engineering Economist*, Vol. 30, No. 2, 1994.
3. Gungor, A. and Gupta, S. M., "Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey," *Computers and Industrial Engineering*, Vol. 36, No. 4, 811-853, 1999.
4. Hopgood, A. A., *Knowledge-based systems for engineers and scientists*, CRC Press, 1993.
5. Pochampally, K. K. and Gupta, S. M., "A multi-phase mathematical programming approach to strategic planning of an efficient reverse supply chain network," *Proceedings of the IEEE International Symposium on the Electronics and the Environment*, 72-78, 2003.
6. Rich, E. and Knight, K., *Artificial Intelligence*, McGraw-Hill, 1992.
7. Tsaur, S., Chang, T. and Yen, C., "The evaluation of airline service quality by fuzzy MCDM," *Tourism Management*, Vol. 23, 107-115, 2002.
8. Vasey, P., Westwood, D. and Johns, N., *Flex Expert System Toolkit*, Logic Programming Associated Ltd., 1996.
9. Wang, M. and Liang, G., "Benefit/cost analysis using fuzzy concept," *The Engineering Economist*, Vol. 40, No. 4, 359-376, 1995.
10. Website: http://www.civeng.nottingham.ac.uk/teaching/h2ci01/kbs_intro.html
11. Zadeh, L. A., "Fuzzy Sets," *Information and Control*, Vol. 8, 338-353, 1965.
12. Zhao, R. and Govind, R., "Algebraic characteristics of extended fuzzy number," *Information Science*, Vol. 54, 103-130, 1991.