Result Fusion in Multi-Agent Systems Based on OWA Operator

Zili Zhang and Chengqi Zhang
School of Computing and Mathematics
Deakin University, Geelong Victoria 3217, Australia
e-mail:{zili, chengqi}@deakin.edu.au

Abstract

In multi-agent systems, each of the agents may have its own expertise. When they are asked to accomplish the same task, the results may be different. In such situations, we need to fuse the results to obtain a final one. In this paper, we propose an information fusion approach when multiple agents are asked to collect the information for the same query from different resources. We also discuss the group decision fusion problem that several agents with similar knowledge make decisions using the same information. Both are based on ordered weighted averaging (OWA) operator.

1 Introduction

In multi-agent systems, each of the agents may have its own knowledge base. They may be delegated the same task. For example, in information retrieval applications, multiple agents may be asked to collect the information for the same query from different resources. In many other applications, each of the agents may make a decision for the same problem using its own knowledge and its inherent subjectivity. In such cases, there is a need to fuse the different results that the agents give.

1.1 Background

In our ongoing project entitled *"Financial Investment Planner Using Intelligent Agent Technologies"*, there are four subsystems: user interface and result visualization agents , decision making agents, information retrieval and information filter agents, and data mining agents. Figure 1 shows the overall framework of this system.

In this system, the user gives his inquiry to the system through the user interface agents. The user interface agents convert the user's inquiry to some kind of internal representation. The decision making agents then ask information retrieval agents to collect some information according to the user's inquiry. The collected information (may be processed by information filter agents and data mining agents) is stored in a database. The decision making agents make decisions according to the user's inquiry, the collected information, and their own domain knowledge. The final alternative decisions are provided to the user by the result visualization agents.

In the decision making subsystem, there are four fusion issues to be solved.

- The first issue is to fuse the possible diverse information retrieved by different agents to single information. All decision making agents use the same information.
- The second issue is to fuse the possible diverse decisions from decision agents even the agents use the same information.
- The third issue is to fuse the diverse decisions when different decision agents use different information (without fusion after retrieval).
- In the fourth issue, there is single information for the agents to use, but each decision making agent uses only part of the information to make decision. In this situation, the diverse part decisions must be fused to form a complete decision.

In this paper, we deal with the first two fusion problems. We propose a method to fuse diverse information collected by multiple agents for the same query from different resources, and discuss the group decision fusion problem that multiple agents use the same fused information. Both are based on Ordered Weighted Averaging (OWA) operator. Our work is another successful application of OWA operator. The implementation technique presented in this paper is flexible, it can be used in other applications as well.

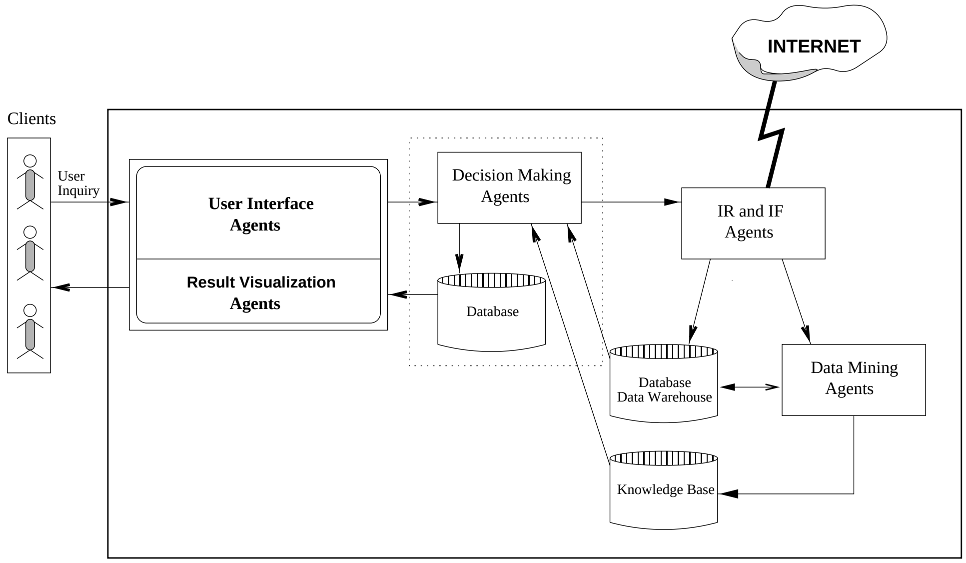

Figure 1. Framework of Financial Investment Planner

1.2 Problem description

Suppose the user (client) wants to invest some money in some fields. Meanwhile, the user also has some constraints on his investment. We represent the user's constraints as $C = \{C_1, C_2, \ldots, C_m\}$, where C_1:*Risk is medium*; C_2:*Gain is high*; C_3:*Term is short* etc.

First, the user gives his *annual income*(AI) and *total networth*(TNW) as well as the constraint set C to the decision making agents through the user interface agents. The decision making agents use their own knowledge to evaluate user's *risk tolerance* (RT) ability.

The decision making agents then delegate the information retrieval agents to collect data concerning the falling or rising of interest rates, the state of the stock market, the trade balance, unemployment rate, level of inventory stockage, etc. We represent these data as a parameter set $P = \{P_1, P_2, \ldots\}$. The parameters collected by different information retrieval agents may be different. So we need to fuse these parameters collected by different information retrieval agents.

The decision making agents then give suggestions on which field the user should invest and the corresponding **Risk, Gain** levels according to the fused parameters and user's risk tolerance ability.

Assume there are k parameters to be collected: $P = \{P_1, P_2, \ldots, P_k\}$, and n information retrieval agents are asked to collect the k parameters independently. The retrieved results are $\{P_{i1}, P_{i2}, \ldots, P_{ik}\}$ $(i = 1, 2, \ldots, n)$.

The fusion problem is to combine $\{P_{i1}, P_{i2}, \ldots, P_{ik}\}$ $(i = 1, 2, \ldots, n)$ together in some reasonable way to obtain $P = \{P_1, P_2, \ldots, P_k\}$.

Assume $DA = \{DA_1, DA_2, \ldots, DA_M\}$ is the set of decision making agents with similar knowledge, $A = \{A_1, A_2, \ldots, A_N\}$ denotes the set of alternative decisions. The decision fusion problem we consider here is to fuse all the alternatives in some way to decide which alternative mostly satisfies the constraint set C.

Because there is much fuzzy information in our application, we need a fusion method that is able to deal with fuzzy information. After we compared several kinds of fuzzy operators for fusion such as fuzzy averaging, weighted MIN, and OWA etc. We find out that OWA is more flexible. The OWA operator can take into account as much information as possible in the fusion process. For example, the fuzzy averaging method can not take the decision maker's attitude into account, but the OWA operator can. This implies that the fused results using OWA are more reliable. This is very important for our financial investment application. That is why we choose OWA in our application.

The rest of the paper is organized as follows: Section 2 gives a brief introduction of OWA operator. Section 3 proposes the information fusion issue using OWA operator. Section 4 discusses the group decision fusion problem. Section 5 is some related work. Section 6 discusses the implementation of OWA fusion in multiagent systems. Finally, Section 5 is the concluding remarks and future work.

2 The OWA operator

Yager introduced the ordered weighted averaging (OWA) operator to provide a family of aggregators having the properties of mean operators[1]. Here, we will briefly provide an introduction to the ordered weighted averaging (OWA) operator[1] [4] [5] [6] [9].

Definitions: A mapping $F : R^n \longrightarrow R$ is called an OWA operator of dimension n if it has an associated weighting vector W of dimension n such that its components satisfy

(1) $w_j \in [0,1]$

(2) $\sum_{j=1}^{n} w_j = 1$

and

$$F_w(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j$$

where b_j is the jth largest of the a_i.

A fundamental feature of this operator is the reordering process which associates the arguments with the weights. This process introduces a nonlinearity into the aggregation process. It should be observed that we can express this aggregation in a vector notation as

$$F_w(a_1, a_2, \ldots, a_n) = W^T B$$

In this expression, W is the OWA weighting vector associated with the aggregation, and B is the ordered argument vector; where the jth component in B, b_j is the jth largest of the a_i.

This operator can easily be seen to be a mean operator in that it is commutative, monotone, and is always bounded by the max and min of the arguments

$$Min_i[a_i] \leq F_w(a_1, a_2, \ldots, a_n) \leq Max_i[a_i]$$

It can be seen that this is idempotent, $F_w(a_1, a_2, \ldots, a_n) = a$.

Expressing the OWA operator $F_w(a_1, a_2, \ldots, a_n)$ in its vector notation form $W^T B$ makes very clear the distinct components involved in the performance of this operation. First, we have a weighting vector W; this is required to have components w_j which lie in the unit interval and sum to one. The second part of the OWA aggregation is the vector B, called the ordered argument vector. This vector is composed of the arguments of the aggregation.

To solve a specific problem using OWA operator, we need to find out the appropriate weighting vector W and the ordered argument vector B.

Generally, the process of obtaining the vector B can be seen as an assigning operation[4]. It is an assigning operation in the sense that it assigns an argument element to a component value in W. There are three general approaches for obtaining W. The first is based upon a learning of the weights from data. An algorithm for learning the OWA weights from data is discussed in [7]. The second approach, which is based upon the close connection between OWA operators and linguistic quantifiers, uses linguistic elements to generate the weight, and is discussed in [3]. The third approach makes use of a single parameter, such as the α measure [8], to obtain the OWA weights.

3 Information fusion

Recall what we have described in Section 1.2 and Section 2. We represent our information fusion problem as the following $n \times k$ matrix.

$$\begin{bmatrix} P_{11} & P_{12} & \ldots & P_{1k} \\ P_{21} & P_{22} & \ldots & P_{2k} \\ \vdots & \vdots & \ldots & \vdots \\ P_{n1} & P_{n2} & \ldots & P_{nk} \end{bmatrix}$$

$$\begin{matrix} \downarrow & \downarrow & \ldots & \downarrow \\ P_1 & P_2 & \ldots & P_k \end{matrix}$$

In this matrix,

$$P_1 = F_w(P_{11}, P_{21}, \ldots, P_{n1})$$

$$P_2 = F_w(P_{12}, P_{22}, \ldots, P_{n2})$$

$$\vdots$$

$$P_k = F_w(P_{1k}, P_{2k}, \ldots, P_{nk})$$

where F_w is an OWA operator.

Now we discuss how to obtain the weighting vector W and ordered argument vector B used in this problem.

3.1 Ordered argument vector

In many practical applications, when an agent provides its retrieved information, it usually associates its own measure of confidence with the information. One can consider the typical situation of paper-review. The referees are usually required to give their confidence of their evaluation for papers. We use u_i ($i = 1, 2, \ldots, n$) to denote the confidence degree of information retrieval agent i about its collected parameters $(P_{i1}, P_{i2}, \ldots, P_{ik})$.

We adopt a more generalization form of OWA operator introduced by Yager[4]. The idea is to associate each of the arguments with an order-inducing variable. In this situation, each of the arguments consists of a two-tuple called an OWA pair which we will denote as $\langle u_i, P_{ij} \rangle$. In this situation, u_i is called the order-inducing variable, and P_{ij} is called the argument variable. Consider the following aggregation of OWA pairs:

$$F_w(\langle u_1, P_{1l} \rangle, \langle u_2, P_{2l} \rangle, \ldots, \langle u_n, P_{nl} \rangle)$$

We can use the following procedure to perform the fusion:

- Let b_j be the P_{il} value which has the jth largest of u_i.
- $F_w(\langle u_1, P_{1j} \rangle, \langle u_2, P_{2j} \rangle, \ldots, \langle u_n, P_{nj} \rangle) = W^T B$.

Here, we order the arguments with respect to the u_i value, but perform the fusion using the P_{ij} values.

Example: Recall the discussion in Section 1.2. Assume there are five parameters concerning *the falling or rising of interest rates, the state of the stock market, the trade balance, unemployment rate, level of inventory stockage*, respectively. Each of the arguments can take a value from $\{1, 2, 3, 4, 5\}$. (For interest rates, for example, 1 may indicate falling much, 2 falling a little, 3 not falling or rising, 4 rising a little, and 5 rising much.) Suppose we have four information retrieval agents to collect the five parameters independently. Each information retrieval agent gives a confidence degree u_i for its retrieved results, u_i also takes value from 1 (very low confidence) to 5 (very high confidence).

Given

$$P_{11} = 4, P_{12} = 3, P_{13} = 3, P_{14} = 2, P_{15} = 3$$

$$P_{21} = 3, P_{22} = 4, P_{23} = 3, P_{24} = 1, P_{25} = 2$$

$$P_{31} = 2, P_{32} = 3, P_{33} = 4, P_{34} = 4, P_{35} = 4$$

$$P_{41} = 5, P_{42} = 4, P_{43} = 4, P_{44} = 5, P_{45} = 5$$

$$u_1 = 4, u_2 = 2, u_3 = 5, u_4 = 3$$

$$\begin{aligned} W &= [w_1, w_2, w_3, w_4] \\ &= [0.4614, 0.2756, 0.1646, 0.0984]. \end{aligned}$$

we then obtain:

$$\begin{aligned} P_1 &= F_w(\langle u_1, P_{11} \rangle, \langle u_2, P_{21} \rangle, \ldots, \langle u_4, P_{41} \rangle) \\ &= P_{31} \times w_1 + P_{11} \times w_2 + \\ &\quad P_{41} \times w_3 + P_{21} \times w_4 \\ &= 2 \times 0.4614 + 4 \times 0.2756 + \\ &\quad 5 \times 0.1646 + 3 \times 0.0984 \\ &= 3.1434 \end{aligned}$$

$$\begin{aligned} P_2 &= F_w(\langle u_1, P_{12} \rangle, \langle u_2, P_{22} \rangle, \ldots, \langle u_4, P_{42} \rangle) \\ &= P_{32} \times w_1 + P_{12} \times w_2 + \\ &\quad P_{42} \times w_3 + P_{22} \times w_4 \\ &= 3.263 \end{aligned}$$

$$\begin{aligned} P_3 &= F_w(\langle u_1, P_{13} \rangle, \langle u_2, P_{23} \rangle, \ldots, \langle u_4, P_{43} \rangle) \\ &= P_{33} \times w_1 + P_{13} \times w_2 + \\ &\quad P_{43} \times w_3 + P23 \times w_4 \\ &= 3.626 \end{aligned}$$

$$\begin{aligned} P_4 &= F_w(\langle u_1, P_{14} \rangle, \langle u_2, P_{24} \rangle, \ldots, \langle u_4, P_{44} \rangle) \\ &= P_{34} \times w_1 + P_{14} \times w_2 + \\ &\quad P_{44} \times w_3 + P_{24} \times w_4 \\ &= 3.3182 \end{aligned}$$

$$\begin{aligned} P_5 &= F_w(\langle u_1, P_{15} \rangle, \langle u_2, P_{25} \rangle, \ldots, \langle u_5, P_{45} \rangle) \\ &= P_{35} \times w_1 + P_{15} \times w_2 + \\ &\quad P_{45} \times w_3 + P_{25} \times w_4 \\ &= 3.6922 \end{aligned}$$

It should be noted that since the u_i's, the order indexing values, do not take part in the matrix multiplication, they are only used to help order the arguments in B; they need not be numbers. Because of this, they can be objects drawn from any scale which has a linear ordering. In our example, the u_i can take values such as *very high, high, medium, low* etc. It is more natural for some practical applications in such a way.

3.2 The Weighting vector

When we fuse the information, we always want to take into account the information in the aggregation as much as possible. Meanwhile, we prefer information of relatively high confidence, i. e., prefer elements at the top of the B vector. To this end, we use O'Hagan's approach[8] to obtain the weighting vector W. Before we go on, we need to introduce two characterizing measures associated with the weighting vector W[1][4].

The first of these, the α value of an OWA operator, is defined as

$$\alpha(W) = \frac{1}{n-1} \sum_{j=1}^{n} w_j (n - j)$$

This measure, which takes its values in the unit interval, is determined by the weighting used in the aggregation. The semantics of α in our information fusion problem is the degree that the aggregation prefers information with high confidence.

The second measure is called the dispersion (or entropy) of W, and is defined as

$$H(W) = -\sum_{j=1}^{n} w_j \ln(w_j).$$

It was shown that this helps measure the degree to which W takes into account all of the information in the aggregation. For our information fusion situation, we hope to take into account the information in the aggregation as much as possible, i.e., maximize $H(W)$.

O'Hagan's method to determine these weights, $w_1, \ldots, w_n$, requires the solution of the following mathematical programming problem.

Maximize

$$-\sum_{j=1}^{n} w_j \ln(w_j)$$

subject to

(1)$\alpha(W) = \frac{1}{n-1}\sum_{j=1}^{n} w_j(n-j)$
(2)$w_j \in [0,1]$
(3)$\sum_{j=1}^{n} w_j = 1$

Suppose the degree to which the fusion prefers high confidence information is 70%, i.e., $\alpha = 0.7$. By solving the above mathematical programming problem with $\alpha = 0.7$, we can obtain the weighting vector $W = [w_1, w_2, w_3, w_4] = [0.4614, 0.2756, 0.1646, 0.0984]$. If $\alpha = 0.8$, we get the the weighting vector $W = [w_1, w_2, w_3, w_4] = [0.5965, 0.2520, 0.1065, 0.045]$.

An *Operations Research* software package called **LINDO** (*http://www.lindo.com/*) is used to solve this mathematical programming problem.

4 Group decision fusion

Recall the user's constraints $C = \{C_1, C_2, \ldots, C_m\}$. C_i are represented by unconditional fuzzy propositions. The corresponding canonical forms [10][11] are

$$\begin{array}{c} X_1 \ is \ E_1 \\ X_2 \ is \ E_2 \\ X_3 \ is \ E_3 \\ \vdots \end{array}$$

where $X_1 = Risk, X_2 = Gain, X_3 = Term$ are variables, X_1 and X_2 take values from $\{very\ high, high, medium, low, very\ low, \ldots\}$; $E_1 = medium, E_2 = high, E_3 = short$ are fuzzy subsets of $[0,1]$.

Now the decision making agent DA_i can use the fused parameters $P = \{P_1, P_2, \ldots, P_k\}$, the user's risk tolerance ability, and its individual knowledge base to infer a result, i.e., an alternative A_i which contains the values of $\{X_1, X_2, X_3, \ldots\}$. Because $DA = \{DA_1, DA_2, \ldots, DA_M\}$, there are M alternatives $A = \{A_1, A_2, \ldots, A_M\}$, where

$$A_1 = \begin{bmatrix} X_1 \ is \ E_{11} \\ X_2 \ is \ E_{21} \\ \vdots \\ X_m \ is \ E_{m1} \end{bmatrix}, \quad A_2 = \begin{bmatrix} X_1 \ is \ E_{12} \\ X_2 \ is \ E_{22} \\ \vdots \\ X_m \ is \ E_{m2} \end{bmatrix}, \quad \vdots \quad A_M = \begin{bmatrix} X_1 \ is \ E_{1M} \\ X_2 \ is \ E_{2M} \\ \vdots \\ X_m \ is \ E_{mM} \end{bmatrix}.$$

We then compute for each pair (E_{ij}, E_j) the distance–the degree of consistency, which indicates the degree of "closeness" between DA_i's decision and the user's constraints. There exist several kinds of interpretations of the "distance"[11]. Here, we use the following definition of the degree of consistency, γ_{ij}.

$$\gamma_{ij} = max\{min[E_{ij}(x_j), E_j(x_j)]\},$$

where $E_{ij}(x_j)$ or $E_j(x_j)$ is a number–the membership value of fuzzy subset E_{ij} or E_j at the sample point x_j.

We store all the $1-\gamma_{ij}$ in a distance matrix G, $g_{ij} = 1-\gamma_{ij}$.

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{km} \end{bmatrix}$$

Now we want to identify the alternatives that mostly satisfy the user's constraints C. We still use OWA operator F_w. What we need to do is to calculate $F_w(g_{i1}, g_{i2}, \ldots, g_{im})$ and provide the user with largest F_w. In the following, we discuss how to choose the weighting vector W and ordered argument vector B in this situation.

4.1 Ordered argument vector

When the user inquires, the interface agents ask the user to provide the important degrees of the constraints C. We still represent the important degree as u_i. Then let b_j be the g_{il} value which has the jth largest of u_i, i.e., the distance with the jth important constraint value. The rest is similar to what we described in Section 3.1.

4.2 The Weighting Vector

The choice of the weighting vector W should depend upon the fused parameters and decision making agent's knowledge involved in the particular decision (the inferred result). With this in mind, we use the technology of approximate reasoning[10] to determine the weighting vector.

Let $L_1, L_2, \ldots, L_q$ be a collection of fuzzy subsets representing linguistic terms describing monetary losses of the investment. In each decision making agent's knowledge base, there are rules such as

$$IF\ P_1 = h_1\ and\ P_2 = h_2\ and\ \ldots$$
$$and\ P_k = h_k\ THEN\ V\ is\ L_i$$

$$IF\ V\ is\ L_1\ THEN\ \alpha_i\ is\ a_1$$
$$IF\ V\ is\ L_2\ THEN\ \alpha_i\ is\ a_2$$
$$\vdots$$
$$IF\ V\ is\ L_q\ THEN\ \alpha_i\ is\ a_q$$

Here, h_i are some specific values of fused parameters. α_i denotes the attitude of decision maker i based on its own knowledge and the retrieved parameters, V is the variable *worst possible loss in this decision*, and $a_i \in [0, 1]$. The final aggregation should depend on the combined attitude of all the decision makers (called α) other than the individual α_i. So we need to combine α_i in some way to obtain α.

Consider the decision making agents play different roles in the decision making process. That means they have different *degrees of importance*. In this case, we still use the OWA operator. We use the degrees of importance to construct the ordered argument vector B.Then we map the degrees of importance into unit interval as the weighting vector, denoted as $W(\alpha)$, to compute the α value (the attitude of M decision makers) as follows.

$$\alpha = F_{w(\alpha)}(\alpha_1, \alpha_2, \ldots, \alpha_M)$$

Obviously, if the decision making agents play equal role in this decision making process,i. e., $W(\alpha) = [1/M, 1/M, \ldots, 1/M]$, the α value can be determined simply as

$$\alpha = \sum_{i=1}^{M} \alpha_i / M$$

After obtaining the α value, we can use O'Hagan's approach (refer to Section 3.2) to decide the weighting vector W used in the final aggregation.

5 Related work

Yager first introduced the OWA operator in decision making. His work[1]is about single agent multi-criteria decision making. Yager also discussed the decision making under *ignorance*[4] to motivate the study of context-dependent OWA weights. In [2], the authors discussed the consensus of multi-agent multi-criteria decision problem based on OWA operators. The emphasis of Ref.[2] is on how to evaluate a consensual judgment and a consensus degree on each alternative given a fuzzy proposition such as *most of the important criteria are satisfied.*

6 Experiment

In Sections 3 and 4, we generally discussed the fusion problems using OWA operator. In this section, we discuss the approach to implement the OWA fusion in our multi-agent system.

The approach is to add a specific fusion agent in our decision making subsystem. All the decision making agents or information retrieval agents inform fusion agent by sending a KQML[12] (Knowledge Query and Manipulation Language–A widely used agent communication language) message with their decisions, degrees of importance or confidence, and decision attitudes etc. We wrote an interpreter to process these KQML messages. The experiment system is under the support of JATLite (Java Agent Template Lite is a set of lightweight Java packages being developed at Stanford University that can be used to build multi-agent systems. For more information on JATLite, see *http://java.stanford.edu/java_agent.*) Figure 2 shows the architecture.

To use this system, all the decision making agents and the fusion agent must register and connect to the JATLite router first. Currently, we used three decision making agents and three information retrieval agents to test the experiment system. It works well.

7 Conclusions and future work

In this paper, we discussed the fusion problems of multi-agent retrieved information and multi-agent multi-criteria decisions using OWA operator. Our work extends the application areas of OWA operators. We

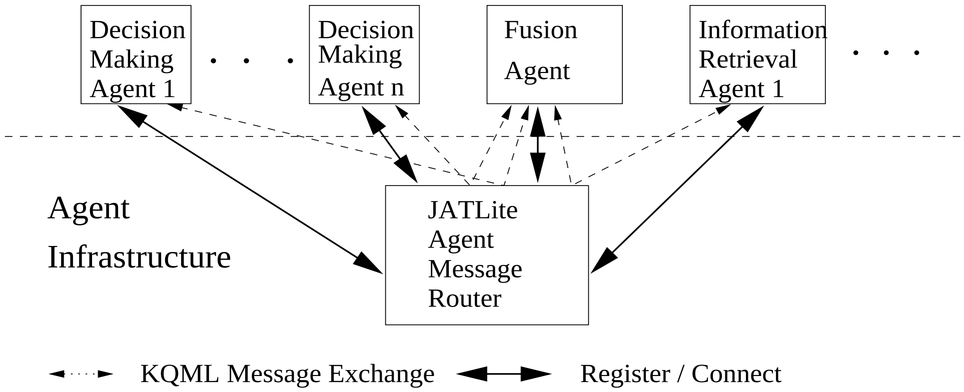

Figure 2. Architecture with Fusion Agent

used the approximate reasoning technology to determine the context-dependent weighting vector. Such fusions take into account as much as information in practical applications and reflect the very nature of human agent behavior.

We also discussed how to implement such fusions in multi-agent environments. Our approach is flexible to add other fusion techniques in the fusion agent.

Recall that there are usually four options for the multi-agent multi-criteria decision fusion. We only discussed the first two. The rest two need further research.

References

[1] R. R. Yager, On Ordered Weighted Averaging Aggregation Operators in Multi-criteria Decision Making, IEEE Trans. Syst., Man, Cybern., Vol. 18, Jan/Feb, 1988, 183-190.

[2] G. Bordogna, M. Fedrizzi, and G. Pasi, A Linguistic Modeling of Consensus in Group Decision Making Based on OWA Operators, IEEE Trans. Syst., Man, Cybern. (Part A), Vol. 27, No. 1, 1997, 126-132.

[3] R. R. Yager, Interpreting Linguistically Quantified Propositions, Int. J. of Intelligent Systems, Vol. 9, 1994, 541-569.

[4] R. R. Yager, New Models of OWA Information Fusion, Int. J. of Intelligent Systems, Vol. 13, 1998, 661-681.

[5] P. A. Schaefer and H. B. Mitcell, A Generalized OWA Operator, Int. J. of Intelligent Systems, Vol. 14, 1999, 123-143.

[6] H. B. Mitcell and D. D. Estrakh, An OWA Operator with Fuzzy Ranks, Int. J. of Intelligent Systems, Vol. 13, 1998, 69-81.

[7] D. P. Filev and R. R. Yager, On the Issue of Obtaining OWA Operator Weights, Fuzzy Sets and Systems, Vol. 94, 1998, 157-169.

[8] M. O'Hagan, Using Maximum Entropy-Ordered Weighted Averaging to Construct a Fuzzy Neuron, Proc. 24nd Annual IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, 1990, 618-623.

[9] R. R. Yager and J. Kacprzyk (Eds.), The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer Academic, Norwell, MA, 1997.

[10] L. A. Zadeh, The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems, in: Approximate Reasoning in Expert Systems, North-Holland, Amsterdam, 1985, 3-31.

[11] Z. L. Zhang, An Approximate Reasoning System: Design and Implementation, Int. J. of Approximate Reasoning, Vol.10, No.9, 1993, 315-326.

[12] T. Finin, Y. Labrou and J. Mayfield, KQML as an Agent Communication Language, in *J. M. Bradshaw (ed.), Software Agents*, AAAI Press/ The MIT Press, Menlo Park, CA, pp.291-316, 1997.