Soft Computing 6 (2002) 320–328 © Springer-Verlag 2002
DOI 10.1007/s00500-002-0185-7

A communication model based on the 2-tuple fuzzy linguistic representation for a distributed intelligent agent system on Internet

M. Delgado, F. Herrera, E. Herrera-Viedma, M. J. Martin-Bautista, L. Martínez, M. A. Vila

Abstract Internet users are assisted by means of distributed intelligent agents in the information gathering process to find the fittest information to their needs. In this paper we present a distributed intelligent agent model where the communication of the evaluation of the retrieved information among the agents is carried out by using linguistic operators based on the 2-tuple fuzzy linguistic representation as a way to endow the retrieval process with a higher flexibility, uniformity and precision. The 2-tuple fuzzy linguistic representation model allows to make processes of computing with words without loss of information.

Keywords Internet, Information retrieval, Intelligent agents, Computing with words, Linguistic modelling

1
Introduction

In the framework of the information retrieval one of the most current problems nowadays for which the fuzzy linguistic approach may be very useful, is the retrieval, handling and identification of relevant information through the Internet.

The fuzzy linguistic approach is an approximate technique, which represents qualitative aspects as linguistic values by means of linguistic variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language [36]. This approach has been applied successfully to different areas as economics [14, 31], planning [1], decision-making [12, 32], information retrieval [3, 9, 18, 19], etc.

Intelligent agents [4, 11, 26, 29, 34] deal with the information gathering process assisting the Internet users to find the fittest information to their needs. Several proposals about intelligent software agents have been emerging in the recent last years, but the lack of connection, communication and consensus among them have lead to a decrease in the quality and suitability of the retrieved information besides the efficiency of the system in the recovering and filtering task. This fact keeps the need of proposals in the field, and emphasizes the importance of the design and development of intelligent software agent organisations, as well as hierarchies and architectures that hold up such structures [7, 10, 21, 28, 29].

However, not only is needed some organization, but also a protocol of communication among the agents. The great variety of representations and evaluations of the information in the Internet is the main obstacle to this communication, and the problem becomes more noticeable when the user takes part in the process. The complexity of all these processes reveals the need of more flexibility in the communication among agents and between agents and the user [9, 34, 35]. For this purpose, several approaches related to mechanisms to introduce and handle flexible information through linguistic labels have been proposed both at levels of agents and users [8, 33].

The main drawback of these approaches is the lack of precision in the final results, due to the fact that appears a loss of information in the processes of computing with words (CW). To overcome this drawback in [15] was presented a linguistic computational model based on linguistic 2-tuples which provides a computational technique to deal with linguistic information in a precise way.

In this paper, we present a distributed intelligent agent model for information gathering on the Internet, where the communication of the evaluation of the retrieved information among the agents is carried out by using linguistic operators based on the 2-tuple fuzzy linguistic representation as a way to endow the retrieval process with a higher flexibility, uniformity and precision.

In order to do so, the paper is structured as follows. Section 2 presents a short review of the fuzzy linguistic approach and the 2-tuple fuzzy linguistic representation model together with its operational resources. Section 3 shows the structure of the distributed intelligent agent model which uses the 2-tuple operational model for information gathering. Section 4 presents an example for illustrating the proposal. Finally, some concluding remarks are pointed out.

2
Linguistic information

In this section we shall make a brief review of the fuzzy linguistic approach and of the 2-tuple fuzzy linguistic representation model.

2.1
Fuzzy linguistic approach

Usually, we work in a quantitative setting, where the information is expressed by means of numerical values.

M. Delgado, F. Herrera, E. Herrera-Viedma,
M. J. Martin-Bautista, M. A. Vila
Department of Computer Science and Artificial Intelligence,
University of Granada, 18071 – Granada, Spain

L. Martínez (✉)
Department of Computer Science,
University of Jaén, 23071 – Jaén, Spain
e-mail: martin@ujaen.es

However, many aspects of different activities in the real world cannot be assessed in a quantitative form, but rather in a qualitative one, i.e., with vague or imprecise knowledge. In that case a better approach may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables [36]. This approach is adequate in some situations where the information may be unquantifiable due to its nature, and thus, it may be stated only in linguistic terms.

We have to choose the appropriate linguistic descriptors for the term set and their semantics. In order to accomplish this objective, an important aspect to analyse is the *"granularity of uncertainty"*, i.e., the level of discrimination among different counts of uncertainty. Typical values of cardinality used in the linguistic models are odd ones, such as 7 or 9, where the mid term represents an assessment of "approximately 0.5", and with the rest of the terms being placed symmetrically around it [2].

One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined. For example, a set of seven terms S, could be given as follows:

$$S = \{s_0 = N, s_1 = VL, s_2 = L, s_3 = M, s_4 = H, \\ s_5 = VH, s_6 = P\} \ .$$

Usually, in these cases, it is required that in the linguistic term set there exist:

1. A negation operator: $\mathrm{Neg}(s_i) = s_j$ such that $j = g - i$ ($g + 1$ is the cardinality).
2. $s_i \leq s_j \iff i \leq j$. Therefore, there exists a *min* and a *max* operator.

The semantics of the linguistic terms is given by fuzzy numbers defined in the [0, 1] interval. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function [2]. The linguistic assessments given by the users are just approximate ones, some authors consider that linear trapezoidal membership functions are good enough to capture the vagueness of such linguistic assessments. The parametric representation is achieved by the 4-tuple (a, b, d, c), where b and d indicate the interval in which the membership value is 1, with a and c indicating the left and right limits of the definition domain of the trapezoidal membership function [2].
A particular case of this type of representation are the linguistic assessments whose membership functions are triangular, i.e., $b = d$, then we represent this type of membership functions by a 3-tuple (a, b, c). An example may be the following (Fig. 1):

$$N = (0, 0, 0.17) \quad VL = (0, 0.17, 0.33)$$

$$L = (0.17, 0.33, 0.5) \quad M = (0.33, 0.5, 0.67)$$

$$H = (0.5, 0.67, 0.83) \quad VH = (0.67, 0.83, 1)$$

$$P = (0.83, 1, 1) \ .$$

Other authors use a non-trapezoidal representation, e.g., Gaussian functions [3, 19].

In the literature we can find different linguistic computational models to accomplish the processes of CW:

- *The approximative computational model based on the Extension Principle* [2, 5]. This model uses fuzzy arithmetic based on the Extension Principle to make computations over the linguistic variables. This model can present the results in two ways:
  1. By means of the fuzzy numbers obtained from the fuzzy arithmetic computations based on the Extension Principle.
  2. Or by means of linguistic labels computed from the fuzzy numbers obtained using a linguistic approximation process.
- *The ordinal linguistic computational model* [6]. This symbolic model makes direct computations on labels, using the ordinal structure of the linguistic term sets. Its results are inherently linguistic labels due to either the operators used, basically max and min operators [32] or because in the computations on the order index there exist an approximation by means of the *round operator* [12].
- *The 2-tuple fuzzy linguistic computational model* [15]. It uses the 2-tuple fuzzy linguistic representation model and its characteristics to make linguistic computations, obtaining as results linguistic 2-tuples. A linguistic 2-tuple is defined by a pair of values, where the first one is a linguistic label and the second one is a real number that represents the value of the symbolic translation. The symbolic translation is the basic concept of the 2-tuple fuzzy linguistic representation model that will be introduced in the following section.

2.2
The 2-tuple fuzzy linguistic representation model based on the symbolic translation

This model and its applications has been presented in [15–17], showing different advantages of this formalism for representing the linguistic information over classical models, such as:

1. The linguistic domain can be treated as continuous, while in the classical models it is treated as discrete.
2. The linguistic computational model based on linguistic 2-tuples carries out processes of computing with words easily and without loss of information.
3. The results of the processes of computing with words are always expressed in the initial expression domain.
4. It is possible to aggregate multigranular linguistic information easily.

Due to these advantages, we shall use this linguistic representation model to accomplish our objective: a higher flexibility, uniformity and precision in the retrieval process.

2.2.1
The 2-tuple fuzzy linguistic representation model

Let $S = \{s_0, \ldots, s_g\}$ be a linguistic term set, if a symbolic method aggregating linguistic information obtains a value $\beta \in [0, g]$, and $\beta \notin \{0, \ldots, g\}$ then an approximation function is used to express the index of the result in S.

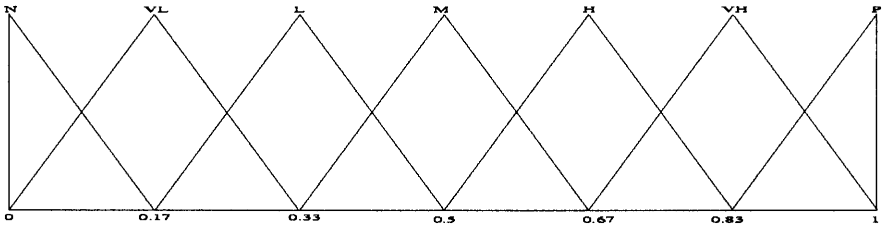

Fig. 1. A set of seven linguistic terms with its semantics

Definition 1. *Let β be the result of an aggregation of the indexes of a set of labels assessed in a linguistic term set S, i.e., the result of a symbolic aggregation operation. $\beta \in [0, g]$, being $g + 1$ the cardinality of S. Let $i = round(\beta)$ and $\alpha = \beta - i$ be two values, such that, $i \in [0, g]$ and $\alpha \in [-0.5, 0.5)$ then α is called a Symbolic Translation.*

Roughly speaking, the symbolic translation of a linguistic term, s_i, is a numerical value assessed in $[-0.5, 0.5)$ that supports the "difference of information" between a counting of information $\beta \in [0, g]$ obtained after a symbolic aggregation operation and the closest value in $\{0, \ldots, g\}$ that indicates the index of the closest linguistic term in $S(i = round(\beta))$.

From this concept we shall develop a linguistic representation model which represents the linguistic information by means of 2-tuples (s_i, α_i), $s_i \in S$ and $\alpha_i \in [-0.5, 0.5)$:

- s_i represents the linguistic label of the information, and
- α_i is a numerical value expressing the value of the translation from the original result β to the closest index label, i, in the linguistic term set (s_i), i.e., the Symbolic Translation.

This model defines a set of transformation functions between numeric values and 2-tuples.

Definition 2. *Let $S = \{s_0, \ldots, s_g\}$ be a linguistic term set and $\beta \in [0, g]$ a value representing the result of a symbolic aggregation operation, then the* 2-tuple that expresses the equivalent information to *β is obtained with the following function*:

$$\Delta : [0, g] \longrightarrow S \times [-0.5, 0.5)$$

$$\Delta(\beta) = (s_i, \alpha), \quad with \begin{cases} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-0.5, 0.5) \end{cases}$$

where $round(\cdot)$ is the usual round *operation, s_i has the closest index label to "β" and "α" is the value of the symbolic translation.*

Proposition 1. *Let $S = \{s_0, \ldots, s_g\}$ be a linguistic term set and (s_i, α) be a 2-tuple. There is always a Δ^{-1} function, such that, from a 2-tuple it returns its equivalent numerical value $\beta \in [0, g] \subset \mathcal{R}$.*

Proof. It is trivial, we consider the following function:

$$\Delta^{-1} : S \times [-0.5, 0.5) \longrightarrow [0, g]$$

$$\Delta^{-1}(s_i, \alpha) = i + \alpha = \beta$$

Remark: From Definitions 1 and 2 and from proposition 1, it is obvious that the conversion of a linguistic term into a linguistic 2-tuple consist of adding a value 0 as symbolic translation:

$$s_i \in S \Longrightarrow (s_i, 0)$$

2.2.2
Linguistic computational model based on linguistic 2-tuples

In this subsection, we present a computational technique to operate with the 2-tuples without loss of information. We shall present the following computations and operators:

1. Comparison of 2-tuples. The comparison of linguistic information represented by 2-tuples is carried out according to an ordinary lexicographic order.

Let (s_k, α_1) and (s_l, α_2) be two 2-tuples, with each one representing a counting of information:

- if $k < l$ then (s_k, α_1) is smaller than (s_l, α_2)
- if $k = l$ then
  1. if $\alpha_1 = \alpha_2$ then (s_k, α_1), (s_l, α_2) represents the same information
  2. if $\alpha_1 < \alpha_2$ then (s_k, α_1) is smaller than (s_l, α_2)
  3. if $\alpha_1 > \alpha_2$ then (s_k, α_1) is bigger than (s_l, α_2).

2. Negation operator of a 2-tuple. We define the negation operator over 2-tuples as:

$$Neg((s_i, \alpha)) = \Delta(g - (\Delta^{-1}(s_i, \alpha)))$$

where $g + 1$ is the cardinality of S, $S = \{s_0, \ldots, s_g\}$.

3. Aggregation of 2-tuples. The aggregation of information consists of obtaining a value that summarizes a set of values, therefore, the result of the aggregation of a set of 2-tuples must be a 2-tuple. In the literature we can find many aggregation operators [30] which allow us to combine the information according to different criteria. The fuzzy linguistic representation model with 2-tuples has defined the functions Δ and Δ^{-1} that transform numerical values into 2-tuples and viceversa without loss of information, therefore any numerical aggregation operator can be easily extended for dealing with linguistic 2-tuples. We shall review several 2-tuple aggregation operators, that are based on classical aggregation operators.

Arithmetic mean. The arithmetic mean is a classical aggregation operator. Its equivalent operator, for linguistic 2-tuples, is defined as:

Definition 3. *Let* $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ *be a set of 2-tuples, the* 2-tuple arithmetic mean $\bar{x}^e$ *is computed as,*

$$\bar{x}^e = \Delta\left(\sum_{i=1}^{n} \frac{1}{n}\Delta^{-1}(r_i, \alpha_i)\right) = \Delta\left(\frac{1}{n}\sum_{i=1}^{n}\beta_i\right)$$

The arithmetic mean for 2-tuples allows us to compute the mean of a set of linguistic values without any loss of information.

Weighted average operator. The weighted average allows different values x_i have a different importance in the nature of the variable x. To do so, each value x_i has a weight associated, w_i, indicating its importance in the nature of the variable. The equivalent operator for linguistic 2-tuples is defined as:

Definition 4. *Let* $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ *be a set of* 2-tuples and $W = \{w_1, \ldots, w_n\}$ *be their associated weights. The* 2-tuple weighted average $\bar{x}^w$ *is:*

$$\bar{x}^w = \Delta\left(\frac{\sum_{i=1}^{n} \Delta^{-1}(r_i, \alpha_i) \cdot w_i}{\sum_{i=1}^{n} w_i}\right) = \Delta\left(\frac{\sum_{i=1}^{n} \beta_i \cdot w_i}{\sum_{i=1}^{n} w_i}\right)$$

Linguistic weighted average operator. This operator is an extension of the $\bar{x}^w$ introduced in Definition 4, in this case the weights are expressed by means of linguistic values [13]:

Definition 5. *Let* $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ *be a set of* 2-tuples and $W = \{(w_1, \alpha_1^w), \ldots, (w_n, \alpha_n^w)\}$ *be their linguistic* 2-tuple associated weights. The 2-tuple linguistic weighted average $\bar{x}_l^w$ *is:*

$$\bar{x}_l^w([(r_1, \alpha_1), (w_1, \alpha_1^w)] \ldots [(r_n, \alpha_n), (w_n, \alpha_n^w)])$$

$$= \Delta\left(\frac{\sum_{t=1}^{k} \beta_i \cdot \beta_{W_i}}{\sum_{i=1}^{n} \beta_{W_i}}\right) ,$$

with $\beta_i = \Delta^{-1}((r_i, \alpha_i))$ *and* $\beta_{W_i} = \Delta^{-1}(w_i, \alpha_i^w))$.

3 A distributed intelligent agent model for information gathering

In this section, we present an application of the 2-tuple fuzzy linguistic representation model as a way to aggregate the evaluations of the information from the intelligent agents on the Internet, in order to obtain the best and newest information for the users.

We shall board the problem of aggregation to match, once an user's weighted query has been given, the fulfillment of the agents to find relevant documents and the degrees of importance of the user.

In the first subsection, the concept of intelligent software agent is presented, in the second one, an architecture for these agents is proposed, and finally, in the last subsection, a way to aggregate this communication through the linguistic 2-tuple weighted average operator is given.

3.1 Concept of intelligent software agent

The intelligent software agents have been defined several times in the literature [23, 26, 29]. However, researches do not seem to agree about a profile of such type of agents, due to the long number of different applications where they are used, and also to the evolution of the single term *agent* towards other terms such as *software agents* and *intelligent agents*. Hence, we are not to give a new definition of this concept, neither to review the ones given previously, but to set the main notions about those characteristics from every of these terms related to our specific purpose.

Therefore, beyond question, the concept of *agent* or rather *autonomous agent* must be the first one to be explained. This term, is strongly associated with the "behaviour-based AI", as opposed to the "knowledge-based AI" [23], led by the expert systems. As Maes defines in [23], an agent is a system that tries to achieve some predefined goals in a complex and dynamic environment. Thus, depending on the environment, we can set the first big gap, by splitting the concept of agent in those called typically "robots", whose environment is basically physical, and those called "software agents", that inhabit in an environment consisting of computers and networks. Both concepts share one main characteristic: they are autonomous, i.e. they are able to operate and decide themselves the way to achieve their goals. However, as this feature is supposed to be inherent in an agent, an *autonomous agent* is usually called simply *agent*. As for the term *intelligent*, there are several discussions [26] about to consider whether an agent is intelligent by nature or not. We shall consider them as intelligent, since they present, in some sense, human behaviour reducing the heaviest work of Internet users.

Hence, the agents which with we are dealing, are *intelligent software agents*, despite keeping sometimes the term *intelligent agent* without specifying the term *software*, as they are supposed to be in an environment of computer and networks, namely Internet.

3.2 A distributed multi-agent architecture on the Internet

Most the designed intelligent agents nowadays are closely connected to the Internet. These agents do not only retrieve and filter information (in the sense of Web documents) [24], but also hand electronic mail, news lists, FAQ lists, ..., [20, 22, 29]. These are properly called *interface agents* [22], since they are more closely to the user. However, all the information that these agents get, come from somewhere or somewhat. There are servers through the Internet that proportionate these services of information, mail, news and FAQs. The agents closest to these data sources are called *information agents* [28]. Since Internet users can access to their interface or personal agents, as well as the general information agents, they feel completely lost and overloaded of information due to this avalanche of agents. This problem reveals the need of an organisation among the agents within Internet that implies both an agent hierarchy and architecture. Since the disposition of the elements taking part in the retrieval information process is distributed, it seems sensible to consider the architecture as a distributed one. Several architectures for these multi-agents distributed models have been proposed and reviewed [21, 25, 28, 29]. However, the architecture that fits better to our model is the one proposed by Sycara et al. in [28]. In this architecture, besides the aforementioned *interface* and *information agents*, the authors consider a third type of agents, the *task agents*. These agents deal with the decision-making process and the exchange of information with the information agents, resolving conflicts and fusing information, in order to release the interface agents of some tasks that make them ineffective.

A hierarchical model with five levels is proposed, as set out below:

- **Level 1:** *Internet Users*, which look for Web documents on the Internet by means of a weighted query where a set of terms related to the desired documents is specified.
- **Level 2:** *Interface Agents* (one for user, generally), that communicate the user's weighted query to the task agents, and filter the retrieved documents from task agents in order to give to the users those that satisfy better their needs.
- **Level 3:** *Task Agents* (one for interface agent, generally), that communicate the user's query to the information agents, and get those documents from every information agent that fulfills better the query, fusing them and resolving the possible conflicts among the information agents.
- **Level 4:** *Information agents*, which receive the weighted query from the task agents, look for the information in the data sources, and give the retrieval documents back to the previous level.
- **Level 5:** *Information sources*, consisting of all data sources within the Internet, such as databases and information repositories.

The scheme of this model can be observed in Fig. 2.

In the next section, an application of the 2-tuple linguistic weighted average operator used by the task agents is proposed as a way of carrying out the decision-making process and communication among these levels.

3.3
Information gathering through the 2-tuple linguistic weighted average operator

In the process of information gathering, as a response of an user's query on the presented model, there are two different parts:

- On the one hand, there is a communication between agents at levels 5–4 and 4–3, which is far from the user's participation, and where the question to be decided by the task agent is about which information agents would satisfy better the user's needs.
- On the other hand, there is a communication between agents at levels 3-2 and the user, where the information element is specifically the set of retrieved documents that will be analyzed and filtered by the interface agents.

Several approaches based on multi-agent models to carry out the information gathering process have been proposed

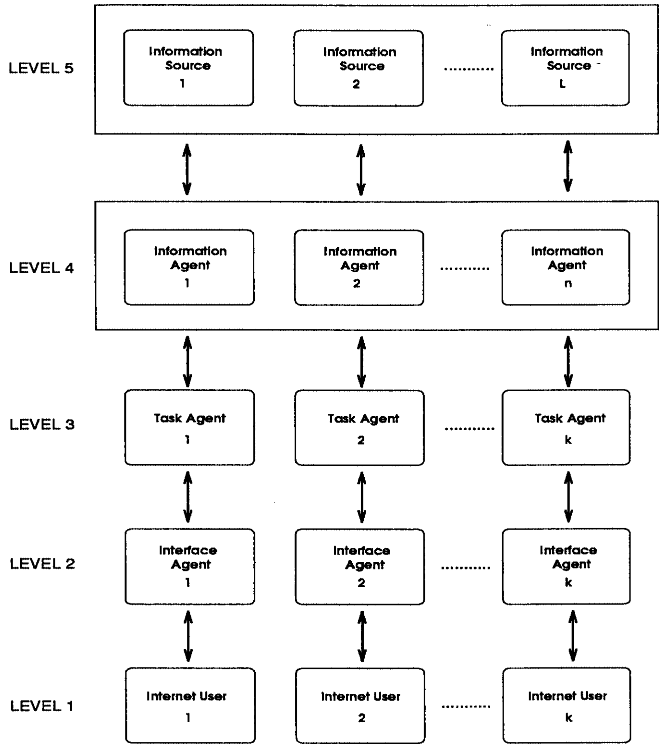

Fig. 2. A general overview of the distributed intelligent agent model

in the literature [28], specially at levels 1 and 2 [22, 24] where only the interface agent and the user take part.

We shall focus now in the first part of the process, detailing the communication among the agents at different levels and proposing the use of the 2-tuple linguistic weighted average operator as a possible aggregation operator between the importance of the criteria of the user's query and the satisfaction of these criteria on the part of each information agent.

In the considered levels (levels 3–4 and 2–3), there are two main flows where the weighted information to be aggregated appears. Such flows are represented in Fig. 3, where the elements related to a single user have been considered.

The description of the information gathering process is as follows:

- **Step 1:** An *Internet user* makes a query to look for those documents related to the terms $\{t_1, t_2, \ldots, t_m\}$, which are weighted by a linguistic degree of importance $\{p_1, p_2, \ldots, p_m\}, p_i \in S$. Both set of values are given by the user to the *interface agent*.
- **Step 2:** The *interface agent* gives the terms and their importance weights to the *task agent*.
- **Step 3:** The *task agent* makes the query to all the information agents to which it is connected, and give them the terms $\{t_1, t_2, \ldots, t_m\}$.
- **Step 4:** All the *information agents* that have received the query, look for the information that better satisfies it in the information sources, and retrieve from them the documents.
- **Step 5:** The *task agent* receives from every *information agent h* a set of documents and their relevances (D^h, R^h)

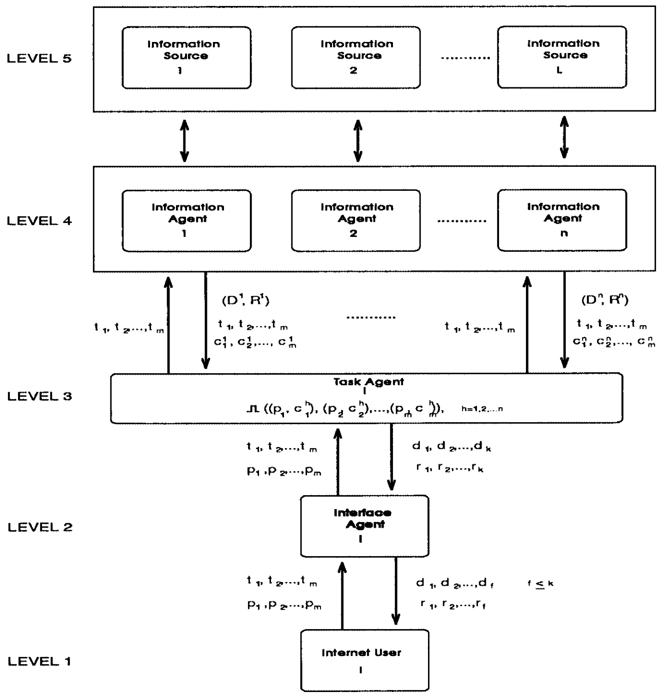

Fig. 3. An overview of information flows in a single user scheme

ordered decreasingly by relevance [27], where every document d_j^h has an associated degree of relevance r_j^h ($j = 1, \ldots, card(D^h)$). It also receives a linguistic degree of satisfaction [3] $c_1^h, c_2^h, \ldots, c_m^h$, $c_i^h \in S$ (whose equivalent 2-tuples are $(c_1^h, 0), (c_2^h, 0), \ldots, (c_m^h, 0)$) of this set of documents with regard to every term of the query.

- **Step 5.1:** The *task agent* aggregates through the 2-tuple linguistic weighted average operator, $\bar{x}_l^w$, both linguistic information weights, the satisfactions of the terms of the query from every *information agent*, (c_i^h, α), and the importance weights that the user assigned to these terms, (p_i, α) as follows:
  Let $\{[(p_1, \alpha), (c_1^h, \alpha)], \ldots, [(p_m, \alpha), (c_m^h, \alpha)]\}$, $p_i, c_i^h \in S$ be the set of pairs of linguistic 2-tuples of importance and satisfaction to be aggregated by the task agent for every information agent h. According to the 2-tuple linguistic weighted average operator definition, the aggregation of the pair associated with every term is obtained as:

$$\lambda^h = \bar{x}_l^w([(p_1, \alpha), (c_1^h, \alpha)], \ldots, [(p_m, \alpha), (c_m^h, \alpha)])$$

- **Step 5.2:** Once the *task agent* has calculated the overall performances $\{\lambda^1, \ldots, \lambda^n\}$, $\lambda^j \in S \times [-0.5, 0.5)$ of the *n information agents* through the aggregation operator, it must decide which agent fulfil better the user's query. For this purpose, the task agent orders the performances decreasingly and obtains the vector $\{\Theta_1, \ldots, \Theta_n\}$, $\Theta_j \in S \times [-0.5, 0.5)$ as follows:

$$\{\Theta_1, \ldots, \Theta_n\} = \sigma(\{\lambda^1, \ldots, \lambda^n\}) = \{\lambda^{\sigma(1)}, \ldots, \lambda^{\sigma(n)}\} ,$$

  where σ is a permutation over the set of linguistic 2-tuples $\{\lambda^1, \ldots, \lambda^n\}$ and

$$\lambda^{\sigma(j)} \leq \lambda^{\sigma(i)} \quad \forall i \quad \leq j .$$

  In order to gather the better documents, the task agent may decide on two alternatives.

  * The first one is the selection of the information agent with the higher satisfaction of the query, Θ_1. This alternative presents a main drawback, as the set of documents of the selected agent contains some documents that, probably, will be less relevant to the query than some of the best documents of the rest of the information agents. This problem leads us to the second alternative, based on the selection of the best documents of every agent.
  * In the second one, with the purpose of selecting a number of documents from every agent being proportional to the degree of satisfaction of such an agent:

$$P_s(\Theta_h) = \frac{\Delta^{-1}(\lambda^h)}{\sum_{i=1}^n \Delta^{-1}(\lambda^i)}$$

  Finally, the number of documents, $k(D^h)$, that the *task agent* would select from such an agent is expressed as:

$$k(D^h) = round\left(\frac{\sum_{i=1}^n card(D^i)}{n} \cdot P_s(\Theta_h)\right) .$$

- **Step 6:** The *interface agent* receives from the *task agent* an ordered list of documents and their relevances $\{(d_j^h, r_j^h)\}$, where $d_j^h \in D^h, r_j^h \in R^h$, $1 \leq h \leq n$ and $j = 1, \ldots, k(D^h)$.
- **Step 7:** The *interface agent* filters these documents in order to give to the user only those documents that fulfill better his/her needs.

4 Example

In the following, an example of the application through this architecture is explained. For this purpose, a view of a single user i will be considered, as it was set out in figure 3. In this example, we will consider four information agents. Let us suppose an user making a query to Internet through an interface agent at the lowest levels of the presented architecture. The user may be interested in 'Agents', and more specifically, in 'Information Agents', to which the terms 'Agents' and 'Information' may be introduced as terms in the query. These terms may be weighted by means of linguistic 2-tuples related to importance. In order to simplify the task of the user to evaluate the documents, a set with three different labels will be considered:

$$S = \{s_2 = H, s_1 = M, s_0 = L\}$$

where H = *High*, M = *Medium*, L = *Low*. Since the user is quite interested in the topic 'Agents' and, explicitly, in 'Information Agents', the labels associated to the query terms may be *High* for the term 'Agents', and *Medium* for the term 'Information'.

Therefore, the parameters which the user will communicate to the interface agent would be as follows:

$$(t_1, (p_1, \alpha)) = (\text{'}Agents\text{'}, (H, 0))$$

$$(t_2, (p_2, \alpha)) = (\text{'}Information\text{'}, (M, 0))$$

The interface agent will go through the task agent, which will merely pass the terms of the query to the information agent level. The information agents search in the information source level those documents related to the terms of the query, and get a list with the most relevant links. For instance, each information agent h ($h = 1, \ldots, 4$) may retrieve a set of five links, D^h and their relevances R^h (see Table 1).

Each information agent h gives back to the task agent a set with the degree of relevance and the linguistic degree of satisfaction c_i^h of the set D^h with regard to every term p_i of the query, according to the following:

$$[(c_1^1, \alpha), (c_2^1, \alpha)] = [(M, 0), (L, 0)]$$
$$[(c_1^2, \alpha), (c_2^2, \alpha)] = [(H, 0), (H, 0)]$$
$$[(c_1^3, \alpha), (c_2^3, \alpha)] = [(H, 0), (M, 0)]$$
$$[(c_1^4, \alpha), (c_2^4, \alpha)] = [(H, 0), (L, 0)]$$

Once the task agent has received this information from the previous level, it aggregates both the satisfaction degrees and the importance degrees which had been obtained through the internet agent in an earlier step. Hence, the pairs of importance and satisfaction are aggregated by the task agent for every information agent h:

$$([(p_1, \alpha), (c_1^1, \alpha)], [(p_2, \alpha), (c_2^1, \alpha)]) = ([(H, 0), (M, 0)], [(M, 0), (L, 0)])$$
$$([(p_1, \alpha), (c_1^2, \alpha)], [(p_2, \alpha), (c_2^2, \alpha)]) = ([(H, 0), (H, 0)], [(M, 0), (H, 0)])$$
$$([(p_1, \alpha), (c_1^3, \alpha)], [(p_2, \alpha), (c_2^3, \alpha)]) = ([(H, 0), (H, 0)], [(M, 0), (M, 0)])$$
$$([(p_1, \alpha), (c_1^4, \alpha)], [(p_2, \alpha), (c_2^4, \alpha)]) = ([(H, 0), (H, 0)], [(M, 0), (L, 0)])$$

Now the Step 5 of the process is carried out by two phases. First, the aggregation of each pair is carried out through the 2-tuple linguistic weighted average, $\bar{x}_l^w$. Therefore, the overall fulfillment λ^h of the information agent h will be determined by the following expressions:

$$\lambda^1 = \bar{x}_l^w([(H, 0), (M, 0)], [(M, 0), (L, 0)]) = (\mathbf{M}, -0.33)$$
$$\lambda^2 = \bar{x}_l^w([(H, 0), (H, 0)], [(M, 0), (H, 0)]) = (\mathbf{H}, 0)$$
$$\lambda^3 = \bar{x}_l^w([(H, 0), (H, 0)], [(M, 0), (M, 0)]) = (\mathbf{H}, -0.33)$$
$$\lambda^4 = \bar{x}_l^w([(H, 0), (H, 0)], [(M, 0), (L, 0)]) = (\mathbf{M}, 0.33)$$

Hence, the overall performances of the information agents is:

$$\{\lambda^1, \lambda^2, \lambda^3, \lambda^4\} = \{(M, -0.33), (H, 0), (H, -0.33), (M, 0.33)\}$$

In the next step,the task agent would order these values decreasingly as follows:

$$\{\Theta_1, \Theta_2, \Theta_3, \Theta_4\} = \{\lambda^2, \lambda^3, \lambda^4, \lambda^1\} = \{(H, 0), (H, -0.33), (M, 0.33), (M, -0.33)\}$$

As it was explained in Sect. 3.3 (Step 5.2), the task agent may decide on choosing the information agent with the highest performance, or select the best documents from all the agents, according to the performance of each one. In general, this last solution is most suitable when all the information agents present similar performances, as it is our case. Therefore, the task agent will calculate the probabilities of selection of the documents of each agent, according to the scheme of selection probabilities referenced in Step 5.2, which expression would set as follows:

$$P_s(\Theta_h) = \frac{\Delta^{-1}(\lambda^h)}{\sum_1^4 \Delta^{-1}(\lambda^i)} \quad ,$$

Table 1. Sets of documents for the terms 'Agents' and 'Information'

(D^h, R^h)	d_j^h	r_j^h
(D^1, R^1)	http://phonebk.duke.edu/clients/tnfagent.html	0.7
	http://webhound.www.media.mit.edu/projects/webhound/doc/Webhound.html	0.7
	http://www.elet.polimi.it/section/compeng/air/agents/	0.6
	http://www.cs.bham.ac.uk/ámw/agents/links/	0.5
	http://groucho.gsfc.nasa.gov/Code_520/Code_522/Projects/Agents/	0.4
(D^2, R^2)	http://lcs.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia.html	0.9
	http://www.osf.org/ri/contracts/6.Rationale.frame.html	0.8
	http://www.info.unicaen.fr/˜serge/sma.html	0.8
	http://www.cs.umbc.edu/˜cikm/1994/iia/papers/jain.html	0.4
	http://www.hinet.com/realty/edge/gallery.html	0.1
(D^3, R^3)	http://activist.gpl,ibm.com/WhitePaper/ptc2.htm	0.9
	http://www.cs.umbc.edu/˜cikm/iia/submitted/viewing/chen.html	0.6
	http://www.psychology.nottingham.ac.uk:80/aigr/research/agents/agents.html	0.6
	http://netq.rowland.org/isab/isab.html	0.5
	http://maple.net/gbd/salagnts.html	0.1
(D^4, R^4)	http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/spetka/spetka.html	0.9
	http://mmm.wiwi.hu-berlin.de/MMM/cebit_engl.html	0.6
	http://foner.www.media.mit.edu/people/foner/Julia/subsection3_2_2.html	0.4
	http://www.cs.bham.ac.uk/˜ämw/agents/index.html	0.4
	http://www.ffly.com/html/About1.html	0.2

Obtaining,

$$P_s(\Theta_1) = 0.1416, P_s(\Theta_2) = 0.4291,$$
$$P_s(\Theta_3) = 0.3562 \text{ and } P_s(\Theta_4) = 0.2854 \ .$$

Finally, the task agent would calculate the number of documents $k(D^h), h = 1, \ldots, n$ to select from each agent. The result of this computation would be:

$$k(D^1) = 1, \ \ k(D^2) = 2, \ \ k(D^3) = 2, \ \ k(D^4) = 1 \ .$$

Hence, the final list of documents ordered by relevance that the interface agent would receive from the task agent would be:

$$(d_1^2, r_1^2) = (http://lcs.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia.html, 0.9)$$
$$(d_1^3, r_1^3) = (http://www.activist.gpl.ibm.com/WhitePaper/ptc2.htm, 0.9)$$
$$(d_1^4, r_1^4) = (http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/spetka/spetka.html, 0.9)$$
$$(d_2^2, r_2^2) = (http://www.osf.org/ri/contracts/6.Rationale.frame.html, 0.8)$$
$$(d_1^1, r_1^1) = (http://phonebk.duke.edu/clients/tnfagent.html, 0.7)$$
$$(d_2^3, r_2^3) = (http://www.cs.umbc.edu/cikm/iia/submitted/viewing/chen.html, 0.6)$$

In the last step of the information gathering process, the interface agent would filter this final ranked list of documents and would give to the user the most relevant documents.

This information gathering process guarantees that the user will receive the most relevant documents for his/her query, due to the fact, in step 5.2 we have chosen the second alternative proposed in the algorithm. Therefore, the ranking list of documents given to the user contains the documents with highest degree of satisfaction (to the query) according to all the agents avoiding a biassed selection of documents.

5 Concluding remarks

We have presented a distributed intelligent agent system where the communication of the evaluation of the retrieved information among the agents is carried out by using the 2-tuple linguistic weighted average operator. We may stand out two main advantages of this proposal:

- The task agent can obtain an overall evaluation of the satisfaction of the query from every information agent, taking into account the degrees of importance that the user assigns to the terms of the query.
- The 2-tuple linguistic weighted average operator allows a higher flexibility and precision in the information gathering process, reducing the effect of low satisfactions of some terms within the overall performance of the information agent.

References

1. **Adamopoulos GI, Pappis GP** (1996) A fuzzy linguistic approach to a multicriteria-sequencing problem, European J Operational Res **92**: 628–636
2. **Bonissone PP, Decker KS** (1986) Selecting uncertainty calculi and granularity: an experiment in trading-off precision and complexity. In: Kanal LH, Lemmer JF (eds), Uncertainty in Artificial Intelligence, North-Holland, pp. 217–247
3. **Bordogna G, Pasi G** (1993) A fuzzy linguistic approach generalizing Boolean information retrieval: a model and its evaluation. J Amer Soc Information Sci **44**: 70–82
4. **Brenner W, Zarnekow R, Witting H** (1998) Intelligent Software Agent, Foundations and Applications. Springer-Verlag, Berlin Heidelberg
5. **Degani R, Bortolan G** (1988) The problem of linguistic approximation in clinical decision making, Int J Approx Reasoning **2**: 143–162
6. **Delgado M, Verdegay JL, Vila MA** (1993) On aggregation operations of linguistic labels, Int J of Intelligent Syst **8**: 351–370
7. **Delgado M, Gómez-Skarmeta AF, Gómez Marín-Blázquez J, Martínez Barberá H** (1999) A multiagent architecture for fuzzy modeling, Int J Intelligent Syst **14**: 305–329
8. **Delgado M, Herrera F, Herrera-Viedma E, Martin-Bautista MJ, Vila MA** (2001) Combining linguistic information in a distributed intelligent agent model for information gathering on the Internet. In: Wang PP (ed.), Computing with Words, John Wiley & Son, in press
9. **Delgado M, Martin-Bautista MJ, Sánchez D, Vila MA** (2001) Aggregating opinions in an information retrieval problem. Proceedings of the Eurofuse Workshop on Preference Modelling and Applications, Granada
10. **Fazlollahi B, Vahidov RM, Aliev RA** (2000) Multi-agent distributed intelligent system based on fuzzy decision making, Int J Intelligent Systems **15**: 849–858
11. **Ferber J** (1999) Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Addison-Wesley Longman, New York
12. **Herrera F, Herrera-Viedma E, Verdegay JL** (1996) Direct approach processes in group decision making using linguistic OWA operators, Fuzzy Sets and Systems **79**: 175–190
13. **Herrera F, Herrera-Viedma E** (1997) Aggregation operators for linguistic weighted information, IEEE Trans Syst, Man and Cybernetics, Part A: Syst **27**: 646–656
14. **Herrera F, López E, Mendaña C, Rodríguez MA** (2001) A linguistic decision model for personnel management solved with a linguistic biobjective genetic algorithm, Fuzzy Sets and Systems **118**: 47–64
15. **Herrera F, Martínez L** (2000) A 2-tuple fuzzy linguistic representation model for computing with words, IEEE Trans Fuzzy Systems **8**(6): 746–752
16. **Herrera F, Martínez L** (2000) An approach for combining linguistic and numerical information based on 2-tuple fuzzy representation model in decision-making, Int J Uncertainty, Fuzziness and Knowledge-Based Systems **8**(5): 539–562
17. **Herrera F, Martínez L** (2001) A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making, IEEE Trans Syst, Man and Cybernetics. Part B: Cybernetics **31**(2): 227–234
18. **Herrera-Viedma E** (2001) Modeling the retrieval process of an information retrieval system using an ordinal fuzzy lin-

guistic approach, J Amer Soc Information Sci Technol **52**(6): 460–475

19. Kraft DH, Bordogna G, Pasi G (1994) An extended fuzzy linguistic approach to generalize Boolean information retrieval, Infor Sci **2**: 119–134

20. Lieberman H (1999) Personal assistants for the Web: A MIT perspective. In: Klusch M (ed.), Intelligent Information Agents, Springer-Verlag, pp. 279–292

21. Lejter M, Dean T (1996) A framework for the development of multiagent architectures, IEEE Expert **11**(6): 47–59

22. Maes P (1994) Agents that reduce work and information overload, Communi ACM **37**: 31–40

23. Maes P (1995) Modeling adaptive autonomous agents. In: Langton CG (ed.), Artificial Life: An Overview, MIT Press, pp. 135–162

24. Martin-Bautista MJ, Larsen HL, Vila MA (1999) A fuzzy genetic algorithm to an adaptive information retrieval agent, J Amer Soc Information Sci **50**(9): 760–771

25. Monkas A, Zacharia G, Maes P (1999) Amalthaea and histos: multiagent systems for WWW sites and representation recommendations. In: Klusch M (ed.), Intelligent Information Agents, Springer-Verlag, pp. 293–322

26. Petrie CJ (1996) Agent-based engineering, the Web and intelligence, IEEE Expert **11**(6): 24–29

27. Salton G, McGill MG (1983) Introduction to Modern Information Retrieval, McGraw-Hill

28. Sycara K, Pannu A, Williamson M, Zeng D (1996) Distributed intelligent agents, IEEE Expert **11**(6): 36–46

29. Wooldridge M, Jennings N (1995) Intelligent agents: theory and practice, The Knowledge Eng Rev **10**: 115–152

30. Yager RR (1988) On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Trans Syst, Man, and Cybernetics **18**: 183–190

31. Yager RR, Goldstein LS, Mendels E (1994) FUZMAR: an approach to aggregating market research data based on fuzzy reasoning, Fuzzy Sets and Systems **68**: 1–11

32. Yager RR (1995) An approach to ordinal decision making, Int J Approx Reasoning **12**: 237–261

33. Yager RR (1996) Protocol for negotiations among multiple intelligent agents. In: Kacprzyk J, Nurmi H, Fedrizzi M (eds), Consensus Under Fuzziness, Kluwer Academic Publishers, pp. 165–174

34. Yager RR (1997) Intelligent agents for World Wide Web advertising decisions, Int J Intelligent Syst **12**: 379–390

35. Yager RR (2001) Fusion of multi-agent preference orderings, Fuzzy Sets and Systems **112**: 1–12

36. Zadeh LA (1975) The concept of a linguistic variable and its applications to approximate reasoning. Part I, Information Sci **8**: 199–249, Part II, Information Sci **8**: 301–357, Part III, Information Sci **9**: 43–80