Improvement of Web-based service information systems using fuzzy linguistic techniques and Semantic Web technologies

Enrique Herrera-Viedma[1], Eduardo Peis[2], José M. Morales-del-Castillo[2] and Karina Anaya[1]

[1]Dpt. Computer Science and AI. University of Granada. viedma@decsai.ugr.es, karina@ugr.es

[2] Dpt. Library and Information Science. University of Granada. epeis@ugr.es, josemdc@ugr.es

Summary. The aim of this paper is to present a model of a web multi-agent system which combines the use of Semantic Web technologies together with the application of user profiles to provide an enhanced Web retrieval service. This system uses fuzzy linguistic techniques to deal with qualitative information in a user-friendly way. The system activity is developed in two phases: retrieval phase to gather the documents from the Web, and feedback phase to update the user profiles and the recommendation values of resources. In this paper we focus on the analysis of the retrieval phase. With this multi-agent system model the retrieval capabilities on the Web can be considerably increased.

Keywords: Web information retrieval, intelligent agents, fuzzy linguistic modeling, Semantic Web technologies, user profiles, information filtering.

1 Introduction

As it is known, one of the main problems of the Web today is to efficiently manage the overwhelming quantity of resources available to Internet users. To avoid arriving to a situation of collapse, it is becoming necessary to develop tools capable to offer solutions to this problem, since the available instruments have shown little efficiency in easing users' time consuming tasks such as gathering and selecting relevant documents. So, it could be useful to develop Web-based service information systems that permit to improve the access to information in a more efficient way. Most of the solutions proposed to face this problem involve different technologies as intelligent software agents [14, 27], information filtering techniques [31], and Semantic Web technologies [4, 5].

Software agents applied to a Web-based framework are usually organized in distributed architectures [7, 13, 14, 25] to mainly perform tasks of intermediation between users and the Web. In other words, we could say that agents play the role of *infomediators* that assist users in the information retrieval process [7, 14, 25, 31]. These agents are entities capable to act in an autonomous way, processing and exchanging results with other agents [19]. Nevertheless, to develop these tasks agents need a knowledge base that can be supported on Web ontologies [18, 19] and/or on implicit or explicit information about the users (obtained by direct observation and imitation of users' behaviour, or by registering users' feedback, respectively [27])

However, the main problem of using agents is to find a flexible and agile communication protocol for exchanging information among agents, and between users and agents because of the great variety of forms the information is represented in the Web. One possibility to facilitate the communication processes consists in the application of the fuzzy linguistic approach [39], that provides a flexible representation model of information by means of linguistic labels. The application of fuzzy linguistic techniques enables us to handle information with several degrees of truth and solving the problem of quantifying qualitative concepts. Some examples of the use of fuzzy linguistic techniques in the design of multi-agent systems can be found in [9, 11, 21, 24].

On the other hand, information filtering techniques ease users the task of sorting out relevant documents from those that are not, thanks to the previous selection (carried out by system) of the resources that better fit users' needs, requirements and preferences. These needs, requirements and preferences are mostly defined in the form of user profiles [26] that can contribute to improve the performance of information systems.

Another possibility to improve the activity of a multi-agent system could be the use of some of the technologies of the Semantic Web project [4, 5]. These semantic technologies allow developing ontology-based infrastructures [29, 35] where agents can operate at semantic level with resources described using RDF (Resource Description Framework)[3] in a manner both interpretable by humans and machines. This common syntactic framework allows us to define a unique communication vocabulary among agents that could also be used to characterize the knowledge base of the system and even the semantics of resources and user profiles.

The aim of this paper is to present a new model of fuzzy linguistic multi-agent system that involves the use of the Semantic Web technologies and user profiles dynamically updated to improve the information access on the Web. The Semantic Web technologies are used to endow the agents with a common communication language, to develop the ontologies that

describe the elements of the system and their interrelation, and to characterize resources and user profiles in a standardized way using RDF. As in [24], the system activity presents two phases, retrieval and feedback. In this paper we focus on the first one and show the structure of the multiagent system that allows develop it.

The paper is structured as follows. Section 2 reviews the technologies employed in this research: the fuzzy linguistic approach, filtering tools and user profiles, and the Semantic Web. Section 3 presents the new multiagent model and describes the retrieval phase. Section 4 shows an example of the system functionality, and finally, some concluding remarks are pointed out in section 5.

2 Preliminaries

2.1 Fuzzy linguistic approach

The *fuzzy linguistic approach* [39] and in particular, the *ordinal fuzzy linguistic approach* [20, 22, 23] are approximate techniques appropriate to deal with qualitative aspects of problems. An ordinal fuzzy linguistic approach is defined by considering a finite and totally ordered label set in the usual sense

$$S = \{s_i, i \in H = \{0, \ldots, T\}\}$$

and with odd cardinality (7 or 9 labels). The mid term representing an assessment of "approximately 0.5" and the rest of the terms being placed symmetrically around it. The semantics of the linguistic term set is established from the ordered structure of the term set by considering that each linguistic term for the pair (s_i, s_{T-i}) is equally informative. Furthermore, for each label s_i could be given a fuzzy number defined on the [0,1] interval, which is described by a linear trapezoidal membership function represented by the 4-tuple (a_i, b_i, α_i β_i) (the first two parameters indicate the interval in which the membership value is 1.0; the third and fourth parameters indicate the left and right widths of the distribution). Additionally, we require the following properties:

1. – *The set is ordered* : $s_i \geq s_j \quad if \quad i \geq j$.
2. – *There is the negation operator* : $Neg(s_i) = s_j$, *with* $j = T - i$.
3. – *Maximization operator* : $MAX(s_i, s_j) = s_i$ *if* $s_i \geq s_j$.
4. – *Minimization operator* : $MIN(s_i, s_j) = s_i$ *if* $s_i \leq s_j$.

To combine the linguistic information we need to define an aggregation operator such as the Linguistic Ordered Weighted Averaging (LOWA) operator [21]. It is used to aggregate non-weighted linguistic information (i.e., linguistic information values with equal importance) and it has been satisfactorily applied in different fields [20, 23, 24]. It is based on the OWA operator [38] and the convex combination of linguistic labels [10]. The LOWA operator is an *"or-and"* operator [21], i.e., its result is located between the maximum and minimum of the set of aggregated linguistic values. Its main advantage is that it allows to aggregate automatically linguistic information without to use linguistic approximation processes [39].

2.2 Filtering techniques and user profiles

Information filtering techniques deal with a variety of processes involving the delivery of information to people who need it. Operating in textual domains, *filtering systems* or *recommender systems* evaluate and filter the resources available on the Web (usually in HTML or XML documents) to assist people in their search processes [32], in most cases through the use of filtering agents [33]. Traditionally, these systems have fallen into two main categories [31]. *Collaborative filtering systems* use the information provided by many users to filter and recommend documents to a given user, ignoring the representation of documents. *Content-based filtering systems* filter and recommend the information by matching user query terms with the index terms used in the representation of documents, ignoring data from other users. These may be a drawback when little is known about user needs, so it becomes a necessity to apply user profiles for providing a fast and efficient filtering [37].

User profiles are the basis for the performance of information filtering systems. User profiles represent the user's long-term information needs or interests. There are two main distinct types of user profiles [26]: i) *collaborative profile*, which is based on the rating patterns of similar users, and hence, it can be represented by a community of similar users; and ii) content-based profile, which is represented by a vector of interest areas.

User profiling is a critical task in information filtering systems. As it is pointed out in [26] "... an improper user profile causes poor filtering performance (for example, the user may be overloaded with irrelevant information, or not get relevant information that has been erroneously filtered out)". Two desire properties that any user profiling should support are the following:

- User profiles should be adaptable or dynamic since user's interests are changing continuously and rapidly over time. This implies the necessity to include a learning module in the information filtering system to adapt the user profile according to feedback from user reaction to information provided by the information filtering system.
- The generating and updating of user profiles should be carried out with a minimal explicit involvement of the users, i.e. by minimizing the degree of the user intervention to reduce user effort and facilitate the system-user interaction.

2.3 Semantic Web technologies

The Semantic Web is an extension of the present Web, in which the information is gifted of a well defined meaning, permitting a better cooperation between humans and machines [4, 5]. It is based on two main ideas: i) semantic mark up of resources, and ii) development of "intelligent" software agents capable to understand and to operate with these resources at semantic level [4, 18].

The semantic backbone of the model is RDF/XML [3], a vocabulary that provides the necessary infrastructure to codify, exchange, link, merge and reuse structured metadata in order to make them directly interpretable by machines. RDF/XML structures the information in assertions (resource-property-value triples), and uniquely identifies resources by means of URI's (Universal Resource Identifier), allowing intelligent software agents the knowledge extraction from and inference reasoning over resources (such as documents, user profiles or even queries) using web ontologies. Ontologies, in the Semantic Web context, represent exhaustively specific knowledge shared by the members of a specific domain structured as a hierarchy of concepts, the relations between these concepts and the axioms defined upon these concepts and relations [6, 15, 16]. Therefore, Web ontologies provide an "invisible" semantic lattice where complex systems can be built over, defining and interrelating the different elements that

form their structure. There exist several ontology languages that can be used for designing web-based ontologies, but the recommendation proposed by the World Wide Web Consortium (W3C) is the Ontology Web Language (OWL) [28], a language with a great expressive capacity that allow defining ontologies maintaining the RDF/XML syntactic convention.

This feature allows querying both resources and ontologies using a common semantic query language (for example [1, 30, 34]), that allows agents extracting information and inferring knowledge from RDF graphs.

3 A model of fuzzy linguistic multi-agent system based on Semantic Web and user profiles

In [24] we define a model of fuzzy linguistic multi-agent system to gather information on the Web with a hierarchical architecture of seven action levels: *internet users, interface agent, collaborative filtering agent, task agent, content-based filtering agent, information agent* and *information sources*. Its activity is developed in two phases: i) *Retrieval phase:* This first phase coincide with the information gathering process developed by the multi-agent model itself; and ii) *Feedback phase:* The second phase correspond with the updating process of recommendations on desired documents existing in a collaborative recommender system.

The main drawback of this model is that it does not utilize user profiles to characterize users' preferences and consequently its retrieval capabilities are clearly handicapped.

To overcome the limitations of the model presented in [24] we define a new and enhanced model of a fuzzy linguistic multi-agent system that improves information retrieval by means of the application of user profiles to enrich the filtering activity, and Semantic Web technologies to set a base for the operation of software agents. This model has been specifically designed for its use in academic environments, although it can be easily adapted for its implementation on different domain-dependant environments where a very specialized information retrieval and filtering is required (such as bio-medical or enterprise information systems). It presents a hierarchical structure with six action levels (*internet users, interface agent, filtering agent, task agent, information access* and *information bases*) and also two main activity phases (see Fig. 1):

1. **Retrieval phase**: This phase involves three processes. The first one is the "semantic retrieval" process and coincides with the information gathering process developed by the multi-agent model itself, although

the query language used is not a Boolean one as in [24], but a semantic query language capable of comparing both literal and semantics structures [17, 36]. In such a way it is possible to obtain more accurate and contextualized answers to queries. The second process is a "filtering" process, which consists on the selection of those resources that better fit both the explicit and the implicit preferences of the users. The third process is the "display" process the filtered resources to users.

2. **Feedback phase**: This phase involves two processes: "recommendation" process and "profile updating" process. In both processes the system needs users to qualitatively appraise the selected documents and the global answer provided by the system to a specific query, respectively. The "recommendation" process is similar to that defined in [24]: users express their opinion about any retrieved documents and with the appraisal provided the system can recalculate and update the recommendations of the desired documents. The second process consists on the dynamic updating of user profiles on the basis of the satisfaction degree the user expresses regard to the global results provided by the system.

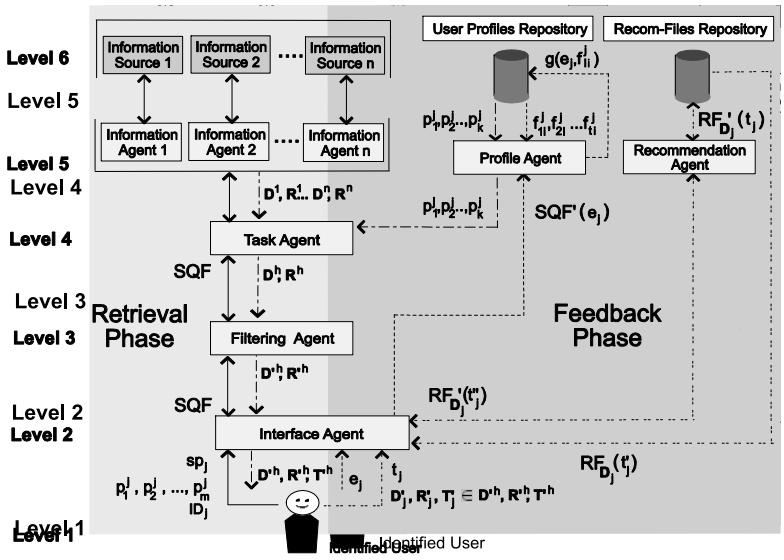

Fig. 1. Model architecture: "Semantic" retrieval and feedback phases

In the following sections, we analyze in detail the retrieval phase together with its action processes.

3.1 Information Retrieval Phase: "Semantic" retrieval process

This process begins when an identified user defines a query and ends when the information agents retrieve sets of relevant resources from the different document repositories. Users define their queries using basically both *preferences* and *search parameters*. *Preferences* refer to the search options that users can select to scope queries, defining constraints upon the characteristics of the documents to be retrieved (such as its semantic context or typology). For example, the user could provide his/her preferences about any of these four categories of basic preferences:

- *Document Type*: This preference establishes the type of document that user prefers to retrieve. For example, we could consider the following F_1= *{SciArticle, Proceedings, BookChapter, all}*.
- *Search Context:* It consists of general topic categories that represent the main areas of the system domain. For example, if the domain of work of our system is "knowledge-based systems" we could define the following set of values [2]: F_2= *{case-based reasoning, knowledge-based intelligent systems, intelligent systems, multi-agent systems, neural networks, fuzzy systems, decision support systems, genetic algorithms, semantic web, all}*.
- *Search aim:* It defines those tasks the user want to carry out with the information to be retrieved. For example, in a scholarship environment we could define different task categories depending on the nature of these tasks and the knowledge level of the different kind of users. Then, a possible set of preference values could be F_3= *{research, teaching_bachelor, teaching_master, teaching_doctorate, studies_bachelor, studies_master, studies_doctorate, all}*.
- *Date:* It refers to the updating or publication date of the resources to be retrieved. A set of different time intervals are defined to cover a wide range of values that vary from few months to several years. For example, F_4= *{3months, 6months, 1year, 3years, 5years,+5years}*.

On the other hand, *search parameters* correspond with both natural language *keywords* that better define user's own information needs, and *structural elements* of the document where the search must be performed (e.g., in the whole document or just in the abstract). The *structural elements* define the logical structure of each document type (which is, in turn, defined and validated through its corresponding XML Schema [12]). Therefore,

the set of structural elements available to define a query depends on the selected value for the *document type* preference. Therefore, the set of structural elements vary from a document type to another. For example, suppose a user choosing the *Scientific article* type, then he/she could select any structural element from the following set that has been exclusively defined for this document type, e.g., E ={*title, authors, abstract, introduction, body, conclusions, bibliography, whole_document*}.

Once the user formulates a query, it is assigned a unique "semantic query file" (SQF) in RDF format that contains the user's idenfier ID_j, the search parameters (sp_j) and the set of selected values $\{d_1^j, d_2^j, ..., d_k^j\}$ for the preferences $\{p_1^j, p_2^j, ..., p_k^j\}$ (see Fig. 2 below).

```
...
<Query rdf:ID="query384">
  <user_ID>http://www.ugr.es/~user/U022005</user_ID>
  <preferences_e>
    <preferences rdf:ID="pref_384">
      <docType>sciArticle</docType>
      <context>user_modeling</context>
      <aim>research_article</aim>
      <date>3months</date>
      <value>NULL</value>
    </preferences>
  </preferences_e>
  <search_parameters_e>
   <keyW>ontologies</keyW>
   <keyW>machine learning</keyW>
   <struct_e>abstract</struct_e>
  </search_parameters_e>
</Query>
...
```

Fig. 2. Semantic Query File (SQF)

The *semantic retrieval process* is developed in the following steps:

- **Step 1:** To define a query any registered user j must specify the search parameters sp_j (keywords and an optional structural element) and a set of k ($0 \leq k \leq m$) preferences $\{p_1^j, p_2^j, ... p_k^j\}$, being m the number of properties used to define a user profile and $p_i^j \in F_i$, being F_i the expression domain associated to the property i. From this query the interface agent generates the associated SQF, storing in it the inputs given by the user (i.e., search parameters and preferences) and his/her ID. Those preferences not explicitly given will appear with NULL value.
- **Step 2:** The SQF is transferred from the interface agent to the task agent.

- **Step 3:** The task agent proceeds to complete the SQF replacing every NULL preference with the value of its expression domain (stored in the user's profile) with a highest associated frequency, obtaining as a result a SQF with no NULL values.
- **Step 4:** Using the keywords and the structural element stored in the SQF, the task agent composes a semantic query using a pre-agreed semantic query language. This semantic query is sent to the different information agents. The query, as in ordinary search engines, is defined using clauses and operators set by default to combine the keywords. In our model, the chosen semantic query language is SeRQL [1] due to its simplicity and flexibility. With this language we can define simple queries using a structure similar to the following:

```
SELECT SciArt
FROM {SciArt} doc:hasAbstract {} doc:abstract {LIT}
WHERE Lit LIKE "keyword1" or Lit LIKE "keyword2"
   IGNORE CASE
USING NAMESPACE
   doc = <http://www.ugr.es/local/kishimaru/SciOnt#>
```

Fig. 3. Semantic query sample

In this case, query indicates that the different information agents should retrieve those scientific articles whose abstracts contain the terms *keyword1* or *keyword2*.

- **Step 5:** The information agents apply these common "semantic" searches in their associated document repository (one per agent), retrieving sets of pertinent documents $\{D^1, D^2, ..., D^n\}$, supposing n information agents. For each resource D^i_j in a set of retrieved documents D^i, being $1 \leq i \leq n$, the information agents calculate a relevance degree $R^i_j \in R^i$ (being R^i the set of relevance degrees for the set of documents D^i). These relevance degrees must be interpreted as the relative importance of the selected keywords in a specific structural element for a particular document. In other words, the retrieved resources will be, for example, documents with relevant abstracts if the chosen structural element was *abstract* or with relevant conclusions if it was the *conclusions* element, and so on (it is obvious that the relevance degree is relative to the whole document when the search is performed using the *whole_document* element). Each information agent sends the resulting sets of relevant documents to the task agent.
- **Step 6:** The task agent aggregates the different sets of resources (obtained from distributed sources) into a single set (D^h, R^h) to make the information more tractable for its filtering.

3.2 Information Retrieval Phase: Filtering process

This process basically consists in performing different "semantic" searches (one per preference) over the set of resources retrieved by the information agents to match users' preferences with the characteristics of the documents. Afterwards, a ranked list of filtered resources is generated.

```
...
<User rdf:ID="user02555">
  <updated>01/06/2005</updated>
  <personalInfo_e>
     <PersonalInfo rdf:ID="inf-pr001">
       <photo>pht0445.jpg</photo>
       <name>Juan</name>
       <surname1>Doe</surname1>
       ...
     </PersonalInfo>
   </personalInfo_e>
   <preferences_e>
     <DocType rdf:ID="docType-pr001">
       <type_e>
         <Type rdf:ID="type1-pr001">
           <type>SciArticle</type>
           <freq>AlmostAlways</freq>
         </Type>
       </type_e>
     ...
     </DocType>
     ...
   </preferences_e>
</User>
...
```

Fig. 4. Representation of a user profile

This process needs three basic inputs:

***The user profile*:** We assume a repository storing a set of user profiles, defined as a structured representation of an individual in RDF format, which is determined by users' ID and characterized by the particular values that each user has assigned to the categories of basic preferences. Each preference has an associated linguistic frequency property (tagged as <*freq*>) representing how often a specific value is used in queries assessed by users as "satisfactory". Thus, being $F_i = \{g_{1i}, g_{2i}, ..., g_{ti}\}$ the set of basic preferences, then we define $f^j{}_i = \{f^j{}_{1i}, f^j{}_{2i}, ..., f^j{}_{ti}\}$ as the set of frequency values associated with each possible value $l \in \{1,..., t\}$ of the property i in the pro-

file of the user *j*. The range of possible values for the frequencies is defined in a set of seven linguistic labels, *S={always, almostAlways, mostTimes, sometimes, aFewTimes, almostNever, never},* i.e., $f^{j}_{li} \in S$. An example is given in Fig. 4. The utilization of user profiles makes necessary a registration process previous to the retrieval phase. When a user logs into the system for the first time, he/she must fill in a simple form with personal information and interests, professional aims, etc., in order to get an approximate structured representation of the individual. On the base of these data, the system is able to assign to each user a basic stereotypic profile (by means of clustering algorithms), that serves as a basis where the user profile can be developed. Each one of these stereotypic profiles describes a specific user type through a set of characteristics, constraints and preferences set by default. For example, in a scholarship environment we could define three different stereotypes: researchers, teachers and students. Although stereotypes may not exactly reflect the characteristics of each individual, they are valid approximations that avoid the problem of "cold start". Another characteristic of these basic stereotypic profiles is that they may evolve over time and be modified when a significant change is detected in the behavior of users pertaining to a specific stereotype. In this process the user is also automatically assigned an ID (a URI) that will uniquely identify him/her in the system, so therefore it will be possible to relate users with other elements and actions they perform (as for example when formulating a query, or giving a recommendation about a resource).

***The Semantic Query Files (SQFs)*:** As it was explained above, this element contains the search parameters and preferences of a particular user for a specific query. To filter the set of retrieved documents is necessary to match concrete documents' characteristics with users' preferences.

***The Description Files of the Documents*:** We assume that each document in a resource repositories has associated a content file (see Fig. 5) and two auxiliary description files, *a classification file (CF)* and a *recommendation file (RF),* in RDF format. This description allows a more flexible and complete characterization of documents. In such a way, we can easily update the recommendation values or classification terms of a given document. Both auxiliary metadata files can be directly referenced from the content file. While RF stores a historical log of the recommendations assigned to the document since it was accessed, the CF (see Fig. 6) contains additional data about the resource, such as its level of complexity, a *document type* property, and a set of content classification categories. The *level of complexity* is an attribute defined in a bid to match the search aim of users with the complexity of the content of the resources. Therefore, the rank of pos-

sible values for the level of *complexity* element must be the same defined for the *search aim* preference.

```
...
<SciArt rdf:ID="S442">
  <hasTitle>
    <Title_e rdf:ID="Title_SA-001">
      <title>Introduction to RDF</title>
    </Title_e>
  </hasTitle>
  <hasAuthor>
    <Author_e rdf:ID="Author_SA-001">
      <author>Martina Branshaw</author>
    </Author_e>
  </hasAuthor>
   ...
</SciArt>
...
```

Fig. 5. Sample of the content file of a document

```
...
<Class_File rdf:ID="CS442">
  <resource>
   http://www.ugr.es/local/kishimaru/SciOnt#S442
  </resource>
  <doctype>Scientific article</doctype>
  <class_e>
    <Class rdf:ID="ce_CS442">
      <catg>web usage mining</catg>
      <dex:mat>minería de datos</dex:mat>
    </Class>
  </class_e>
  <complex>Researcher</complex>
</Class_File>
...
```

Fig. 6. Representation of a classification file (CF)

Then, the filtering process is developed in the following steps:

- **Step 1:** The task agent sends the ranked set of aggregated resources (D^h, R^h) to the filtering agent.
- **Step 2:** The filtering agent matches the characteristics of the resources (stored both in the content file and in its associated CF) with the preferences stored in the SQF generated for the current query, discarding

those documents that don't fit user's requirements. To do so, defines a set of queries (one per preference) to filter the resulting ranked set of resources. For example, according to the *search context* preference, the filtering agent could define a query with a structure similar to the following one:

```
SELECT Class_file
FROM {Class_file} cf:class_e {} cf:catg {"data mining"}
USING NAMESPACE
   cf = <http://www.ugr.es/~CF/class#>
```

Fig 7. Filtering query sample

In this example, the search is carried out in the CF of each retrieved resource, matching the classification topics with the "*Search Context"* preference in the SQF. As a result, those resources that doesn't include in their classification categories the term "data mining" are discarded.

- **Step 3:** The filtering agent seeks the URIs of the documents that have matched all the preferences defined in the SQF, therefore generating a ranked list with the set of already filtered resources (D'^{h}, R'^{h}), that is consequently sent to the interface agent.

3.3 Information Retrieval Phase: Display process

Once the retrieved documents are filtered, the system proceeds to display the results to the users. They can choose those particular documents of their interest and their display format.

This process requires the following input:

-Recommendation Files of the Documents: As aforementioned, we assume that each document has associated a RF in RDF format (see Fig. 8) where is contained information about all the appraisals made by users that have read it previously. The RF contains data as its URI, the last recommendation value displayed and a set of log items containing previous appraisals about that document. Each log item is defined by an user's ID, his/her corresponding appraisal and the search context used in the query formulated by the user to retrieve that document. This representation enables the adoption of different recommendation policies, allowing us, for example, to recalculate recommendation values based on the opinion of all

the users, or just of some of them (e.g., using the appraisals given by those users who looked for information by the same topic).

```
...
<RecomFile rdf:ID="recomf001">
  <resource>
   http://www.ugr.es/local/kishimaru/SciOnt#S442
  </resource>
  <accesed>2005-08-17T13:25:42Z</accesed>
  <modified>2005-03-08T16:32:11Z</modified>
  <recom_value>High</recom_value>
  <recom_history>
    <R_history rdf:ID="histf001">
      <item>
        <RecomItem rdf:ID="form01-pr001">
          <appraissal>VeryHigh</appraissal>
          <topic>Data mining</topic>
          <user_ID>user-pr022005</user_ID>
        </RecomItem>
      </item>
      ....
    </R_history>
  </recom_history>
</RecomFile>
...
```

Fig. 8. Representation of a Recommendation File (RF)

This process is developed in four steps:

- **Step 1:** The interface agent asks the recommendation agent for the recommendation values corresponding to each one of the documents to be displayed.
- **Step 2:** For each particular document, the recommendation agent checks out if its corresponding recommendation file $RF_{D'j}$ has been modified since the last time it was accessed. If not, the interface agent receives the last recommendation value displayed t'_j that was stored in the <*recom_value*> tag of the $RF_{D'j}$. If in the $RF_{D'j}$ was added a new log item, the recommendation agent proceeds to recalculate a new global recommendation value by means of the LOWA operator [21] from the set of historical values, generating a new recommendation value (t''_j) that replaces the old one t'_j.
- **Step 3:** The documents are displayed coupled with their respective recommendation value.

– **Step 4:** Once the user selects a document of his/her interest, it is displayed by the interface agent in the preferred file format defined by the user (e.g., txt, xml, html, pdf, etc.) by means of XSLT stylesheets [8].

4 Application example

Suppose the following framework. John Quest, a researcher member of RECSEM (an academic institution specialized in the study of "information systems"), has to write a paper about "*semantic information systems*" for a prestigious scientific journal. John decides to search in RECSEM's documents repositories for recent resources about this topic and, in such a way, to build a solid knowledge background for his work. He proceeds to log into RECSEM's site and reaches the search interface page. Then, he writes a pair of keywords ("*information systems*" and "*ontologies*") in the search box, and explicitly specifies that these keywords must be searched in the abstract of each document. To scope the search, John selects *"knowledge representation"* as preferred search context, *"research"* as search aim and *"3months"* as preferred date of publication, but doesn't specify any value for the *"Document type"* preference. The system checks his profile and works out that the most satisfying value for the *"Document type"* preference is *"Scientific Article"*.

The task agent composes a semantic query that is sent to the different information agents. Each information agent retrieves a set of relevant documents and calculates their corresponding relevance degrees. The task agent aggregates all the retrieved documents into a set of relevant resources. This relevant set is filtered, being discarded those documents that don't match the preferences that appear in the SQF.

Before displaying the filtered documents to John, the task agent proceeds to check the RF of each resource and dynamically calculates their recommendation value. For example, let *H={high, medium, veryHigh, medium, low, high}* the set of historical recommendations values of a specific resource. If this set has been modified since the last time the document was accessed then the recommendation agent aggregates the different historical recommendation values using the LOWA operator and calculates a new recommendation value for the resource.

As a result of this search, on John's laptop screen are displayed a list of ranked documents that fit his explicit requirements (see Table 1).

Table 1. Query answer sample

D_j^h	R_j^h	T_j^h
http://www.ugr.es/~Arep/N0021	0.95	high
http://www.ugr.es/~Arep/L57641	0.93	NULL
http://www.ugr.es/~Erep/P70435	0.87	medium

Each document appears with an associated relevance degree and a recommendation value. This extra information eases John's task of deciding which resources can be more useful for him.

After he has used a document, he is asked to voluntary provide an opinion about its quality that will be stored in the list of historical recommendation values of that document. Furthermore, and before John can leave the current search session, he must provide his satisfaction degree relative to the answer provided by the system to his query, thus triggering the user profile updating mechanism.

After the search process, John finishes his session with the feeling that he has saved a lot of time, and with the certainty that he has obtained relevant and useful resources for his purposes.

5 Concluding remarks

We have described the architecture and elements of a fuzzy linguistic multi-agent system specially designed to perform information retrieval and filtering tasks in domain dependant environments (specifically in academic environments).

The key aspect of the system is the joint application of Semantic Web technologies and the use of user profiles. Semantic Web technologies provide the system with the necessary semantic infrastructure to improve inference and communication capacities of agents and with means to represent the information (resources and user profiles) in a common vocabulary both human and machine interpretable. The use of user profiles allows a better users' characterization and a better performance of the system can be achieved.

In the future, we shall study the development of an enhanced user profile updating process based on web usage analysis and rule discovery techniques, and the adaptation of this system to different work contexts.

Acknowledgements

Authors wish to acknowledge support from the Spanish Ministry of Education and Culture (project ref. TIC-2003-07977).

References

1. B.V. Aduna, A. Sirma, The SERQL query language (revision 1.2). In *User guide for Sesame* (2005). Available at http://www.openrdf.org/doc/sesame/users/ch06.html. Accessed 10/05/2005.
2. G. Avram, Empirical study on knowledge based systems, *The electronic journal of Information Systems Evaluation*. 8 (1) (2005), 11-20.
3. D. Beckett (ed.), RDF/XML Syntax Specification (Revised). W3C Recommendation. (2004). Available at *http://www.w3.org/TR/rdf-syntax-grammar/*. Accessed 02/19/2005.
4. T. Berners-Lee, Semantic Web Road map (1998). Available at *http://www.w3.org/DesignIssues/Semantic.html*. Accessed 02/15/2005.
5. T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, *Scientific American* May (2001).
6. Y.A. Bishr, H. Pundt, W. Kuhn, M. Radwan, Probing the concept of information communities: a first step toward semantic interoperability. In M. Goodchild, M. Egen-Hofer, R. Fegeas, C. Kottman (eds), *Interoperating Geographic Information Systems*, (Kluwer Academic, 1999), 55–69.
7. W. Brenner, R. Zarnekow, H. Witting, Intelligent Software Agent, Foundations and Applications, (Springer-Verlag. Berlin Heidelberg, 1998).
8. J. Clark (ed.), XSL Transformations (XSLT), Version 1.2 (1999). Available at http://www.w3.org/TR/xslt. Accessed 07/16/2005.
9. M. Delgado, F. Herrera, E. Herrera-Viedma, M.J. Martín-Bautista, M.A Vila, Combining linguistic information in a distributed intelligent agent model for information gathering on the Internet. In P.P. Wang (ed), *Computing with Word*, (John Wiley & Son, 2001), 251-276.
10. M. Delgado, J.L. Verdegay, and M.A. Vila, On aggregation operations of linguistic labels, *International Journal of Intelligent Systems*, 8 (1993) 351-370.
11. M. Delgado, F. Herrera, E. Herrera-Viedma, M.J. Martín-Bautista, L. Martínez, M.A. Vila, A communication model based on the 2-tuple fuzzy linguistic representation for a distributed intelligent agent system on Internet, *Soft Computing* 6 (2002), 320-328.
12. D. C. Fallside, P. Walmsley (eds.), XML Schema Part 0: Primer Second Edition, (2004). Available at http://www.w3.org/TR/xmlschema-0/. Accessed 09/10/2005.

13. B. Fazlollahi, R.M. Vahidov, R.A. Aliev, Multi-agent distributed intelligent system based on fuzzy decision making, *Int. J. of Intelligent Systems* 15 (2000), 849-858.
14. J. Ferber, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence (Addison-Wesley Longman, New York, 1999).
15. T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, *Int. J. of Human-Computer Studies* 43 (5-6) (1995), 907-928.
16. N. Guarino, Formal ontology and information systems. In N. Guarino (ed) *Formal Ontology in Information Systems, Proceedings of FOIS'98*. Trento (Italy), (IOS Press, Amsterdam, 1998), 3–17.
17. R. Guha, R. McCool, E. Miller, Semantic search. *12th Int. World Wide Web Conference 2003 (WWW2003)*, Budapest (Hungary), (2003) 700 – 709
18. J. Hendler, Is there an intelligent agent in your future? (1999). Available at http://www.nature.com/nature/webmatters/agents/agents.html. Accessed 02/20/2005.
19. J. Hendler, Agents and the Semantic Web, *IEEE Intelligent Systems*, March, April (2001), 30-37.
20. F. Herrera, E. Herrera-Viedma, J.L. Verdegay, A model of Consensus in Group Decision Problems under Linguistic Assessments, *Fuzzy Sets and Systems* 78 (1996), 73-87.
21. F. Herrera, E. Herrera-Viedma, J.L. Verdegay, Direct Approach Processes in Group Decision Making using Linguistic OWA operators, *Fuzzy Sets and Systems* 79 (1996), 175-190.
22. E. Herrera-Viedma, Modelling the Retrieval Process of an Information Retrieval System Using an Ordinal Fuzzy Linguistic Approach, *J. of the American Society for Information Science and Technology (JASIST)* 52 (6) (2001), 460-475.
23. E. Herrera-Viedma, E. Peis, Evaluating the informative quality of documents in SGML format using fuzzy linguistic techniques based on computing with words, *Information Processing & Management* 39 (2) (2003), 195-213.
24. E. Herrera-Viedma, F. Herrera, L. Martínez, J.C. Herrera, A.G. López, Incorporating Filtering Techniques in a Fuzzy Multi-Agent Model for Gathering of Information on the Web, *Fuzzy Sets and Systems* 148 (1) (2004), 61-83.
25. N. Jennings, K. Sycara, M. Wooldridge, A roadmap of agent research and development, *Autonomous Agents and Multi-Agents Systems* 1 (1998), 7-38.
26. T. Kuflik, P. Shoval, Generation of user profiles for information filtering-research agenda. *Proc. of the 23rd Annual Int. ACM SIGIR Conf. on Research and Development Information Retrieval*, Athens (Greece) (2000), 313-315.
27. P. Maes, Agents that reduce the work and information overload, *Communications of the ACM*, 37 (7) (1994), 30-40.
28. D.L. McGuinness, F. van Harmelen (eds.), OWL Web Ontology Language Overview. W3C Recommendation. 10 February 2004, (2004). Available at http://www.w3.org/TR/2004/REC-owl-features-20040210/. Accessed 02/16/2005.
29. Ontoknowledge Project. Available at http://www.ontoknowledge.org/. Accessed 02/18/2005.

30. E. Prud'hommeaux, A. Seaborne (eds.), SPARQL query language for RDF (2004). Available at www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/. Accessed 03/07/2005.
31. A. Popescul, L.H. Ungar, D.M. Pennock, S. Lawrence, Probabilistic models for unified-collaborative and content-based recommendation in sparse-data environments, *Proc. of the Seventeenth Conf. on Uncertainty in Artificial Intelligence (UAI)*, San Francisco (2001), 437-444.
32. P. Reisnick, H.R. Varian, Recommender Systems. *Special issue of Comm. of the ACM* 40 (3) (1997).
33. J.B. Schafer, J.A. Konstan, J. Riedl, E-Commerce recommendation applications. *Data Mining and Knowledge Discovery* 5 (1/2) (2001), 115-153.
34. A. Seaborne (eds.), RDQL: A query language for RDF, (2004). Available at http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/. Accessed 02/02/2005.
35. Semantic Web Advanced Development for Europe (SWAD-Europe). Available at http://www.w3.org/2001/sw/Europe/. Accessed 02/17/2005.
36. U. Shah, T. Finin, Y. Peng, J. Mayfield, Information Retrieval on the Semantic Web, *Proc. of the 10th In.l Conf. on Information and Knowledge Management* (2002), 461-468.
37. B. Shapira, U. Hanani, A. Raveh, P. Shoval, Information filtering: A new two-phase model using stereotypic user profiling, *J. of Intelligent Information Systems* 8 (1997), 155–165.
38. R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics* 18 (1988) 183-190.
39. L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part I, In *Information Sciences* 8 (1975), 199-249. Part II, *Information Sciences* 8 (1975), 301-357. Part III, *Information Sciences* 9 (1975), 43-80.