An agent-based Collaboratory

Alberto Vázquez, Ignacio Barrio, Javier Vázquez-Salceda,

Josep M. Pujol, Ramon Sangüesa

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
c/ Jordi Girona 1-3. E-08034, Barcelona, Spain
{avazquez, ibarrio, jvazquez, jmpujol, sanguesa}@lsi.upc.es

Abstract

Collaboration for distributed groups can be supported by intelligent agents in charge of several functionalities, partly addressed by Computer Supported Co-operative Work (CSCW) and Knowledge Management Systems. Collaboratories are special type of systems covering the needs of research communities. We describe here an agent-based architecture and a real collaboratory implementation that is a natural evolution of a set of recommender systems. Each agent acts as an assistant agent, co-ordinated with the rest of agents in the system.

Keywords: Machine Learning, Multi-Agent Systems, Recommender Systems, User Modelling, CSCW, Knowledge Management.

1 Introduction

The present nature of work assumes the existence of groups gathered around a given project or set of projects, and not as a result of a hierarchically defined function. Most areas of activity, specially in knowledge-intensive sectors, resort to flexible teams that share and organise their competences for a given project and a given span of time. The detection and gathering of the relevant competences (within a company or across cooperating enterprises) or the way that scientific cooperation is carried on are just two examples of this new organization of work. Groups are geographically distributed and they need fast and easy detection of competences. The sharing of knowledge and the acceleration of group learning abilities is a must in many areas. An important realisation of these ideas is known under the name of "Collaboratories" [20], which started as an initiative of the American National Science Foundation to improve cooperation between scientists by means of telecommunications and CSCW.

Some computer-supported systems have been proposed or are in current use in order to foster the development of group-based knowledge intensive projects. The tradition of Computer Supported Co-operative Work [3] has remarked such aspects as information sharing, communication among group members, and a common group document repository. From the point of view of Knowledge Management [8] the goal is to create systems that sustain the whole knowledge management cycle within a geographically distributed group. That is: a common repository or organisational memory [19] to store the relevant knowledge created in the system exists as well as facilities to ease the *access* to that relevant knowledge from the people that may need it.

However, this approach has revealed some flaws in its usage and adoption since it assumes an active role on the part of the users that should be actively searching documents and establishing the relevance of the found information as well as being continuously aware of which of their actual or potential group partners can cooperate in a given project or be interested in new knowledge. The emphasis is on the searching and accessing actions by the user instead of a more proactive action on the part of the system. For this reason some systems adopt a different approach, trying to create environments that proactively work for their users on the basis of the previously generated knowledge as well as on their known competence areas. One framework under this idea is the Knowledge Pump approach [11] where an intermediate layer exists between the users and the knowledge (documents) generated within a group in order to detect which new piece of knowledge may be of interest to each user.

This type of anticipatory task to help a group of people seems well suited to be solved by intelligent adaptive agents. We describe in the following the Collaboratory[1], a prototype system that supports an actual research community, the researchers designing and developing the Catalan Internet2 project, I2-Cat [12].

The core of the Collaboratory is a multi-agent system that manages document sharing, finding potential partners and establishing the relevance of newly generated knowledge. In this way it gives support to the basic functions needed for cooperation and group learning.

2. System Description

Users of the Collaboratory log on to the system via an Internet browser. If the user is not currently registered, when he enters he has to register himself as a new member. In the registration the user should choose a set of topics of his interest, so the system can build an initial user profile. Once the user has entered, the system offers the following functionalities:

- Recommender functionalities
- Search functionalities
- Expert finder functionalities

We describe each in the following sections.

2.1 Recommender functionalities

The user does not always need to search for useful information, he can see the new document recommendations the system has computed for him automatically according to his topics of interest. Then those new documents that the user finds interesting can be voted as good documents and the not interesting ones can be voted as bad documents. This feedback can be used to improve the user profile, as we will see in section 4. In doing so, the system implements a collaborative filtering metaphor [25][2].

Recommended documents can be currently documents written in txt, html, rtf, ps or pdf formats. These documents are written or recommended by other registered users who thought that such document could be interesting to other people in the community. The system only distributes documents to those users that are potentially interested not only in the topic of the document but in the document as a whole. This filtering can be made thanks to the a) *content-based filtering component*, that finds the most important words in a text document, and b) the *social filtering component* (also called collaborative filtering), based on information extracted from votes issued by users of the system. This second kind of filtering allows the Collaboratory to recommend other kinds of information, not only text documents but also video or audio files. These functionalities are supported by a *Recommending Agency* (described in section 3.1)

2.2 Search functionalities

Searches are possible on the Internet and the Collaboratory internal repository of documents. When an user wants to make a search then *the Search Agency* shows to him a list of useful thesauri, one for each topic. Then the user can decide to adjust his query to one of the words in the thesauri. The system then applies a query expansion [15] algorithm to choose other words to restrict the query. As in section 2.1 searches can find not only textual but also multimedia information.

2.3 Expert finder functionalities

Sometimes, it is more useful and fast to find an expert on a topic related to the problem at hand than spending time looking for, finding and interpreting documents.

The Collaboratory allows to search for experts in a certain area or topic. The user can find experts by making a query as in the case of searching documents, but now the system retrieves people whose personal profiles match the query word. It also retrieves other expert related to him, what is known as the social or knowledge network [7], which is a graph that represents the people that one registered user knows ("Who knows who") as in the Referral Web system [13].

Once the user gets a list of people, he can contact with one of them using the built-in e-mail, chat or videoconference functionalities. Retrieving an expert social network is useful as it allows to find the paths inside the social network, that is, the referral chains a user should use to reach the expert. The simplest example would be the following: user A asks the system for an expert in certain topic. The system

[1] The Collaboratory can be found in the following addresses: `http://upclsi1.i2.cat/colaboratorio/` and in `http://hercules.lsi.upc.es/colaboratorio/`.

retrieves a list of people, and A chooses one of them, the expert B, that user A does not know, but that is friend of another user C that A knows as well. In this case if this hidden information is made explicit it is easier for the user A to ask the expert B for information, going through their common relationship with C. For further details see [24]. All theses functionalities are supported by the corresponding set of agents, the *Expert Finder Agency*.

2.4 CSCW functionalities

Apart of the above-explained functionalities, there are some typical peer-to-peer tools as short message, chat and videoconference communication among a pair of users. These tools allow users to work together even thought the users are not working in the same location. In Collaboratory there are users that work in different departments, even in different universities, but they can put in contact with each others using the broad bandwidth of Internet 2. All functionalities are tied to the corresponding agencies that can be seen as a set of cooperating agents.

3 Architecture

The Collaboratory general architecture is shown as a three-layer architecture depicted in Fig. 1:

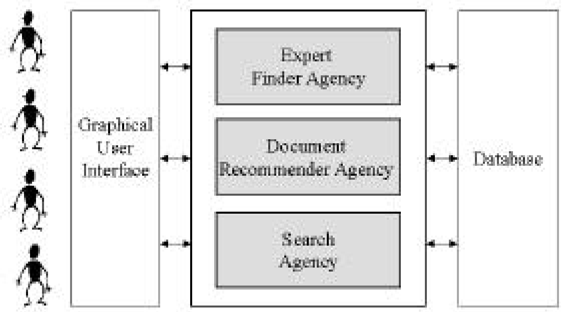

Fig. 1: General Architecture.

- The **Graphical User Interface**, composed by the Browser and the client-side Interface Agents.
- The **Agent Architecture**, proper composed by three Agencies: the *Expert Finder Agency*, the *Document Recommender Agency* and the *Search Agency*.
- The **Database**, which stores information about the users (user profiles) and information about the registered documents (the document repository currently is a Mysql database [18])

The most important layer is the *Agent Architecture*, which is built as a multi-agent system. Every agent in the system has a specific goal to reach and it interacts with other agents through KQML messages, in order to make requests and send results. The Collaboratory core is, in fact, the combination of three cooperating agencies: the *Expert Finder Agency*, the *Document Recommender Agency*, and the *Search Agency*.

3.1 The Document Recommender Agency

The *Document Recommender Agency* implements the recommendation functionalities. Its architecture is shown in Fig. 2 and is composed by the following agents:

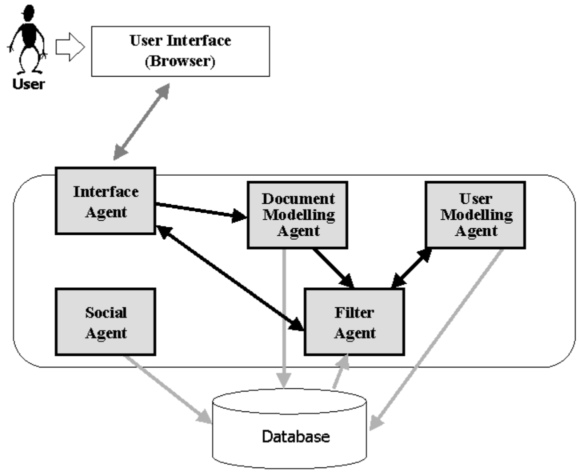

Fig. 2: Document Recommender Agency

- **Interface Agent**: As explained above there is an agent for every user connected to the Collaboratory. This agent is the adapter between the user interface, the *Document Modelling Agent* and the *Filter Agent*.
- **Document Modelling Agent**: Every time a user enters a new document in the system, the *Document Modelling Agent* builds a document profile for it. This model is a word list based on the Space Vector Model [23].
- **Filter Agent**: It uses the previously computed document and user profiles to decide which documents a user should receive. These decisions can be seen as a filtering process, and are based on the cosine inner product similarity to decide

whether both profiles (the user's and the document's) are similar or not.

- **User Modelling Agent**: Each time one user votes a document this Agent is activated to create or modify the user's content-based profile. This profile is a list of the most important words extracted for that user. This word selection is done by using the TF-IDF measure as in other recommender systems such as Syskill&Webert[21], Fab[2] and Raap [9].
- **Social Agent**: This agent creates and manages the collaborative-based profile of each user. To figure out the relations among users it creates clusters of similar users that are correlated in their voting patterns, that is, the votes made previously by the users.

In section 4 we will give more details about the creation of the user profiles.

3.2 Search Agency

The *Search Agency* allows the user to search for documents in the internal database or through Internet using an ad-hoc thesaurus for each area or topic registered in the Collaboratory. By now we have two experimental thesauri, one for agent technology and the other one for the collaborative technology. The utility of such thesaurus to improve the relevance and accuracy of the information retrieval process has been proved (see [5]).

The agency is depicted in Fig. 3. It is composed by the following agents:

- **Interface Agent**: As we previously explained, there is an agent for every user connected to the Collaboratory. This agent is the adapter between the user Interface and the *Clever Agent*.
- **Clever Agent**: The *Clever Agent* expands queries by means of the Boolean Spread Algorithm, widely used in several search systems (see [26] and [6]). Once the query has been expanded, the *Clever Agent* sends a message to the *Main Agent* , which then takes the control. [15]
- **Main Agent**: The *Search Agency* has a co-ordinator, the *Main Agent*. This agent supervises the work of the others agents, distributing the tasks to be done.
- **Search Agent**: there are currently three kinds of search agents: the *Lycos Search Agent.*, the *Altavista Search Agent* and the *Excite Search Agent*. Each one is a meta-searcher[2] from the public-known search engines Lycos[14], Altavista[1] and Excite[10].
- **Fusion Agent**: When the query results have been retrieved, the *Fusion Agent* joins and filters the URLs found to show them to the user with an unique layout.

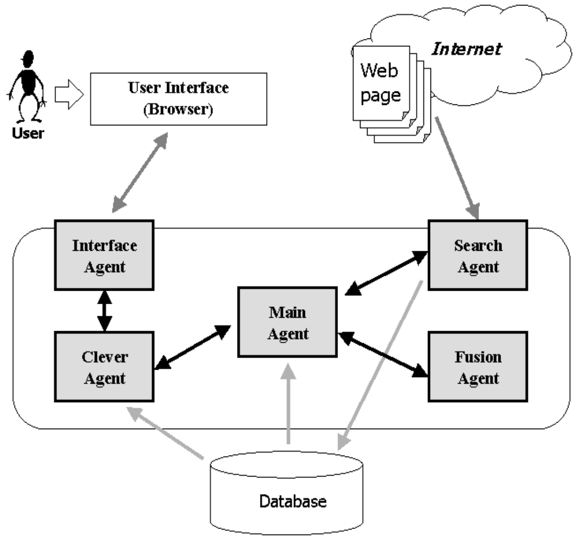

Fig.3: Search Agency.

3.3 The Expert Finder Agency

As we can see in Fig. 4 this agency interacts with the Internet and the *Database*. It serves the expert finder functionalities explained before.

It has the following agents:

- **Interface Agent**: There is an agent for every user connected to the Collaboratory. This agent is the adapter between the user Interface and *the Graph Agent*, and can be placed at the user's machine or the system's machines depending on the user's browser. If the user enters to the system through an Internet Browser, the *Interface Agent* is inside a Java applet placed in the user's machine. If the user connects through a WAP mobile phone, the interface agent is placed at the server, building the information the WAP phone will receive and display.
- **Graph Agent**: This agent serves information to the *Interface Agent* making queries to the database.

[2] A search system that uses the search results from other search systems to get its own results.

- **Spy Agent**: The *Spy Agent* has the aim to spy the users' behaviour and discover if there are changes over the user community. In this case the *Spy Agent* alerts the *RelationshipMaker Agent*.
- **RelationshipMaker Agent**: This agent discovers and maintains the relations between the users in the community. This task is done from the information that it keeps inside the database and the information extracted by the *WebSnooper Agent*.
- **WebSnooper Agent**: The *WebSnooper Agent* captures information from the personal web pages of every user in the community. The results are sent to the *RelationshipMaker Agent*.

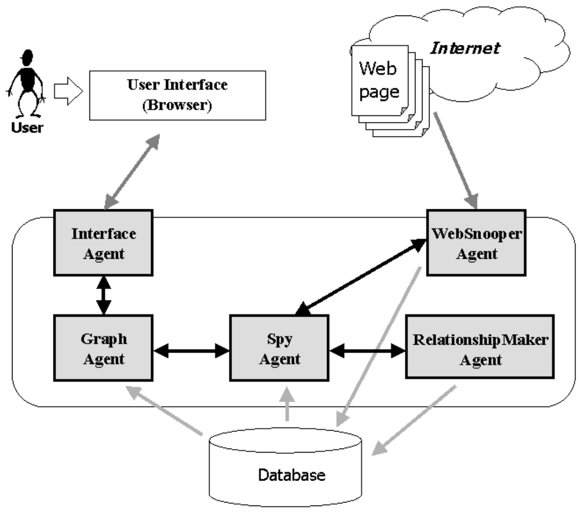

Fig. 4: Expert Finder Agency

More details about each of these agents can be found at [24].

4. Learning a user profile

In the Collaboratory we have used agent-based technology in order to model users' interests. Each user that is connected to the system has an *Interface Agent* that represents him and learns to act on his behalf. To reach this goal the agent has to learn a formal representation of the user interests from the feedback between user and system. This formal representation, the *user profile*, is built from two sources of user feedback:

a) *Implicit voting*: The system is continuously "looking over the shoulder" [16] and is aware of the kind of documents sent by the user to the document repository. It also takes into account URLs or suggestions contributed to the community by each user. The changes the user has introduced in his homepage as well as if the user has been cited in a document sent by other colleague.

b) *Explicit voting*: The user can vote in favour or against the documents presented by the system to confirm or deny their potential interest to him.

From the collected information the system can build a user profile that is subdivided in three parts:

- Content-based profile
- Collaborative-based profile
- Acquaintance-based profile

4.1 Learning the content-based user profile

The content based profile is built from the analysis of the words of the textual documents voted or suggested by the user. This profile is a set of vectors, each one a list of the 100 most informative words for the user for a given topic.

In order to extract the most informative words we use the Information Gain [22] as depicted in (1) .

$$E(W,S) = I(S) - \left[P(W_{present}) I(S_{w_{present}}) + P(W_{\neg present}) I(S_{w_{\neg present}}) \right]$$

where

$$I(S) = \sum_{c \in \{+,-\}} - p(S_c) \log_2 (p(S_c))$$

(1) Extracting relevant words from a document.

W is the actual word and *S* is the set of documents. $P(W_{present})$ is the probability that *W* was present in a page. $Sw_{present}$ is the set of pages where word W occurs one or more times. S_c are all pages of class *c*.

For each selected word we compute the *tf* (term-frequency[3]) and the *idf* (inverse-document-frequency[4]) to calculate the TF-IDF algorithm that measures the relevance of each word in the document, in a similar way as it was done in the ACE system. For details about the overall process, see [25].

[3] The number of times the word appears in the documents the user is interested in.

[4] The number of times the word appears in all the documents of the database.

The major disadvantage of this kind of profiles is that is not possible to learn them from multimedia documents with no associated textual information or metadata. The major advantage is that this kind of profile does not depend on the users feedback and it is very useful when there is a small user community or the community gives little feedback.

4.2 Learning the collaborative-based user profile

The collaborative-based profile uses the votes a user has issued for each document to find related similar users.

For each user the learnt profile is a weighted vector of related users. We apply the inner product vector similarity depicted in (2)

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

(2) Learning the collaborative profile

This formula measures the vector similarity between two users: *a* is the actual user, *i* is some other of the rest of users, and $V_{a,j}$ is the vote *a* has made to the document *j*. This formula computes the cosine of the angle formed by the two vectors. The squared terms in the denominator are used to normalise votes.

If the value obtained is higher than a fixed threshold, user *i* is included in the social part of the vector model for user *a*, and user *a* is included in the social vector of the user *i*. For further justification for this formula see [4].

As explained in [17] and [9], this profile is complementary with respect to the content-based one, because its major advantage is the possibility of learning from multimedia documents via other similar users votes. Otherwise this kind of profile is fully dependent of the quality and amount of feedback given to the system by the user community.

4.3 Learning the acquaintance based user profile

The system needs another profile to model the fact that a user knows other users, that is, to make explicit the social network. As briefly explained in section 2.3, the social network is extracted from two kind of sources.

a) personal web pages
b) documents sent to the Collaboratory repository

From the a) source we applied the formula (3):

$$w(a,b) = K_1 f(a,b) + K_2 g(a,b) + K_3 \sum_{i \in R(a,b)} 2 / depht(i,a) + depth(i,b)$$

(3) Learning the acquaintance profile (I)

w(a,b) is the acquaintance weight between the user a and b. K_1,K_2,K_3 are constants to establish the weight of each sub-formula. The sub-formula f(a,b) returns a value between {0,1}, 1 if exists an e-mail link of one user to the other and 0 if does not exist. The sub-formula g(a,b) is similar to f(a,b) but it returns 0 when the two users are reciprocally cited in their web pages. R(x,y) is the set of URLs shared between the two users. Finally this formula is pondered by the depth the URLs have been found in the personal web page of the users (higher values if the link is in lower depth).

For documents coming from source b) we apply formula (4):

$$w(a,b) = K_1 heav(n(a,b),1) + K_2 n(a,b)$$

(4) Learning the acquaintance profile (II)

Where w(a,b) is the acquaintance weight between users *a* and *b*. K_1,K_2 are constants to establish the weight of each sub-formula. The sub-formula n(a,b) is the number of documents the two users have co-authored, and heav(x,y) is the heavyside function[5].

5 Conclusions

We presented an agent-based Collaboratory and an actual implementation of it currently serving a community of research users. It is in current used, supporting the i2cat research community and some parts of it have been experimentally put to test by using data from the Software Department (LSI) of the Technical University of Catalonia (UPC). In spite of being a prototype, its usefulness seems well established as well as the use of a multiagent-system to give support to collaborative activities

[5] if x < y return 0, otherwise return 1.

The Collaboratory is now under improvement in order to let users work concurrently with objects like blackboards and other types of shared objects. The next evolution of the Collaboratory is to adapt the actual generic system to a specific and commercial tool in the medical area.

Further work has to be made to use some advanced machine learning techniques in order to model users interests. We are currently working to use automatic thesaurus construction algorithms to learn a personal user thesaurus that would allow us to model the user interests in a more structured fashion.

Another important issue to study is the improvement of the tailorability of the whole system. By now the system is adaptable to the user interests but it is not tailorable as in other CSCW systems such as the Evolve Project [27].

References

[1] Altavista. http://www.altavista.com

[2] Balabanovic, M. and Shoham, Y. *Fab: Content-Based, Collaborative Recommendation. Communications of the ACM*, vol. 40, nº 3, pags. 66-72. (March 1997)

[3] Borghoff, U. M. and Schlichter, J. H. *Computer-Supported Cooperative Work*. Introduction to Distributed Applications. Springer Verlag; ISBN: 3540669841

[4] Breese, J. S.; Heckermann, D.; Kadie, C. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Proceedings of the 14th conference on Uncertainty in Artificial Intelligence, Madison, Winsconsin (July, 1998). Morgan Kaufmann Publisher.

[5] Crouch, C. J. and Yang, B. *Experiments in Automatic Statistical Thesaurus Construction.* SIGIR 92, 77-88, 1992

[6] Cohen, P.R. and Kjeldsen, R. *Information retrieval by constrained spreading activation in semantic networks.* Information processing and Manegement, 23(4):255-268,1987

[7] Contractor, N. S.; O'Keefe, B. J.; Jones, P. M. *IKNOW: Inquiring Knowledge Networks On the Web.* Computer software. University of Illinois. (1997).

[8] Davenport, T. H. and Prusak, L. *Working Knowledge*. Harvard Business School Pr., 2000. (ISBN: 1578513014)

[9] Delgado,J; Ishii, N.; and Ura, T. *Intelligent Collaborative Information Retrieval*. 6th Iberoamerican Conference on Artificial Intelligence (IBERAMIA 98), Lisbon, Postugal (October 1998).

[10] Excite. *http://www.excite.com*

[11] Glance, N.; Arregui, D.; Dardenne, M. *Knowledge Pump: Supporting the Flow and Use of Knowledge* In: Springer Verlag, Borghoff, U.and Pareschi, R. (Eds), Information Technology for Knowledge Management, 1998

[12] I2Cat. *http://www.i2-cat.net*

[13] Kautz, H.; Selman, B.; Shah, M. *The Hidden Web.* AI Magazine vol. 18 (Summer 1997).

[14] Lycos. *http://www.lycos.com*

[15] Macho, S. *Intelligent Searcher for Heterogeneus Domains.* Master Thesis LSI, 1998.

[16] Maes, P. *Agents that reduce work and information overload.* Communications of the ACM vol. 37, n. 7, pags. 31-40.

[17] Miyahara, K. and Okamoto. T. *Collaborative Information Filtering in cooperative communities*. Journal of computer Assisted Learning 14 (1998), pags. 100-109.

[18] MySQL *http://www.mysql.com*

[19] Nagendra Prasad, N. M. and Plaza, E. *Corporate memories as distributed case libraries*. In Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop KAW`96, Special Track on CorporateMemory and Enterprise Modeling. (November 1996).

[20] National Coordination Office for Information Technology Research & Development: Collaboratories *http://www.ccic.gov/pubs/blue99/collaboratories.html*

[21] Pazzani, M.; Muramatsu, J.; Billsus, D. *Syskill & Webert: Identifying interesting web sites.* Proceedings of the National Conferenceon Artificial Intelligence, Portland, OR. (1996).

[22] Quinlan, J. R. Induction of decision trees. *Machine Learning*, 1 (1986), 81-106

[23] Salton, G. and McGill, M J. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, 1983.

[24] Sangüesa, R. and Pujol, J. M. *NetExpert: A multiagent system for expertise location* (IJCAI'01) Workshop on Organizational Memories and Knowledge Management) [Accepted]

[25] Sangüesa, R.; Vázquez A; Vázquez-Salceda, J. "*Mixing Collaborative and Cognitive Filtering in Multiagent Systems*". Presented at the 3rd Workshop on Agent-Based Recommender Systems (WARS 2000), Barcelona, june 2000

[26] Shoval, P. *Principles, procedures and rules in an expert system for information retrieval.* Information processing and Manegement, 21(6):475-487,1985.

[27] Stiemerlig, O. and Cremers, A. B. *The Evolve Project: Component-Based Tailorability for CSCW Applications.* AI & Society (2000) 14:120-141