

Architecture of a Recommender System to Support Collaboration in a Software Environment

Daniel Lichtnow¹, Stanley Loh^{1,2}, Ramiro Saldana Garin¹, Augusto Caringi¹, and Pablo Lucas dos Anjos¹

¹Catholic University of Pelotas/ESIN, Rua Félix da Cunha, 412 CEP 96010-000 Pelotas/RS, Brazil. *Tel:* +55 (53) 284.8227. *Fax:* +55 (53) 225.3105.
{lichtnow, loh, rsaldana, pablo}@atlas.ucpel.tche.br
caringi.sul@terra.com.br

²Lutheran University of Brasil, Rua Miguel Tostes, 101 CEP 92420-280 Canoas/RS, Brazil. *Tel:* +55 (51) 477-4000. *Fax:* +55 (51) 477.1313.

Abstract. Within organizations, people learn through exchanging knowledge. This kind of task (named collaboration) is important for the organizational learning. Collaboration can be supported by Information Technology tools as chats, newsgroups, forums and e-mailing lists. However, this kind of support only enables message exchange, lacking to help people in the learning process. This work presents the architecture of a recommender system to support collaboration among people in an software organization. The system analyzes textual messages sent during the session, identifies the context of the discussion and suggests documents, authorities (people with competence in a subject) and past discussions within the same context. **Keywords:** collaboration, learning organizations, text mining, groupware, recommender systems.

1 Introduction

Learning organizations are those that use knowledge as one of its main resources [Se94]. They know how to use knowledge for improving business, and they know that knowledge, as other resources, need to be acquired and managed. Knowledge Management (KM) [DP97] is the area responsible for capturing, storing and retrieving knowledge to support business processes.

One of the great challenges for the KM is to share knowledge between people or departments. Technology-based tools as e-mail and chats can help this task. However, these tools do not allow recording the knowledge shared by people to later reuse it. This lack can lead to re-discussion of themes or to the lost of important information. Even the tools that store knowledge (for example, the history of discussions) do not use this knowledge in a proactive way. In these cases, people needs to query the knowledge base to find what is interesting.

This work presents an architecture of a recommender system to support people in collaboration tasks within a software development environment. Collaboration happens within the system through synchronous interactions (as chats in Internet) and

recommendations are made by the system in a proactive way during the online discussion (automatic recommendations). The system analyzes the messages exchanged by the participants in a session and identifies the context or theme of the discussion. After that, the system selects items from a digital base. Items classified in the specific theme or context of the discussion are suggested to the participants. Recommendations are made automatically in a proactive way, without people needing to decide. Items may be electronic documents, Web sites or pages, past discussions, program codes or authorities in that theme (people with competence). A *thesaurus* is used for the identification of themes in discussions and for the classification of items.

Section 2 of this paper deals with some concepts related to collaboration and recommender systems. Section 3 presents the architecture of the system and its components, as well its functionality. Concluding remarks discuss contributions and the ongoing implementation of the system.

2 Collaboration and Recommendation

Usually organizational knowledge is provided and maintained by people. New knowledge may be generated from existing one. For example, when people read documents, they obtain knowledge (tacit knowledge). However, generally this knowledge is not stored in formal and accessible ways (explicit knowledge). The consequence is the difficulty for sharing and reusing knowledge.

The majority of the organizational knowledge comes from interactions between people [NT97]. People tend to reuse solutions from other persons in order to gain productivity. For example, during the development cycle, some problems are discussed more than once and the same information is analyzed many times by people. If knowledge is not adequately recorded, organized and retrieved, the consequence is re-work and lost of productivity. Thus, it is important to store knowledge and to create efficient ways for people retrieving this knowledge.

In a software environment, people use to share knowledge about source codes, system functionalities, implementation difficulties, task schedules, language details and so on. This sharing task, called collaboration, is made through synchronous interactions (i.e., exchange of messages in a chat), asynchronous interactions (i.e., electronic mailing lists or forums), direct contact (two persons talking) or indirect contact (one person stores an electronic document and another one reads this document).

The collaboration task in software environments may be support by software tools that capture and store information and retrieve relevant information. Retrieval may be automatically performed by software tools (in a proactive way). Recommender systems are responsible for this task. Their goal is to supply people with information useful for decision making. This information may be about books, documents, music CDs or restaurants [Re97]. Recommender systems are broadly used in electronic commerce for suggesting products or providing information about products and services, helping people to decide in a shopping process [La01], [Sc01]. The advantage is that people do not need to request recommendation or to query an information base, but the system decides what and when to suggest. The

recommendation is usually based on user profiles and reuse of solutions. The profile includes information about the user's interests, habits, history of relationship and demographic data. Information to be recommended may be the one that was useful for a similar person in another time or useful for a different person in a similar situation (reuse).

3 Architecture and Functionality of the Recommender System

The proposed system supports collaboration between persons in a software environment. Only collaborations performed through synchronous interactions are supported. The main goal of the system is to recommend items to participants of discussions. To do that, the system needs to recognize the context of the discussion or the theme being dealt, analyzing the messages exchange in a software tool like an Internet chat. The system recommends items from a digital base classified in the same subject of the discussion (context or theme). Figure 1 presents an overview of the system architecture, detailing its main components and some interfaces.

The first module is the Session Analyzer, responsible for identifying the subject of the discussion. This is made using a Text Mining tool that analyzes texts in the messages exchanged in the chat tool. A *thesaurus* should be defined to represent the subjects possible to be discussed (including an hierarchy of the subjects). The *thesaurus* also contains the terms used to express those subjects in the written language. This *thesaurus* is specific to the local environment and needs to be defined by people of the environment using automatic and manual methods, e.g., [Ch96].

The recommender module receives the current subject of the discussion and select items from a digital base to suggest for the discussion participants. It is possible to exist more than one subject in the same session, as will be explained later. The recommendation intends to accomplish the knowledge reuse, suggesting to the participants information or solutions that were useful to other persons of the software environment. To do that, the system has to store a knowledge base that will be maintained by people of the organization. The environment personnel must add items to the digital base, classified in subjects according to the *thesaurus*.

3.1 Session Analyzer

Every message sent by a participant in a discussion session will be analyzed to find keywords. Keywords represent subjects as defined in the *thesaurus*. The text classification method is based on Rocchio's and Bayes' algorithms (details in [Lo00]). The method analyzes the context of the words and not only the presence of keywords, eliminating ambiguities.

There is a subject pointer, indicating what subject of those in the *thesaurus* is the current one being discussed. The pointer navigates over the *thesaurus* structure as different subjects are being dealt. The list of subjects discussed in a session will be stored in the discussion history for later analyses.

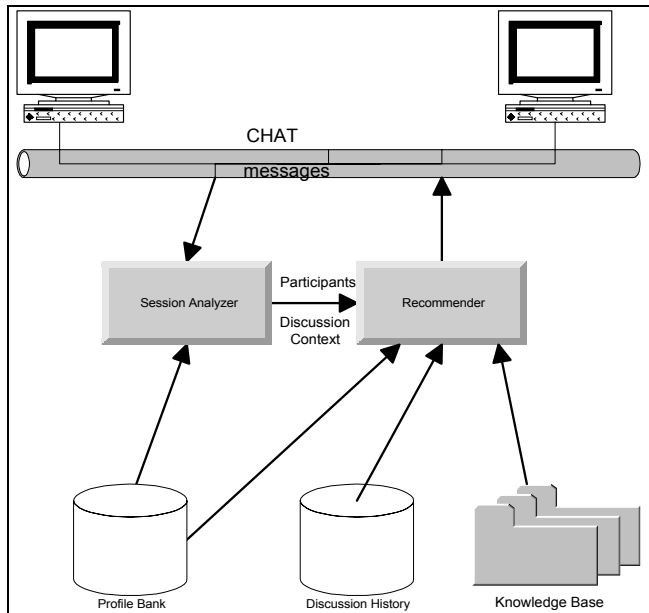


Fig. 1. Overview of the System Architecture

3.2 Knowledge Base

The organizational knowledge may be explicit through the following items:

- documents (electronic documents in formats as doc, rtf, pdf, html, etc., corresponding to papers, manuals, Web pages, etc.);
- software source code (routines, procedures, functions, objects, etc.);
- authorities: persons with recognized competence in some subjects, extracted from a profile bank;
- past discussions, extracted from a historical log file.

The maintenance of the base is responsibility of the organization personnel. Each person in the environment may include new items whenever he/she wants. This is the key for the approach success, since the base quality and volume are important for the recommendation quality (garbage-in, garbage-out).

Each item has to be classified in one or more subjects, according to the *thesaurus* hierarchy. This task may be performed manually by people or automatically by software (for example, text classifiers).

3.3 Profile Bank

The profile bank contains identification of the organization personnel. Besides usual data as name, department, etc., this bank stores the competence areas of each person, as well a degree informing how much this person “knows” the subject area. This

degree is not absolute but relative from one person to the others. The profile bank is similar to a Knowledge Map (or *Yellow Pages*), useful to indicate who owns what knowledge [St98].

Each time somebody participate in a discussion, the subjects discussed in that session are incremented in the profile of that person, indicating that his/her competence in that area is increasing. In addition, when somebody reads, adds or downloads items from the knowledge base (documents, Web references, source codes, etc.), the corresponding subjects (where the items are classified) are incremented in the person profile (this latter module does not appear in the architecture and will not be explained in this paper).

The profile bank is used by the recommender module to indicate authorities in a subject (people with high competence in the subject). The profile may be created with initial degrees (manually defined by authorized people).

The profile bank also records the items accessed (added, read or downloaded) by or recommended to each person, in order to eliminate duplicated recommendations or suggestion of known items.

3.4 Recommender Module

The recommender module receives the current subject of a discussion (it may receive many subjects during the same session). To do the recommendations, the module searches:

- in the knowledge base, items classified in the same subject (the current one);
- in the profile bank, persons with competence degree in the same subject (only degrees above a specified threshold);
- in the discussion history, past sessions where the same subject was discussed (present in the subject list of the session).

Recommendations are made in a separate window to not broke the interaction. Recommendations may be different to each people, since the recommender module verifies if the items or persons or sessions were not yet recommended to the same person. An hypothesis being considered is whether the recommendations may or not be redone to the same person.

3.5 Report Module

There is a module that generates some reports to the system administrator. Reports include:

- the most dealt subjects according to accesses in the knowledge base and subject lists of past sessions: this is important to verify which areas are being more worked by the personnel and which deserves more attention;
- the competence areas of each person and a general list of most competent people by subject;
- the most active people, according to volume of participation in discussions and access to the knowledge base.

4 Conclusion and Discussion

This work described the architecture of a system to support collaboration between people in a software developing environment. The architecture is applicable to others organizations, e.g., research groups. Using the tool, a member can communicate with others. Part of tacit knowledge becomes explicit and can be retrieved for reuse.

There are some challenges related to the use of this tool. First it is necessary to fill the Knowledge Base with some initial content for enable recommendations. Second a thesaurus has to be built by humans and this is a time consuming task. However once the thesaurus is defined the cost to maintain it is low. Thesaurus is strongly dependent on the domain of the environment. Another point is whether recommendations should or should not be repeated, because in some cases the user does not remember previous recommendations.

In the future, it will be necessary to consider the different levels of knowledge between people. Persons with high level of knowledge in an area do not have interest in recommendations related to introductory material.

It is very important to evaluate the strategy of recommendation. In some cases people do not want to have your work broken. The challenge is to foresee which users wish to receive recommendations.

Currently, the system is being developed using PHP and PostgreSQL there is a first prototype that enable which users sharing messages. This messages are classified by keywords existing in messages. In this process a pre-built thesaurus is used. At the moment, this first prototype does not have all proposal functionalities.

Future research includes the evaluation of productivity in a software development environment which uses this tool against an environment without this tool.

References

- [Ch96] Chen H.: A concept space approach to addressing the vocabulary problem in scientific information retrieval: an experiment on the worm community system. *Journal of the American Society for Information Science*. Arizona: Aug 1996 v.47,n.8.
- [DP97] Davenport, T. H; Pruzac, L.: *Working Knowledge – How organizations manage what they know*. Harvard Business School Press, 1998.
- [La01] Lawrence, R. D. et al.: Personalization of supermarket product recommendations. *Journal of Data Mining and Knowledge Discovery*, v.5, n.1/2, January, 2001; p.11-32.
- [Lo00] Loh, S. et al. : Concept-based knowledge discovery in texts extracted from the Web. *ACM SIGKDD Explorations*, v.2, n.1, July, 2000; p.29-39.
- [NT97] Nonaka, I. & Takeuchi, T.: *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, Cambridge, UK, 1995.
- [Re97] Resnick, P. Varian, H.: Recommender systems. *Communications of the ACM*, v.40 n.3, March, 1997; p.56-58.
- [Sc01] Schafer, J. Ben et al. : E-commerce recommendation applications. *Journal of Data Mining and Knowledge Discovery*, v.5, n.1/2, January, 2001; p.115-153.
- [St98] Stewart, T. A.: *Intellectual Capital: The New Wealth of Organizations*. Bantam Books, 1998
- [Se94] Senge, P. M.: *The Fifth Discipline*. New York: Currency Doubleday, 1994.