Sharing Engineering Information and Knowledge

Mauro F. Koyama, Olga Nabuco, Francisco E. D. Pereira, Khalil Drira *
*Research Center Renato Archer- CenPRA – Brazil, * LAAS / CNRS - France*
*{mauro.koyama , olga.nabuco, edeneziano.dantas}@cenpra.gov.br; * khalil@laas.fr*

Abstract

This paper describes an agent-based recommendation system developed to support knowledge acquisition and sharing processes. Its purpose is to aid the process of community building, specially in the early phase of peer search and recognition in a distributed engineering environment, where multidisciplinary teams must be established. In this kind of problem information about possible candidates to make up the teams must be gathered from all available channels, including the electronic ones. Due to privacy issues, web based data mining systems are not efficient for the purpose of constructing candidate's profiles, even if they can offer some help; therefore a distributed solution that permits each candidate's workplace to be explored was developed. As a means of aggregating collected knowledge for sharing, a conceptual model, relating candidate's profiles and their knowledge domain was developed in the form of an ontology. This strategy allows the system to be specialised in terms of the candidates' main knowledge domain.

Keywords: agent-based system, ontology, knowledge acquisition, knowledge sharing, distributed engineering

1. Introduction

Innovation is an important survival strategy for manufacturing enterprises, permitting them to face competition and market demands, however a strategy like this must be accompanied by organizational and structural changes in the way companies work [3]. Innovation management is a multidisciplinary process that congregates people with different background within an enterprise. In the case of manufacturing enterprises these people could be, for instance, experts in management, marketing, engineering design, production, shop-floor control and logistics. All of them could supply viewpoints and insights that could lead to the development of a new product or process that is an import outcome of the innovation process within enterprises.

New Product Development process is fundamental in manufacturing. Today, with the advent of the extended and virtual enterprises, it is clear that this process must be distributed among enterprises, as part of a Distributed Engineering process.

Clearly, team establishment is an enabler for innovation and knowledge sharing. Teams have a life cycle that encompass phases of forming (establishment), brainstorming, norming and performing (operation). The system proposed in this paper is intended to help in the distributed engineering team establishment phase. In this phase, the team has not as yet been formed, and there is a set of candidates with multidisciplinary backgrounds that possibly do not know each other and have not as yet developed social relations. The awareness of others (experts from another enterprise) is low; and in addition, the use of searching tools or recommendation systems based on the web offer little help as, due to enterprises' privacy policies, employees usually are not allowed to maintain web pages containing their skills and historical technical data from their past assignments.

As a solution to the problem of finding candidates with expertise in the domain of interest, circumscribed to the geographical domain of the enterprises / institutions participating in the distributed engineering effort, the proposal herewith is to develop a recommendation system for candidates with similar technical profiles.

After the establishment phase the new team /community could use the system to keep informed in terms of other members' interests changes, in a way that could enhance dynamics and learning within the team.

SHEIK (Sharing Engineering Information and Knowledge) system was developed to fulfil these objectives and it will be described and discussed in the remaining sections of this paper that are: related work, system description, system testing and conclusion.

2. Related work

The problem of cooperative engineering design is not new, as Willaert et al. [13] pointed out; the novelty is in the forms enterprises are adopting. No matter the name: extended, virtual or fractal, a common denominator is in its increasing distributed nature.

The problem of knowledge sharing in engineering was

Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)
0-7695-2416-8/05 $20.00 © 2005 IEEE

the motivation for SHADE / PACT projects [14]. SHADE treated the engineering design problem and used the concepts of meta-knowledge and ontologies; PACT was used to pack design tools in the form of an agent-based system.

Bradshaw et al. [2] showed how to use (KAos) agents for managing loosely-coupled information sources through a matchmaking approach in that "an agent desiring to use a service may ask a Matchmaker to recommend available agents that have previously advertised that service".

Davies & Edwards [4] discussed agent-based knowledge discovery, where they showed some organizational approaches for agents' distributed learning and the resulting need for knowledge integration.

Dieng [6] et al. discussed the materialization of Corporate Memories, including the case for virtual enterprises, and related how existing techniques can aid this process.

Recently, a number of recommendation systems emerged, some of them can be classified as being knowledge sharing systems and are ontology-based: OntoShare [5] to share opinions, OntoCOPI [1] to recommend Communities of Practice and QuickStep [9] to share selected web pages.

SHEIK shares many ideas with these works. It is targeted at the distributed engineering domain such as SHADE / PACT, but in a previous phase of the engineering process; team establishment. It is an agent-based system such as PACT and KAos and uses a form of matchmaking to recommend peers for collaboration.

SHEIK's aim is not to directly construct Corporate Memories, its approach is less time enduring and focused on team life cycle rather than on enterprise life cycle. Nevertheless, SHEIK uses some of the techniques listed in [6] such as: knowledge & ontology engineering, natural language processing and multi-agent systems (MAS).

3. SHEIK description

SHEIK architecture is presented in Figure 1. There are two types of agents in the architecture: Erudite agents and Sheik agents. Each Erudite agent is in charge of maintaining knowledge in a particular domain (for instance, manufacturing); each Sheik agent uniquely represents a candidate so as to set up a new team in this domain.

Sheik agents construct queries (asking for collaboration peers) according to their candidate's profiles and submit those queries to Erudite agents. Erudite agents receive queries from many Sheik agents, match the queries by means of a search for similar profiles and answer the requesting candidate about them, recommending a list of possible peers for future contact.

If the queries cannot be solved in their knowledge domain, Erudite agents can redirect them to another domain for instance the Computer Science Domain.

The relationship between Sheik and Erudite agents is described via a set of messages (Figure 2.), the endings Req and Rsp means request and response.

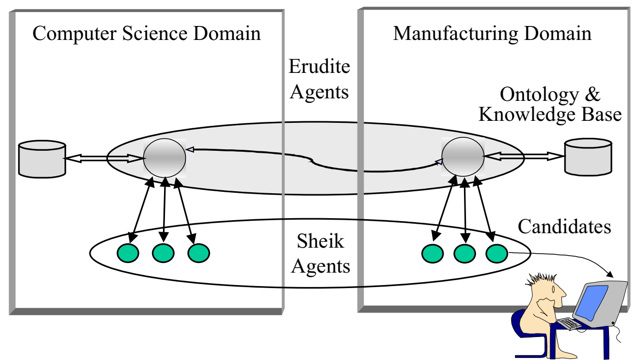

Figure 1. SHEIK Architecture

These messages are *ad-hoc* performatives and are not compliant with emerging FIPA standards (www.fipa.org) because compatibility would require the definition of ontologies in the knowledge domains of interest; as this was not possible, due to the lack of standard ontologies, this compatibility was postponed. The messages are:

subsribeReq and **unsubscribeReq** - are used by Sheik to request subscription / quitting of Erudite functionality for the finding of peers for cooperation, trying to join a potential new team.

subsribeRsp and **unsubscribeRsp** - convey the answer from Erudite.

queryReq – Sheik requests the Erudite agent to evaluate the candidate's current profile.

queryRsp - is the answer from the Erudite agent, that will try to search for a similar profile to that of the requesting candidate's and, if compatible profiles are found, it will send a list containing recommended candidates and contact information.

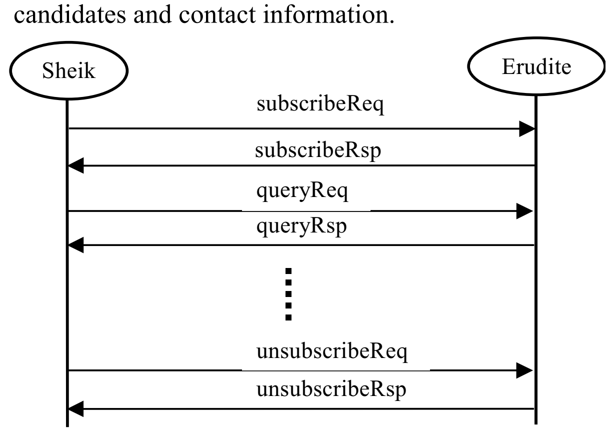

Figure 2. Sheik & Erudite UML sequence diagram

Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)
0-7695-2416-8/05 $20.00 © 2005 IEEE

A scenario of normal operation is shown in Figure 2. Sheik requests subscription to Erudite's functionality and, upon receiving a positive response, addresses queries to it asking peers for its local candidate. Similar scenarios were developed for other cases of interaction between Sheik and Erudite and between diverse instances of Erudite agents.

A state-machine was developed to control the dynamics of both the internal operation and messages exchange for each agent. Three sets of data were obtained: a set of messages exchanged among agents, a set of states for each agent in the MAS and a set of actions that were executed by each agent. These were the inputs for the implementation phase carried out using Java language.

Figure 3. shows Sheik and Erudite agent's interaction. A simplified model is used for each agent description, where there are three interacting parts: Behavior represents agent dynamics, how it responds to messages and events; Intelligence (knowledge acquisition - KA & knowledge base – KB) depicts how agents solve more complicated tasks; Communication is responsible for message exchange with other agents.

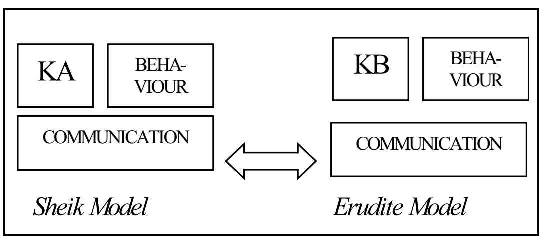

Figure 3. Sheik & Erudite models

Initial requirements analysis of SHEIK was made in 2001 [10] and used the Gaia methodology [12]. As Gaia was not directly supported by any integrated development environment, the specification was reengineered in order to enable fast prototyping, using a simplified methodology for MAS development (SMASDM) compatible with the simplified model shown. In a late phase of development, the system profited from the use of an agent prototyping tool (MASPT) developed in-house.

3.1 How it works

Sheik agents reside at the candidates' work places, on their computers and acting as interface agents. Each candidate has one Sheik agent instance and allows it to search his/her workspace, analyze working documents and prepare a local image of candidate's profile containing information upon his/her professional interests. To perform part of this job, the KEA tool [11], that contains an intelligent algorithm, based on a bayesian networks approach, was selected (Figure 3., KA). Sheik uses the candidate's workspace (Figure 4., left side, workDir) as input and infers what are the keywords (key-phrases) that characterize this space and thus the candidate's profile (Figure 4., right side).

Information coming from Sheik agents is contextualized and should be considered as knowledge in a loose sense. To enhance its quality it is filtered and classified according to an ontology that relates candidates' profiles and a taxonomy of manufacturing developed at the Integrated Manufacturing Technology Initiative - IMTI [8]. A meta-model that relates personal (technical) profiles and knowledge domains, structured according to the manufacturing taxonomy, configuring an ontology, was developed. To support ontology editing, Protégé environment was used [7]. It is an ontology editor and knowledge base, developed in Java and was embedded as part of the Erudite agent (Figure 3., KB block).

3.2 System testing

SHEIK's prototype testing consisted of supporting a team development process in the manufacturing automation domain. The test environment was made up of two laboratories; one at the Renato Archer Research Center - CenPRA and another at the State University of Campinas - Unicamp, School of Mechanical Engineering. Manufacturing is the main interest' area to both laboratories.

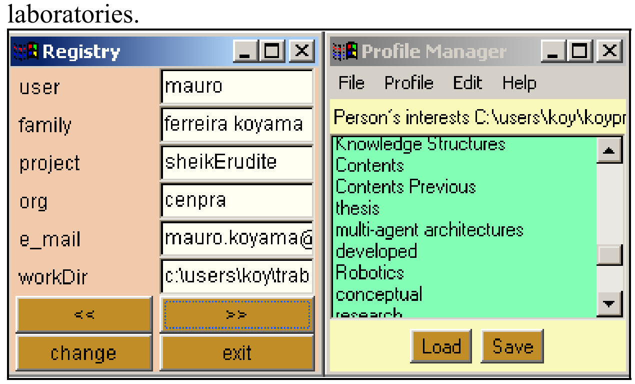

Figure 4. User's Interface

Candidate's profiles were filled out containing references to research areas and working domains defined in the IMTI taxonomy already entered in Protégé. An Erudite agent was installed at Unicamp, containing the related Protégé system. It was accessible via Internet.

At the CenPRA and Unicamp sites, Sheik agents gathered local user's data daily. Simulations of team formation was made by starting all agents and verifying what was the initial recommendation provided by Erudite. Local data presentation (Figure 4. – optional interface) was carried out using external modules.

To test Sheik's ability to request candidate's recommendation from Erudite, changes were made in the candidates' workspaces, by adding new technical documents related to others sub-domains of knowledge. Sheik agents tested the workspaces, verified the modifications and generated a new local profile containing the keywords related to the new documents. Again they requested Erudite to furnish a list of potential peers and received it.

Then candidates evaluated if the new list was meaningful to the team establishment purpose.

For instance, candidate Mauro from CenPRA, Figure 4., inserted a document containing the **robotics** keyword into his workspace. Sheik agent detected it and Erudite agent, upon receipt of Sheik's query, replied with a list containing a robotics' expert named João (not shown in the figure). Mauro then checked the list and contacted João, trying to establish a team to discuss robotics.

4. Conclusion

SHEIK is a ontology-based recommendation system and presents a novelty in both the way of doing the recommendation, using automatically collected specialist profiles and in the application area: distributed engineering. It differs from other ontology-based recommendation systems [1][5][9] in the recommended subject: persons with similar profiles, possibly peers for a collaboration and in the main purpose: to establish a new engineering team.

SHEIK allowed its users to gather information in their own working domain and increased their awareness of other specialists that worked in similar areas. That was done in a totally automatic way from users' viewpoint, as Sheik agents that resided in the users' machines were totally autonomous and did not require any intervention while analyzing users' documents and when requesting peers' list from the Erudite agent. Its purpose was not to develop knowledge acquisition and knowledge base components (the Intelligence part in the agent model) but instead to provide a frame that can accept the existing artificial intelligence techniques in the area, while also being compatible with related new developments; it intentionally uses public domain applications.

It integrated many concepts, such as knowledge management, knowledge acquisition & sharing, ontology and multi-agent systems, in an application to support part of the distributed engineering effort. Arrangements are being made to enable its use in a Brazilian national community of researchers, in the manufacturing field (around one hundred people). Such an experiment would allow the re-assessment of its validity and unveil its merits and drawbacks.

5. References

[1] H. Alani, , K. O'Hara, and N. Shadbolt, "ONTOCOPI: methods and tools for identifying communities of practice", Proc. of 2002 World Computer Congress, Intelligent Information Processing Stream (Montreal) 2002.

[2] J. M. Bradshaw, R. Carpenter, R. Cranfill, R. Jeffers, L. Poblete, T. Robinson, A. Sun, Y. Gawdiak, I. Bichindaritz, K. Sullivan, "Roles for Agent Technology in Knowledge Management: Examples from Applications in Aerospace and Medicine", Proc. of AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, 1997, pp. 9-16.

[3] Davenport, T. H., "Process Innovation - Reengineering Work through Information Technology", Harvard Business School Press, Boston, USA, 1993

[4] W. H. E. Davies, & P. Edwards, "Agent-Based Knowledge Discovery", Proc. of the AAAI Spring Symposium on Information Gathering from Distributed Heterogeneous Environments, Stanford University, California, March 1995.

[5] J. Davies, A. Dukes, and A. Stonkus, "OntoShare: Using ontologies for knowledge sharing", Proc. of the WWW2002 International Workshop on the Semantic Web, Hawaii, May 7, 2002.

[6] R. Dieng, A. Corby, O. Giboin, and M. Ribière, "Methods and Tools for corporate knowledge management", Int. J. Human – Computer Studies, 1999, v. 51, pp. 567-598.

[7] J.H. Gennari, M.A. Musen, R. Fergerson, W.E. Grosso M. Cruzby, H. Eriksson, N.F. Noy, S.W. Tu, "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development", Int. Journal of Human-Computer Studies, v. 58 (1), 2003, pp. 89-123.

[8] IMTI - Integrated Manufacturing Technology Initiative. 21st Century, "Manufacturing Taxonomy A Framework for Manufacturing Technology", Knowledge Management, v. 1.0, Dec. 2000.

[9] S. Middleton, D. De Roure, N. Shadbolt, "Capturing Knowledge of user preferences: ontologies in recommender systems", Proc. of First International Conference on Knowledge Capture, KCAP2001, Victoria, Canada, 2001, pp. 100-107.

[10] O. Nabuco, K. Drira, E. Dantas, "A Layered Design Model for Knowledge and Information Sharing Cooperative Systems", Proc. of tenth IEEE International Workshops of Enabling Technologies: Infrastructure for Collaborative Enterprises, MIT, Cambridge, USA, june 2001, pp. 305-310.

[11] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning, "KEA: practical automatic keyphrase extraction", Proc. Digital Libraries '99, Berkeley, CA, USA, August 1999, pp. 254-265.

[12] M. Wooldridge, N. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design", Journal of Autonomous Agents and Multi-Agent Systems, 3930, 2000, pp. 285-312.

[13] S. Willaert, R. de Graaf, S. Minderhoud, "Collaborative engineering: A case study of Concurrent Engineering in a wider context", Journal of Engineering and Technology Management JET-M, 1998, v. 15, pp. 87-109.

[14] M. Tenenbaum, J. Weber, and T. Gruber, "Enterprise integration: Lessons from SHADE and PACT", C. Petrie (editor), Enterprise Integration Modeling, MIT Press, 1993.

[15] M.S. Fox, "The TOVE project: Towards a common sense model of the enterprise", C. Petrie (editor), Enterprise Integration Modeling, MIT Press, 1993.

0-7695-2416-8/05 $20.00 © 2005 IEEE