



Journal of Artificial Intelligence

ISSN 1994-5450

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Multi-Agent Framework to Facilitate Knowledge Sharing

¹Abeer EL-Korany and ²Khaled EL-Bahnasy

¹Department of Computer Science, Faculty of Computer and Information,
Cairo University, 5 Dr. Ahmed Zoweil Street, Orman, Giza, Egypt

²Department of Information Systems, Faculty of Computer and Information Sciences,
Ain Shams University, Abbasia, Cairo, Egypt

Abstract: The aim of the study is to develop an intelligent web-based system that allows integration, easy access and sharing of domain specific Knowledge-Based Systems (KBSs). Development of KBSs requires continuous adaptation and extension of already deployed knowledge bases. Often, these modifications consider the improvement of the Knowledge Base (KB). Furthermore, examination of the KBSs in its actual user environment could broaden the KB. The proposed knowledge sharing environment allows different clients to access, maintain and upgrade domain specific knowledge bases. The proposed framework is generally applicable for KADS-based KBSs. As a matter of fact, we have applied it to agriculture diagnostic KBSs.

Key words: Knowledge base system, multi-agent system, community of practice, bayesian classification

INTRODUCTION

The dynamic nature of knowledge has long been realised: knowledge evolves over time as experiences accumulate; it is revised and augmented in the light of deeper comprehension; entirely new bodies of knowledge are created while at the same time others pass into obsolescence. However, this dynamism has rarely been acknowledged in the engineering of knowledge-based systems: systems of static knowledge elements are assumed to exist in unchanging environments. The increasing desire to deliver knowledge across distributed environments such as the World-Wide Web highlights the extent of this gulf: the Web is ever-changing and systems must be able to accommodate change if they are to succeed and thrive. To be able to describe the dynamics of knowledge, there seems to be a need for some level of formality. The role of formality is twofold: to give a concise account of what is going on and to use this account for practical purposes in maintaining and analyzing the life cycles of knowledge components from their creation, through successive phases of modification, to their eventual retirement (Schorlemmer *et al.*, 2002).

The presented research focuses on enhancing the process of knowledge-based systems construction by providing an integrated development, deployment and sharing environment that aims to provide up to date KBSs. This research proposes a knowledge sharing and collaboration environment with a prototypical application in the agriculture field. It gives a way to collaborate and share knowledge between web-users. The environment covers many aspects related to the content, the editing facilities and exchange of knowledge. It provides system users (client) with different views that enable each of them to access permitted knowledge according to user profiles. Two types of users are involved in this system, the first users type is domain experts that belongs to different agriculture disciplines who constitute the main source of knowledge. The other system client is the end users

(extension workers) who use the KBSS to support him in making his daily decision in giving the appropriate advice to growers. Based on their daily life expertise, users may require to update the domain knowledge. As a result, a need for specific domain experts to review and confirm user modifications arises. The proposed knowledge sharing environment allows both end users and domain experts to apply her/his modifications in structured manner without needs of knowledge engineer. It is obviously difficult to bring together information from different knowledge sources about a subject of common interest. Therefore, the proposed environment aims to close the gap between feedback knowledge obtained from the usage of the KBSSs in real world environment and the domain expert opinion. This is an important step in KBSSs development because it allows monitoring the system in its actual user environment. Then, specific domain experts extend the KB in their area of expertise based on experiment of KBSSs in real life environment.

Applying Multi-Agent Systems (MAS) (Wang *et al.*, 1999) that provide a degree of flexibility (generic) to different application areas. Software agents are broadly defined as software entities capable of getting information from their environment, reacting to it, communicating with other agents and achieving a goal of performing some complex task on behalf of their users (Ustyomenko, 2008). Effectiveness of the proposed knowledge sharing platform rely on its ability to automatically accumulate and refine end users modifications and transfer it to specialized domain experts to be reviewed. Knowledge exchange and sharing is necessary in this case to build an intelligent service that carried out by specific agent. The society of agents in our system consists of multi-agent diagnostic expert system, set of communication agents and global coordination agent. Collaborative communication between different agents is based on JADE platform. (<http://jade.tilab.com>).

Knowledge sharing framework such as the Rapid Knowledge Formation project (RKF) (Tecuci *et al.*, 2004) has been developed. RKF aims at building power-tools for knowledge acquisition. The central objective of this study was to enable distributed teams of subject matter experts to enter and modify knowledge directly and easily, without the need of prior knowledge engineering experience. Disciple-RKF, the implementation of the most recent version of the disciple approach, is an agent shell with a knowledge base structured into an object ontology that describes the entities from an application domain and a set of task reduction and solution composition rules expressed with these objects. The development of the knowledge base of a disciple-RKF agent is based on importing ontological knowledge from existing knowledge repositories and on teaching the agent how to perform various tasks, in a way that resembles how subject matter experts teaches a human apprentice.

Furthermore, a conceptual framework for facilitating the modelling and development of heuristic reasoning services such as diagnosis, therapy planning and patient monitoring in clinical domains was developed by Taboada *et al.* (2002). The purpose of that framework was to provide support for (1) the representation of medical knowledge in a structured manner, (2) the design of clinical tasks, (3) simplifying the development of executable clinical reasoning services from their design and (4) making available its use on the web, with two main objectives: helping to increase the use of decision-aid systems in the clinical practice; facilitating the design of these systems by reusing information and knowledge sources distributed in the Web.

Bioinformatics Knowledge Management System (BKMS) has been developed by Abdullah *et al.* (2008) using multi-agent systems in order to facilitate knowledge sharing among bioinformatics communities as well as to demonstrate it into the system wise. The researchers provided how the agent technology could be utilized in the BKMS as a system model for serving the communities that is developed by using groupware such as Lotus Notes software. The achievement in conducting this framework of the BKMS is an added value for the any organization that need to implement the BKMS as a system, which could be helped the CoPs to work together in achieving their aims and mission statements.

An agent mediated community of a practice system using MAS-CommonKADS methodology was developed by Peyravi *et al.* (2007). Each single agent, even those in the same community, needs its own autonomy in order to model an organization and its individuals correctly using this approach, the natural model for an agent based on knowledge sharing system has been resulted. All models of MAS-CommonKADS methodology required for developing the multi-agent system was described. An implementation resulted design model was presented. Rebeca was used for modeling and verifying this system.

PROPOSED KNOWLEDGE SHARING FRAMEWORK

The Knowledge Sharing Framework (KSF) focuses on providing an efficient framework for accessing, maintaining and refinement generic knowledge bases.

Knowledge Sharing

Knowledge sharing involves the transfer or dissemination of knowledge from one person or group to another. World wide web can be effectively used for knowledge transfer between individual of the same domain. The aim of the proposed Knowledge Sharing Framework (KSF) is to incorporate different knowledge sources in order to be able to build a complete Knowledge Base (KB) that has at least as much information as each KB components. This implies that knowledge is collective and able to evolve since it is liable to be extended through the interactions of the various system entities which store various parts of the available knowledge. This knowledge sharing framework is generally applicable to different domain. As an example, the system is applied in the agriculture domain. Expert systems are recognized as an appropriate technology for knowledge sharing because they address the problem of transferring knowledge and expertise from highly qualified specialists to less knowledgeable personnel. Moreover, the development of an agriculture expert system requires the combined efforts of specialists from many fields of agriculture domain and must be developed with the cooperation of the farmers and extension officers who will use them (Broner *et al.*, 1992). Knowledge transfer is always taking place from research to extension, from extension to farmers and even from farmer to farmer (Rafea and Shaalan, 1996). Different knowledge sources are considered here. The first one (the core) is a centralized generic diagnostic knowledge based-system. This generic model has the ability to derive diagnoses for differently structured individual systems from the agriculture domain. The generic diagnostic model (El-Korany *et al.*, 2004) was developed based on CommonKADS model of expertise (Schreiber *et al.*, 2000) that integrates the domain knowledge and the problem solving knowledge. Domain knowledge contains all knowledge about diagnosis in the agriculture domain, regardless of the crop. The problem solving process describes the reasoning process. The second source of knowledge is the knowledge acquired from end user (extension worker) based on their daily life expertise. End users can extend the domain knowledge based on feedback from the usage of the ESs in real world environment. Researchers have agreed that end users are only allowed to specify certain kinds of modifications, i.e., domain-specific knowledge regarding concepts and domain relation but not modifying problem-solving knowledge (Kim and Gil, 2000). The remaining knowledge source is domain specific experts in different agriculture diagnostic fields who review and confirm end user modifications. The role of the domain experts is to decide to include end users modifications in the centralized knowledge base.

Use of Multi-Agent to Support Community of Practice

According to Ferber (1999), a multi-agent system is a system composed of a population of autonomous agents which cooperate with each other to reach common objectives, while simultaneously each agent pursues individual objectives. Multi-agent system can be viewed as a system in which agents can communicate, exchange their individual knowledge and cooperate in order to solve complex

problems and to achieve collective or individual goals. Multi agent system could be developed and used to facilitate and support the community of practice (CoP) (Maier *et al.*, 2005). In this case, agent also could be categories in terms of its roles in collaborative context like in knowledge communication, knowledge monitoring and others. Three main agent categories are involved in the proposed system like:

- Service agent that to assist in local production activities, like ES agent
- Monitoring agent that aims to monitor, collect and publish modified knowledge, like global coordination agent
- Interaction agent to assist with cooperative work and to mediate communication among people, like user community agent and domain expert community agent

A prototype for knowledge sharing environment was designed and implemented using JADE platform. JADF (Java Agent Development Framework) is a software framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the Foundation for Intelligent Physical Agents (FIPA) specifications. Collaborative communication between agents occurs in the form of request and reply messages. Messages between different system components are exchanged based on JADE platform. Agents provide two important capabilities in the combined system:

- Communication: Ability of sharing information among agents
- Cooperation: The joint effort of different agents that decompose the system aiming at achieving one goal (Kone *et al.*, 2000). This goal originally would go beyond their individual capabilities. Cooperation-agents are responsible for coordination the overall operation

SYSTEM ARCHITECTURE

From the architectural point of view, the system provides a framework to coordinate the behavior of several specific agent types. The proposed architecture has the advantage to be fully generic, only the type of knowledge should be modified. The society of agents in this system consists of multi-agent diagnostic expert system, set of communication agents and global coordination agent. The diagnosis knowledge is distributed among several agents, each agent is an autonomous expert for a certain agriculture crop. The end user community agent and the domain expert community agent employ intelligent data collector working on the different client sides. Those agents communicate by passing messages to the global coordination agent.

Multi-Agent Diagnostic Expert System

Expert systems running on the Internet can support a large group of users who communicate with the system over the network. The applied multi-agent expert system constitutes a set of agents capable of diagnosing disorders of different crops in the agriculture domain. Each expert system agents provides the reasoning capabilities as well as the domain knowledge which is required to carry out its tasks for a certain agriculture crop. A diagnosis agent is responsible for producing diagnosis of disorders based on a set of input observations.

All agent communication takes place through by passing messages to the Expert System Coordinator Agent (ESCA) (Fig. 1). The main functionality of this agent is to react to external request (normally from the global cooperation agent) and select the appropriate expert agent and assigning the request to it. A communication protocol that aims to ensure concurrent interactions between ES agents working on the resolution of a problem, together and interact with the environment is provided through ESCA.

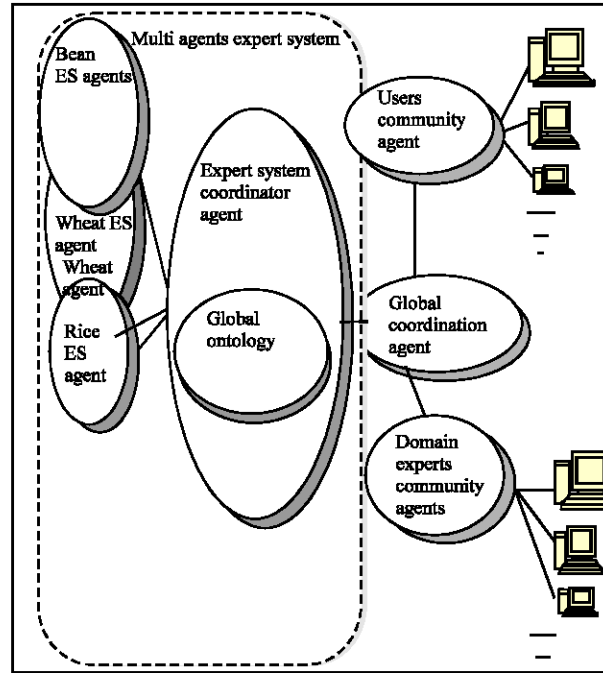


Fig. 1: The overall structure of multi agents system

Domain Expert Community Agent

Diagnostic KBS combines different specializations that are well defined in the agricultural domain. The agricultural diagnosis domain has been classified into six groups, namely: fungal, insect agent, nematode, snail, mites and nutrition deficiency. This classification is done according to the diagnosis taxonomy in the agriculture domain (Shalan *et al.*, 2004). It has the following advantages that contribute to the suggested framework. First, it tries to simulate what may happen in real world in a very fine-grained manner. Second, partitioning of domain knowledge makes it easy to maintain and refine the domain knowledge. Each of diagnosis class is assigned to a specific domain expert (s) who is authorized to review and confirm any user modifications. The main functions of the Domain Expert Community Agent (DECA) are:

- Establish a community of experts specialized in different agriculture diagnostic fields
- Receive the set of guilty rules from the global coordinator agent and distribute among domain experts according to their specialization
- Retrieve the confirmed modifications from domain experts and transport it to global coordinator agent

It is also responsible for authenticating the domain expert of the KSE and authorizing them to the appropriate resources. It manages these activities through a database of domain experts.

End Users Community Agent

The knowledge sharing framework allows users to register and deploy their local KBS, modify it and receive updates from their colleagues in the community based fashion. Therefore, the two main functionality of End User Community Agent (EUCA) are:

- Track users modifications
- Establish a community of practice

Each of these functions is illustrated in the following subsections.

Track Users Modifications

Users need to access and interact with KBS from different perspectives depending on the tasks they are completing. End users (extensions, highly experienced growers) interact with the system to complete their sessions by applying their own diagnostic cases and obtaining the system result. When the system does not (does no longer) function as required, users may require to update any of the domain specific knowledge. Since in real life, the symptoms of disorders are likely to change, also may others disorders need to be added/removed from the original KB. When the End User Community Agent (EUCA) receives a user request to join the community, it authenticates the user to the KSE, provide him with a modifiable local KBs, user refinement module and create log files for this user to keep track of her/his modifications. When the end user applies any knowledge modification, the refinement module provides him with a complete guidance about the effect of her/his modifications on the domain knowledge to avoid any knowledge inconsistency (Bahnsy and El-Korany, 2008). When the user confirms her/his modification, the log file is activated. Regularly, the end user community agent send approved user modifications to the global cooperation agent. End user community agent manages all user activities through data base. It is also responsible for applying the collaboration mechanism between the global cooperative agent and end users.

Establish a Community of Practice

The shared knowledge building and experiences from a community of practice could provide the technical advisor with the confidence to support other academics with similar knowledge of practice (Molphy *et al.*, 2007). Knowledge update done by each individual user solely may be of added value to other users in the community. Therefore, there is a need to provide system users with the contribution of the updates applied by different community members. End users community agent aims to constitute a community of users with joint expertise to form a Community of Practice (CoP). Since each individual user applies her/his modification on her/his local KB, this modification is opaque to her/his colleague. Therefore, the main functionality of End User Community Agent (EUCA) is to allow other users in the community to share each knowledge update (of their interest) and to decide whether to include it in their individual KB or to neglect it. This is accomplished by first classifying users into a set of group based on their interest. Then, knowledge modification is supplied to group of users of the same interest. User interest is obtained based on the customary modification type that is frequently done by each user. As describe earlier, the agriculture disorder taxonomy is decomposed into the following clusters namely: fungal, insect, nematode, snail, mites and nutrition deficiency. End user profiling is done based on this taxonomy i.e., depends on weight (heaviness) of users modifications applied on a specific diagnostic class(es).

Global Cooperative Agent

Agent cooperation is a key issue in a multi-agent system (Lejouad-Chaari and Mouria-Beji, 1998). The Global Cooperative Agent is (GCA) the core of the presented environment. It provides the deployment and cooperation services to its clients over the network. The process of knowledge monitoring, exchange and integration is the main responsibility of the global cooperative agent. Moreover, it is also responsible for publishing the modified versions to corresponding users. This can be done through:

- Monitoring knowledge exchange
- Update Generic KBs

Monitoring Knowledge Exchange

In the proposed architecture, the central KBs have two types of users: the end users and the domains experts, each working on his corresponding side. While the end users apply their changes on their local KBs, domain experts would refine these changes before applying it to the centralized KBs. When the end user community agent sends the accumulated users modification to the GCA, it proceeds and executed the incremental refinement algorithms. A set of guilty rules is then transmitted to domain expert community agent which in turn distributes it among domain expert based on their specialization. Moreover, when expert community agent responds and sends confirmation of domain experts back to the GCA, it provides expert system coordination agent with these modifications to be applied to original KBs and publish the new version. Knowledge exchange is transferred between agents by applying message-passing protocol. In multi-agent environments, messages should be schedulable, as well as event driven. They can be sent in synchronous or asynchronous modes. Furthermore, the transportation mechanism should support unique addressing. Applying direct communication protocol, each agent want to communicate must explicitly name the receipt or sender of the communication (Kone *et al.*, 2000). In this scheme the following are required.

- Describe the prototypical scenarios between agents, the conversations are identified through the following primitives:
 - Send (agent, message) A message is sent to agent
 - Receive (agent, message) receive a message from agent
- Represent the events (interchanged messages) between agents which reflect the relationships between the agents via their services

The use of global ontology helps us to define the interchanged knowledge structures. These functionality are fully supported through using JADE platform.

Update Generic KBs

The knowledge engineering process involves (at least) three parties: the intended end-user of the system, the domain expert who provides the expertise that forms the basis of the system and the knowledge engineer, who acquires the knowledge of the domain expert and uses it to construct the system. The hope of knowledge engineering has long been to remove the knowledge engineer from the adaptive loop and provide tools that enable the domain expert to adjust the system to changing knowledge requirements. This functionality is applied through the GCA that is responsible for regularly gathering all user modifications kept by the end user community agent. These modifications are stored in a global modifications database to be used by the incremental refinement algorithm. When the domain experts approve user modifications, it is transmitted to the original KB to be integrated and published to end users again.

OVERALL PROCESS

The intended form of representation for the KB repository is XML structures. Due to the structural nature of XML and its high interchangeability, representing domain knowledge in XML would provide syntactical standardization well suited for domain knowledge exchange when there is an agreement on their intended semantics. The suggested knowledge sharing platform decomposed the following functionality:

- Local refinement
- User profiling
- Incremental refinement

Each of these functionalities is described in details in the following subsections

Local Refinement

At the user site, the local refinement module handles user modifications that affect the semantic of the knowledge base. For example, when a user wants to add new rule (or any KB component) and as she/he is not familiar with the classification criteria upon which the rule clustering taxonomy was built. As a result, she may add this rule at incorrect rule cluster. The objective of the local refinement module is to advice the user about her/his modification request and its effect on knowledge semantic to avoid any inconsistency. When any user submits her/his modification to her/his local KB, the local log file is activated. The details of the local refinement module is explained in by Bahnasy and El-Korany (2008).

User Profiling

After the user admits her/his modification to her/his local KB, this modification is circulated to the community of users based on their profiling. Thus, only users of common interest are notified about the newly added knowledge. This helps users to save their time as they only get particular modifications instead of receiving all knowledge update. Moreover, when more than one user confesses the same modification on her/his local KB, this provides an indication about the importance of this modification. Accordingly, this ratio is transferred to the global cooperative agent to be considered by incremental refinement module. User profiling is decomposed from two main phases.

- User profile phase
- Ranking procedure

User Profile Phase

User profile phase is used to identify persons inside a community with similar interests. Users in the same group are held together by a common interest in a body of knowledge and are driven by a need to share problems, experiences and best practices (Kim and Gill, 2000). Thus, in the proposed environment, the process of user profiling is based on user frequency of applying modifications to the same knowledge cluster. This helps locate and cluster similar users to facilitate relevant knowledge sharing. Moreover, it helps to find users who apply the same changes on their local KB. User profile algorithm is shown in Fig. 2.

```
1 FOR ACTION all member is accessed by CoP
2 Stamp user id, knowledge cluster id
3 If new knowledge cluster id THEN
4     Create new knowledge profile
5 If new user id THEN
6     Create user-status = 1
7     ELSE
8         Update user-status by increased 1
9 ENDIF
10 If user-status >3 THEN
11     append user in knowledge profile
```

Fig. 2: User profiling algorithm

```
1 FOR ACTION all users is access by the user profile
2 stamp knowledge modification token
3 Stamp user id
4 IF save is OK THEN
5     Increment knowledge status by 1
6 IF knowledge status >3
7     Store all particulars in Log-File
8     Transfer knowledge modification to GCA
END IF
NEXT ACTION
```

Fig. 3: Ranking algorithm

Ranking Procedure

A Ranking procedure is used to indicate the ratio of approved modifications by the community of users. It represents the number of end users of the same profile who apply the same modification to their local KBs. This ratio represents the confidence of the applied modification and is used by the incremental refinement techniques. In order to calculate this ratio, a token that represents the knowledge modification is circulated to the all users in the same profile. When a user selects to include this modification in her/his local KB, the token is updated. When the number of users exceed certain threshold (for example 3), this modification is transferred to the Global Coordinative Agent (GCA) to be used by incremental refinement. Figure 3 presents the ranking procedure used.

Incremental Refinement

The aim of the incremental refinement phase is to recommend to the domain expert the possibly of including the collected user modification to the original KBs. This is done based on two important factors: knowledge similarity and ranking ratio. The former indicate the similarity measure (Rong and Edward, 2007) of the updated knowledge w.r.t knowledge consistency. While the later represents the ratio of approved modifications by more than one user in the same community. The incremental refinement consists of two important phases:

- Knowledge localization
- knowledge inconsistency removal

Knowledge Localization

Knowledge localization is the process of locating the updated knowledge in the appropriate knowledge cluster. When a knowledge base component is updated, the system suggested the suitable knowledge cluster that should include this newly knowledge component based on the structure and semantic similarities of knowledge. For example, newly diagnostic rules are obtained from user in form of antecedence-consequence pairs. Antecedence represents the symptoms (finding) that lead to certain disorder, while consequence presents the disorder. The user can freely selects any rule cluster to add in this new rule. The new rule must have at least one dominant finding match with the specified rule cluster from the user. Otherwise, it has no similarity and cannot be added to that rule cluster. The knowledge localization phase is responsible measuring the similarity of any modified rule (for example) to all rule clusters. The task of classifying a rule to the nearest (most similar) rule clusters, is most likely to explain on a firm theoretical foundation using Bayesian statistics. The key problem here is to get the likelihood of the given description (findings) given the classes. Applying Bayesian Classification theory (Hanson *et al.*, 1991) to get the most confidence knowledge cluster that

matches each newly added knowledge components. The formula used attempts to understand more about the knowledge base component by considering its semantic similarity w.r.t other knowledge clusters.

$$p(C|F) = \frac{P(F|C)P(C)}{P(F)}$$

- P(c) prior belief (probability of rule cluster)
- P(F|C) likelihood (probability of the finding match inside each cluster)
- P(F) = $\sum P(F|C) P(C)$ Finding evidence (marginal probability of the finding overall the classes)
- P(C|F) posterior belief (probability of class C to include finding F)

The mathematical formulas say how likely any particular finding would be confidence to a certain cluster. Next, confidence of each finding to all cluster is calculated. The confidence values of each finding indicate whether it is dominant and subsidiary features w.r.t diagnosis class. Dominant finding are those prototypical (Rosch, 1975) for the diagnosis class membership, more importantly, while subsidiary finding are those overlaps between classes.

Knowledge Inconsistency Removal

At the domain expert site, each domain expert (according to his specialization) receives a set of guilty rules. Those guilty rules are categorized with two different beliefs. The first belief represents the ranking ratio (votes) which is the actual number of users that select to apply this modification in their local KB. The second is the confidence measure obtained from the knowledge localization. Now, based on these values the domain expert can decide whether to accept these modifications or not. Then the domain expert community agent will send the accepted modification to the global coordination agent to be transmitted and included in the centralized KBS.

CONCLUSION

The aim of this study is to facilitate the sharing, access, updating of generic diagnosis agriculture KBSs residing on server and allow both end users and domain experts to apply her/his modifications in structured manner without needs of knowledge engineer. The society of agents in this system consists of multi-agent diagnostic expert system, set of communication agents and global coordination agent. Unlike other knowledge sharing platform, the proposed knowledge sharing environment is capable of gathering and integrating feedback knowledge from end users. This feedback knowledge reflects the real life experience is included with the original KBs in order to enhance the process of constructing knowledge bases without a need for knowledge engineer. Moreover, the existence of specialized domain experts who confirm any modification before updating the centralized KB robust the construction process of domain specific KBSs.

REFERENCES

- Abdullah, R., H. Ibrahim, R. Atan, S. Napis, A. Jaafar, M.H. Selamat, 2008. Applying agent technology to facilitate knowledge sharing among bioinformatics communities of practice. *IJCSNS Int. J. Comput. Sci. Network Security*, 8: 310-317.
- Bahnasy, K. and A. EL-Korany, 2008. An evolutionary model for knowledge-based management system development. *Proceedings of the International Conference of Information and Computer System*, Mar. 27-29, Cairo, Egypt, pp: 56-65.

- Broner, I., A. Parente and K. Tompson, 1992. Knowledge-based systems for crop management. Proceedings of the 4th international conference on Computers in Agricultural Extension Programs, Jan. 28-31, Orlando, Florida, St. Joseph, pp: 117-122.
- El-Korany, A., E. El-Azhary and M. Yehia, 2004. An approach for building generic diagnosis model in agricultural domain. Proceedings of the 5th International Workshop on Artificial Intelligent in Agricultural, Mar. 8-10, Cairo, Egypt, pp: 75-80.
- Ferber, 1999. Multi-Agent Systems: An Introduction to Artificial Intelligence. Addison-Wesley, Redwood City, CA., ISBN: 0-201-36048-9.
- Hanson, R., J. Stutz and P. Cheeseman, 1991. Bayesian classification theory. Bayesian classification theory. Technical Report FIA-90-12-7-01. ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19920017651_1992017651.pdf.
- Kim, J. and A. Gil, 2000. Acquiring problem-solving knowledge from end users: Putting interdependency models to the test. Proceedings of the 7th National Conference on Artificial Intelligence, (AAAI-2000), American Association for Artificial Intelligence, pp: 223-229.
- Kone, M.T., A. Shimazu and T. Nakajima, 2000. State of the art in agent communication languages. Knowledge Inform. Syst., 2: 259-284.
- Lejouad-Chaari, W. and F. Mouria-Beji, 1998. High-level communication protocol in a distributed multi-agent system. Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems: Methodology and Tools in Knowledge-Based Systems, LNCS 1415, June 01-04, Springer-Verlag London, UK., pp: 606-616.
- Maier, R., T. Hadrich and R. Peinl, 2005. Enterprise Knowledge Infrastructures. Springer Berlin, Heidelberg, ISBN: 978-3-540-23915-4.
- Molphy, M., C. Pocknee and T. Young, 2007. Online communities of practice: Are they principled and how do they work? Proceedings of the Providing Choices for Learners and Learning, 2007, Ascilite Singapore, pp: 710-716.
- Peyravi, F., K. Pashaei and F. Taghiyareh, 2007. A multi agent community of practice. Neural Network World, 17: 397-413.
- Rafea, A. and K. Shaalan, 1996. Using expert systems as a training tool in the agriculture sector in Egypt. Exp. Syst. Appl., 11: 343-349.
- Rong, C. and H. Edward, 2007. Clinical diagnosis based on bayesian classification of functional magnetic-resonance data. Neuroinformatics, 5: 178-188.
- Rosch, E., 1975. Cognitive representations of semantic categories. J. Exp. Psychol., 104: 192-233.
- Schorlemmer, M., S. Potter, D. Robertson and D. Sleeman, 2002. Knowledge life-cycle managover a distributed architecture. Expert Update, 5: 2-19.
- Schreiber, G., H. Akkermans, A. Anjewierden, R. De-Hoog, N. Shadbolt, W. Van-De-Velde *et al.*, 2000. Knowledge Engineering and Management: The Common KADS Methodology. MIT Press, Cambridge, MA., ISBN: 0-262-19300-0.
- Shaalan, K., M. El-Badry and A. Rafea, 2004. A multiagent approach for diagnostic expert systems via the internet. Exp. Syst. Appl., 27: 1-10.
- Taboada, M., M. Argüello, J. Des, J. Mira and D. Martínez, 2002. Heuristic reasoning services for medical environments. Proceedings of the 3rd ACIS International Conference on Software Engineering, 2002, Madrid, Spain, pp: 1-7.
- Tecuci, G., M. Boicu, D. Marcu, B. Stanescu, C. Boicu and M. Barbulescu, 2004. Parallel knowledge base development by subject matter experts. Proceedings of the 14th International Conference, EKAW 2004 Whittlebury Hall, UK., LNCS 3257, Oct. 5-8, Springer Berlin/Heidelberg, pp: 265-279.

- Ustyomenko, S., 2008. Multiagent dynamic reasoning about belief and trust. College of Arts and Sciences, Florida State University. <http://etd.lib.fsu.edu/theses/available/etd-04152008-001210/unrestricted/d.pdf>.
- Wang, A., R. Conradi and C. Liu, 1999. A multi-agent architecture for cooperative software engineering. Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, June 17-19, Kaiserslautern, Germany, pp: 1-22.