International Journal of Engineering &Technology, 7 (3.12) (2018) 616-620

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



KM Trajectory Schema Service Frame Work for Software Development Organizations

M.A. Krishna Priya[1], Justus Selwyn[2]

[1]Research Scholar, Computer Science Department, Bharathiar University, Coimbatore, India
[2]Professor, SCSE, VIT University, Chennai, India
**Corresponding Author Email: priya_krishi@yahoo.co.in[1], justus.s@vit.ac.in[2]*

Abstract

Software Development Organizations are in the need to follow an effective KM strategy for Knowledge acquisition and Knowledge utilization. Even though, knowledge is available in the forms of documents, it is time consuming for the developers to identify and utilize the knowledge according to their need. Hence, we present a KM Trajectory Schema - Service Frame work which offers services for systematic Knowledge acquisition and Knowledge utilization. Knowledge is acquired from the development activity, and then the developer's code is stored in knowledge base. Knowledge Base Administrator designs a repository; knowledge base is appended by knowledge workers. Assessment and Evaluation of this Knowledge base is been done through the subject experts or knowledge champions. Finally Solution Knowledge Base is built and managed by Knowledge Base Administrators. KM service Framework facilitates Knowledge utilization by providing proactive semantic help for the software developers by the time of the development. KM Service Framework is an experimental collaborative platform that provides great service to the Software Development Organizations.

Keywords: *Knowledge Management (KM); Knowledge Management System (KMS); Knowledge Base (KB); Software Development Organizations (SDO)*

1. Introduction

In this competitive environment, knowledge management in software development organizations is essential. Organizations are in the great need to frame a sustainable knowledge management strategy with an effective framework which offers services for the knowledge workers in their highly sensitive, knowledge intensive software development activity. Grant (1997:451) argues that 'knowledge is the overwhelmingly important productive resource in terms of its contribution to value added and its strategic significance'. However, knowledge in itself will not ensure the success of an organization. This paper gives the idea to frame a sustainable knowledge management strategy with an effective framework which offers services for knowledge workers. Here, focus is on knowledge acquisition to develop an effective solution knowledgebase for explicit utilization of knowledge. Software developers /Knowledge workers can avail its best service using solution knowledgebase; can achieve solutions by using the proactive suggestions from the help assistant.

2. Research Method

The research methodology followed to address the above mentioned objective consisted of three main steps. The first step is acquisition of knowledge to collect data; asses evaluate and create a solution knowledge base. The second step is appropriate utilization of knowledge at the time of the need with the help of the solution knowledge base. The final step is to combine these ideas to derive a knowledge management service framework for providing the necessary knowledge to the software developers using blackboard systems. This framework is designed for software development organizations to manage the essential knowledge resources and it provides functional solutions for improving the software development process, its services enable the software developers to complete their task successfully within the stipulated time period.

3. Overview of KM Schema, Pattern, Repository and Technologies

1. Schema:

A schema is a broad and conceptual plan or scheme for a class of situations. This is a generalized concept which defines our understanding of the underlying structure, nature, or principles of a general type of story, situation, or "system". This is a spatially or temporarily organized structure in which the parts are connected on the basis of contiguities that have been experienced in space or time. A schema is formed on the basis of past experiences with the objects, scenes or events and consists of a set of non conscious expectations about what things look like and the order in which they occur. The parts or units of schema consists of a set of variables or slots which can be filled or instantiated in any given instance by values that have greater or lesser degrees of probability of occurrence attached to them. Schemas vary greatly in their degree of generality-the more general schema, the less specified, or the less predictable, are the values that satisfy them. [Mandler, 1979, p.263] Schemata are concepts or mental models by which a static or dynamic situation can be characterized and understood. Schemata are typically abstract models of a generalized situation. Scripts are more concrete and specific than schemata and can be

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

generated from schemas to form more definite expectations for evolutions of specific situations.

2. Patterns:

The main idea behind using patterns is to provide documentation to help, categorize and communicate solutions to recurring problems. A definition that closely reflects its use within the patterns community by Riehle and Ztillighoven: A Pattern is instructive information that captures the essential structure and insight of a successful family of proven solutions to a recurring problem that arises within a certain context and system of forces. Coplien explains that a good pattern will do the following: It solves a problem with solutions, It is a proven concept with a track record, the best patterns generate a solution to a problem indirectly a necessary approach for the most difficult problems of design, patterns do not just describe modules, but describe deeper system structures and mechanisms, the best patterns explicitly appeal to aesthetics and utility. Initially patterns were developed to solve design problems. However, recent patterns encompass all aspects of software engineering, including requirements engineering, project planning, software development process and software configuration management. Every pattern has to be expressed "in the form of a rule-template which establishes a relationship between a context, a system of forces which arises in that context and a configuration, which allows these forces to resolve themselves in that context. A pattern should contain certain essential components like Name, Problem, Context, Forces, Solution, Examples, Resulting Context, Rationale, Related Patterns, Known uses; these components should be clearly recognizable on reading a pattern. A good pattern begin with an abstract that provides a overview, that gives a clear picture to the users and quickly informs them of its relevance to any problems they may wish to solve. A pattern represents a "best practice" where as anti pattern represents "worst practice". Anti patterns describe bad solution to a problem that resulted in a bad situation and shows how to get out of a bad situation and how to proceed from there to a good solution. Creating knowledge patterns is very difficult, explains Appleton. Patterns are a reference manual that provides facts and also tell a story that captures the experience they are trying to convey. Knowledge patterns help its users comprehend existing systems, customize systems to fit user needs and construct new systems. Buschmann says that pattern writers do not need to be original inventor or discoverer of the solutions that they document. Pattern submissions are shepherded rather than reviewed.

3. Knowledge Repository:

In the context of knowledge management literature 'repository' means a place or person regarded as a store of information for future retrieval towards knowledge capturing purposes. A knowledge repository is a computerized system that systematically and continuously captures, organizes, categorizes and analyses the knowledge assets. The repository searches data and quickly retrieves. It's a collaborative system where people query; browse both structured and unstructured information. The scope of repository system is storing unstructured form of knowledge. Such as syntax rules for writing syntax and semantics for writing procedures that are yet to be written. Knowledge repository aims to retrieve data in a context sensitive way substituting the simple keyword based retrieval. Highly effective knowledge repository, serve as a single point of contact for all software needs. Knowledge sharing through repository happens in different layers, knowledge sharing through terms gives facts, specific details and elements. Sharing concepts gives theories, models, principles and generalizations. Sharing Procedures is for skills, algorithms, techniques and methods. Sharing cognitive knowledge is about learning, thinking and problem solving. Knowledge repositories offer a range of options for users to explore access and identify knowledge sources. Repositories may have a different set of processes and roles by which its content is created, refined and stored. Complex knowledge management problems naturally require multiple repositories, segmented by a degree of interactivity, volatility of content, or the structure of the knowledge itself.

4. Knowledge Management Technologies:

Knowledge management technologies support acquisition, dissemination, sharing and application of knowledge. KM Applications supports knowledge sharing among users. And also these applications exhibit a sequential flow of knowledge into and out of the repository. The repository becomes a premium medium for knowledge exchange. KM tools enable and enhance knowledge generation, codification and transfer of knowledge. Knowledge generation facilitates knowledge availability to users. Knowledge transfer decreases problems with reference to time and place by adopting proper communication within the schema. Intellisense is a Microsoft's Intellisense is a sophisticated completion technology, it can save a lot of typing and it often let the developer to avoid looking up stuff in the online help. Most probably developers are using this technology at work already while writing the code. Intellisense pops up little list boxes with helpful information about the objects. The developer can accept the suggestion of the Intellisense by pressing the Enter key.

5. Knowledge Management Framework:

KM Framework provides predictive logic i.e idea and supports the software developers with its proactive suggestions as service. Service framework comprises of two parts, Capture and Synthesize, Extract and Use. In the first part, solution schema is synthesized with solution patterns. In the second part, semantic help assistant is used for knowledge utilization. Our framework is modeled to improve the software development process. Functional knowledge solutions are extracted from the solution repository, Semantic help assistant supports the software developers with its proactive suggestions.

4. KM Trajectory Schema

Knowledge Management Trajectory Schema shows the traces of methods and solutions for a particular problem scenario, software developers could opt for the best choice in their software development activity. The schema can be articulated with three layers called Knowledge Harvesting, Knowledge Mapping and Knowledge sharing.

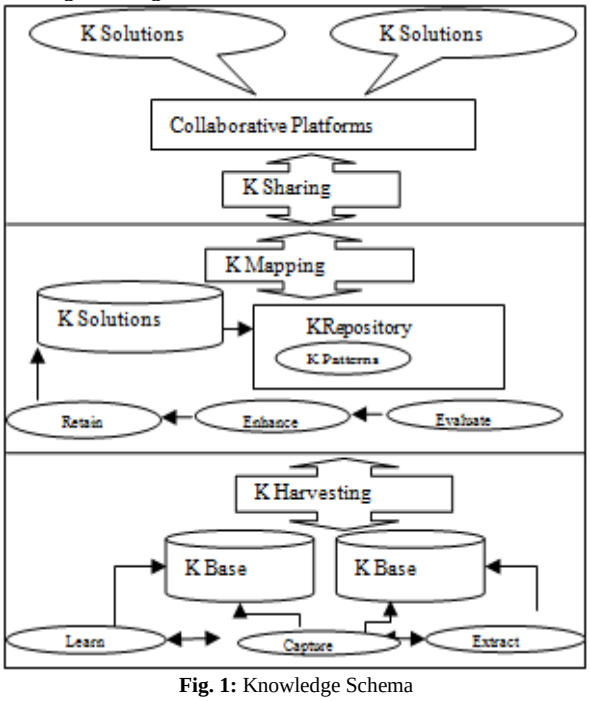

Fig. 1: Knowledge Schema

4.1 Knowledge Harvesting:

K Harvesting consists set of methods for making tacit knowledge more explicit-getting people's knowledge into documents, so that it can be more easily shared with others. Knowledge workers harvest the available knowledge from various resources. Knowledge base is a basic component for the knowledge based system which contains system's domain knowledge in some representation suitable for the system. Knowledge in knowledge bases is typically represented in a standard format. Initially, Knowledge base is created and knowledge acquisition happens through learn capture and extract data from knowledge workers and stores it in knowledge base. Here, codes are written by software developers. Knowledge extraction algorithms could be utilized for capturing the code patterns. There are different approaches to construct knowledge extraction systems; Primary approach is to develop reliable knowledge extraction rules by coding conditional expressions that identify the desired knowledge patterns. Due to the development of various software languages and its code formats, developing extraction rules manually is tedious but in this paper we used simple extraction methods for experimental purpose.

4.2 Knowledge Mapping:

Knowledge mapping helps to navigation between explicit information and tacit knowledge. This shows the relationships between knowledge stores and dynamics. Knowledge mapping could be approached in two ways, one is where the knowledge exists and the other is how knowledge flows or moves. Knowledge assets or the knowledge resources are the places where we can find the knowledge. Attention is required to identify the need of the knowledge and supply it from the available resource to the needy. Knowledge mapping could be helpful to maintain the knowledge flow. After knowledge harvesting, Evaluation of Code data is been done by the technical experts and enhancement of code is been done by knowledge champions if required. And finally code data is retained in Knowledge Solutions base. Knowledge management team members are the pattern writers. They write knowledge patterns, which are shepherded by subject experts and knowledge champions, they discuss with the team members that might be clarified or improved. K Solutions provide knowledge patterns to knowledge repository. Knowledge patterns are stored in the knowledge repository for sharing the knowledge.

4.3 Knowledge Sharing:

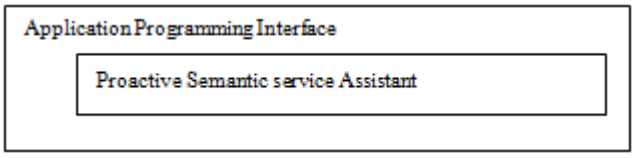

Fig 2: Collaborative Platform

Efficient and effective knowledge sharing in our schema could take place with the help of IT tools. Our Collaborative platform is an experimental proactive semantic help assistant tool that can be used as plug-in with any application programming interface. This development interface supports the developers by providing appropriate knowledge. Tool research perhaps helps us in the synthesis of km solutions

5. Generation& Management of Knowledge Bases

The procedure involved in the processing and management of code data has been simplified with the development of a parser to acquire the code block as an attribute and store it as knowledge pattern files typically done by learning algorithms. This removes the need to perform the conversion manually. The files contain the data used to create final pattern set and information concerning what this data describes.

5.1 Analysis of Knowledge Base Management Process

To develop an agent based system for the management of knowledge bases, the manual processes involved in knowledge gathering and processing procedures should be identified, tasks and steps that are carried out in real life are analyzed. From this analysis, the actual tasks that need to be performed by the different agents within the system can be determined and specified. The decision making and physical processes involved have to be understood before the structure and functionality of the agents can be specified. As such, the workflow process for gathering and processing code data are generated from an analysis of the processes involved at these two stages. It is essential to separate these processes knowledge harvesting; knowledge sharing even though the latter is dependent upon the former. This is because they take place at different locations, this process involve different users knowledge workers / software developers and knowledge champions / subject experts in the organizations. We analyze and define processes for the collection and processing of code data/pattern respectively.

5.2 Creation and Utilization of Knowledge Base

Agent system combined with machine learning techniques is used for knowledge base generation. Agent systems are ideally suited for information processing tasks and have been employed in many different information processing applications (Alty et al., 1994; Etzoni and Weld, 1994; Maes, 1994; Sheth, 1994; Jennings et al., 1996; Davies et al., 1997; O'Brien and Wiegand, 1997; Papazoglou and van den Heuvel, 2000; Leiberman, 1999; Pham et al., 2001). Machine learning techniques re applied to the generation of pattern sets for use in knowledge based systems. The proposed agent based system performs both information gathering and processing and user assistance. It could be perceived as an interface agent. As the intention is also to provide a generic framework to allow any learning algorithm to be employed, agent based system that encompasses both Creation and utilization of knowledge base increase the potential applicability of the proposed architecture. Software developers write different code sets and they store it in knowledge base. These code sets are assessed and evaluated by subject experts. Solution base is created with solution patterns. Solution patterns could be utilized by the developers by simple search keyword. Here an experimental prototype is been created to satisfy the need of software developers. The details are shown below.

5.3 Service Framework for Knowledge Acquisition and Management

The proposed service framework is used for acquiring and managing code data / patterns. Our Proactive Semantic service Assistant has two agents, the service agent (SA) and user interface agent (UIA). These agents perform two separate but interdependent roles. It can be envisaged that UIA operates within the end user's computer and interacts with end user. A "form" in UIA collects and stores codes as patterns in knowledge base, Knowledge champions and subject experts' asses and evaluates this data and incorporates it as solution patterns into the solution knowledge base. Solution knowledge base sends code data / solution patterns to SA. SA supplies solution patterns to software developers in the middle of the software development activity. The system is structured such that it allows SA manages the solution knowledge base for different product variants, in order to facilitate the retrieval of code data. In the implementation of this proposed application,

there is a requirement for communication between assistants.

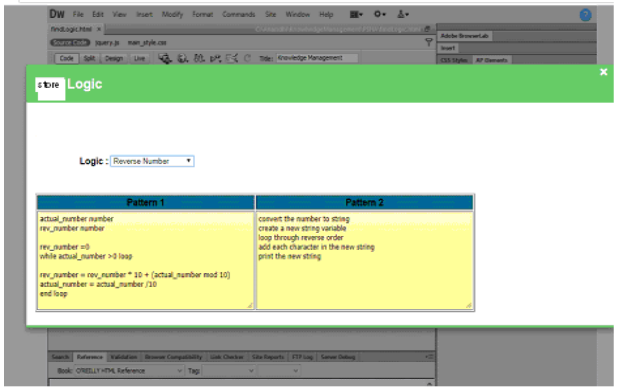

Fig. 3: UIA collects and stores solution patterns

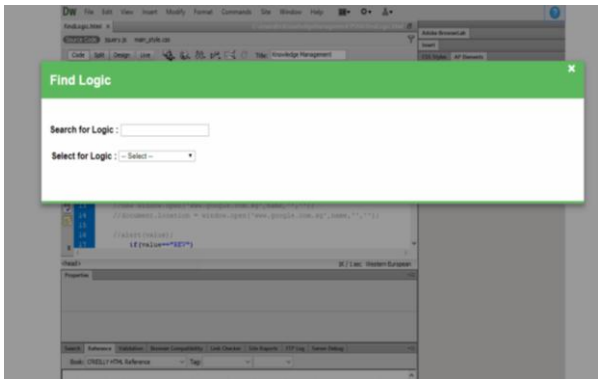

Fig. 4: UIA search screen

This communication is envisaged to take place via knowledge mapping techniques. The system as a whole is not purely agent based but can be considered to be a hybrid of an agent-based system and a BB system in a manner similar to Lander et al. (1990), Hayes-Roth et al. (1994) and Holt and Rodd (1994). This is because there are certain BB features and concepts are particularly appropriate to this system for knowledge mapping. The feature of BB systems that is of particular relevance is the globally accessible data structure known as a BB as mentioned in Englemore and Morgan (1988) and Craig (1995). The utilization of concepts from BB systems manifests itself in the BB server that is used as a channel or medium for all communications. The BB server performs knowledge-message management and thus frees Agents to concentrate upon performing information processing related tasks.

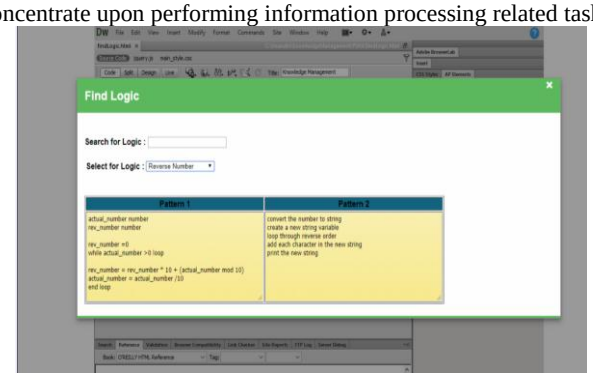

Fig. 5: SA supplies solution patterns to UIA

The purpose of User Interface Agent is to facilitate the acquisition of code data from knowledge workers. This is achieved by UIA by providing a user interface. Interface is used for the entry and submission of code to the knowledge base. And another functionality of UIA is retrieving code from the service agent which is stored in knowledge base previously. Users can retrieve the semantic patterns using keywords. UIA gets keyword from user, submits query to the service agent. SA acts as a query processor thus gives semantic patterns for the requirement. Our Collaborative platform provides km solutions from solution knowledge base.

6. Conclusion

KM Trajectory Schema – Service Frame work is helpful to reduce the dependency on human resources in software development organizations. Knowledge Base Administrator, Knowledge Champions and KM team members has to contribute their time in creation and maintenance of solution knowledge base along with their regular work. Organizations could treat KM Service Frame work as a resource asset that supports the pressuring need of software developers. Functional KM solutions for software process improvement positively improve the business process in the software industry. This work is the basement to construct the architecture for the km implementation. Our service framework's implementation will satisfy the developer's essential needs by using extraction algorithms in order to improve the developer's performance. KM Trajectory Schema – Service Frame work will be helpful in the software process improvement which in turn will increase the business of software organizations.

References

[1] Ali bahrami, "Object oriented systems design", Addison-Wesley pp 77-79,249 2002
[2] Aksoy MS, "New algorithms for machine learning," in PhD. thesis, Cardiff University, UK, 1993.
[3] Clark P, Boswell R, "Rule induction with CN2: some recent improvement" in Proceedings of the 5th European conference on machine learning EWSL-91, 151–163.1991
[4] Craig I, "Blackboard systems", Ablex Publishing Corporation, Norwood,1995.
[5] Englemore RS, Morgan AJ, "Blackboard systems",Addison-Wesley, Wokingham,1988.
[6] Fisher M, "Representing and executing agent-based systems" in Wooldridge M, Jennings NR. (eds) Lect. Notes in Artif. Intell., 890:203-218,1995.
[7] Forsyth R, "Machine Learning: Principles and Techniques" Chapman and Hall, London,1996.
[8] Franklin S, Graesser L, "Is it an agent, or just a program?: A taxonomy for autonomous agents" in Muller JP, Wooldridge MJ, Jennings NR, (eds) Lect. Notes in Artif. Intell., 1193: 21–35,1996.
[9] Genesereth MR, Nilsson NJ, "Logical foundations of artificial intelligence" Morgan Kauf-mann, San Mateo,1988.
[10] Genesereth MR, Ketchpel SP, " Software agents" Communications of the ACM 37(7):48–53,1994.
[11] Hunt EB, Marin J, Stone PJ, "Experiments in induction" Academic press, New York, USA,1966.
[12] Ito A, Yano H, "Can machines ever learn to cooperate? The emergence of cooperation in asociety of autonomous agents" J. Comm. Res. Lab., 42(3):243–249,1995.
[13] Jennings NR, "Controlling cooperative problem solving in industrial multi-agent systems using joint intentions" Artif. Intell., 75:195–240,1995.
[14] Jennings NR, Mamdani EH, Laresgoiti I et al. "GRATE: a general framework for co-operative problem solving", IEE-BCS J. Intell. Sys. Eng., 1(2):102–114,1992.
[15] Knight K, "Are many reactive agents better than a few deliberative ones? " in Proceedings of the 13th international joint conference on artificial intelligence (IJCAI'93), Chambery, France, 432–437, 1993.
[16] Kowalski R, Sadri R,"Towards a unified agent architecture that combines rationality with reactivity" in Proceedings of the international workshop logic in databases LID, San Miniato, 137–149,1996.
[17] Kumar D, Shapiro SC, "Architecture of an intelligent agent in SNePS", ACM SIGART Bulletin, 2(4):89–92,1991.
[18] Labrou Y, Finin T, "A proposal for a new KQML specification" The University of Maryland, Baltimore County report TR CS-97–03, USA,1997.
[19] Lashkari Y, Metral M, Maes P, "Collaborative interface agents" in Proceedings of the 12th national conference on artificial intelligence AAAI'94, Seattle, USA, 1:444–449,1994.
[20] Maes P, "Modelling adaptive autonomous agents" in Langton CG (ed) Artificial life, MIT Press, Cambridge, USA, 1995.
[21] Mayfield J, Labrou Y, Finin T, " Evaluation of KQML as an agent communication language" IJCAI'95 workshop, Montreal, Canada, 347–360,1996.

[22] Meghini C, "A reactive logical agent" in Proceedings of the 1st international workshop on cooperative information agents CIA'97, Kiel, 148–158, 1997.
[23] Michalski RS, "A theory and methodology of inductive learning" in Michalski RS, Car-bonell JG, Mitchell TM (eds) Machine learning– an artificial intelligence approach, Morgan Kaufmann, Los Altos, 83–134,1983.
[24] Michalski RS, Mozetic I, Hong J, et al "The multi-purpose incremental learning algorithm AQ15 and its testing, application in three medical domains" in Proceedings of the 5th na-tional conference on artificial intelligence, Philadelphia, USA, 1041–1047, 1986.
[25] Muller JP, "The design of intelligent agents: A layered approach", Lect. Notes in Artif. In-tell., 1177,1996.
[26] O'Brien P, Wiegand M, "Agents of change in business process management" in Nwana HS, Azarmi N (eds) Software agents and soft computing, concepts and applications, Springer-Verlag, Berlin, Germany, pp. 132–245, 1997.
[27] Pham DT, Aksoy MS, "An algorithm for automatic rule induction", Artif. Intell. Eng., 8:277–282,1993.
[28] Pham DT, Aksoy MS , "RULES: a simple rule extraction algorithm", Expert Sys. App. 8(1):59–65,1995.
[29] Pham DT, Dimov SS, "An efficient algorithm for automatic knowledge acquisition. Pattern Recognition" in Proceedings of the Institution of Mechanical Engineers PartA : J. Eng. Manuf., 30(7):1137–1143,1997.
[30] Pham DT, Dimov SS, "An algorithm for incremental inductive learning" in Proceedings of the Institution of Mechanical Engineers Part B: J. Eng. Manuf., 211:239–249, 1997.
[31] Pham DT, Dimov SS, Soroka AJ, "Knowledge acquisition for intelligent product manuals" in Proceedings of the Institution of Mechanical Engineers Part B: J. Eng. Manuf., 215:97– 103, 2001.
[32] Pham DT, Soroka AJ, "ISLE – A novel immune-system inspired rule extraction algorithm" in Proceedings of the 16th IFAC World Congress, Prague, 2005.
[33] Pham DT, Soroka AJ, "RULES_IS immune-network inspired machine learning algorithm" in Proceedings of the 1st I*PROMS virtual conference on intelligent production machines and systems, 2005.
[34] Quinlan JR, "Induction of decision trees", Machine learning Vol. 1. Kluwer Academic Pub-lishers, Boston, USA, 81–106, 1986.
[35] Quinlan JR, "Generating production rules from decision trees" in Proceedings of the 10th international joint conference on artificial intelligence, Milan, Italy, pp. 304–307 ,1987.
[36] Sycara KP, "In context information management through adaptive collaboration of intelli-gent agents" in Klusch M (ed) Intelligent information agents: agent-based information dis-covery and management on the internet, Springer, Berlin, Germany, pp. 78–99,1999.
[37] Utgoff PE , "ID5: an incremental ID3" in Proceedings of the 5th international conference on machine learning, Ann Arbor, Michigan, USA, pp. 107–120, 1988.
[38] Voorhees EM, "Software agents for information retrieval" in Etzioni O (ed) Software, agents – spring symposium, AAAI Press, pp. 126–129, 1994.
[39] Wooldridge MJ, Jennings NR, "Agent theories, architectures and languages: a survey" in Wooldridge MJ, Jennings N. (eds) Lect. Notes in Artif. Intell., 890:1–39, 1995.
[40] Zlotkin G, Rosenschein JS, "Negotiation and task sharing amongst autonomous agents in cooperative domains" in Proceedings of the 11th international joint conference on artificial intelligence IJCAI'89, Detroit, USA, 2:912–917, 1989.