Extracting and Maintaining Project Knowledge Using Ontologies

Panos Fitsilis[1], Vassilios Gerogiannis[1] and Achilles Kameas[2]

[1] Project Management Dept., Technological Education Institute of Larissa,
41110 Larissa, Hellas.

[2] Hellenic Open University, 23 Sahtouri str., 26222 Patras, Hellas

Abstract. One of the most valuable resources for organizations today is knowledge developed and held within their teams during the execution of their projects. Although the need for maximal reuse of lessons learned and knowledge accumulated for keeping companies at the leading edge is evident, this knowledge is often lost because it can be difficult or impossible to articulate. k.PrOnto[1] framework infuses the process of project management with knowledge management technology and provides project managers with concepts and tools to support them in decision making and project control. The tools operate at a stand-alone mode, but, in the context of k.PrOnto architecture, can also be used as components of a distributed system operating at a higher organizational level. Thus the k.PrOnto framework assists large organizations in identifying best practices, metrics and guidelines, starting from individual projects and in amplifying their efforts to achieve organizational maturity and build corporate culture and memory.

1 Introduction

In a competitive and fast changing business environment, an organization's ability to efficiently align resources and business activities with strategic objectives can mean the difference between succeeding and just surviving. For achieving strategic alignment, organizations are increasingly managing their activities as projects —in essence, becoming project-based organizations— in an attempt to monitor performance more closely and make better business decisions about their overall work portfolio. By planning and tracking projects with clarity and precision, organizations can respond with greater agility to the demands of a fast-changing business environment.

Contemporary project management science has become a multi-disciplinary research field influenced by disciplines as diverse as psychology, pedagogy, business administration, organization theory, industrial engineering and sociology [22]. The

[1] k.PrOnto framework has been partially funded by the project MISSION-SPM which is co-funded by the European Social Fund and Hellenic national resources (EPEAEK II/ Archimedes II programme).

Fitsilis P., Gerogiannis V. and Kameas A. (2006).
Extracting and Maintaining Project Knowledge Using Ontologies.
In *Proceedings of the 1st International Workshop on Technologies for Collaborative Business Process Management*, pages 72-83

discipline of project management has evolved because the more traditional, well-established industrial age principles and methods for managing our classical functional organizations (involving on-going, repetitive operations of various kinds) do not work well for planning, controlling, and managing projects, programs, or project portfolios. Projects are comprised of diverse tasks that require diverse specialist skills, and hence cut across the traditional functional organizational lines. They are temporary endeavors with a finite lifetime and so do not provide stable organizational homes for the people involved.

Thus, one of the aims of project management is to capture the knowledge developed before, during and after projects, as, after the project ends, it is usually kept in the minds of the project teams or hidden in the project deliverables. At the same time, especially in multi-partner projects, one has to deal with different background and culture, incompatible procedures and distributed experience, which must be combined with the need to protect each organization's procedures and project knowledge.

k.PrOnto (extracting **k**nowledge using **Pr**oject **Onto**logies) is a framework that applies recent developments in knowledge engineering, software engineering, on existing project management processes in order to support organizational networking and process integration and thus assist organizations in improving adaptability and responsiveness to rapidly changing market demands and customer requirements.

The k.PrOnto framework infuses the project management process with knowledge management technology and provides project managers with tools to support them in decision making process and project control. These tools are deployed as components within a web-based component framework, which enables them to operate at a stand-alone mode (i.e. a project "dashboard"), or as components of a networked system operating at a higher organizational level. Thus k.PrOnto framework can assist large organizations in identifying best practices, metrics and guidelines, starting from individual projects and in amplifying their efforts to achieve organizational maturity, to build corporate culture and memory and to establish secure and trusted knowledge-based collaboration practices with peer organizations, achieving in the end real-time governance.

k.PrOnto provides the infrastructure, so that a project organization can gain visibility, insight, and control its portfolio of projects, whether they are executed on a single site, or on multiple distributed sites, either by a single organization or by a business network. k.PrOnto allows organizations to improve productivity, reduce cycle times, decrease costs, and increase quality, because it enables them to:

- Select projects and programs that are aligned with the organization's strategies and objectives.
- Make the best use of available resources by applying to the highest priority projects.
- Regularly assess how projects and programs are contributing to project portfolio health.
- Take management action to keep the portfolio in compliance with business objectives.

In brief, k.PrOnto toolkit supports:

- Project managers in the decision making process in real time.
- Individual groups in project knowledge exchange across the same organization.

- Organizations in the identification of their best practices and metrics targeting to effectiveness and performance improvement.
- Business inter-networking based on project knowledge exchange.

Summarizing, the main objective of k.PrOnto is to provide a project management *specific* knowledge management framework that can be used primarily to implement organization's project memory and as collaboration tool among different projects and secondarily as a training tool for complex project cases and scenarios.

In the following sections we outline the project management knowledge areas and practices, project processes and project information representation, and we present how k.PrOnto framework is addressing project knowledge management. Finally we present k.PrOnto high level architecture and we are closing by presenting the conclusions.

2 Background

The practice of project management has evolved over half a century and permeates all industries, institutions and governments throughout the world. In response to the perceived need to organize thinking about project management a number of frameworks have been produced. Two kinds of frameworks are broadly identifiable, both of which have sought to model the subject area by presenting only what is "generally" agreed. These are:

- *Life-cycle or maturity models*. Common examples include the ISO series (especially BS ISO10006:2003 [5]), "Project Excellence Model" by the Association of UK Project Managers [30], "Project Management Maturity Model" [29], the Japanese designed P2M modal and "Projects In Controlled Environments 2 (PRINCE 2)" [25], the family of Software Engineering Institute "Capability Maturity Models" in general [6], etc.
- *Bodies of Knowledge*. They provide the standards against which would-be project managers aspire and form the basis for training courses from which such managers may become certified. More fundamentally, they also provide a knowledge framework for understanding the elements that comprise project management. In these areas we have APM Body of Knowledge, PMI Guide to the Project Management Body of Knowledge [31], BSI BS6079 Guide to Project Management [4], Japanese Project Management Body of Knowledge. At the same time there are numerous project management methodologies for software development and construction.

Project processes represent knowledge about software development activities. Capturing, storing and using process knowledge of an organization has to deal with several typical characteristics of software development. The fact that we are referring to software development projects does not limit k. Pronto framework since it could support other domains such as construction projects provided that relevant ontologies have been developed.

Processes are inherently nondeterministic, concurrent and distributed. The non-determinism of processes results from the fact that the sequence of development steps cannot be predicted in advance. Reasons are the existence of many creative development steps (e.g. design steps), possible choices among different alternative paths for plan execution, and product changes triggered from inside or outside the development organization. Concurrency and distribution of processes result from interacting development activities that can be performed in parallel. Especially, outsourcing of development activities and the pressure to incorporate distributed agents enforces the distribution of tasks to different partners-contractors.

Within process models, metamodels are useful for specifying the concepts, rules and relationships used to define a family of related methodologies. Although it is possible to describe a methodology without an explicit metamodel, formalizing the underpinning ideas of the methodology in question is valuable when checking its consistency or when planning extensions or modifications. A good metamodel must address all of the different aspects of methodologies, i.e. the process to follow and the work products to be generated.

In turn, specifying the work products that must be developed implies defining the basic modeling building blocks from which they are built [18].

In the software market sector, a number of metamodels have been constructed to both underpin and formalize methodologies. Examples are: the Object Management Group's Software Process Engineering Metamodel (SPEM) [27], the OPEN Process Framework (OPF) [13], the OOSPICE (developed by a European Commission funded project) metamodel for capability assessment [15] and the LiveNet [17] approach for Computer Supported Collaborative Work (CSCW) are the most prominent.

The SPEM was created by the Object Management Group as a de facto, high-level standard for processes used in object-oriented software development. Initially, it was created as a stand-alone metamodel but later it was reformulated to be a UML [26] (Unified Modeling Language) Profile. This means that the authors recast the process-oriented concepts into model-oriented concepts.

There have been efforts to produce a standard project representation data model without significant success. Three are the main reasons for this failure: unwillingness of project management tools providers to collaborate for commercial reasons; lack of strong technical project management community to push for the development of the standard; lack of consensus at the level conceptual project modeling.

Further, there have been various attempts from academic institutions [8, 20, 34] to produce XML Document Type Definitions (DTD) for project modeling.

As result, a project management XML specification [32] has been proposed by Pacific Edge Software Inc. in 2000 under the auspices of the PMXML consortium. Primavera Systems, Welcom, eProject.com, Great Plains, PlanView, NASA, Oracle and others joined the consortium, which maintains the PMXML standard. Using PMXML standardized schema each compatible application can exchange data with each other and interpret the data accurately. Currently the standard consists of data definitions and specifically the four major project management data types (project, resource, task and assignment) and a few minor ones.

The definition starts with a ProjectManagementSchema and it contains: *InstanceData*, a collection of user and application specific data; *PoolResources*, a collection of resource definitions; *Projects*, a collection of project definitions.

From an outsiders point of view the PMXML standard currently seems to be a pure data definition. Messaging or relations to other standards, especially to the broader integration standards, like ebXML, that could help to embed the project management activities into the broader business processes between organizations, are not publicly available.

At the same time tool vendors are developing interfaces to XML. Microsoft with Project Server 2003 is the clear market leader in the market of project management tools. Similarly with PMXML, Microsoft Project Server 2003 defines data types for projects, WBSs, Calendars, tasks, resources and assignments [23].

So far projects have been regarded as scheduling problems from an IT point of view: there is a broad range of standard software packages (project management tools) available on the market supporting various network analysis techniques such as Program Evaluation and Review Technique (PERT) or Critical Path Method (CPM). Project management tools like Microsoft Project, Primavera, SuperProject, Artemis or even larger integrated ERP systems like SAP R/3 or ORACLE Business Suite etc. are not being optimized from the knowledge point of view.

Project knowledge can be classified as either explicit or tacit. Generally speaking:

- *Explicit knowledge* is that which has been codified and expressed in formal language; it can be represented, stored, shared and effectively applied [24]. Explicit information is the information that enables or facilitates the execution of particular information, including contracting, drawing, solving problems or approving proposals.
- *Tacit knowledge* is personal, rooted in action with commitment and involvement in specific context. It consists of paradigms, viewpoints, beliefs and concrete skills. Consequently, it is difficult to model and cannot be documented in formal language.

The distinction between these two types of knowledge is important because each must be managed in a different way. This implies that the problems for acquiring and using tacit project knowledge are different from those faced in managing explicit knowledge. For example in the case of reusing tacit project knowledge the main problems are related with knowledge, experience and know how loss while in the case of explicit knowledge the problem areas include project knowledge representation, incomplete information etc.

Therefore, in order to achieve project knowledge management and knowledge reuse, several enabling activities could be considered. By collecting explicit knowledge and tacit knowledge, a knowledge management system can store information and knowledge about these activities. The use of associated information/knowledge makes the activity-based knowledge management system [34] substantially different from traditional project scheduling systems. Consequently, each activity in the activity-based knowledge management system involves two types of information, which correspond to explicit knowledge and tacit knowledge. Tacit knowledge records the forms of resources and information as well as statements of experience and domain knowledge.

Further, project knowledge can be classified according to [9] in:

- *Knowledge about projects*, which concerns methodological knowledge on how to manage projects. Usually methodological knowledge is related with project processes, methods, templates, skills etc.
- *Knowledge in projects*, which is knowledge that members of project team acquire during the execution of the project. This type of knowledge includes informal information that is exchanged through e-mail, meetings, personal discussions etc or it is the outcome of the project itself, the project deliverables and documentation.

- *Knowledge from projects*, which has been generated in projects that have already finished. During the entire project lifecycle, efforts have been made by the project team for solving problems. These experiences should flow into a company's organizational knowledge base in order to provide input for future projects. Although, project experiences is regularly requested in the sense of final project reports, literature and experience shows that this is done incompletely and superficially.

Project experiences produced by post project reviews, post project appraisals, after action reviews, project postmortem review, debriefings, reuse planning, experience factory, post implementation-installation evaluations constitute a significant asset for every knowledge organization and therefore their management attracted a lot research attention the last years [2, 3, 7, 19, 35].

A number of different projects and works have addresses similar or partially the same problem areas addressed by k.PrOnto. Among them, Caramba system implements a Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams [11]; TeamLog system implements knowledge mining of ad-hoc processes [12]; FRODO project studied the methods and tools for building and maintaining distributed organizational memories in an enterprise environment (www.dfki.de/frodo); MILOS system supports dynamic coordination of distributed software development teams by integrating project planning and workflow technologies over internet [21], etc.

3 k.PrOnto Framework

k.PrOnto an ontology-based approach is used for knowledge management. Specifically, we use ontologies to represent project knowledge and semantics. An ontology is a formal specification of a shared conceptualization, as defined by Gruber [16]. Ontologies allow the specification of concepts with attributes of a specific type. Concepts can be organized in a hierarchy (using the specialization relationship between two concepts). General information regarding ontological engineering foundations and a survey of most well-known ontologies can be found in [14]. An illustration of the relationship between ontological engineering and other disciplines (software engineering and object oriented software development, in particular) is given in [10].

Considering the large number of ontologies developed, ranging from generic and core ontologies to domain and application specific ontologies, and the lack of standardization, an evolution of methodologies and supportive tools for "ontology engineering" is expected. In k.PrOnto, we use standardized ontology languages [28] and development tools, Protégé ontology development tool [33].

In order to address the full spectrum of knowledge in project management, tacit and explicit knowledge, knowledge about projects, knowledge in projects and knowledge from projects a number of different complementary ontologies has to be developed. In k.PrOnto, for each subject domain there are three sub-ontologies covering three distinct project knowledge areas; content, experience and process [9]. More specifically:

- *Experience ontology*: The experience ontology describes the skills and qualifications required for performing specific task types. Example skills could be "code reading" and "Java programming". Further it describes the know-how that is produced by each project.
- *Process ontology*: The process ontology allows defining a hierarchical process type structure and alternative process decompositions. For example, it is possible to state that "white box testing" is a subtype of "testing". In addition, it is possible to annotate each process type with required skills and information from the project ontology.
- *Project content ontology*: The project ontology allows representing information about the project context. Examples are: "Size of the project in person years = 9", "average skill level of employees = experienced", "application domain = real time communication systems", or "Goal for uptime = 99.999%".

Knowledge management life cycle includes building knowledge, organizing and holding, distributing and pooling, and applying knowledge to work object. According to k.PrOnto approach, project knowledge management is activity centric. This implies that knowledge is associated with specific project activities. Knowledge and information associated with activities in previous projects may be reused and applied in future projects. Information and domain knowledge from all projects are divided and saved as "activity" units in categories related to the projects for collection and management. The main advantage of activity based knowledge management is the ease with which the information and knowledge can be understood and reapplied. An overview and conceptual framework of activity-based knowledge management used in k.PrOnto is presented in figure 1.

According to k.PrOnto approach, knowledge lifecycle consists of the following steps:

- *Knowledge Acquisition*: Knowledge acquisition is the collection of related data and information, concerning of a typical project.
- *Knowledge Extraction*: Knowledge extraction is the process of translating data and information into knowledge.
- *Knowledge Storage*: Knowledge has been stored under a centralized and safe environment.
- *Knowledge Sharing*: Knowledge sharing enables the engineer to share the valuable knowledge and information which has stored in the system by using the internet or intranet.
- *Knowledge Update*: The feedback from various users which has put back to the knowledge management system and updates the knowledge base for reuse.

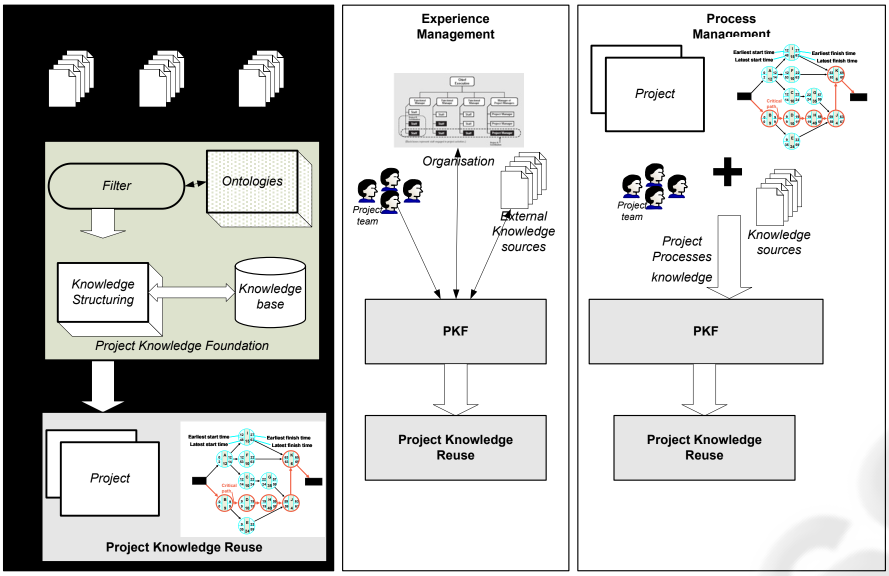

Fig. 1. k.PrOnto knowledge management framework.

4 k.PrOnto Architecture

Component based software is designed with standard, clearly defined interfaces which tend to hide changes in the software environment outside its boundaries. Component based applications are composed, i.e. assembled, at run-time from components selected from a component pool. The fact that components communicate only through well-defined interfaces means that when an application needs to be modified, a limited number of components either need to be modified or replaced by others; all this without fear of disturbing the other components making up the application.

In effect, the component based programming model is the evolution from the monolithic application development to applications built from increasingly more modularized pieces with the addition of the framework. The framework is the glue that binds components together. It is used to compose (combine) separate components from a component pool into a running application. It allows components to be linked together and to make calls on specific component interfaces. Additionally, the framework can provide information about the run-time environment [1].

k.PrOnto consists of a number of concrete architectural layers implementing an open architecture. These layers are presented in figure 2:

- Data Exchange Layer (DEL). It is a collection of APIs and filters that extracts project data from existing project management tools. It is also responsible for formatting data to be exchanged with various Ambient Intelligence devices (i.e. PDAs, mobile phones etc) according to their profiles. DEL implements a number of XML translators, translating COTS produced XML data to k.PrOnto standard project XML DTD.

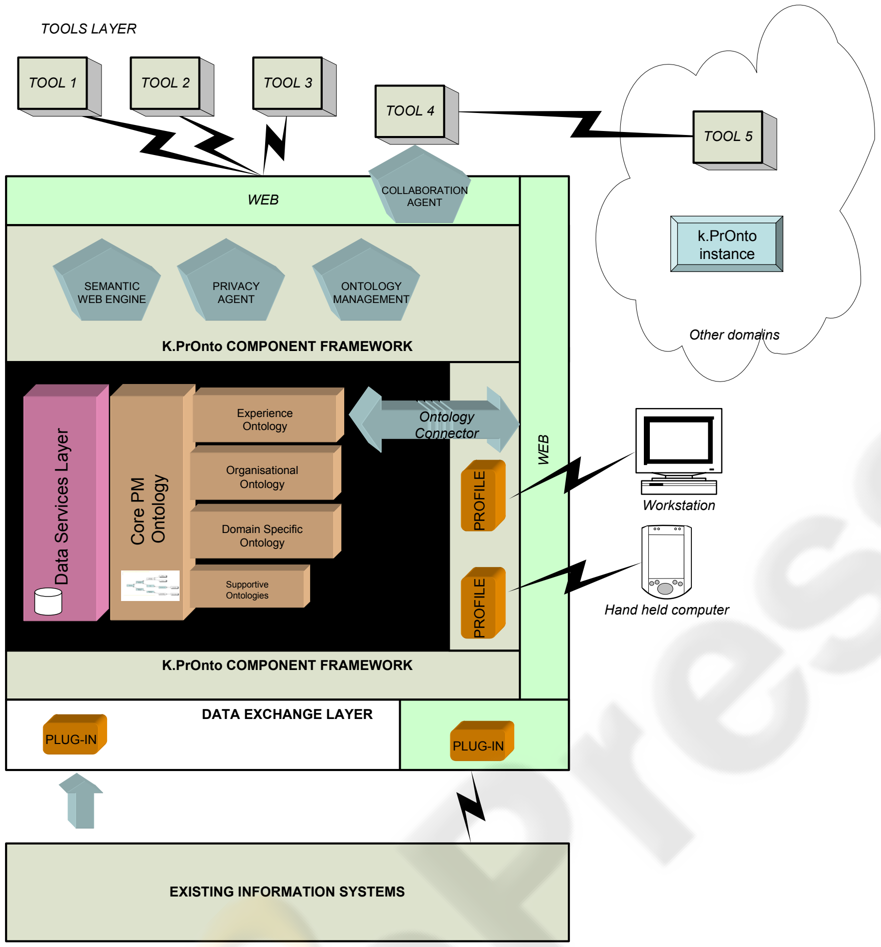

Fig. 2. k.PrOnto layered architecture and components.

- Project Knowledge Framework (PKF), which includes:
  - Project Management XML DTD data exchange interfaces.
  - Project Management ontologies including: core ontology, domain specific ontologies, organizational ontologies and experience ontologies
  - Knowledge mining mechanisms for extracting knowledge from past projects
- Project Management Layer (PML) or k.PrOnto tools layer. It is an application layer and consists of the k.PrOnto project management toolkit. The k.PrOnto tools are:
  - Project Management Dashboard: Presents the current status of the project based on previous project knowledge. In order to identify if a project is in a critical stage we need to define first what is critical.
  - Project Management Advisor allows project manager to built hypothetical scenarios and based on previous project knowledge and current project status it simulates and presents the project alternatives. Based on the simulation results it

  proposes best alternative scenarios to the project manager in order to take final decision.
  - Project Management Browser is used for browsing existing knowledge. Further it can be used to see knowledge that has been captured automatically and to validate the automatic extracted rules, guidelines etc.
  - Project Management Collaborator is used to enable and monitor collaboration sessions among networked businesses.
- Component Framework: It is a web-based layer that uses server-side logic to implement various service components. These include the semantic web engine, the ontology manager and the privacy enforcement component. It also provides collaboration services enabling networked business to exchange project data and knowledge using specific proprietary agents, which make use of the knowledge and policies stores in the local ontologies and the reference definitions included in the core ontology in order to negotiate transactions with peer agents who belong to other organizations.

5 Conclusions

k.PrOnto system explores the process of project management by assisting large distributed project organizations in their efforts to capitalize on the acquired project knowledge.

At the same time, k.PrOnto advances state of the art in the area of real-time project governance especially in the area of virtual project organizations. This research area attracts increasing attentions since more and more organizations are subscribing to this model.

Of the many factors involved in project management, three are of paramount importance: a) exponential growth of knowledge b) growing demand for complex and customized products and services c) the evolution of worldwide markets.

In addition, the introduction of internet technologies brings new challenges in the field as virtual organizations assemble and disassemble on an opportunistic manner, while real-time governance of dynamic organizational structures is required.

k.PrOnto addresses these challenges using a holistic approach based on automatic knowledge acquisition. Distributed project information is attained, processed using domain specific project ontologies and stored in a knowledge base in order to be used in project toolkits implementing real-time governance. It enables enterprises to capitalize on their expertise and to manage new projects as simple new instances in a continuous enterprise project timeline. More specifically, k.PrOnto enables organizations to:

- continuously evaluate project status and quickly identify at-risk and underperforming projects using roll-up scorecard reports that graphically display key project metrics
- gain insight into the performance of the project since results are directly compared with similar projects. Thus, trends and problem areas can be easily identified
- enable the exchange of knowledge and data between networked business/virtual organizations either in the form of good practices or in the form of quantitative data.

- continuously identify valuable resources within the organization or in high market demand and help project managers in optimizing their utilization or performance.

References

1. Bachman, F., Bass, L., Buhman, C., Comella-Dorda, S., Long, F., Robert, F., Seacord, R., & Wallnau, K.: Technical Concepts of Component-Based Software Engineering, Carnegie Mellon University (2000)
2. Basili, V.R., Caldiera, G., & Rombach, H.D.: Experience Factory, Encyclopedia of Software Engineering, Vol. 1. Wiley, New York (1994)
3. Bergmann R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications. Springer, Germany (2002)
4. BS 6079-1:2002 Project management.: Guide to project management, British Standards Institution (2002)
5. BS ISO 10006:2003. Quality management systems: Guidelines for quality management in projects, British Standards Institution (2003)
6. Capability Maturity Model Integration: CMMISM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1), Carnegie Mellon University (2002)
7. Collier, B., DeMarco, T., & Fearey. P.: A defined process for project postmortem review. IEEE Software, (1996) 13(4)
8. Curran, K., Flanagan, L. & Callan, M.: PMXML: An XML Vocabulary Intended for the Exchange of Task Planning and Tracking Information. Information Technology Journal, 3(2): (2004) 192-195
9. Damn, D., & Schindler. M.: Security issues of a knowledge medium for distributed project work, International Journal of Project Management, Vol. 20, (2002)
10. Devedžic, V.: Understanding ontological engineering. Communication of the. ACM 45, 4, (2002)
11. Dustar, S.: Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams, Distributed and Parallel Databases, Vol. 15, (2004) 45–66
12. Dustar, S., Hoffmann, T., & van der Aalst, W.: Mining of ad-hoc business processes with TeamLog, Data & Knowledge Engineering, Vol. 55, (2005) 129–158
13. Firesmith, D.G., & Henderson-Sellers, B.: The OPEN Process Framework: An Introduction. Addison-Wesley (2002)
14. Gomez-Perez, A.: Ontological engineering: A state of the art. Expert Update, 2, (1999) 33-43
15. Gonzalez-Perez, C., McBride, T., & Henderson-Sellers, B.: A metamodel for assessable software development methodologies, Software Quality Journal (2004)
16. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Int. Journal of Human-Computer Studies, Vol. 43, (1995) 907-928
17. Hawryszkiewycz, I.T.: Knowledge networks in administrative system, Proceedings of the Working Conference on Advances in Electronic Government, Zaragoza, Spain, (2000) 59-76
18. Henderson-Sellers, B., & Gonzalez-Perez C.: A comparison of four process metamodels and the creation of a new generic standard. Information and Software Technology, 47, (2005) 49-65
19. Kerth, N.: The ritual of retrospectives—how to maximize group learning by understanding past projects. Software Testing & Quality Engineering, Vol.2(5), (2000)

20. Liu, D.R., & Hsu, C.: Project-based knowledge maps: combining project mining and XML-enabled topic maps, Internet research, Vol. 14(3), (2004) 254-266
21. Maurer, F., Dellen, B., Bendeck, F., Goldmann, S., Holz, H., Kötting, B., & Schaaf, M.: Merging Project Planning and Web-Enabled Dynamic Workflow Technologies. IEEE Internet Computing, May-June (2000)
22. Meredith, J., & Mantel, S.: Project Management: A managerial approach. Wiley, 4th edition (2000)
23. Microsoft Office 2003 Edition XML Schema References (2003). Available at http://msdn.microsoft.com
24. Nonaka, I., & Takeuchi H.: The Knowledge-Creating Company. Oxford University Press, Oxford (1995)
25. Managing Successful Projects with PRINCE2, Office of Government Commerce, London: The Stationary Office (2005)
26. Unified Modeling Language Specification, Version 2.0, Object Management Group document formal/05-07-04 (2004). Available at: http://www.omg.org.
27. Software Process Engineering Metamodel Specification, Version 1.1, Object Management Group, formal/05-01-06 (2005). Available at: http://www.omg.org.
28. OWL Web Ontology Language: W3C recommendation 10 February 2004 (2004). Available at http://www.w3.org/TR/2004/REC-owl-features-20040210/
29. Programme Management Maturity Model, The Programme Management Web Site. Available at http://www.e-programme.com/
30. Project Excellence Model, International Project Management Association. Available at: http://www.apm.org.uk/ProjectExcellence.asp
31. A guide to the Project Management Body of Knowledge, Project Management Institute Standard Committee (2003).
32. Project Management XML Schema: Cover Page, Hosted by OASIS, On-line resources for markup languages techniques (2000). Available at http://xml.coverpages.org/projectManageSchema.html
33. Protégé ontology editor: Stanford University Medical Informatics (2005). Available at http://protege.stanford.edu/
34. Tserng, H., & Lin Y.C.: Developing an electronic acquisition model for project scheduling using XML-based information standard. Automation in Construction, 12, (2003) 67-95.
35. Williams, T., Eden, C., Ackermann, F., & Howick, S.: The use of project post-mortems. Annual Symposium, Nashville, Tennessee USA. Project Management Institute (2001).