Journal of Software Engineering and Applications, 2014, 7, 1096-1110
Published Online December 2014 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2014.713097

Ontologies for Software Project Management: A Review

Panos Fitsilis, Vassilis Gerogiannis, Leonidas Anthopoulos

Business Administration Department, Technological Educational Institute of Thessaly, Larissa, Greece
Email: fitsilis@teilar.gr

Received 10 October 2014; revised 8 November 2014; accepted 4 December 2014

Academic Editor: Yashwant K. Malaiya, Colorado State University, USA

Abstract

Software Project Management is a knowledge intensive process that can benefit substantially from ontology development and ontology engineering. Ontology development could facilitate or improve substantially the software development process through the improvement of knowledge management, the increase of software and artefacts reusability, and the establishment of internal consistency within project management processes of various phases of software life cycle. A large number of ontologies have been developed attempting to address various software engineering aspects, such as requirements engineering, components reuse, domain modelling, etc. In this paper, we present a systematic literature review focusing on software project management ontologies. The literature review, among other, has identified lack of standardization in terminology and concepts, lack of systematic domain modelling and use of ontologies mainly in prototype ontology systems that address rather limited aspects of software project management processes.

Keywords

Software Project Management, Ontologies, Project Management

1. Introduction

Project Management (PM) is widely accepted today as an important management tool in business development and business success. In this context, a large number of PM frameworks, methodologies and approaches have been developed over the past few decades. Among the most popular are the "Project Management Body of Knowledge" (PMBOK) from Project Management Institute [1], the "IPMA Competence Baseline" (ICB) from International Project Management Association [2] and the "Projects IN a Controlled Environment" (PRINCE2)

How to cite this paper: Fitsilis, P., Gerogiannis, V. and Anthopoulos, L. (2014) Ontologies for Software Project Management: A Review. *Journal of Software Engineering and Applications*, **7**, 1096-1110. http://dx.doi.org/10.4236/jsea.2014.713097

from the Office of Government Commerce in UK [3].

Similarly in the area of software engineering, the goal is to find repeatable processes that improve both productivity and quality. For this reason, a large number of software process models have been developed, namely Waterfall, Prototyping, RAD (Rapid Application Development), Incremental, Spiral, UP (Unified Process) XP (Extreme Programming), Scrum, etc. [4] attempting to address the optimal way to develop software based on the specialties of each problem domain and the characteristics of each software development paradigm. Even though software project management attracted significant attention from both industry and academia, a great number of projects still fail to meet their requirements in terms of time delays, cost overrun and quality restrictions. These failures are attributed to the facts that software projects are complex undertakings, relying heavily on human knowledge and human interaction. Many studies on various types of software projects have proven that their outcomes are far from the complete fulfilment of the initial requirements [5] [6].

Srikantaiah *et al*. [7] argued on the need of convergence of project management and knowledge management subject areas since knowledge management is becoming a very important element for project success and organizational performance. In order to achieve project knowledge management and knowledge reuse, several enabling activities and alternative approaches should be considered. One of these approaches is considered to be ontological engineering.

Gruber [8] defined ontology as the formal specification of a shared conceptualization. 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). A general review regarding ontological engineering foundations and a survey of most well-known ontologies can be found in [9]. An illustration of the relationship between ontological engineering and other disciplines (software engineering and object oriented software development, in particular) is given in [10]. The most prominent and standardized ontology languages is OWL 2 [11] and among the development tools, Protégé ontology development tool has been widely used in many cases.

Ontology development in the field of project management and software engineering could benefit substantially the software development process through the improvement of knowledge management process, the increase of software and artefacts reusability, and the establishment of internal consistency within project management processes, to be used for specialized purposes in various phases of the software life cycle.

Considering the large number of ontologies developed, ranging from generic ontologies to domain and application specific ontologies and the lack of taxonomy and standardization, it is quite difficult for the researchers and the practitioners to evaluate and use these ontologies in a successful way and to develop further the concepts, the methodologies and the tools needed for applying the ontological paradigm to real life projects.

This is the problem that this survey is attempting to address: to present in a systematic way the available in the literature ontologies addressing project management, as well as ontologies modelling specific software development lifecycles or addressing specific problem areas of software projects from the software project management perspective.

The remainder of the paper is structured as follows. Section 2 provides an overview of project management methodologies and of ontological engineering and defines the context of this survey. Section 3 presents the research methodology and Section 4 introduces the ontologies and offers a classification of them based on various criteria. The final section presents the conclusions and possible future research directions.

2. Project Management and Ontological Engineering

Traditional software development methodologies grew out of a need to control large development projects, and from the difficulties of estimating and managing these efforts to reliably deliver results. These difficulties are inherited in the nature of software and they were identified from the early years of software system development and unfortunately most of them still remain. Most of the scepticism expressed in the legendary book of Frederic Brooks, "the mythical man-month" thirty years ago is still a reality [12].

Agile methodologies attempt to overcome these obstacles by changing the approach used to develop software and manage projects. Agile software development attempts to put the software being developed first and to acknowledge that the user requirements change, responding quickly to the users' needs, producing frequent and regular, software releases, etc.

The Manifesto for Agile Software Development was released in February 2001 by a group of 17 software process methodologists, who attended a summit meeting to promote a better way of developing software and then formed the Agile Alliance. The Manifesto for Agile Software Development can be found on the Agile Alliance website[1].

Since then, a number of software development methods subscribed to this approach. The list varies depending on different viewpoints and interpretations, but in general the list in the literature includes Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Adaptive Software Development (ASD), Crystal Clear Methodology, etc.

Most agile development methods were created within corporations by software process experts as an attempt to improve existing processes. For example, XP was created by Kent Beck during his work on the Chrysler Comprehensive Compensation System payroll project. Kent Beck refined the development method used and the result was published in his book "Extreme Programming Explained" [13]. Similarly, FDD was initially introduced by Jeff De Luca to meet the specific needs of a 15 month, 50 person software development projects at a large Singapore bank in 1997. FDD was influenced by ideas of Peter Coad on object modelling. The description of FDD was first introduced in the book "Java Modelling in Colour with UML" by Peter Coad, Eric Lefebvre and Jeff De Luca in 1999 [14]. A more generic description of FDD decoupled from Java can be found in the book "A Practical Guide to Feature-Driven Development" [15].

On the other end, more conventional methodologies rely heavier on processes, linear development cycles and waterfall like software development life cycles. Along with predictability, they inherited a deterministic, reductionist approach that relied on task breakdown, and was predicated on stability—stable requirements, analysis and stable design. This rigidity was also marked by a tendency towards slavish process "compliance" as a means of project control [16].

Project Management Body of Knowledge (PMBOK) developed by Project Management Institute is the best representative of this approach [1]. PMBOK formally defines a large set of project processes that describe activities throughout a project's life cycle. These processes are organized into two axes: into five process groups and into nine knowledge areas that will be described briefly in the following section. Within PMBOK each process is described in terms of inputs (documents, plans, design, other data, etc.), outputs (documents, products), tools and techniques (mechanisms that are applied to inputs for producing outputs) and without being too specific provides guidance to someone that wishes to apply the processes [1].

Similar approaches have been introduced by other international bodies or associations such as International Project Management Association (IPMA)[2] and the Association for Project Management in UK (APM)[3]. IPMA defined a Competence Baseline (ICB), which describes the necessary competences for project management. According to IPMA ICB three are the basic capabilities areas: a) *contextual* (project orientation, objective, portfolio; implementation of plans and portfolio; permanent organization; business; system, product and technology; personnel management; health, safety, security and environment; finance and legality); b) *behavioural* (leadership, liability and motivation; self-control; self-confidence; relaxation; openness; creativity; outcome orientation; efficiency; consultation; negotiation; conflicts and crisis; reliability, appreciation values, etiquette); c) *technical* (successful project management; interested parties; project requirements and objectives; risk and opportunity; quality; project organization; teamwork, problem resolution; project structure; scope, area and capability of fulfilment; time and project phases; resources; costs and finance; delivery and contract; changes of management and administration; information and documentation; communication; initialization; termination) [2].

Software engineering community has recognized quite early, ontologies as a promising way to address current software engineering problems [17] [18]. For example, ontologies are proposed to be used in requirement engineering, software design, software maintenance, software reuse and knowledge management etc. Moreover, software engineering technologies have been proposed and developed for modelling and reasoning with the use of ontologies. These synergies between ontologies and software engineering have also attracted attention of standardization bodies and have some on-going activities.

In order to facilitate the work of researchers and practitioners Happel and Seedorf [18] defined a simple classification scheme that allows a better differentiation among the various software engineering ontologies. They use two dimensions for ordering ontologies. Firstly, according to their position in the Software Engineering life cycle (analysis, design, development, testing, etc.) and, secondly, according to their usage.

Ruiz and Jilera [19] in their survey presented the state of the art with regard to use of ontologies in software

[1] http://www.agilemanifesto.org
[2] http://www.ipma.ch
[3] http://www.apm.org.uk

engineering and software technology. They offered a taxonomy for software engineering ontologies and they presented representative cases of such ontologies.

On the same topic Zhao *et al*. [20] classified available ontologies for semantic web based software engineering, and presented the overall picture in this area. According to Zhao, each ontology is characterised by:

- The software process model used (e.g. Waterfall, Incremental, Spiral, Agile, etc.);
- The software development phase (e.g. Requirements Engineering, Design, Implementation, Testing, Maintenance, All);
- The artefact type (e.g. Software Architecture, Code and Document).

In this paper, and according to the taxonomies presented, we utilize Zhao [20] classification to review various software development ontologies presented in the relevant literature, according to the project management approach used, the software process employed, and the software development phase used.

3. Research Methodology

The aim of this paper is to study software project management ontologies in order to reuse/develop an upper software project management ontology that will be used for automated knowledge extraction from social networks. Having this as the main objective it becomes obvious that existing literature had to be reviewed in order to:

- To better understand the fundamental concepts and the knowledge that constitute the software project management knowledge area;
- To examine and if possible to evaluate software project management ontologies;
- To determine the main application areas of the existing software project management ontologies.

The examined area is quite wide since it refers to the intersection of project management, software engineering, software life cycle processes and knowledge management

In order to achieve these objectives, a systematic literature review, was conducted by the authors. This survey relied on surveying and using secondary sources [21].

The study consisted of two steps. The first step involved literature search, while the second step was focused on the taxonomy of the works and their critical analysis.

At the first step of the literature search, the authors attempted to locate and retrieve articles focused on ontologies that are related to:

a) Project management in general;

b) Project management knowledge areas based in PMBOK;

c) Project management methodologies (e.g. ICB IPMA, PRINCE);

d) Software process models;

e) Software life cycle phases.

The study was mainly conducted with the use of Google Scholar[4], since this tool is indexing texts from all major on-line repositories (e.g. Science direct, IEEE, and ACM). The study examined only books, articles in academic journals/conferences and grey literature while we excluded industrial white papers and articles on IT professional websites.

Table 1 presents all the terms that were used for this research along with the number of works that satisfied the research queries.

Subsequently, the retrieved articles and materials were systematically and critically analysed. The inclusion criteria were:

The study should be focused mainly on project management topics;

The presented ontology or system should have a project management perspective;

The study should be focused on software development or software maintenance processes (other studies were excluded).

The set of remaining works is presented in the next sections.

4. Ontologies Focusing on Project Management Process

The ontologies presented in this section are focusing on project management models without focusing software

[4] http://scholar.google.com

Table 1. Search terms.

Search Query in Google Scholar	No. of Results
"Project Management Ontology" AND PMBOK	41
"Project Management Ontology" AND ICB	11
"Project Management Ontology" AND PRINCE	8
"Project Management Ontology" AND Agile	17
"Project Management Ontology" AND Scrum	3
"Project Management Ontology" AND XP	4
"Project Management Ontology" AND FDD	0
"Project Management Ontology" AND CMM	14
"Project Management Ontology" AND SPEM	4
"Project Management Ontology" AND (UP OR RUP)	79
"Project Management Ontology" AND SWEBOK	7
"Project Management Ontology" excluding the words (-PMBOK -ICB -PRINCE -Agile -Scrum -XP -FDD -CMM -SPEM -UP -RUP -SWEBOK)	12
"Project Scope Management" AND Ontology	49
"Project Time Management" AND Ontology	67
"Project Cost Management" AND Ontology	60
"Project Quality Management" AND Ontology	63
"Project Communication Management" AND Ontology	24
"Project Human Resource Management" AND Ontology	36
"Project Risk Management" AND Ontology	367
"Project Procurement Management" AND Ontology	44

projects. A short introduction to the main concepts of each ontology along with possible applications is presented.

PROMONT [22] is a project management ontology developed to model project management specifications. It provides the basis of common vocabulary of terms and methods and thus enables better project management. PROMONT formalizes the typical elements used for project structuring (such as task, milestone, resource or checklist). Further PROMONT defines a set of relationships. The categorization builds on the fact that, just like concepts, relations can be arranged into a "is derived from" hierarchy.

Hughes [23] developed an ontology based on the PRINCE2 project management method [3]. The prototype ontology has been developed in OWL as a collection of axioms and using the Protégé 4 ontology editor. The produced ontology is used in checking the compliance of suppliers stated methods with standards. However, it presents difficulties for domain experts as its reasoning rules can be counter-intuitive.

Sheeba *et al*. [24] presented an ontology to be used for the automatic classification of learning materials according to PMBOK structure of knowledge. This facilitates the search for learning materials within the given domain. The Rational Unified Process (RUP) was used to define axioms describing the relationships between the main concepts.

Bodea *et al*. [25] developed SinPers system which is a web based personalized e-learning system focused on project management. The learning material that was used was structured with the development of an ontology that modelled International Project Management Association Competence Baseline (ICB) [2]. The ontology developed in SinPers contained 200 concepts, 406 competences, 366 learning objects, and various relationships.

Ruiz-Bertol *et al*. [26] used OWL and SWRL languages for defining the PM ontologies and rules, respectively. They presented an example of different types of rules that can be applied on this specific ontology. In this way, they demonstrate how further knowledge can be derived and, thus, decision-making for managing projects can be improved.

Aramo-Immonen [27] explored project management by providing a PM ontology for managers. This ontology is a classification of management disciplines for project managers. The objective of this study is to help system integrators to manage the evolution of projects during their life-cycles in terms of this ontology. The disciplines defined in this ontology are project integration management, project scope management, project time management.

Dong *et al*. [28] proposed an ontology for a real-time project monitoring system (ORPSM). Their objective was to resolve the ambiguity issue in project monitoring processes produced by a number of factors. This ontology is part of a framework that incorporates a series of ontologies for knowledge management and term disambiguation by focusing on project monitoring processes, and a number of metrics for assisting management of project organizations to monitor projects.

Wong *et al*. [29] presented in their paper "Developing PMIS for Business Projects Based on Social Science Research Findings and Ontology Modelling" a model, to be used as a reference, to guide the business project management practice in order to increase chances of project success. Its representation as the Business Project Management (BProjM) ontology and expedites the injection of this new domain knowledge into a software solution. The ontology has been defined in Unified Modelling Language (UML); and provides an example of how the resulting model can be used as the foundation to support the development of an integrated Project Management Information System (PMIS) for business projects.

Further, a number of PM ontologies have been developed focusing on specialized PM subject areas.

For example, Project Metrics Ontology (PMO) was an ontology originally developed by BBN Technologies/Verizon in 2002 and is not meant to cover all typical project management issues. Instead of this, it focuses on providing an ontology that represents metrics for a specific project. This allows for example to perform performance measurements of specific projects or sub-projects[5].

IT-CODE is an ontology that was defined to describe the project team of a building project within the IC-CODE project itself. This ontology focuses on providing typical project management classes such as "Task", "Actor", "Project" or "Activity". It contains 32 different classes and an additional set of 26 properties[6] [30].

The OZONE ontology [31] is an ontology that focuses on task scheduling. It provides a generic perspective for building scheduling systems and was used for configuring constraint-based scheduling systems. In OZONE, scheduling is defined as a process for feasibly synchronizing the use of RESOURCES by ACTIVITIES to satisfy DEMANDS over time.

Similarly, Rajpathak *et al*. [32] developed a generic task ontology again focusing on scheduling. It is referred as a "task ontology" for emphasising that it describes the class of scheduling tasks, independently of the various ways by which these tasks can be solved. Further, it takes into account cost related issues.

On the subject area of human resource management, personnel selection, competence management, etc. are common issues addressed by Human Resource Ontologies. Mochol, Oldakowski and Heese [33] have developed a Human Resource Ontology to support the Recruitment Process. This HR-ontology is mainly based on the German version of the HR-XML standard developed by the HR-XML Consortium[7]. HR-XML is a library of more than 75 interdependent XML schemes which define data components for various HR transactions, as well as options and constraints governing the use of these components. The ontology was adapted so the Industry sub-ontology to use both the German Classification of the Industry Sector (WZ2003) and the North American Industry Classification System (NAICS). For the occupational classification the German Occupation Codes (BKZ) are used along the Standard Occupational Classification (SOC) System.

Dorn *et al*. [34] developed an ontology containing concepts of HRM for two different projects: a meta-search engine for job searching in job portals and for a university competence management system. In their paper, they presented the requirements derived from these two projects and they describe the design of the ontology. This ontology is characterized by its integration of job descriptions, concepts for evaluating competencies on different levels and evidences for competencies. The definition is also aligned to HR-XML for the definition of competence profiles.

[5] http://www.daml.org/ontologies/349
[6] http://www.daml.org/ontologies/306
[7] http://www.hr-xml.org/

Schmidt and Kunzmann [35] [36] describe an ontology that integrates concepts from skill management and learning. Their approach is similar to Dorn *et al*. [34], however, they do not consider job descriptions composed of required competencies. In the same paper a holistic view of the organisation and of the HR problem area is presented showing how the ontology is covering different organizational aspects.

Gómez-Pérez *et al*. developed an ontology for HR based on standards [37]. More specifically, they developed SEEMP ontology as a common "language" as a controlled vocabulary in order to describe the details of a job posting and the CV of a job seeker. SEEMP Ontology is composed of thirteen sub-ontologies: Competence, Compensation, Driving License, Economic Activity, Education, Geography, Job Offer, Job Seeker, Labour Regulatory, Language, Occupation, Skill and Time.

Peng and Nunes [38] systematically analysed research works and proposed a total of 40 ERP post-implementation risks related to diverse operational, analytical, organization-wide and technical aspects. A risk ontology was subsequently established to highlight these ERP risks, as well as to present their potential causal relationships. An overview of the project management ontologies, their focus and their application is presented in **Table 2**.

Table 2. Project management related ontologies.

Authors/Ontology Name	Main Focus	Applications
Abels *et al*. [22]/PROMONT	Project Management	Knowledge management in virtual organizations. Promote common understanding between virtual teams.
Hughes [23]/Prinny	Prince Project Management Methodology	Used for checking the compliance of suppliers stated methods with standards and supporting process model tailoring.
Sheeba *et al*. [24]	PMBOK	Used for automatic classification of learning materials to the Project Management knowledge area domain.
Bodea *et al*. [25]/SinPers system	IPMA ICB	Used within a web based personalized e-learning system focused on project management for the classification of the learning material.
Ruiz-Bertol *et al.* [26]	General approach of Project management	Demonstrated how knowledge can be derived in order to support project management decision process.
Aramo-Immonen [27]	Mainly on PMBOK	Offers a classification of project management concepts.
Dong *et al*. [28]	General approach of Project management	To be used for knowledge capture, storage, sharing and term disambiguation in project monitoring processes.
Wong *et al*. [29]/Business Project Management (BProjM) ontology	General approach of Project management	Reference for development of an integrated Project Management Information System (PMIS).
BBN Technologies-Verizon/ Project Metrics Ontology (PMO)	General approach of Project management	The ontology represents project metrics in order to perform project performance measurements.
Smith and Becker/OZONE [31]	Scheduling problem domain	The OZONE ontology is focusing on scheduling problems and can be apply from manufacturing production scheduling, project planning and it mainly concerned in modelling scheduling problem domain.
Rajpathak *et al*. [32]	Scheduling problem domain	The task ontology can now be used as a knowledge capturing tool in various domains for increasing reusability. The given ontological framework provides a fairly fine-grained structure that is needed to build the scheduling systems.
Mochol and Oldakowski [33]	Human Resource Management/ Recruitment process	To be used for the development of a recruitment platform.
Dorn *et al*. [24]	Human Resource Management	To be used for a meta-search engine for searching for jobs in job portals and for a university competence management system.
Schmidt and Kunzmann [35] [36]	Human Resource Management/ Skill management	The ontology is used for competency management and knowledge management.
Gómez-Pérez *et al*. [37]/SEEMP	Human Resource Management/ Recruitment	SEEMP is a reference ontology to be used for modelling job offers and curriculum data.
Peng and Nunes [38]/Repo	Risk Management for post implementation of ERP Systems	The risk ontology can be used as a systematic tool and checklist for risk identification, assessment and management, as well as for strategic planning and decision-making.

5. Ontologies Focusing on Software Process Models

A software process ontology [20] defines software activities, phases, and process models. Each phase is specified by a sequence of activities, each process model is described by process phases and each activity is associated with the artefacts it produces. Generally, a software project regardless its size and type, follows a Software Development Life Cycle (SDLC). In this section, we will discuss ontologies that are coupled with specific SDLC, such as Scrum, RUP, etc. In the next paragraphs ontologies based on SDLCs are presented.

Lin *et al*. [39] developed an ontology, named K-CRIO that allows the description of a specific kind of business processes: those that are dedicated to the design of a product. K-CRIO ontology describes the key concepts and relationships of software development process and it illustrates the use of the ontology by taking the example of the Scrum development process.

Santana [40] attempted to model Scrum [41] software methodology in OntoSCRUM ontology. The main purpose according to the author for this ontology was to reuse project artefacts and to share tacit knowledge within organizations and project teams.

Zualkernan [42] presented an ontology for Generating Assessments for the Scrum Process. The motive is to create assessments for a correct understanding of a process that can be used in a software development company.

Similarly Valaski *et al*. in [43] developed an ontology so called OntoRUP having as purpose to classify the learning material in software engineering. For this purpose they combined Software Engineering Body of Knowledge (SWEBOK) [44] and the Rational Unified Process (RUP) [45]. SWEBOK was used to define the structure of the software engineering knowledge area, while RUP was used for defining the axioms that represent relationships between concepts and to enable the reasoning to SWEBOK knowledge areas.

An alternative proposal for software process ontologies was made by Parson [46], who designed an ontology based on an analysis of a number of commonly used agile methods like Scrum [41], XP [13], FDD [14], etc. Parson took seven agile methods and attempted to summarize their terminology, and built an initial ontology of agile methods that attempts to encompass the various characteristics of commonly used methods. On this sense, his work can be considered as a generic ontology for agile process. Advancing on his research, Parsons [47] deals with Aspect Oriented Software development and agile methods integration. The author proposed an ontology enabled development based on an analysis of existing ontologies of aspect oriented programming, a proposed ontology of agile methods, and a derived ontology of agile aspect oriented development.

Ceravolo *et al*. [48] defined Extreme Programming Ontology (XPO), a formal model that describes the concepts of Extreme Programming (XP) methodology [13]. XPO is based on a set of core components modelling three main concepts: Organisational Role, Product and Phase. XPO can be used for indexing relevant documents, XP artefacts and Wiki pages, in order to mine and analyse agile processes, programmers' activity and repositories content.

Siddiqui and Alam [49] developed an ontology for Feature Driven Development (FDD) life cycle that can be used for application model development to feature design and implementation. Features are precisely defined in the OWL-based domain model. Transition from OWL based domain model to feature list is directly defined in transformation rules.

Capability Maturity Model Integration (CMMI) is a process reference model that was developed by the Software Engineering Institute (SEI) of Carnegie Mellon University. CMMI addresses the development and maintenance activities applied to both products and services. This model could be used for improving processes, and measuring the capability of a process or the maturity of an organization. CMMI components (including a model, its training materials, and appraisal related documents) are designed to meet the needs of some specific areas of interest, which is called constellation. There are three constellations as supported by the framework: CMMI for Development [50], CMMI for Services, and CMMI for Acquisition.

There are only a few studies on CMMI ontology in the literature. Soydan and Kokar [51] developed an ontology of the capability maturity model, the staged representation CMMI-SW (focusing on Software Engineering). The ontology has been developed in OWL-DL[8]. The ontology includes 309 classes and 4 kinds of properties (consistsOf, satisfiedBy, achievedBy and subClassOf).

Similarly, Sharifloo *et al*. [52] introduced an ontology for CMMI for Acquisition based on SUMO [53] upper ontology.

[8] The code is located at http://www.ece.neu.edu/groups/scs/onto/CMMI/cmmi.owl.

Gazel *et al*. [54] presented an ontology-based software process assessment tool which was developed to support data collection phase of process assessment and to track conformance of software processes to CMMI as the process reference model.

Using on the work of Soydan *et al*. [51], Rungratri and Usanavasin [55] developed Project Assets Ontology (PAO), an ontology which merges CMMI process areas and project assets. PAO was developed in the context of a framework called "CMMI v.1.2 based Gap Analysis Assistant Framework (CMMI-GAAF)" that has a primary objective the automatic gap analysis with respect to CMMI.

Lee *et al*. [56] [57] developed an ontology-based intelligent decision support agent (OIDSA) for project monitoring and control of capability maturity model integration (CMMI). Liao *et al*. [58] aimed to create generic Software Process Ontology (SPO) and strived to ensure that it covered the requirements of both CMMI and ISO/IEC 15504 [59]. A process was represented by atomic practices and to that end; an organization's process model could be represented by using SPO.

Líška and Navrat [60] presented an approach that uses Software Process Meta-Model (SPEM) [61] for improvements that are rooted in knowledge engineering approaches. They show how SPEM can be used in the Semantic Web technical space by developing a SPEM ontology. Consequently, they use this SPEM ontology for project plan generation and verification

Falbo and Bertollo [62] based on the idea of standard software processes, created the Software Process Ontology -SPO. This ontology evolved over the last years, with help of a Foundational ontology (UFO, unified foundational ontology). In addition, Falbo *et al*. [63] evolved all these artefacts and created what he called Ontology Language Patterns for Software Process (OLP-SP). The idea lying behind was to define pieces of software process ontologies that can be reused by other ontologies. This ontology is shown to be expressive enough to be used as a common ground for mapping the software process fragments of standards such as ISO/ IEC 12207-ISO 9001:2000-ISO/IEC 15504, CMMI, RUP and SPEM.

In [64] an ontology model of software engineering to represent the knowledge needed in software development is presented. They have analysed the characteristics of software engineering ontology and defined graphical notations of modelling software engineering ontology as an alternative formalism.

Klieber *et al*. [65] introduced an ontology to automatically support the software documentation process. Therefore, they introduced a software documentation ontology that is populated automatically through analyzing source code, documentation and code executions and made accessible to developers via a wiki.

Guizzardi *et al*. [66] presented the latest developments in the UFO foundational ontology. In particular, they presented new versions of two fragments of UFO, namely UFO-B (concerned with events) and UFO-C (dealing with social and intentional concepts).

Sarantis and Askounis [67] developed a project management ontology as a reference for e-Government projects. Their model enables knowledge reuse, collaboration and interoperability between all stakeholders related with the implementation of such type of projects. On the same topic, Magoulas *et al*. [68] developed a quality ontology that formalizes all the needed knowledge for the realization of a multi-perspective and adaptive evaluation of e-government services.

ICONS project [69], even though it was a research project, developed a Structural Fund Project Knowledge Portal for managing projects of European Commission Structural Funds that included a domain ontology. This Domain Ontology provides the conceptual foundation for all knowledge management applications, mainly PM related, and included the Concept Glossary, which defined the Structural Fund domain ontology. Moreover, this ontology provided support for both expert and non-expert use of the underlying knowledge representations and knowledge management services, as well as support for the automatic inference engine helping users find and merge information.

Mendes and Abran [70] presented a prototype of an ontology, which is able to represent the domain of Software Engineering, based on the SWEBOK guide. A literal extraction from the guide results in approximately 4000 concepts.

Sicilia *et al*. [71] also proposes a SWEBOK based ontology with a descriptive part in order to identify artefacts and activities and prescriptive part, with approaches and concrete activities' rules for "commonly accepted" practical activities.

Hilera *et al*. [72] proposed an ontology called OntoGLOSE based on the Software Engineering Terminology Glossary, published by IEEE. OntoGLOSE includes about 1500 concepts, corresponding to 1300 glossary terms with their different meanings.

Table 3. Ontologies focusing on software process models.

Authors/Ontology Name	Focus	Main Application
Santana [40]/OntoSCRUM	SCRUM	To reuse project artifacts and to share tacit knowledge within organizations and project teams.
Zualkernan [42]	SCRUM	To create assessments for a correct understanding of a process that can be used in a software development company.
Valaski *et al*. [43]	RUP, SWEBOK	To classify the learning material in software engineering.
Parson [46] [47]	SCRUM, XP, FDD,etc.	To summarize the terminology of agile software development methodologies terminology, and built an initial ontology of agile methods that attempts to encompass the various characteristics of commonly used methods.
Ceravolo *et al*. [48] defined Extreme Programming Ontology (XPO)	XP	For analysis and mining of agile processes in order to identify critical factors in agile software development.
Siddiqui and Alam [49]	FDD	The ontology will be used for semantic web application, for feature design and implementation.
Soydan and Kokar [51]	CMMI-SW	Modelling CMMI-SW.
Gazel *et al*. [54]	CMMI	The ontology was used to support the data collection phase of process assessment and to track conformance of software processes to CMMI as the process reference model.
Rungratri and Usanavasin [55]/Project Assets Ontology (PAO)	CMMI	To develop a framework called "CMMI v.1.2 based Gap Analysis Assistant Framework (CMMI-GAAF)" that has a primary objective the automatic gap analysis with respect to CMMI.
Lee and Chen [56]/OIDSA	CMMI	To be used for intelligent decision support for project monitoring and control of capability maturity model integration.
Liao *et al*. [58]	CMMI, ISO/IEC 15504	To represent a process model.
Sharifloo *et al*. [52]	CMMI for Acquisition	To be used for an application that evaluates the maturity level of an organization and offers recommendations.
Lin *et al*. [75]	CMMI, IEEE/EAI 12207	To develop a model that integrates concepts from both standards.
Líška and Navrat [60]	SPEM	Demonstration on how SPEM can be used for project planning.
Falbo and Bertollo [62]	Standard software processes	A model for helping software development organisations to use standards in their software process improvement efforts.
Falbo *et al*. [63]/OLP-SP	Standard software processes	An ontology that can be reused from other ontologies.
Mendes and Abran [70]	SWBOK	For modelling software engineering discipline.
Sicilia *et al*. [71]/Onto-SWEBOK	SWBOK	To be used as review tool for SWBOK itself.
Klieber *et al*. [65]	Expert Location	To be used for knowledge discovery within "KnowMiner" framework with the primary goal to ease usage for non-knowledge discovery experts.
Sarantis and Askounis [67]	E-Government	To be used for e-Government projects to enable knowledge reuse, collaboration and interoperability between stakeholders.
Magoulas *et al*. [68]	Quality Management	A quality ontology to be used for the evaluation of e-government services.
ICONS Project [69]	Project Knowledge	To be used in a Structural Fund Project Knowledge Portal for managing projects of European Commission Structural Funds.
Hilera *et al*. [72]/OntoGLOSE	Document Annotation	OntoGloss ontology is used for a document annotation tool that uses pre-defined concepts in ontology to mark–up a document with software engineering terms.
Ruiz *et al*. [74]	Software maintenance projects	The ontology is part of MANTIS software engineering environment having as objective to reuse information, knowledge and expertise generated during the maintenance of software projects.
García *et al*. [76]	Software measurement	For standardizing the software measurement terminology.
Girardi and Faria [77]/GRAMO	Requirements management, domain modelling	Used from ONTODUM for domain engineering.
Settas and Stamelos [73]/PROMAISE	Software antipatterns	Used for software project managers to select antipatterns.
González-Pérez and Henderson-Sellers [78]	Software Development	An ontology for software development methodology that include a metamodel and an architecture.
Lin *et al*. [39]/K-CRIO	Process management	K-CRIO is an ontology of organizations for their understanding, analysis and also enables reasoning. The K-CRIO ontology is illustrated by an example of a software development organization that follows the waterfall process.
Fable *et al*. [79]/ODE	Software Development	To be used in an ontology based software development.

Settas and Stamelos [73] developed a Software Project Management Antipattern Intelligent System (PROMAISE) that uses antipattern ontologies in order to provide intelligent and up to date advice to software project managers regarding the selection appropriate antipatterns in a software project.

Beyond the above, the literature contains a large number of ontologies can be found for various engineering aspects such as metrics, software maintenance, software quality, various software processes, specialised subjects or sectors, etc. [74]-[78]. However, these ontologies are not focused on project management and as such they are not included in this paper. An overview of the ontologies focusing on software process models is presented in **Table 3**.

6. Conclusions

This paper confirmed that a large number of ontologies in the areas of software project management and of software engineering have been developed. The authors performed a systematic literature review and they focused on ontologies that address the project management process or on ontologies that enable Ontology-Enabled Development (ontologies are used as a support tool for developers). Further, even though in the case software project management lower level processes have been examined (e.g. project time management and project cost management), we have selected to examine only generic software engineering ontologies addressing well established and well known software engineering processes (e.g. RUP and CMMI) or practices (e.g. XP and Scrum). This choice was made since a very large number of ontologies have been developed to address software engineering specific subdomain (e.g. requirements management, analysis and design). In many cases, there are overlaps making it extremely difficult for the research to categorize ontologies in disjoint categories.

This systematic research has illustrated that the most important issue in ontology usage in the area of software project management is standardization. The large number of project management methodologies and of software process models creates an extremely difficult environment to comprehend and to use. The reasons that contribute to this complexity are the existence of various overlapping standards both in the areas of project management and in the areas of software engineering. Quite commonly ontologies based on SWEBOK or PMI include hundreds of concepts and terms. Further, the large number of available software process models makes knowledge sharing difficult. Secondly, in many cases studied ontologies are experimental which implies that they have not been used in real cases or they do not cover all subject areas of PM or all phases of software process.

From this study, it was made clear that a reference ontology in the subject area of software project management is missing. This ontology should be developed from an international and well established organization in order to increase the likelihood of increased adoption and of usage in real life cases.

Acknowledgements

The research presented in this paper has been co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework. In particular, the research work has been financed by the R & D project “ONSOCIAL” which take place in the context of the “ARCHIMEDES III” National Research Programme.

References

[1] Rose, K.H. (2013) A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. *Project Management Journal*, **44**, e1. http://dx.doi.org/10.1002/pmj.21345

[2] International Project Management Association (2006) ICB-IPMA Competence Baseline Version 3.0. International Project Management Association, Nijkerk.

[3] Office of Government Commerce (2009) Directing Successful Projects with PRINCE2. The Stationery Office, Norwich.

[4] Sommerville, I. (2004) Software Engineering. *International Computer Science Series*.

[5] Standish Group (2009) Chaos Summary 2009: The 10 Laws of CHAOS. http://www.portal.state.pa.us/portal/server.pt/document/690719/chaos_summary_2009_pdf

[6] Charette, R.N. (2005) Why Software Fails [Software Failure]. *IEEE Spectrum*, **42**, 42-49. http://dx.doi.org/10.1109/MSPEC.2005.1502528

[7] Srikantaiah, T., Srikantaiah, T.K., Koenig, M.E. and Al-Hawamdeh, S., Eds. (2010) Convergence of Project Management and Knowledge Management. Scarecrow Press, Plymouth.

[8] Gruber, T.R. (1995) Toward Principles for the Design of Ontologies Used for Knowledge Sharing? *International Journal of Human-Computer Studies*, **43**, 907-928. http://dx.doi.org/10.1006/ijhc.1995.1081

[9] Suárez-Figueroa, M.C., García-Castro, R., Villazón-Terrazas, B. and Gómez-Pérez, A. (2011) Essentials in Ontology Engineering: Methodologies, Languages, and Tools. *CIB Conference W*078-*W*012—*2nd Workshop on EEB Data Models*, Sophia Antipolis, 26-28 October 2011, 9-21.

[10] Devedzić, V. (2002) Understanding Ontological Engineering. *Communications of the ACM*, **45**, 136-144. http://dx.doi.org/10.1145/505248.506002

[11] Motik, B., Patel-Schneider, P. and Parsia, B. (2009) OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. World Wide Web Consortium. W3C Candidate Recommendation. http://www.w3.org/TR/2009/CR-owl2-syntax-20090611

[12] Brooks Jr., F.P. (1995) The Mythical Man-Month, Anniversary Edition: Essays on Software Engineering. Pearson Education, Upper Saddle River.

[13] Beck, K. (2001) Planning Extreme Programming. Addison-Wesley Professional, Boston.

[14] Coad, P., Luca, J.D. and Lefebvre, E. (1999) Java Modeling Color with Uml: Enterprise Components and Process with Cdrom. Prentice Hall PTR, Upper Saddle River.

[15] Carmichael, A. and Haywood, D. (2002) Better Software Faster. Prentice Hall Professional Technical Reference.

[16] Augustine, S., Payne, B., Sencindiver, F. and Woodcock, S. (2005) Agile Project Management: Steering from the Edges. *Communications of the ACM*, **48**, 85-89. http://dx.doi.org/10.1145/1101779.1101781

[17] Gašević, D., Kaviani, N. and Milanović, M. (2009) Ontologies and Software Engineering. In: Staab, S. and Studer, R., Eds., *Handbook on Ontologies*, Springer, Berlin, 593-615. http://dx.doi.org/10.1007/978-3-540-92673-3_27

[18] Happel, H.J. and Seedorf, S. (2006) Applications of Ontologies in Software Engineering. *Proceedings of Workshop on Sematic Web Enabled Software Engineering* (*SWESE*) *on the ISWC*, Athens, 6 November 2006, 5-9.

[19] Ruiz, F. and Hilera, J.R. (2006) Using Ontologies in Software Engineering and Technology. In: Calero, C., Ruiz, F. and Piattini, M., Eds., *Ontologies for Software Engineering and Software Technology*, Springer, Berlin, 49-102. http://dx.doi.org/10.1007/3-540-34518-3_2

[20] Zhao, Y., Dong, J. and Peng, T. (2009) Ontology Classification for Semantic-Web-Based Software Engineering. *IEEE Transactions on Services Computing*, **2**, 303-317. http://dx.doi.org/10.1109/TSC.2009.20

[21] Saunders, M.N., Saunders, M., Lewis, P. and Thornhill, A. (2011) Research Methods for Business Students. 5th Edition, Pearson Education, Essex.

[22] Abels, S., Ahlemann, F., Hahn, A., Hausmann, K. and Strickmann, J. (2006) PROMONT—A Project Management Ontology as a Reference for Virtual Project Organizations. In: *On the Move to Meaningful Internet Systems* 2006: *OTM* 2006 *Workshops*, Springer, Berlin, 813-823.

[23] Hughes, R.T. (2010) Project Management Process Ontologies: A Proof of Concept. *Proceedings of the* 15*th Annual Conference of UK Academy for Information Systems*, Oxford, 23-24 March 2010.

[24] Sheeba, T., Krishnan, R. and Justin Bernard, M. (2012) An Ontology in Project Management Knowledge Domain. *International Journal of Computer Applications*, **56**, 1-7. http://dx.doi.org/10.5120/8884-2881

[25] Bodea, C., Trandafir, I. and Borozan, A.M. (2008) Assessment of Project Management Knowledge Using Semantic Networks in the SinPers System. *Proceedings of the* 22*nd IPMA World Congress on Project Management*, Rome, 9-11 November 2008, 1314-1319.

[26] Ruiz-Bertol, F.J., Rodríguez, D. and Dolado, J. (2011) Applying Rules to an Ontology for Project Management. *Proceedings of the* 16*th Spanish Conference on Software Engineering and Databases*, Coruña, 5-7 September 2011, 5-7.

[27] Aramo-Immonen, H. (2009) Project Management Ontology—The Organizational Learning Perspective. Tampereen teknillinen yliopisto, Julkaisu-Tampere University of Technology, Tampere, 836.

[28] Dong, H., Hussain, F.K. and Chang, E. (2011) ORPMS: An Ontology-Based Real-Time Project Monitoring System in the Cloud. *Journal of Universal Computer Science*, **17**, 1161-1182.

[29] Wong, S.M., Lafaye, J.Y. and Boursier, P. (2010) Developing PMIS for Business Projects Based on Social Science Research Findings and Ontology Modelling. *Proceedings of the International Conference on Information Systems* (*IADIS*'10), Porto, 18-20 March 2010, 283-291.

[30] Lai, Y.C., Christiansson, P. and Svidt, K. (2002) IT in Collaborative Building Design (IT-CODE). *Proceedings of the European Conference on Information and Communication Technology Advances and Innovation in the Knowledge Society*, Salford, 18-21 November 2002, 323-331.

[31] Smith S.F. and Becker, M.A. (1997) An Ontology for Constructing Scheduling Systems. In: *Working Notes of* 1997

AAAI Symposium on Ontological Engineering, AAAI Press, Palo Alto, 120-127.

[32] Rajpathak, D., Motta, E. and Roy, R. (2001) A Generic Task Ontology for Scheduling Applications. *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, 25-28 June 2001.

[33] Mochol, M., Oldakowski, R. and Heese, R. (2004) Ontology Based Recruitment Process. In: Dadam, W.P. and Reichert, M., Eds., *GI Jahrestagung* 2, Springer, Heidelberg, 198-202.

[34] Dorn, J., Naz, T. and Pichlmair, M. (2007) Ontology Development for Human Resource Management. *Proceedings of the* 4th *International Conference on Knowledge Management*, Vienna, 27-28 August 2007, 109-120.

[35] Schmidt, A. and Kunzmann, C. (2006) Towards Human Resource Development Ontology for Combining Competence Management and Technology-Enhanced Workplace Learning. *Lecture Notes in Computer Science*, **4278**, 1078-1087. http://dx.doi.org/10.1007/11915072_10

[36] Schmidt, A. and Kunzmann, C. (2007) Human Resource Development with Ontology-Based Competency Catalogs. In: Cunningham and Cunningham, Eds., *eChallenges* 07, Den Haag.

[37] Gómez-Pérez, A., Ramírez, J. and Villazón-Terrazas, B. (2007) An Ontology for Modelling Human Resources Management Based on Standards. In: Apolloni, B., Howlett, R.J. and Jain, L., Eds., *Knowledge-Based Intelligent Information and Engineering Systems*, Springer, Berlin, 534-541.
http://dx.doi.org/10.1007/978-3-540-74819-9_66

[38] Peng, G.C. and Nunes, M. (2009) Surfacing ERP Exploitation Risks through a Risk Ontology. *Industrial Management and Data Systems*, **109**, 926-942. http://dx.doi.org/10.1108/02635570910982283

[39] Lin, Y., Hilaire, V., Gaud, N. and Koukam, A. (2012) Scrum Conceptualization Using K-CRIO Ontology. In: *Data-Driven Process Discovery and Analysis*, Springer Berlin Heidelberg, Berlin, 189-211.
http://dx.doi.org/10.1007/978-3-642-34044-4_11

[40] Santana, A., Lóscio, B.F. and Freitas, F. (2013) A Scrum Application Ontology for Support Historical Data Management of Projects. Centro de Informática—Universidade Federal de Pernambuco, Recife.

[41] Schwaber, K. (2004) Agile Project Management with Scrum. Microsoft Press, Redmond.

[42] Zualkernan, I.A. (2008) An Ontology-Driven Approach for Generating Assessments for the SCRUM Process. In: Fujita, H. and Zualkernan, I.A., Eds., *New Trends in Software Methodologies*, *Tools and Techniques*. IOS Press, The Netherlands.

[43] Valaski, J., Malucelli, A., Reinehr, S.S. and Santos, R. (2011) Ontology to Classify Learning Material in Software Engineering Knowledge Domain. In: *ONTOBRAS-MOST*, UFPE, Recife, 37-47.

[44] Bourque, P. and Fairley, R., Eds. (2014) SWEBOK 3.0: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society Press. http://www.computer.org/portal/web/swebok

[45] Rational Software (2011) Rational Unified Process Best Practices for Software Development Teams, Rational Software. White Paper TP026B, Rev 11/01, Rational Software, Cupertino.
https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

[46] Parson, D. (2011) Agile Software Development Methodology, an Ontological Analysis.
http://www.academia.edu/1020810/Agile_software_development_methodology_an_ontological_analysis

[47] Parsons, D. (2011) An Ontology of Agile Aspect Oriented Software Development. *Research Letters in the Information and Mathematical Sciences*, **15**, 1-11.

[48] Ceravolo, P., Damiani, E., Marchesi, M., Pinna, S. and Zavetarelli, F. (2003) A Ontology-Based Process Modelling for XP. *Proceedings of the Tenth Asia-Pacific Software Engineering Conference*, Chiang Mai, 10-12 December 2003, 236-242. http://dx.doi.org/10.1109/APSEC.2003.1254376

[49] Siddiqui, F. and Alam, M.A. (2012) Ontology Based Feature Driven Development Life Cycle. *International Journal of Computer Science Issues*, **9**, 207-212.

[50] CMMI Product Team (2010) CMMI for Development, Version 1.3.

[51] Soydan, G.H. and Kokar, M. (2006) An OWL Ontology for Representing the CMMI-SW Model. *Proceedings of ISWC* 2006 *Workshop on Semantic Web Enabled Software Engineering*, Athens, 6 November 2006.

[52] Sharifloo, A.A., Motazedi, Y., Shamsfard, M. and Dehkharghani, R. (2008) An Ontology for CMMI-ACQ Model. *Proceedings of the* 3rd *International Conference on Information and Communication Technologies*: *From Theory to Applications* (*ICTTA*), Damascus, 7-11 April 2008, 1-6.

[53] Suggested Upper Merged Ontology (SUMO) http://www.ontologyportal.org

[54] Gazel, S., Akçapinar Sezer, E. and Tarhan, A. (2012) An Ontology Based Infrastructure to Support CMMI Based Software Process Assessment. *Gazi University Journal of Science*, **25**, 155-164.

[55] Rungratri, S. and Usanavasin, S. (2008) Project Assets Ontology (PAO) to Support Gap Analysis for Organization Process Improvement Based on CMMI. In: *Making Globally Distributed Software Development a Success Story*, Springer,

Berlin, 76-87.

[56] Lee, C.S., Wang, M.H. and Chen, J.J. (2008) Ontology-Based Intelligent Decision Support Agent for CMMI Project Monitoring and Control. *International Journal of Approximate Reasoning*, **48**, 62-76. http://dx.doi.org/10.1016/j.ijar.2007.06.007

[57] Lee, C.S. and Wang, M.H. (2009) Ontology-Based Computational Intelligent Multi-Agent and Its Application to CMMI Assessment. *Applied Intelligence*, **30**, 203-219. http://dx.doi.org/10.1007/s10489-007-0071-1

[58] Liao, L., Qu, Y. and Leung, H. (2005) A Software Process Ontology and Its Application. *Proceedings of ISWC* 2005 *Workshop on Semantic Web Enabled Software Engineering*, Galway, 6 November 2005, 6-10.

[59] ISO (2004) ISO/IEC 15504–2:2003/Cor.1:2004(E). Information Technology—Process Assessment—Part 2: Performing an Assessment. International Organization for Standardization, Geneva.

[60] Líška, M. and Navrat, P. (2010) An Approach to Project Planning Employing Software and Systems Engineering Meta-Model Represented by an Ontology. *Computer Science and Information Systems*, **7**, 721-736. http://dx.doi.org/10.2298/CSIS091110030L

[61] OMG (2008) Software & Systems Process Engineering Meta-Model Specification, Version 2.0. Object Management Group, Needham. http://doc.omg.org/formal/08-04-01.pdf

[62] Falbo, R.A. and Bertollo, G. (2010) A Software Process Ontology as a Common Vocabulary about Software Processes. *International Journal of Business Process Integration and Management*, **4**, 239-250.

[63] Falbo, R.A., Barcellos, M.P., Nardi, J.C. and Guizzardi, G. (2013) Organizing Ontology Design Patterns as Ontology Pattern Languages. *ESWC*'2013.

[64] Wongthongtham, P., Chang, E., Dillon, T. and Sommerville, I. (2009) Development of a Software Engineering Ontology for Multisite Software Development. *IEEE Transactions on Knowledge and Data Engineering*, **21**, 1205-1217.

[65] Klieber, W., Sabol, V., Muhr, M., Kern, R., Öttl, G. and Granitzer, M. (2009) Knowledge Discovery Using the KnowMiner Framework. *Proceedings of the IADIS International Conference Information Systems*, Barcelona, 25-27 February 2009, 9.

[66] Guizzardi, G., de Almeida Falbo, R. and Guizzardi, R.S. (2008) Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The Case of the ODE Software Process Ontology. *Proceedings of CIbSE*, Recife, 13-17 February 2008, 127-140.

[67] Sarantis, D. and Askounis, D. (2009) A Project Management Ontology as a Reference for E-Government Projects. *Proceedings of International Conference for Internet Technology and Secured Transactions*, London, 9-12 November 2009, 1-8.

[68] Magoutas, B., Halaris, C. and Mentzas, G. (2007) An Ontology for the Multi-Perspective Evaluation of Quality in E-Government Services. *Electronic Government*, **4656**, 318-329.

[69] Staniszkis, E. and Nowicki, B. (2004) ICONS Based Knowledge Management in the Process of Structural Funds Projects Preparation. *Proceedings of e-Challenges e*-2004 *Conference*, Vienna, 27-29 October 2004.

[70] Mendes, O. and Abran, A. (2005) Issues in the Development of an Ontology for an Emerging Engineering Discipline. *Proceedings of the First Workshop on Ontology*, *Conceptualizations and Epistemology for Software and Systems Engineering* (*ONTOSE*), Alcalá de Henares, 9-10 June 2005, 139-144.

[71] Sicilia, M., Cuadrado, J.J., Garcia, E., Rodriguez, D. and Hilera, J.R. (2005) The Evaluation of Ontological Representation of the SWEBOK as a Revision Tool. *Proceedings of the* 29*th Annual International Computer Software and Application Conference* (*COMPSAC*), Edinburgh, 26-28 July 2005, 26-28.

[72] Hilera, J.R., Sánchez-Alonso, S., García, E.A. and Del Molino, C.J. (2005) OntoGLOSE: A Lightweight Software Engineering Ontology. *Proceedings of the* 1*st Workshop on Ontology*, *Conceptualizations and Epistemology for Software and Systems Engineering* (*ONTOSE*), Alcalá de Henares, 9-10 June 2005, 9-10.

[73] Settas, D. and Stamelos, I. (2007) Towards a Dynamic Ontology Based Software Project Management Antipattern Intelligent System. *Proceedings of the* 19*th IEEE International Conference on Tools with Artificial Intelligence*, Patras, 29-31 October 2007, 186-193.

[74] Ruiz, F., Vizcaíno, A., Piattini, M. and García, F. (2004) An Ontology for the Management of Software Maintenance Projects. *International Journal of Software Engineering and Knowledge Engineering*, **14**, 323-349. http://dx.doi.org/10.1142/S0218194004001646

[75] Lin, S., Liu, F. and Loe, S. (2003) Building a Knowledge Base of IEEE/EAI 12207 and CMMI with Ontology. *Proceedings of the Sixth International Protégé Workshop*, Manchester, 7-9 July 2003. http://protege.stanford.edu/conference/2003/abstract.html

[76] García, F., Bertoa, M.F., Calero, C., Vallecillo, A., Ruíz, F., Piattini, M. and Genero, M. (2006) Towards a Consistent Terminology for Software Measurement. *Information and Software Technology*, **48**, 631-644.

http://dx.doi.org/10.1016/j.infsof.2005.07.001

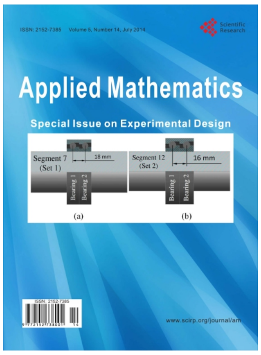
[77] Girardi, R. and Faria, C. (2003) A Generic Ontology for the Specification of Domain Models. *Proceedings of the* 1*st International Workshop on Component Engineering Methodology* (*WCEM'*03) *at Second International Conference on Generative Programming and Component Engineering*, Erfurt, 22-23 September 2003, 41-50.

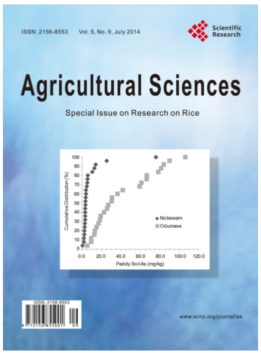
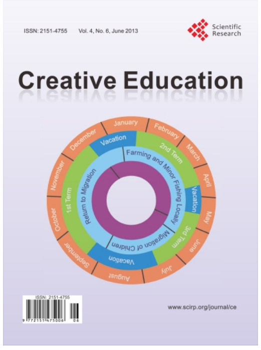
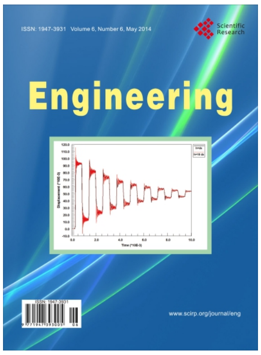
[78] González-Pérez, C. and Henderson-Sellers, B. (2006) An Ontology for Software Development Methodologies and Endeavours. In: Calero, C., Ruiz, F. and Piattini, M., Eds., *Ontologies for Software Engineering and Software Technology*, Springer-Verlag, Berlin, 123-151.

[79] Falbo, R.D.A., Natali, A.C.C., Mian, P.G., Bertollo, G. and Ruy, F.B. (2003) ODE: Ontology-Based Software Development Environment. *IX Congreso Argentino de Ciencias de la Computación*, La Plata, 6-7 October 2003, 1124-1135.

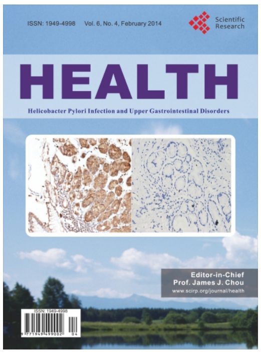
Advances in Bioscience and Biotechnology

American Journal of Analytical Chemistry

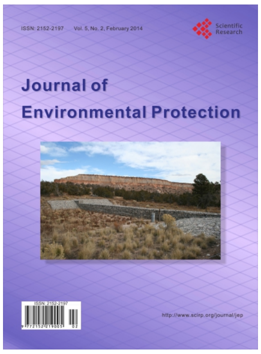
American Journal of Plant Sciences
Special Issue on Photosynthesis

Applied Mathematics
Special Issue on Experimental Design

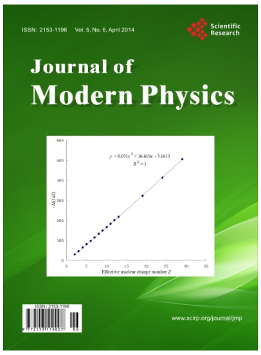

Agricultural Sciences
Special Issue on Research on Rice

Creative Education

Engineering

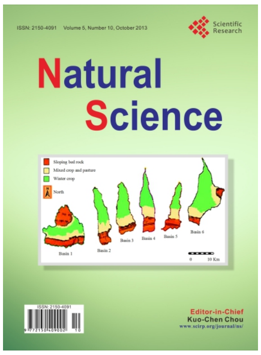
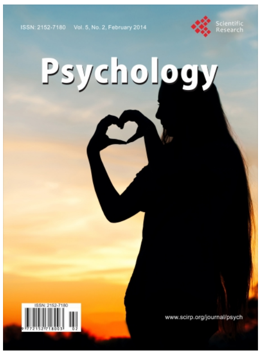
Food and Nutrition Sciences


HEALTH
Helicobacter Pylori Infection and Upper Gastrointestinal Disorders
Editor-in-Chief
Prof. James J. Chou

Journal of Computer and Communications

Journal of Cancer Therapy
Special Issue on Bone Tumor Research

Journal of Environmental Protection


Journal of Modern Physics

Modern Economy

Natural Science
Editor-in-Chief
Kuo-Chen Chou

Psychology