ABSTRACTS

Chapter 1. Tools supporting generation of a data intensive web applications. One of the fundamental problems of software engineering of business systems are the speed of delivery and minimization of maintenance cost. Either of these goals can be achieved with minimizing application complexity. Concerning systems, whose main task is processing persistent data, one of the most critical complexity factor is their persistence mechanism. The most popular approach is to map an application object model onto a relational database data model. Accelerating construction of an application is achieved by employing tools automating the date mapping process and generating programming language structures enabling manipulating the data. The paper presents problems appearing on the junction of incompatible models, despite of a really fast development of various mapping technologies. Furthermore, it proposes an alternative approach to persistence issues.

Chapter 2. Application Logic Patterns – reusable elements of requirements specifications. An application of patterns (pertaining to design, or architecture) is a common practice used towards reduction of costs and improvement of quality in the software engineering domain. Reoccurring schemes can be also utilized in the area of systems' functionality, but reuse of such patterns is currently infrequent. This chapter fills this gap by presenting a library of Application Logic Patterns. The elements of the library are precisely described by a language which is defined by a strict meta-model and which extends the notation and semantics of well-known UML structures. Typical activities supported by the business applications included in the library are represented in a way allowing their easy adaptation in practice. The patterns are supplemented by static, abstract domain models. The presented idea is supported by the open-source ReDSeeDS engine allowing for seamless inclusion of Application Logic Patterns in projects developed according to the principles of software modeling.

Chapter 3. Module supporting designing and development of complex information systems based on Spring and OSGi. This section presents a module extending functionality of CASE tool - Enterprise Architect. The module supports designing and implementation of complex computer systems based on Spring Framework and OSGi. Background of OSGi idea is presented in this work. The work describes proposed UML profiles providing extensions for building UML models of systems based on Spring and OSGi. A simple system described by class, package and component diagrams with customized UML element is shown. The work also presents a new additional software module integrated with Enterprise Architect (plug-in). The plug-in may be used for automatic generation of Spring and OSGi configuration files basing on automatic analysis of models built with mentioned UML extensions. The main contributions of this work are set of resources packed in MDG Technology file useful for designing Spring-OSGi-based system, and Enterprise Architect plug-in for programmatic generating of configuration files.

Chapter 4. Some rules of an effective design for semantic Web services. In the chapter, some key rules of semantic services in the Internet have been described. Especially, principles of the knowledge formalisms by the ontological systems have been considered, due to the OWL and KIF programming languages. Moreover, semantic networks as artificial intelligence techniques have been studied, in the context of the Web 3.0 frameworks. Finally, rules of using XML, XML Schema, RDF, RDF Schema, OWL, SPARQL as well as RIF have been discussed.

Chapter 5. Information systems integration in Service Oriented Architecture. The aim of this chapter is to investigate the usage of Service Oriented Architecture for integration of information systems. The first part contains presentation of integration technologies and applied solutions. Especially the usage and place of the Service Oriented Architecture in this domain is discussed. Further the technology of Web Services is presented as an implementation of this architecture. Finally the analysis and conclusions of the integration using Service Oriented Architecture, particular Web Services are presented.

Chapter 6. Automatic Consistency Checking of UML Models by using the three dimensional Document Circulation Diagram (DOD). Checking the consistency and completeness of UML diagrams by automatic transformation into three dimensional DOD model is presented. The three dimensional DOD model can be used to design the functionality, structure and even the behavior of an application. UML diagrams (use case, class and state machines) can be automatically transformed into the DOD model. In DOD model the consistency can be checked. Reverse transformations – from DOD into UML diagrams, are also implemented in Dodocum modeler.

Chapter 7. Consequences of modeling in software project - a survey with practitioners. The chapter presents the most interesting findings of the survey conducted to explore practitioners' opinions about modeling technology. The goal of the study was to explore the state of practice related to the use of modeling in software projects. The study has taken into account the areas of application, benefits of using modeling and UML tools as well as the problems with current state of technology and the problems which appear when modeling is not in use.

Chapter 8. Diophantine Model of Reachability Problem. Real-life experience provides us many examples of decision making inspired by cyclic scheduling problems**.** Because of an integer nature of data and decision variables some of them belong to a class of so called Diophantine problems. In general, that kind of problems is unsolvable. The cyclic scheduling problems are imposed by many however of different nature tasks such as scheduling of timetables, duties, training activities, and so on. In that context it seems to be quite obvious the decision support systems software has to express the Diophantine characters of problems consider. In particularly it means that a programmer involved in dedicated decision support software packages development has to remember that the system's structure determines however do not constraint its behavior. In other words, the system's structure do not allows encompassing the all behaviors assumed.

Chapter 9. Usage of the quality requirements in the model-driven development process Growing complexity of modern software systems raises need to search for methods that shorten the path leading from requirements to design and implementation. The MDA concept and its practical applications simplify this path by applying model transformations. Through the application of certain transformation rules to a precise functional requirements model we can conduct automatic generation of the architectural model fulfilling these requirements. However, such a process does not assure the fulfillment of quality requirements. These requirements, being the key driver of the architectural decisions, should drive the transformation configuration process. In this chapter we introduce a concept of quality requirements-driven transformation of functional requirements. There is presented an extension of the Requirements Specification Language (RSL) with the NFR Framework language for specifying quality requirements, combined with a concept of compound transformations.

Chapter 10. MWING and IMES systems and their applications to performance measurements of WWW network. The chapter presents two computer systems MWING and IMES designed to measure Internet and WWW network. The systems were designed in research projects supervised by the author. They were used in several performance active measuremets of WWW network installed in public Internet and in PlanetLab distributed environment. The measured data set was used to study file transfer time prediction methods and algorithms in real-life cases, including those which are based on data mining techniques. The author's new proposal named Web performance mining to be used for such prediction was also studied with a success.

Chapter 11. Making multiobjective decision using fuzzy ratings in the OLAP environment.
This chapter formulates the task of multiple criteria making decision in unclear conditions in OLAP environment. In addition, presents ways of solving this problem taking into account a relation of order in the space of fuzzy sets, assuming a finite number of permissible decisions. Fuzzy OLAP Multicriteria decision model, which was developed by the fuzzy model, OLAP, enables the extension of the traditional approach through the description of complex and imprecise relationships, combining a standard approach to decision support problems with the theory of fuzzy sets in a multidimensional OLAP environment.

Chapter 12. Balancing agility of the Scrum methodology with maturity of the CMMI model. The chapter presents an approach to balancing the agility of Scrum with the maturity of the CMMI model. At first, the Scrum methodology and the CMMI maturity model are outlined. Then, a model of mapping the CMMI specific practices to the Scrum practices is proposed. Wherever a CMMI practice is not covered by Scrum, extensions and additional practices are defined in a way that sustains agility while introducing discipline and maturity. To assist in the practical use of the presented approach to the identification of CMMI and Scrum compatible set of practices, the application process is defined supported with a questionnaire and a software tool. Finally, the results of the validation in two case studies with small and medium IT companies as well as by expert opinion are reported.

Chapter 13. Team programming. Supported practices by the TFS system (Team Foundation Server). Software development is complex process. Success of software development effort depends on many factors. Software engineering is discipline to achieve better quality of IT product. This chapter describes methodologies that helps mitigate potential risks and can guarantee success in IT projects. Chosen methodologies: Scrum and MSF belongs to agile methodologies group. Following paragraphs in that chapter characterizes software development environment from Microsoft – Team Foundation Server. It also shows how TFS applies Forrester definition of Application Lifecycle Management (ALM), and which different software development practices can be implemented and utilized when using TFS. Additionally it shows what kind of benefits can be achieved after implementing certain software development practices. Chapter ends with experience from Volvo Information Technology utilizing TFS for last 5 years.

Chapter 14. Software Project - from idea to implementation. Teamwork with use of Internet. This chapter is about software project education of information technology students which study with use of internet. The Authors discuss some methods and tools that are helpful teaching software engineering and software project realization with team.

Chapter 15. System creation based on cooperating state machines. The chapter presents a system development supported by model driven engineering. Complex logic of a system can be specified using state machines associated with particular classes. Integration of different system parts can be specified at earlier stages of the system development. A model including class and state machine models is transformed into a source code. An executable application is built with usage of a library implementing all single elements of the UML behavioral state machine. This task was accomplished using the FXU environment (Framework for eXecutable UML), which transforms UML class and state diagrams into C# code. The approach was illustrated with a case study devoted to a social network of mobile users. The model and created application realized a presence server for the statuses services in the network. The server completes three main tasks: subscription of a status of another user, publication of a new status with given rules and notification another user about a status.

Chapter 16. Practices in software engineering – team work perspective. The chapter presents an overview of practices recommended by popular development methodologies, and analyses their usability in the context of a software team project. The analysis was conducted on the basis of selected agile methodologies (XP, Scrum, OpenUp), and one plan-driven methodology – RUP. Moreover, the definition of the notion *practice* (*best practice*) is given as its meaning is usually misunderstood and wrongly considered as the development (sub)processes.

Chapter 17. Group decision-making in IT process management. The chapter is devoted to ways of group decision-making (GDM) for IT project management. Detailed attention is paid to the projects which deal with corporate management support systems. Purposefulness of GDM application with regard to IT projects is justified and conditionings for its application are presented. Utilisation of AHP/ANP for GDM implementation is proposed. Different detailed ways are considered with this regard. Practical dimensions of GDM application for IT project management complement theoretical discussion. A sample decision-making problem deals with the selection of appropriate IT project management tool. Levels of attributes which describe the project are included in the analysis.

Chapter 18. Integration of a business process architecture and an ERP system. Methods of the integration of a business process architecture of a company and an ERP (Enterprise Resource Planning) system are presented. The "integration" means such an adaptation of an ERP system in a company, that the system fully supports a management at the operational level, i.e. it supports the business processes in the best possible way. The choice of methods and tools described in the chapter is based on the authors' experiences in several dozen of implementation projects, realized in large Polish enterprises. The methods can be used to select the best ERP system for a company, to design its implementation, and to customize the system accordingly to the requirements of the company.

Chapter 19. Modelling Integration Services Using OfficeObjects® Service Broker Platform. The paper proposes an application platform developed originally within the European research and development project eGovBus and further extended commercially by Rodan Systems S.A. The platform offers a flexible architecture enabling design and implementation of generic services in numerous areas of the ITC applications. The paper discusses platform architectures, presents various tools available to users and overviews the available functionality. Designing integration services using OfficeObjects®ServiceBroker is based on the dedicated methodology briefly described in the paper, as well as on the set of visual tools supporting all stages of design and development of services in question. All available user tools are integrated within Services Design Environment which is accessible from Web browser level. Final section of the paper presents real-life example of the integration service designed and implemented using the discussed platform.

Chapter 20. Testing in Incremental and Evolutionary Software Development Process. The article presents experience from the process of testing a complex Internet based system. The system is under development for the last five years. The development process follows incremental and evolutionary lifecycle model and the system is delivered in subsequent releases. Main releases and intermediate releases are distinguished where a main release undergoes a full range of regression tests whereas for an intermediate release the scope of regression testing is limited. The article positions the testing process within the context of change management and describes the applied testing environment. A format for documenting test cases is given and explained. The problem of automation of tests is discussed in more detail. Experimental data presenting the costs and benefits related to tests automation are given. Plans for further development of the described approach are presented.

Chapter 21. PN ISO/IEC 12207 Standard as a Guaranty of Software Rollout
In this chapter described practical aspects of application PN ISO/IEC 12207 standard to guarantee dedicated software buyer interests. Standard can be used to software acquisition agreements in which project method is not defined and supplier have ISO PN ISO/IEC 9001:2001 quality system certificate. Chapter contents description how standard can be used to rollout assessment in detail how decide in controversy around rollout progress or result. PN ISO/IEC 12207 standard can be also apply in the legal proceedings to analyze "conservation of adequately accuracy" in the sense of civil law 527 article. Confirm by the court judgment "lack of the conservation of adequate accuracy" denote not only lawsuit lose with their consequence (payback of rollout costs with interest) but in the case of project sponsored from the public founds means also exclusion of public auction during the 3 years.

Chapter 22. OPINION on "New, Modern, Innovative" Technology. Chapter describes practical aspects – possibility of obtaining public found or tax reduction which can be obtain proper investment in the "new, innovative" or "modern" technology. Opinion on that notion can be emit by independent cells of the university. In that kind of opinion it is necessary to prove the compatibility with "legal" definitions. Chapter describes profits from having opinion about "new technology " in the sense of the low from the 25.02. 1992 – income tax from the law persons. Also opinion on the "innovative" technology is discussed. This kind of opinion is sometimes required by the Polish Agency of Enterprise Development and Regional Financing Institutions. At the end possibility of tax reduction in the case of "modern technology acquisition and rollout" in the protected employment enterprise is presented.

Chapter 23. Model transformation and comparison as an approach to validation of consistency of implementation and its architectural design. The idea of a popular Model-Driven Engineering (MDE) software development methodology focuses on creating abstractions ranging from very general to very detailed ones. One can assume that the usual final outcome of the project – the implementation code – is the most detailed view. Maintaining consistency between the software model and the code is the key factor of successful software development project. On the other hand the disorder in the software and so inconsistency between design model and code grows in time. It is an obligation of the software project managers to assess and limit those discrepancies as often as possible and necessary. This approach requires availability of a software tool requiring reasonable effort and cost to perform such comparison. The paper presents a concept of such a tool which supports quick assertion of consistency of model and related code. Implementation Java code is reverse engineered to a model which is compared to the initial design model. A set of similarity metrics is also proposed and experimentally evaluated.

Chapter 24. Prediction of defects on the basis of software metrics - the identification of classes of projects. The design elements of the model system indicating burdened with the highest number of defects, although it's not complicated, it is not the act is often done in industrial production processes of the software. Probably the main reason for this is a negligible amount of work investigating the possibility of re-use the model in another project. This paper presents research in which classes were identified projects which can be used the same model prediction of defects. The hypothesis that the environment in which the design is derived which determines the type of model should be replicated, and then statistically verified this hypothesis using a defect prediction models for different types of projects and assessing the accuracy of prediction obtained

Chapter 25. Analysis of the possible realizations of Operations Registration Module in the System Assisting Operational-Processing Activities. Chapter presents the analysis of possible realizations of the module responsible for registration of all system's users' activities. Possible architectures of such system, as well as solution used in prototype of the module, were presented. The prototype is a part of a project currently realized by Silesian University of Technology in cooperation with Wasko company.

Chapter 26. Business process verification using semantic tableaux. This work applies formal verification of business process models to deductive reasoning. BPMN notation is widely used to describe the business processes of enterprises, but can also provide visualization of the BPEL language, which could be considered as a basic language of SOA. For business models and specifications temporal logic is used as well describing the issues of liveness and safety system properties. The method of semantic tableaux is applied as a method of inference. It can run automatically, and is an interesting alternative to the traditional approach to deductive reasoning. It enables relatively easy to identify errors in the specification. Methods for obtaining the temporal logic formulae directly from BPMN models is proposed.

Chapter 27. Automatic verification of the model at the stage of requirements analysis. This article applies to issues of formal verification of use cases and scenarios of use cases written in the form of activity diagrams using deductive reasoning. Methodology was proposed to describe the transition from use case diagrams, activity diagrams. For the specification of the system and requested his property has been used temporal logic, and deductive inference is done using the method of semantic tables. The proposed method of obtaining the formulas that were temporal logic directly from use case scenarios stored in UML'owych activity diagrams.

Chapter 28. Indexing in Processing of Heterogeneous Resources. In this paper, we present a volatile indexing technique which addresses a difficult query optimisation domain concerning processing global queries addressing heterogeneous resources. In contrast to regular indices, which are redundant structures stored at a server, a volatile index is materialised only during a query evaluation. Therefore, a performance of this technique manifests when the index is invoked multiple times which mainly concerns processing of complex and laborious queries. The volatile indexing has been implemented and tested in the prototype ODRA OODBMS implementation.

Chapter 29. A model driven tool for design and simulation of Probabilistic Faults with Time Dependencies. Probabilistic Fault Trees with Time Dependencies (PFTTD) enable efficient analysis of technical systems, in which duration of faults might be described by random variables. PFTTD provide not only high expressive power, but also lucidity and intuition to engineers, because they refer to respected standards. In order for PFTTD to become acclaimed, automation of dependability evaluation must be facilitated by designing a dedicated tool for their design and analysis. Taking advantage of intermodel transformations available at the Eclipse platform, a model driven application was built to design the trees graphically, validate them syntactically and verify semantically. Furthermore, two approaches to behavioral specification transformation were presented: using Groovy language and JBoss rule engine. After verifying simulation results, the implementations' nonfunctional characteristics were considered. Keeping in mind future research of PFTTD extensions, a final choice was rationalized.

Chapter 30. On two methods of simulating concurrent tasks. In the chapter two methods of simulating concurrent tasks are presented and analyzed. In simulation every task is described by set of differential equations and control law, which directs task to the target and allows to avoid obstacles. Every task is activated in synchronous way with fixed sampling period. The first method builds simulation objects as Timer objects from Swing package. In the approach all tasks are executed by the same dispatch thread of Swing. Potential usage of multicore processor will have no effect on simulation efficiency. The second methods uses the specialized class from java.util.concurrent package. The Executor type class has thread pool with declared number of working threads and assigns them simulation tasks. With such construction Java Virtual Machine can allocate different threads to be executed by different processor cores. In such case the simulation tool should work more effectively.

Chapter 31. Budding – the software development method of interoperable e-services for distributed environments. Two modern software development methods are discussed: the iterative-incremental methodology and agile techniques, their advantages and disadvantages in the context of building interoperable platforms and distributed environments. There is introduced the budding software development methodology. Its advantages and disadvantages are mentioned. There are presented some technologies on the basis of which the budding software development method works: Software Product Line, Enterprise Application Integration and Open-Source. Their strong points and usefulness are also mentioned. The procedure of the budding software development method is presented in a formal language. The steps are placed in the multidimensional space. There is described the use case of the budding software development method while building the interoperable service-oriented platform at Gdansk University of Technology. Some parameters connected with such a method are also discussed. The comparison of the iterative-incremental methodology and the budding software development methodology is presented. Finally, there are discussed the results of the introduction of the software development methodology at Gdansk University of Technology.

Chapter 32. Integration between process design and verification faze of designing object with using neural network. Proposed solution in area of informatics technology CAE has been experimentally used in limited range. In first phase, based on collected rich experimental material, has been built expert system used to prediction load of construction fixed power transmission system of helicopter. This solution, based o simulation method, eliminate collection of parameters which can be potentially dangerous for construction. Expert system used to modernization of construction make possible determine predict tensions for new area of parameters of the flight. Described solution can make shorter construction process, based previously on interactive testing all versions of construction on research stands or by flight. It was vary impotent, particularly for this areas of parameters of flight when relation between them and load of responsible part of helicopter are nonlinear.

Chapter 33. Standards of medical information as a form of data integration in hospital information systems. This chapter describes the development of standards in collecting and processing medical information. Two basics Standards – HL7 and DICOM are described along with the advantages and disadvantages of their use. Hospital Information System and it's components are also discussed.

Chapter 34. Research on information technology for internet services' provider. The paper presents research on information technology which is being used to manage portal content (CMS). The company finds current tool insufficient in the face of growing needs. Therefore it was decided to verify if it is required to change the system and estimate the possibility and efficiency of its modification. Authors analyzed the functionality of the CMS divided into separate categories of modules and evaluated each of them. At the same time they performed audit of whole company using CobiT methodology. Different business processes of the four CobiT domains were analyzed. The result of the research was supposed to support managers in making decision regarding possible changes in current situation.

Chapter 35. Models of information systems integration in companies. The article indicates the need for supporting IT organizations with incident and change management IT tools. The article presents the characteristics of a computer system for comprehensive management of projects, which can be used to resolve the issue of version management, releases and incidents.
Key words: Rational Team Concert, integration of IT systems

Chapter 36. Implementing change management processes in an enterprise: This article demonstrates how to implement change management processes in a company, based on the suitability of IBM software. Company workflow processes were studied and procedures for changes were presented. In relation to these changes, the implementation of three concepts of workflow documentation were assessed. This fulfills the pragmatic goal of the study (support for the managers of a company) as well as the scientific goal (the development of a multi-criteria system for decision support in management).

Chapter 37. Closing the Gap Between Industry and Science with Smart Knowledge Engineering Based Management Support. Knowledge Engineering (KE) techniques are becoming useful and popular components of hybrid integrated systems used to solve complicated practical problems in different disciplines. Knowledge Engineering techniques offer features such as: learning from previous knowledge, handling noisy and incomplete data, helping with decision making, and predicting capabilities. Within KE, its applied tools are exposed to public deliberation and analysis upon the enhancement of management support. A panel discussion to examine industry and science on the research and application programs would help researchers and practitioners in developing goal paths for the future of KE. Real "SMART" technologies require novel solutions and joint work between industry and science can contribute to resolving issues surrounding such technologies. Deliberation and analysis may help the public, researchers and enterprises better understand the issues, motivate their involvement, develop better ideas for addressing the issues, and enhance the achievement of novel solutions. Several researchers and industries are continuously examining the above mentioned problems; in particular, as part of a panel of experts in different fields, from industry and science, I will examine KE research questions, methods, experiences, efforts and results.

Chapter 38. Design concept of Intelligent Management System. In this paper the conception of designing and building of the intelligent decision support systems in production management is introduced. New approach to designing of the intelligent management support systems is proposed, based on integration of technologies of artificial intelligence (fuzzy logic, artificial neural networks, expert systems and genetic algorithms) with exact methods and models of search of decisions, and also simulation techniques. The proposed approach allows creating intelligent system, providing the decision of complex, unstructured problems of management in fuzzy conditions, which learn by accumulated data and adapt to changes for conditions of operation.

Chapter 39. The queue based message exchange platform – design pattern. The contents of the chapter presents design pattern (components) for communication platform in the enterprise systems. By introducing of various components, like ConfigurationService, MessageService and ClientGateways we construct comprehensive message exchange system. The system is able to achieve high reliability, security and scalability.

Chapter 40. Partial trace selection in time-travel debuggers. Time travel debuggers look similar to traditional debuggers but they support stepping back in time. This can be achieved by logging analyzed program history at run-time and browsing a simulated program execution afterward. Logging full program trace is space-consuming. We consider a partial trace debugger, a tool similar to time-travel debugger except that it logs only a part of full trace information. We propose two approaches to strategies of partial trace selection: one based on code coverage, and one based on clustering algorithm.

Chapter 41. Requirements Driven Regression Testing. The title in English. Skilful selection of regression tests is an important aspect of software quality assurance which also influences its costs. This chapter introduces a technique of automatic regression test selection from the complete test suite, based on requirements traces designated through MDA-style transformations. The requirements model, written in RSL, is transformed into design models, and then into code. Following the requirements traces, all the elements, which are responsible for the implementation of the chosen requirement (or group of requirements), are found. It is also possible to find all the requirements, which might be affected by changes in implementation, through the discussed traces. The implementation of this complementary regression testing method is based on the ReDSeeDS tool and can be used for the development of software with high level of interaction between the users and the system.

Chapter 42. Semiautomatic document clustering with given topic. Common clustering algorithms learn without supervision, proposed method lets the user influence the clustering flow by giving a topic. Described method consists of following stages: topic input, data tagging based on topic and semantic relations, using hierarchic, optimizing clustering algorithm with a criterion function to compute best division of documents.

Chapter 43. Knowledge Management – decision support in a workflow system. An exemplar decision support feature implementation is presented in a workflow system. A simplified loan decision process is chosen as a business process and the rule logic and the case-based reasoning as decision support methods. Implementation notes are also included especially in area of enabling efficient knowledge management in such system.

Chapter 44. Improvement of the website of Bank Zachodni WBK S.A. based on usability tests. The chapter describes the application of usability tests in an industrial case study. The tests were aimed at improving the usability of main website of one of Polish biggest banks. The introduction of the chapter outlines the state-of-the-art methods and tools for usability assessment and optimisation. Selected methods and tools were applied in the series of usability evaluation activities including: heuristic evaluation, scenario-based usability tests involving a number of volunteers and analysis of real users behaviour with clicktracking. The final part of the chapter presents the change proposals issued on the basis of evaluation results, the process of change management and implementation as well as the conclusions.

Chapter 45. Integration of ontological models for IT project management methodologies on example of change management. For many IT enterprises, whose primary activity consists in realization of projects, application of defined project management methodologies, either classical, or agile, is often considered as a key solution that allows increasing their efficiency and competitiveness. Very often an enterprise uses both classical and agile methodologies for projects run in parallel; there are also many efforts to adapt methodologies to local environments or to integrate them. In the chapter we propose to use formal ontological models to describe concepts in project management methodologies and to perform integration by integrating their ontological models. We discuss basic elements of two ontologies modeling the classical methodology PMBOK and the most popular agile methodology Scrum: artifacts, roles, processes and events. Those concepts together with a set of specific relations are used to define formally change management in both methodologies.

Chapter 46. Web Application Security Active Testing Method HttpValider and its Effectiveness Assessment. The chapter concerns web application security assessment, in particular passive and active approach to security testing. Active testing HttpValider method is proposed. It uses HTTP request generation for detecting 13 types of security vulnerabilities. A software tool was designed and implemented. The tool performs automatic tests for a web application usage scenario recorded by the tester. The method effectiveness was proved by the validation conducted for a set of test web applications with and without any security vulnerabilities. A test has been conducted to compare Httpvalider effectiveness with two other security testing tools. The comparative test was performed for two web applications. The results presented in the chapter confirm HttpValider effectiveness for detecting a set of 13 security vulnerabilities.

Rozdział 47. Smart knowledge engineering based management support. We discuss the problems related to the maintaining of semantic load and in general of knowledge between the different stages of the PLM and how these techniques indeed help in the smart management. We address implementation issues like legacy systems integration and web based applications taking into account security and reliability of the knowledge produced. Questions and answers can be also from the industry perspective as many of the presented and discussed techniques are indeed applied in real world applications running in managed environments in different types of industries. The implications of knowledge embedded in engineer objects are not only an open issue from a technical perspective, but also an ethical discussion will arise in the next years as knowledge and experience from users will be transferred to intelligent objects providing foundations to the Internet of things paradigm.