ONTOLOGY-BASED DATA ACCESS AND MODEL TRANSFORMATIONS FOR ENTERPRISE INTEROPERABILITY

ОНТОЛОГИЧНО БАЗИРАНИ ДОСТЪП ДО ДАННИ И ТРАНСФОРМАЦИЯ НА МОДЕЛИ ЗА ОПЕРАТИВНА СЪВМЕСТИМОСТ НА ПРЕДПРИЯТИЯТА

Assist. Prof. Dr. Gocheva D. G.[1], Prof. Dr. Batchkova I. A.[1], Prof. D.Sc. Popov G. T.[2]
[1]Dept. of Industrial Automation, University of Chemical Technology and Metallurgy
Bul. Kl. Ohridski 8, Sofia, Bulgaria
[2]Dept. Technology of Machine Tools and Manufacturing, Technical University – Sofia,
Bul. Kl. Ohridski 8, Sofia, Bulgaria

dani@uctm.edu, idilia@uctm.edu, gepop@tu-sofia.bg

Abstract: *Enterprise interoperability is the ability of disparate and diverse information and control systems to work together efficiently towards mutually beneficial common goals. Some of the main tasks for achieving enterprise interoperability are connected with data and knowledge access and sharing and by use of model transformation. In this paper are presented and discussed several model transformations between different Technological Spaces: RDB, UML and OWL in order to enable ontology-based access to different models of enterprise data. At the core of the approach is the usage of reference models, based on the standard for enterprise integration ISO/IEC 62264.*

Keywords*:* ONTOLOGY, INFORMATION, INTEGRATION, OWL, UML

1. Introduction

Information access and data retrieval from distributed heterogeneous data sources are important, but difficult tasks in the modern industrial enterprises. Some of the possibilities in this direction are standard frameworks and data models, based on the best practices, which separate the information and knowledge from specific commercial products and methods of implementation. In this context, the ISO/IEC 62264 standard [1] (former ANSI/ISA S95) provides common terminology and consistent reference models that represent the best practices for integration, applicable the entire life cycle of enterprises. The reference models, provided by the standard, are built on a high level of abstraction, independent from the specific industry and might be used as a basis for development of any enterprise systems as well as for standard compliant interfaces between the legacy systems.

A major trend in enterprise systems over the last decade has shifted from the traditional programming of imperative code to the creation of rich, interchangeable representations of information and knowledge [2]. Two significant paradigms (also known as Technological Spaces (TS) in [3]) have emerged in recent years to support this activity, each with its own terminology, standards bodies, research communities and flagship language: Model Driven Development (MDD) evolved primarily in the software engineering community and has the OMG's Unified Modeling Language (UML) [4] as its flagship language. The other is the so-called ontology engineering paradigm that places ontologies at the center of the development process. Ontology engineering primarily evolved from the artificial intelligence community and has the W3C's Web Ontology Language (OWL) [5] as its flagship language. Several publications discuss the relation of UML and OWL in general and the transformations of UML class diagrams into OWL ontologies. In [6] the authors investigate whether and how conceptual models written in UML can be automatically transformed into models written in OWL 2. Differences and similarities of various model elements in static UML data models and OWL 2 ontologies are shown and a transformation for similar elements is provided, indicating that data models written in UML can be represented as OWL ontologies quite well. But the different extent of the meta-models clearly suggests that OWL provides much more complex means of modelling already, so the transformation of general ontologies in UML data models is not always possible [6]. The transformation into an OWL 2 ontology allowed to check the consistency of the UML models and the data [6]. In [7] we discussed the two different approaches for a transformation between UML and OWL 2: XML-based transformations and transformations built upon meta-models using transformation languages (ATL, QVT-R). In this paper we introduced transformation rules in order to provide semantic descriptions of the object models proposed in ISO/IEC62264 standard, which use only few UML modeling elements. The suggested rules are built upon the meta-models of the UML class diagrams and the language OWL 2. The paper [8] also investigates providing semantic descriptions for particular enterprise models with limited UML elements, in order to promote system interoperability, enterprise information integration and information retrieval.

Relational databases are considered as one of the most popular storage solutions for various kinds of data. But according to Honeywell Process Solutions [9] relational data models can be very complex, difficult to report, inflexible. The variety of DBMS implies synchronization across data sources. Using data warehouses duplicates data and is not suitable for real time. Contrariwise, semantic data models are easier to understand, easier to report, flexible and can merge any data without replication; they are able to perform true federation: support structural change and system discovery across data sources [9]. The problem of bridging the gap between relational databases and semantic web ontologies has attracted the interest, and has been recognized as a key factor in generating huge amounts of data for Semantic Web applications based on standards RDF, OWL, SPARQL [10]. Transforming structured data sources into universal description model, as RDF aspires to be, enables seamless integration of their contents with information already represented in RDF [10]. A lot of methods and tools, classified in Knowledge Extraction Tools Survey Ontology (KETSO) are analyzed in [11]. Since data management according to the relational data model is still an order of magnitude faster than RDF data management and RDBs are the dominant data storage solution for enterprise data, relational data management will be prevalent in the next years.

Compared with relational databases, object data models are easier to understand and also can merge measurement's data [9]. The Object-Relational Mapping (ORM) enables transformations between object models and relational databases. For application development it is easy to start from creating the model by using the UML Class Diagram to generate the executable persistence layer for the model. By Model Driven Development (MDD) approach using models at different levels of abstraction and applying model transformation to code generation, it is useful also and to modify the Entity Relational (ER) model, which comes from reverse engineering of an existing database, transform into object model and generate persistence layer. Mutual software tools automate the Object-Relational Mapping enabling the MDD of different software systems.

The aim of the paper is to present and discuss several model transformations between the three Technological Spaces (TS): the DBMS TS (with a representative RDB), the Model Driven Architecture TS (with a representative UML) and the Ontology Engineering TS (with a representative OWL) in order to enable ontology-based access to different models of enterprise data. The basis of the approach is the usage of a standard for information integration during all phases of the life cycle of systems.

The paper is organized in 5 parts. In part 2 a short overview of the basic technologies and tools used in the approach is given. The third part of the paper discusses the approach for ontology based data access and information retrieval. In Part 4 the applicability of the suggested approach is illustrated. Finally some conclusions are made.

2. Basic technologies and tools used in the approach

2.1. Web ontologies

Web ontologies are used to model and share knowledge among various applications in a specific domain facilitating interoperability between different systems. The Web Ontology Language OWL 2 [5] accepted as a World Wide Web Consortium (W3C) is a powerful knowledge representation language; it has been applied successfully for knowledge modelling in many application areas. As a descriptive language, OWL 2 is used to express expert knowledge in a formal way, and as a logical language, it is used to draw conclusions from this knowledge. Every OWL 2 ontology is a machine - processable formal description of a domain of interest and consists of the following three different syntactic categories: *Entities*, *Expressions* and *Axioms*. *Entities*, such as classes, properties, and individuals are identified by IRIs. They form the primitive terms of an ontology and constitute the basic elements of the ontology. *Expressions* represent complex concepts in the domain being described. *Axioms* are statements that are asserted to be true in the domain being described and allow relationships to be established between *Expressions*. The ability to infer additional knowledge is of great importance for designing and deploying OWL ontologies. Sound and complete OWL 2 reasoning is of high complexity, but the OWL 2 new profiles (OWL 2 EL, OWL 2 QL, OWL 2 RL), known also as lightweight ontology languages, restrict the used syntactic categories to improve complexity and practical performance [12]. The profile OWL 2 RL enables the implementation of polynomial time reasoning algorithms using rule-extended database technologies operating directly on RDF triples [12]. As any OWL 2 Ontology can be mapped to RDF, for ontology-based data access can be used SPARQL - a query language across diverse data sources, whether the data is stored natively as RDF or viewed as RDF for RDF models [13].

2.2. A standard for information integration

ISO/IEC 62264 is a multi-part standard that defines the interface content between enterprise activities and control activities [1]. This standard is an agreement between leading companies to create a common framework, and guidelines for design and integration of systems providing common models and terminology for structuring and describing the exchanged information. The standard provides UML models, which are the basis for the development of information systems and standard compliant interfaces between different systems.

The semantic approach, presented in this paper, uses the models and terminology, provided by the standard ISO/IEC 62264 together with a meta-ontology [14], conceptualizing knowledge embedded in the standard as a target OWL ontology in order to use together data presented in different data models. The mapping rules between UML class diagrams (provided by the standard) and OWL elements of the meta-ontology are presented in [15]. The standard conformed meta-ontology (Fig.1) is comprised of 103 classes, structured in four categories of resources: *Personnel*, *Equipment*, *Material* and *ProcessSegment* (marked on the figure) and five main interface classes, defining the relations Production-Process-Product: *ProductDefinition*, *ProductionCapability*, *ProductionSchedule*, *ProcessSegmentCapability* and *ProductionPerformance*. Based on the main resources, *ProductDefinition* information indicates what must be defined to make a product, *ProductionCapability* information defines what resources are available, *ProductionSchedule* information defines what actual production will be executed and *ProductionPerformance* information states what actual production was achieved.

The meta-ontology has the following favourable features: (i) it establishes clearly defined standardized terminology and structure for general enterprise systems; (ii) it is independent of specific industry or technology; (iii) it facilitates building domain models. The meta-ontology might be used for design of new enterprise systems or reengineering the legacy systems ensuring the interoperability or integration of these systems.

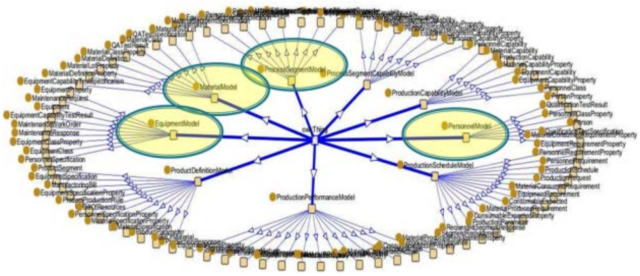

Figure.1: *Meta-ontology, based on ISO/IEC 62264*

2.3. Software tools, used in the approach

MDD guarantee that business functionality is complete and correct, end-user needs are met and program design supports requirements for scalability, robustness, security, extendibility, and other characteristics, before implementation in code. The process of gathering and analyzing an application's requirements, and incorporating them into a program design, is a complex one and the industry currently supports many methodologies that define formal procedures specifying how to go about it. One characteristic of UML - in fact, the one that enables the widespread industry support that the language enjoys - is that it is methodology-independent. And, using XMI (XML Metadata Interchange), transfer the UML model from one tool into a repository, or into another tool for refinement or the next step in the development process. The so called Object-Relational Mapping (ORM) serves as a bridge between object models, data models and relational databases to automate the transformation between object models and relational data models. The IDE Visual Paradigm 13.0 (https://www.visual-paradigm.com/) features all the UML diagrams and Entity Relational Diagrams (ERD) tools essentially in system and database design. Visual Paradigm provides a model driven platform for application development; allows not only to start from creating the models by using UML Class Diagram or Entity Relationship Diagram (ERD) to generating the executable persistence layer from the models, but also to modify the ER model which comes from reverse engineering of an existing database, transform into object model and generate persistence layer. The IDE supports a wide range of databases, including Oracle, DB2, Microsoft SQL Server, PostgreSQL, MySQL and more.

TopBraid Composer Maestro Edition (TBC-ME) (http://www.topquadrant.com/) combines world's leading semantic web modeling capabilities with the most comprehensive data conversion options and is a powerful IDE for building, managing and testing configurations of ontologies, for operations on ontologies (merging, alignment, integration, inference, etc.), for automated conversion of spreadsheets, Excel, UML and other data sources, for dynamic connections to relational databases, or running SPARQL queries over relational data.

The D2RQ Platform (http://d2rq.org/), integrated in TBC-ME is a matured system for accessing relational databases as virtual, read-only RDF graphs. It offers RDF-based access to the content of relational databases without having to replicate it into an RDF store. The system includes a SPARQL-to-SQL rewriter that can evaluate SPARQL queries over the mapped database.

3. Description of the suggested approach

The approach, suggested in this paper aims to transform, represent and integrate different data and data models into the technological space of ontology engineering for achieving interoperability between OWL ontologies, relational databases and UML models (Fig.2). At the core of the approach is the usage of reference models, based on the standard for enterprise integration ISO/IEC 62264. One of the benefits is using in a maximum degree the legacy data and data models (relational data models, UML models, Excel data sheets, documents, etc.) presenting them as RDF/OWL structures. Thus the basic advantages of the ontology based system development might be used: intelligent integration, semantic in data management, reasoning upon data and data models, reusable reference models, etc.

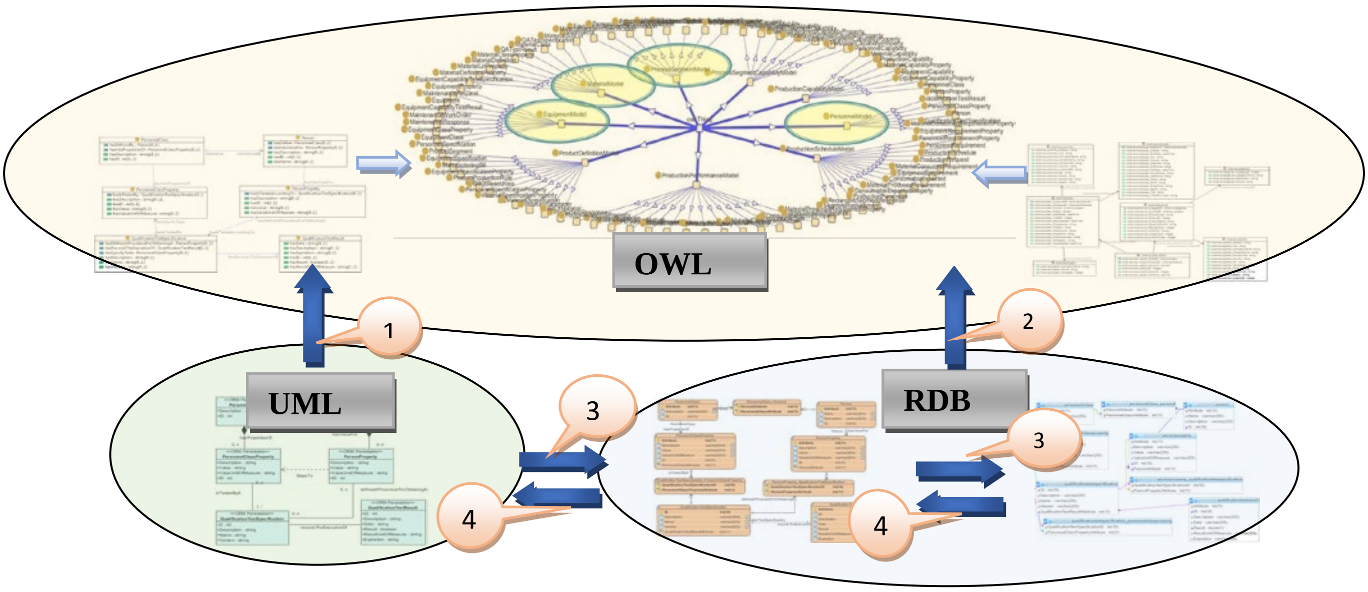

Figure.2: *Illustration of the suggested approach*

The presented approach is realized with two mutual commercial software IDEs for automatic and personalized model transformations: Visual Paradigm 13.0 for UML and ERD transformations and Top Braid Composer ME 4.2.0 used for ontology engineering. As a DBMS mySQL is used and the data connection is done with the free software tool to handle the administration of MySQL over the Web: phpMyAdmin (https://www.phpmyadmin.net/). The approach states on the usage of a reference standard based meta-ontology as a target ontology based on the models and terminology, provided by the standard ISO/IEC 62264.

4. Applicability of the suggested approach

The realized model transformations are pointed on the figures with the numbers 1-4.

4.1. UML class diagram to OWL ontology

The transformation from UML class diagram (Visual Paradigm) to OWL ontology (TopBraid Composer) is pointed with (1) on Figure 2. One of the main resource models of ISO/IEC 62264 is used as it is presented in the standard – the Personnel model. The Personnel model contains the information about specific personnel, classes of personnel, and qualifications of personnel. The *PersonnelClass* is a class to describe a grouping of persons with similar characteristics. Each *PersonnelClass* may have zero or more recognized properties. The class *Person* represents a specifically identified individual. A person may be a member of zero or more personnel classes. These specify the current property values of the person for the associated personnel property. A *qualification test specification* may be associated with a *personnel class property* or *person property*. A *qualification test result* records the results from a qualification test for a specific person. Using a model-based transformation approach, implemented in TopBraid Composer ME 4.2.0, based on SPARQL Rules, XMI model of the UML Personnel model, created in Visual Paradigm 13.0 is converted to OWL (Figure 3). The transformation is not still personalized, the names of the classes and properties are generated automatically.

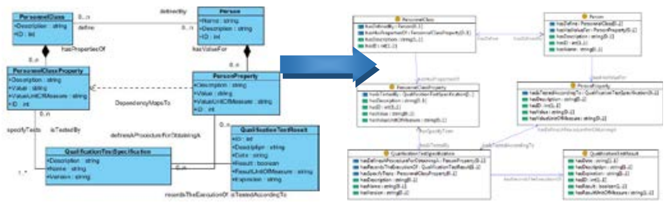

Figure.3: *Transformation from UML class diagram to OWL ontology*

4.2. Relational database to OWL ontology

The transformation of a relational database (PhpMyAdmin) to OWL ontology (TopBraid Composer) is pointed with (2) on Figure 2. The transformation is done (Figure 4) with the integrated D2RQ mapping language in the TopBraid Composer ME 4.2.0.

Figure.4: *Transformation from relational database to OWL ontology*

A sample mySQL database for Inventory/Orders/Customers system - northwind.sql is used and with D2RQ mapping language a static import of schema and dynamic import of data is done. The schema and the data from the used database might be used along with a domain ontology which can be integrated into the meta-ontology for *PersonnelModel* or *ProductionSchedulingModel*, or a suitable mapping directly to the meta-ontology might be done. SPARQL queries to the content of the relational database can be performed without having to replicate it in ontology data store.

4.3. Object-Relational Mapping

The transformation of an UML class diagram (Visual Paradigm) to ERD data model (Visual Paradigm) and to MySQL (PhpMyAdmin) is pointed with (3) on Figure 2. One of the main resource models of ISO/IEC 62264 is used as it is presented in the standard – the Personnel model. The UML Personnel model, created in Visual Paradigm 13.0 is converted to ERD physical model and a database Personnel.sql is generated. The UML class diagram, represented by the stereotype <<ORMPersistable>>, the physical ERD data model and the created relationships in the target relational database in PhpMyAdmin tool are shown on Figure 5.

The transformation of an relational database (PhpMyAdmin) to ERD data model (Visual Paradigm) and to UML class diagram (Visual Paradigm) is pointed with (4) on Figure 2. In this case, the same sample mySQL database for Inventory/Orders/Customers system - northwind.sql is used in PhpMyAdmin, the corresponding ER model is created and the new object model is created according the mapping rules. Figure 6 presents the automatic transformations; additional personalization might be done in both cases: object to relational or relational to object mappings.

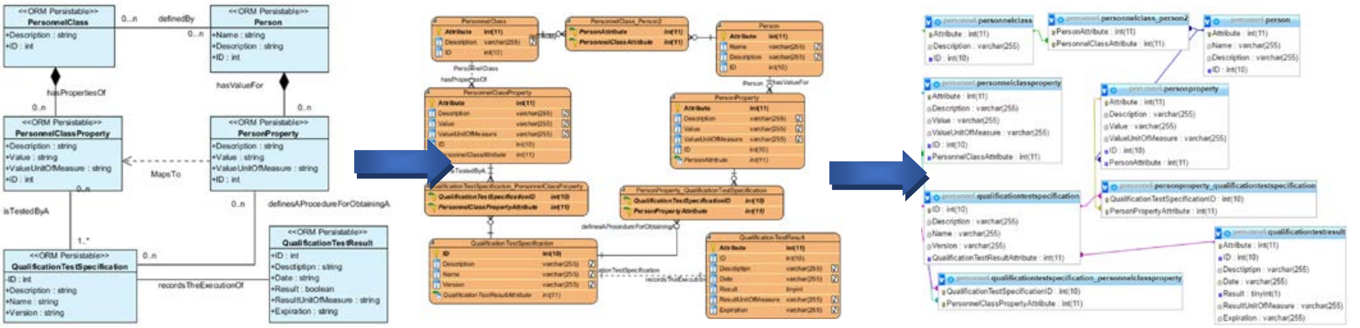

***Figure 5.**: Transformation from UML class diagram to ERD data model and to MySQL database*

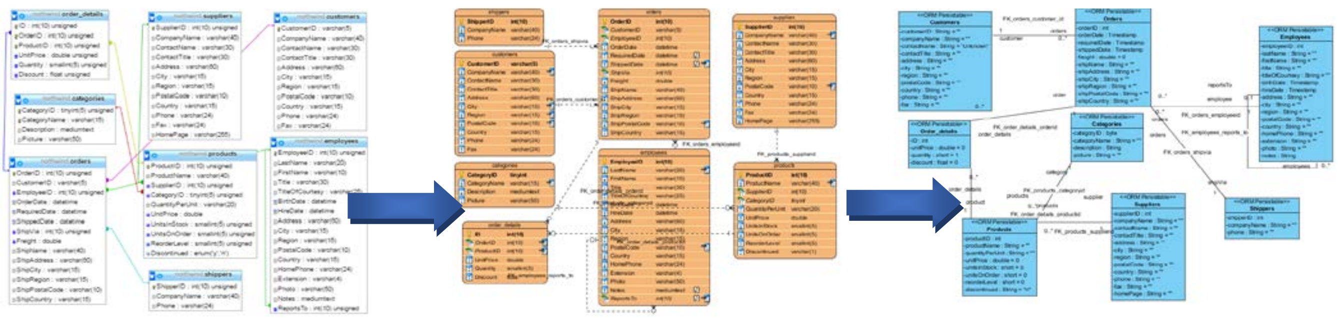

***Figure 6.**: Transformation from relational database to ERD data model and to UML class diagram*

5. Conclusions

The suggested approach, based on the joint use of standards for integration, relational databases, UML models and OWL ontologies aims to reduce the effort, cost, and errors during the design, implementation and integration of systems. Ontologies ensure easy merging, extending and sharing of heterogeneous data models, rich semantics, and reusable modules. Semantics shows enormous potential in making software systems more efficient, adaptive and intelligent. Ontology-based systems allow humans and computer systems to exchange knowledge without ambiguity and also makes it possible to use logical deduction to infer additional information from the facts stated explicitly in ontologies.

6. References

1. ISO/IEC 62264, Enterprise-control system integration, Part 1: Models and terminology, Part 2: Object model attributes, 2015.
2. K. Kiko and C. Atkinson, "A Detailed Comparison of UML and OWL," Mannheim, 2008, technical report.
3. Kurtev I., Bézivin J., Aksit M., Technological Spaces: an initial appraisal, CoopIS, DOA'2002 Federated Conferences, Industrial track, Irvine, 2002.
4. OMG (2014), Unified Modeling Language, Superstructure, Version 2.4, http://www.omg.org/spec/UML/2.4.1/Superstructure/pdf.
5. W3C OWL Working Group, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition) W3C Recommendation, 11 December 2012, http://www.w3.org/TR/owl2-syntax.
6. Zedlitz, J., & Luttenberger, N. (2014). Conceptual Modelling in UML and OWL-2. International Journal on Advances in Software, 7(1), 182-196.
7. D. Gocheva, I. Batchkova (2014), Transformation of UML class diagram to OWL ontology, International Conference "Automatics and Informatics'2014", 03-05 October, Sofia.
8. Viademonte S., Cui Z., Deriving OWL Ontologies from UML Models: an Enterprise Modelling Approach, on-line available on: http://www.researchgate/profile/Sergio_Viademonte.
9. Jay Funnell, J., Dickson, D., Chacon, J., Demystifying the Intuition Semantic Model, Control Engineering, 25 March 2012, http://www.controlengeurope.com
10. Spanos, D. E., Stavrou, P., & Mitrou, N. (2012). Bringing relational databases into the semantic web: A survey. Semantic Web, 3(2), 169-209.
11. Unbehauen, J, Hellmann, S., Auer, S., Stadler, C., Knowledge Extraction from Structured Sources, in Stefano Ceri & Marco Brambilla, ed., Search Computing Broadening Web Search, Springer, pp. 34-52, 2012.
12. Krötzsch, M., OWL 2 Profiles: An Introduction to Lightweight Ontology Languages, Proc. 8th Reasoning Web Summer School, LNCS 7487, Springer, pp. 112–183, 2012.
13. SPARQL 1.1 Overview W3C Recommendation 21 March 2013 http://www.w3.org/TR/sparql11-overview.
14. Dobrev, M., Gocheva, D., Batchkova, I., An ontological approach for planning and scheduling in primary steel production, In Proceedings of the 4-th International IEEE Conference on Intelligent Systems, Vol.1, pp.6-14: 6-19, Varna, Bulgaria, September 6-8, 2008.
15. Gocheva D, Batchkova I. (2014), Transformation of UML class diagram to OWL ontology, International Conference "Automatics and Informatics'2014", 03-05 October, Sofia.