Computer Assisted Mechanics and Engineering Sciences, **12**: 231–246, 2005.

Contemporary problems connected with including Standard for the Exchange of Product Model Data (ISO 10303 – STEP) in designing ontology using UML and XML

Wojciech Skarka
Silesian University of Technology, Department of Fundamentals of Machinery Design
Konarskiego 18a, 44-100 Gliwice, Poland

(Received March 31, 2005)

Standard for the Exchange of Product Model Data (STEP-ISO 10303) contains product information models covering most of the aspects of product lifecycle management (PLM). In designing ontology of knowledge base concerning any aspects of PLM it is necessary to use such product information models. Unfortunately, a model based on STEP is based on EXPRESS language which is not compatible with common technologies for ontology creation. The paper presents contemporary methods of ontology description and it describes the importance as well as current level of advancement in development of STEP standard. Attention has been paid to drawbacks in possibilities of already existing methods, included in STEP for ontology creation, which could form a foundation for knowledge base designing methodology founded on the defined ontologies. A method of ontology creation has been proposed, including STEP model and examples of this method application in knowledge base implementation have been given.

Keywords: knowledge base, ontology, object-oriented modeling, STEP, UML, XML

1. Introduction

Development of knowledge representation and usage of knowledge base in the whole life cycle of a product enables taking decisions consciously during its development and becomes an essential factor of fast progress. Due to this fact one can see great increase in tasks aiming at ordering computer aided techniques and methods, by means of building commonly accepted models of different knowledge domains, which could further constitute a base for building aiding applications in various processes of product life including knowledge bases. Ontologies are one of such forms of description of a given domain including vocabulary of concepts with their definitions and relations between them. In the recent years an interest has been put in knowledge bases and forms of knowledge representations which resulted in development of many incompatible computer technologies, which could be treated as methods of ontology record. Such a drawback of having many different incompatible methods of description of knowledge base is strengthened by intensive studies on building models which constitute forms of description of a given knowledge range; beginning with minor ordering models e.g. of personal address data [32] which create a base for recording and exchanging such data and up to description of data model in the whole process of product development [5]. Models defined in different forms of record which are difficult to transfer, can not be applied without problems by means of commonly used methods of model representation. In particular, such a situation deals with product data models describing basically many domains from a cycle of product development defined in STEP standard [5, 7].

2. STEP standard

ISO 10303 standard [5] is known as Standard for the Exchange of Product Model data (STEP) informally. It covers broad range and is constantly developed. It can be said that it is not a standard in a classical formulation but a modern family of standards defining methodology of description of product data throughout the lifecycle of the product. STEP is widely used in Computer Aided Design (CAD) and Product Data Management (PDM) systems. It is used in automotive, ship building and aerospace industries. Main targets of STEP standard creation, in historical context were defined at four levels of application:

1. Passive File Exchange.
2. Active File Exchange.
3. Shared Database.
4. Shared Knowledgebase.

They enable, at different levels of application, product data and knowledge exchange, sharing and archiving. For these purposes the standard has accepted a proper architecture of structure including suitable architectural components.

One of the main elements which differentiate this standard from standards used up till now is application of common mechanisms for specifying the data constructs of STEP (Fig. 1). Description methods include the formal data specification language – EXPRESS, a graphical form of the language and a mapping language for EXPRESS. The second component of architecture of STEP is implementation method series. These are standard implementation techniques for the information structures specified by the only STEP data specification intended for implementation – application protocols. Implementation methods include physical file exchange structure, the standard data access interface and its language bindings. It separates data specification from the implementation methods and enables upward and downward compatibility of implementation of STEP. The third part of STEP is conformance testing. It includes methodology describing how testing of implementations of various parts of STEP is accomplished. The fourth basic component of STEP architecture is the data specifications. All of the data specifications are documented using the description methods. There are four part series of data specifications: integrated resources, application protocols, application constructs and generic resources. According to STEP architecture data model is captured in semantic way (Fig. 2). Concept of model must include proper resources which identify the product, the product type, the version of the product, the product definition and applicable lifecycle stage. Such model ensures possibility of mapping the same requirements in different Application Protocols assuming proper mapping resource constructs specializations and constraints.

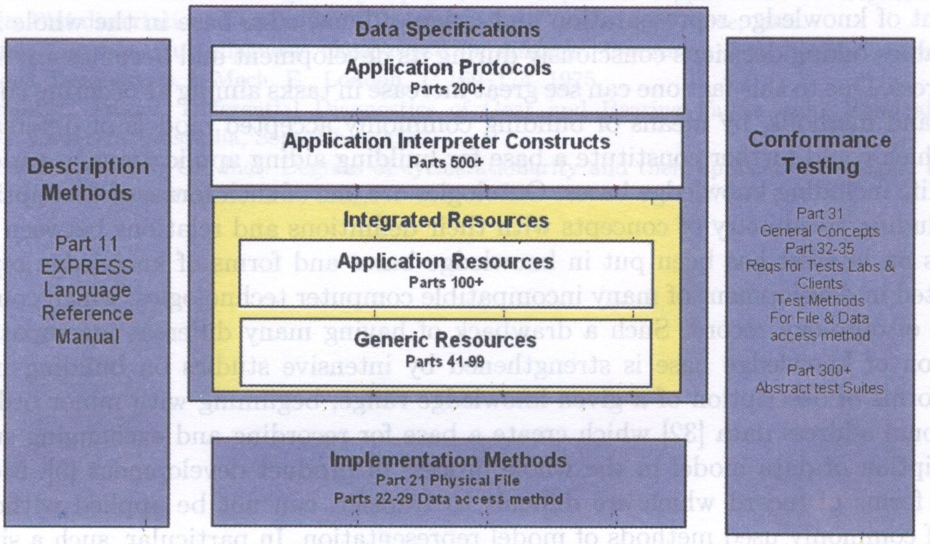

Fig. 1. General STEP Document Structure

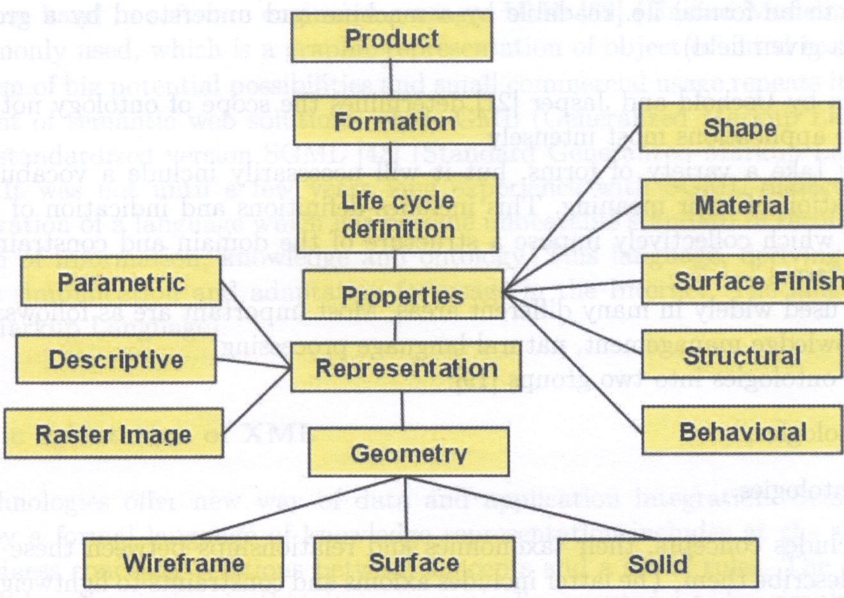

Fig. 2. Semantic STEP data model (example)

3. Ontology definition

The term ontology appears when we think of methodic approach to creation of knowledge bases. The meaning of that term in knowledge engineering has been established in the last decade [46] but it differs from its original meaning, which was taken from philosophy. In philosophy ontology describes the theory of existence and the structure of reality [34]. In knowledge engineering one of the first definitions of ontology was given in a descriptive way [11]. Ontology was determined as description of basic terms and relations including dictionary of a given domain with rules of connecting terms and relations for defining extension of this dictionary. This descriptive definition of ontology gives us guidelines on what to do in order to define ontology of a given domain.

According to that definition in order to define ontology it is necessary to define basic terms and relations between them, rules between terms, definitions of those terms and relations. Current definition of ontology in knowledge engineering applications can be divided into two groups: first one characterizes this term by defining the process of ontology construction, latter one is formed by more general description of term relating to description of phenomena and a given domain. Definition which characterizes the process of ontology construction is particularly connected with knowledge engineering because it relates also to relations between ontology and knowledge base.

According to that definition [4] in knowledge base knowledge representation is determined whereas in ontology the very domain of knowledge and relations between elements of that domain are defined. Therefore, undoubtfully ontology includes dictionary (a set of terms) and corresponding definitions. More detailed requirements for ontology construction are different and depend, most frequently, on the way of its formulating. The very definitions of terms can be determined in a formal way or in a natural language. Natural language is used while constructing or agreeing ontology between experts whereas formal languages are recommended and desired system of record when ontology has to be processed by machines. The second way depends largely on formalism of used language and is usually more difficult but necessary in knowledge base applications.

In the second group of definitions ontology is determined as specification of some conceptualization. Such specification should have some features and among them the most important are [3, 20]:

- specification should be explicit,
- it should describe formal model of some phenomena identifying terms connected with these phenomena,

- it is supposed to be formal i.e. readable by a machine and understood by a group of people (experts from a given field).

Definition given by Uschold and Jasper [21] determines the scope of ontology not only in engineering knowledge applications most intensely.

"Ontology may take a variety of forms, but it will necessarily include a vocabulary of terms and some specification of their meaning. This includes definitions and indication of how concepts are intern-related which collectively impose a structure of the domain and constrain the possible interpretations of terms."

Ontologies are used widely in many different areas. Most important are as follows: e-commerce, semantic web, knowledge management, natural language processing.

Experts divide ontologies into two groups [19]:

- lightweight ontologies,
- heavyweight ontologies.

The former includes concepts, their taxonomies and relationships between these concepts and properties which describe them. The latter includes axioms and constraints to lightweight ontologies.

4. Building ontology methods

There are many methods of ontology creation, each of them is dependable on tools which were used for that purpose. In majority of methods three main steps in ontology creation can be differentiated [6]:

- Creation.
- Extension.
- Maintenance, service and usage.

In creation process a creator must first of all fulfill purely theoretical tasks connected with conceptual defining of ontology of a given domain, which consist in formulation of a given domain, defining terms which are used and corresponding phenomena, their classification and verification. In those tasks literature research from a given field, discussions with experts, etc. are very useful. Formalization of ontology record which consists in a choice of formal representation and implementation of ontology description in a given formal language of ontology description are also very important.

Extension of ontology is done by verification of description coherence based on methods available in a given technology of description (searching, browning). During extension new vocabulary elements and definitions are added but also they can be deleted or changed.

After these modifications ontology goes into maintenance, service and usage phase when ontology record is transformed into other formal languages, depending on the aims of usage. Also during that phase there are changes and verification of the scope of ontology domain.

4.1. Standard languages for knowledge modelling

Contemporary state of art of usage and methods of ontology representation results from the development of three independent trends: knowledge representation, semantic web and software development. In those seemingly unrelated directly domains the problem of ontology is defined. In a way which is specific for a given language the problem of ontology is defined in the usage of languages of knowledge representation (KL-ONE, CLIPS, KIF) and as a result great possibilities are not followed by common usage.

On the other hand in software engineering usages UML [39] (Unified Modeling Language) language is commonly used, which is a graphic representation of object oriented techniques.

The problem of big potential possibilities and small commercial usage repeats itself in the history of development of semantic web solutions where GML (Generalized Markup Language) language and its later standardized version SGML [43] (Standard Generalized Markup Language) were not widely used. It was not until a few years long experience with SGML dialect-HTML language allowed elaboration of a language which has become unbeatable standard of record and information representation of information, knowledge and ontology. This language, deriving from SGML was formed by its simplification and adaptation for usage in the Internet. The language is XML [28] (eXtensible Markup Language).

4.2. Semantic orientation of XML

Semantic technologies offer new way of data and application integration. Semantic data model represented by a formal language of knowledge representation includes at the same time domain concepts, business concepts, relations between concepts and a set of rules. The biggest advantage of this knowledge description is that it is semantically oriented i.e. by simple formalization of description it is human and machine understandable and can be easily visualized for the purpose of ontology creation and analysis by a man and at the same time easily processed and transformed to other formalized languages.

XML (eXtensible Markup Language) is such a formal, commonly used language. The most important XML dialect is XML Schema [44] which allows planning of document structure, in other words allows knowledge structuralization of a given domain – ontology creation. The XML itself is a young standard of broad usage and it is foreseen to use it further and intensely for new semantic communication applications. Both granularization and structuralization of knowledge from a given domain allow detailed description of a given knowledge domain, forming a base of systematic knowledge record of that domain however, it does not assure effective knowledge gathering neither its usage. XML language is an extensible language of markups. Data, by means of that language, is recorded in a form of a text. Data is described by markups. Marking up is a mechanism which allows putting metainformation in a document i.e. information on data.

Markup as opposed to markuped data alone is distinguished in a special way by means of < and > mark e.g. <familyName>Brown</familyName>. Initial markup in a marked text forms <familyName>. In a description of a final mark-up back slash appears </familyName>, marked text *Brown* is between initial and final mark-up. Such a couple of markups with marked text create an element of a document. Document's elements may be of a complex structure and can be nested one in another e.g. <person> <name>John</name> <familyName>Brown</familyName> </person>.

Formal description of markups and its structure is defined in XML Schema [44]. XML Schema has been an approved standard of WWW Consortium [43] since 2001. XML Schema was created on the basis of DTD (Document Type Definition) experience and therefore it is free of majority of drawbacks of DTD. It is not the same form of schema defining; it does not complement DTD but substitutes it, creating a new quality in schema defining. From each DTD a corresponding schema can be generated however, a reverse action is not possible. At the first glance XML Schema is a more complex record which results from using syntax of XML document in XML Schema record. As a rule such schemas are much bigger and DTD syntax allowed more compressed record of analogical declarations as compared with XML Schema.

In XML Schema, similarly to DTD elements' and attributes' declarations are described. Additionally, patterns and types of contents of both elements and attributes can be also placed. XML Schema itself gives a big number of inbuilt types of data, which can be modified by constraining or extending them according to will of a document's designer. XML Schema exists independently of XML document. Connection with a given schema (schemas) is declared in XML document. The schema itself is an XML application (dialect) i.e. a language used for defining XML Schemas, derived from metalanguage XML. It is built on a Namespace specific for XML Schema, which constitutes a set of names for declaration of elements of XML document structure.

4.3. XML-based knowledge and ontology modeling languages

Some specific languages of conceptual modeling have been developed on the basis of XML technology. One of the main dialects of XML – XML Schema language which can be used to define other dialects allows ontology creation for a given domain. The XML Schema itself does not have dictionaries or extensions especially defined for the purpose of ontology creation or definition determination neither dependences between them. Dependencies between domain elements are formed by means of widely available possibilities of elements defining itself and attributes of a defined dialect. DTD (Document Type Definition) has similar properties but the syntax of this language is different from XML syntax. Specific extensions which allow ontology creation have other languages based on XML and among them the most important are:

- Topic Maps, XTM [23].
- Resource Description Framework (RDF) [36].
- DARPA Agent Markup Language + Ontology Interface Layer (DAML+OIL) [24].
- Web Ontology Language (OWL) [40].

Topic Maps is an approved ISO standard and historically it is the oldest of the above mentioned languages. Topic Maps uses SGML language for representation of network of concepts. This net is placed on knowledge sources (this can be documents of different types) it allows management of documents contents as a whole, substituting semantically sources of contents. In particular it is useful for documents which assumingly do not allow such operations on the documents - e.g. multimedia. By generalization it can be said that Topic Maps is a net representing topics which are mentioned in the documents and correlations between topics whereas XTM is XML representation of Topic Maps.

RDF (Resource Description Framework) serves for representation and processing of metadata, aiding exchange of knowledge on the Web. RDF is mainly used in description of WWW contents by means of library catalogues, world wide directories, separating news, software and contents of personal collections. RDF Schema is vocabulary description language which constitutes extension of RDF itself. It allows description of groups related resources and relationships between these resources. RDF Schema plays the same role for RDF as XML Schema for XML.

DAML+OIL is semantic markup language for Web resources. It extends RDF and RDF Schema with richer modeling features. DAML+OIL became the basis for extending possibilities of XML, RDF and RDF Schema while constructing a new language of ontology description OWL by W3C Web Ontology Working Group (WebOnt).

OWL (Web Ontology Language) is a new language which learns on experience of previously presented languages. Despite potential abilities of the language itself we will have to wait for its practice tests results since its final version has been just finished. (W3C Recommendation - 10 Feb 2004).

5. Comparison of description forms and methods of conversion of models defined in STEP standard

The STEP modeling language – EXPRESS is an independent language elaborated especially for that standard. Its own language for data description has been used which is not connected with any already existing or commonly used languages e.g. UML or XML. EXPRESS language is a modeling language combining ideas from the entity-attribute-relationship family modeling languages with object modeling concepts. It fulfills all the requirements for representation and construction of models included in STEP. Unfortunately, it does not have suitable support which makes it difficult to use in the process of application creation. UML and XML are commonly used for that purpose. In particular the latter one, due to its easy usage in Internet applications and vast possibilities of extension of usage included in the structure of the language itself, result in unquestionable priority in the process of application creation beginning with creation of data models and finishing with user's interface. As a result in some usages individual models are built based on XML vocabularies created ad-hoc which are in conflict with other applications and with STEP. Transformation of one vocabulary into the other does not create a problem when using XML language but in order to guarantee data's longevity it is favorable to rely on one model of data which ensures compatibility with many related domains, offers greater concept range and forms an open standard accepted for a bigger group of users. In this situation the usage of EXPRESS language forms an obstacle for development of knowledge base application in lifecycle of the product based on widely accepted STEP models. In order to facilitate application of STEP models defined in EXPRESS schemas ISO is developing implementation methods concerning XML - ISO 10303-28, Implementation methods: XML Schema governed representation of EXPRESS schema governed data and UML – ISO 10303-25, EXPRESS to OMG XMI binding for transforming EXPRESS schemas into XML Schemas or UML models.

6. Contemporary UML, XML and EXPRESS mapping methods

Waiting for final results of research on ISO 10303-STEP implementation methods concerning XML and UML it is necessary to deal with problems of transfer of STEP schema in the XML and UML language as well as transferring UML models to XML Schema. Until quite recently this problem has not been solved but some possibilities have appeared lately.

6.1. The STEP module repository tagged in XML

The STEP module repository forms a collection of resources tagged in XML [38] to serve as the core of a modular environment for developers of STEP, a family of product data exchange standards, and PLCS, a family of standards related to supporting the complete product life cycle.

For repository a simple definition of document type EXPRESS which enables documents generation is used. It is not suitable for transformation or models mapping for XML representation. However, repository is a full document description of modules ISO 10303 Specification.

6.2. exff (exPRESS for free)

exff is a project [37] designed to enable the integration and use of the EXPRESS language, and therefore the STEP standards, with UML-based modeling and implementation tools.

The exff goal is to be met by:

- enabling the re-use of models across many domains, and
- enabling the use of a variety of modeling and implementation systems across the industrial data and traditional software engineering communities.

Exff uses STEP Module Repository project DTD and generates UML representation through XSLT Stylesheets. It is possible to use UML and XML versions, forming a base for a format for

UML files record and enabling the usage of popular programs for designing UML models Poseidon CE [29] and ArgoUML [22].

Exff is accessible on the basis of Common Public License

6.3. XML Profile

XML profile is an extension of UML language which enables conversion of UML models to XML Schema. Reverse transformation of XML Schema to UML models is also possible. For the transformation commercial application is required [42].

While two first methods use rigidly defined models in DTD form and XSLT Stylesheets, the third method is the most flexible and allows to influence the way of mapping of UML representation elements to XML Schema. This method enables exclusively, usage of XML Schema representation as schema and does not allow the usage of Document Type Definition. It is not a serious drawback since DTD is an older form of representation of document schema than XML Schema. Additionally, DTD has much smaller possibilities of representation of data types.

Apart from these general methods of mapping (Fig. 3) it is possible to use commercially available programs which convert EXPRESS models to XML Schemas [30] or schemas which are compatible with STEP, some of them are just being created [33].

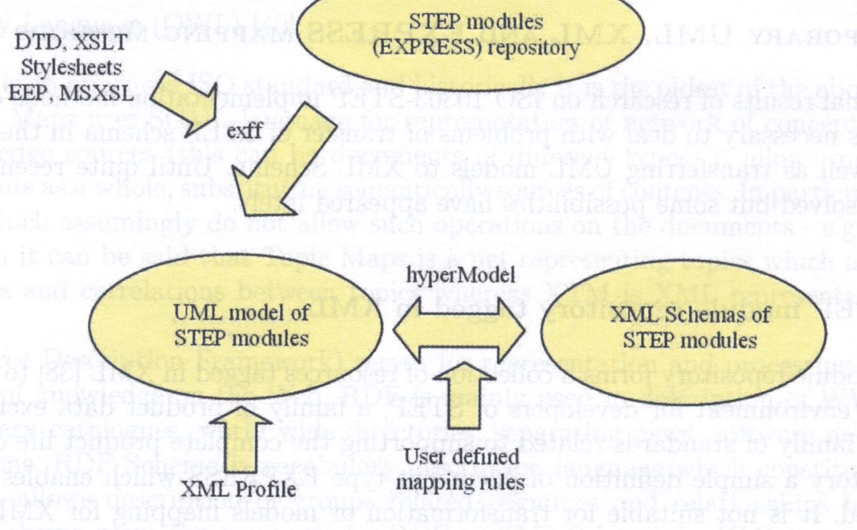

Fig. 3. Repository and the methods of STEP schemas transformation

7. Building ontology – an example

Ontology building for a given domain is a process tightly connected with a range of this domain and thus with the size of ontology description itself. For a narrow specialized domain where a new description is created this process is completely different from a domain which is described by means of different requirements. A wide range of domains which are described by means of ontologies results in the fact that almost always within such a task there are both classes of issues of ontology description. Paradoxically, an author of ontology which is not described with requirements has an easier task since he operates in the area of newly created definitions and the lack of constraints facilitates the task. Problem has to be faced when ontology is created with formally existing descriptions of a given domain. Then an author tries to assure consistence with already existing description. Informal description of a domain e.g. in a natural language does not form an obstacle here, but formalization of a form of description can result in an increase in the number of problems connected with ontology creation.

Such problems occurred when creating ontology for knowledge base designed for usage in machine designing process. In this base knowledge is gathered in a structuralized form with its roots in exact description of knowledge domains in a form of ontology. Knowledge base environment was created in two phases, the former one covered general division of the whole range of environment into autonomous and complimentary domains, and formalization of ontology for designing process and general product structure. On the basis of that ontology an application was produced serving as model application for expanding knowledge base.

This model environment was called Product Development Environment (PDE) [14, 15]. Extension of knowledge base is done by differentiating activities sensitive to knowledge in the process of product development, where knowledge can be intensively obtained or used. These activities known as scenarios [12, 16, 17], define, in a specific way, domains where it is necessary to define own ontologies. Each domain forms their own specific name space where similar notions can be understood in a different way. It prevents hypothetical conflicts of notions and definitions, which when extending ontology range, would be unavoidable. While creating ontology for PDE and applications extending the range of knowledge base, analysis has been carried of the existing requirements for ontology. Including already existing ontologies caused problems resulting from non complete or contemporary form of development of a given ontology and different forms of ontology record. While there is no cure for the former group of problems, the latter ones have a methodology elaborated which is partly presented in this paper and in full form in a book [16]. The full range of ontology for domain defined by PDE environment is presented by STEP standard [5, 7, 26, 35, 38], there are also other sources of models for the product itself as well as data connected with processes carried in the product development and related domains [30–33]. They all should be included in ontology creation for PDE environment. However, it is not feasible due to the fact that some defined models (STEP) are being created, there is no coherence of the already existing models and a form of description is different for different models which unables formal comparison of them. Generally, two basic forms of ontology description have been distinguished i.e. EXPRESS and XML languages, with the latter one having two separate forms of description Document Type Definition (DTD) and XML Schema. Finally, the following main aspects have been taken into consideration while choosing the way of ontology creation

- Possibly the easiest integration of ontology in the process of creation of the desired application of knowledge base in a form of Web Service.
- Availability of software tools.
- Possibly easy edition of ontology (graphical interface).
- Commonness of usage of the language of ontology representation.
- Easy converting from existing models (EXPRESS, DTD, XML Schema).

Although the usage of XML language and in particular XML Schema is becoming quite popular nowadays, it has been decided to use UML language for ontology creation. It is due to graphic interface of that language, designing possibilities of other application elements apart from ontology of knowledge base and greater prospects of description of ontology itself.

For ontology creation in UML language mainly Class Diagram has been used as one of 9 available diagrams. In the process of transformation of Class Diagram (UML) to XML Schema XML Profile has been used [1, 16] (Fig.4), which is a description of syntax of XML Schema language in a form of UML. Although UML language was a formal representation of ontology it was XML Profile which assure at the same time congruence and transformability of such representation to XML Schema language. The transformation itself to XML Schema format was done by transitional format XMI [42] (XML Metadata Interchange), which is typical for this transformation. It was additionally controlled by available and adjustable rules of mapping elements of UML model to XML Schema model [16], which facilitates the very process of adjustment of the final form of XML Schema.

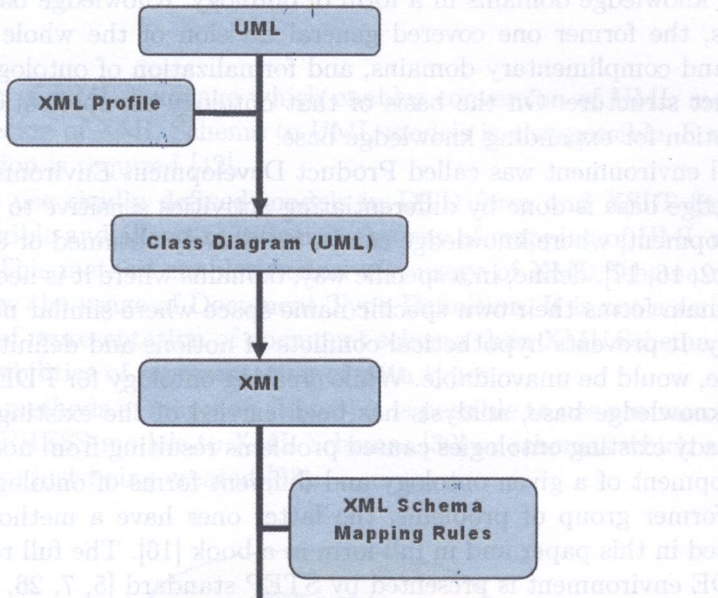

Fig. 4. Framework of ontology transformation from UML to XML Schema representation

Due to existing discrepancies in the same models, forms of record, realization of full coherence of created ontology has been abandoned and in particular with STEP standard following only available models. Coherence of data models is to be guaranteed by ultimate form of application - Web Service, which by definition allows easy exchange and conversion of data models. It results from final form of data description record in XML format and simplicity of their transformation by means of transformation language XSLT (Fig. 5).

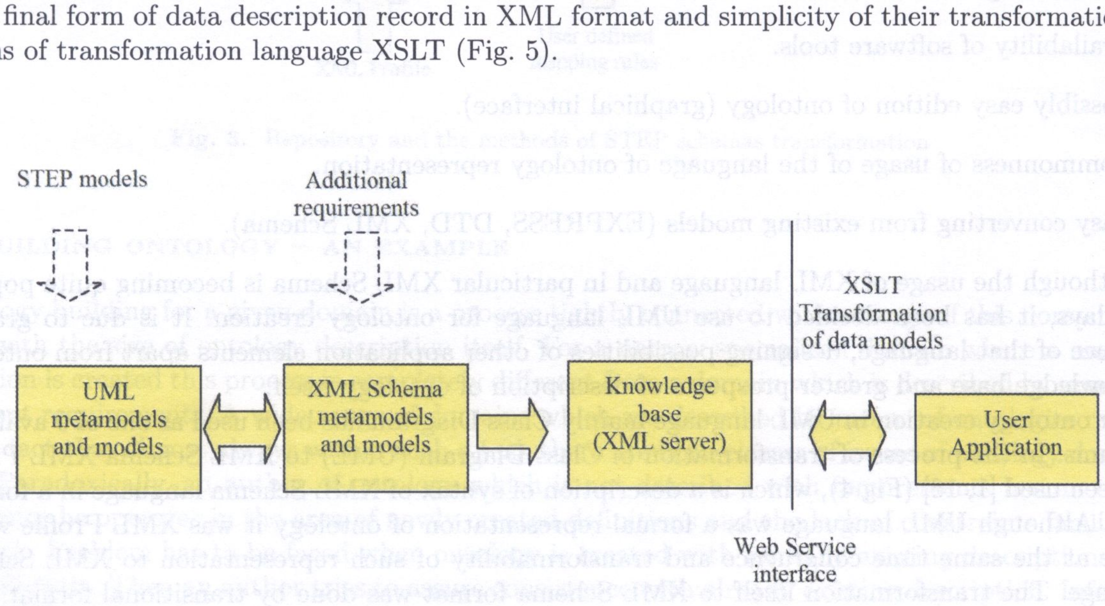

Fig. 5. Ensurance of ontology coherence with existing models

7.1. The structure of domain ontology

In the whole ontology domain formal ranges have been created, which allow its ordering in a hierarchical way. The suggested division it not a finished one and it can be supplemented freely by detailed process of already existing ranges or by defining new ranges. These ranges create separate namespaces and formally, exist as Packages in UML notation. At the highest level 5 separate packages have been distinguished (Fig. 6):

- *Structure* – structure of the product.
- *Designing* – designing process.
- *OperationandMaintenance* – operation and maintenance.
- *Manufacturing* – manufacturing information.
- *LifecycleSupport* – additional processes from product lifecycle.

These packages may have other packages in them which additionally differentiate a range in a given space.

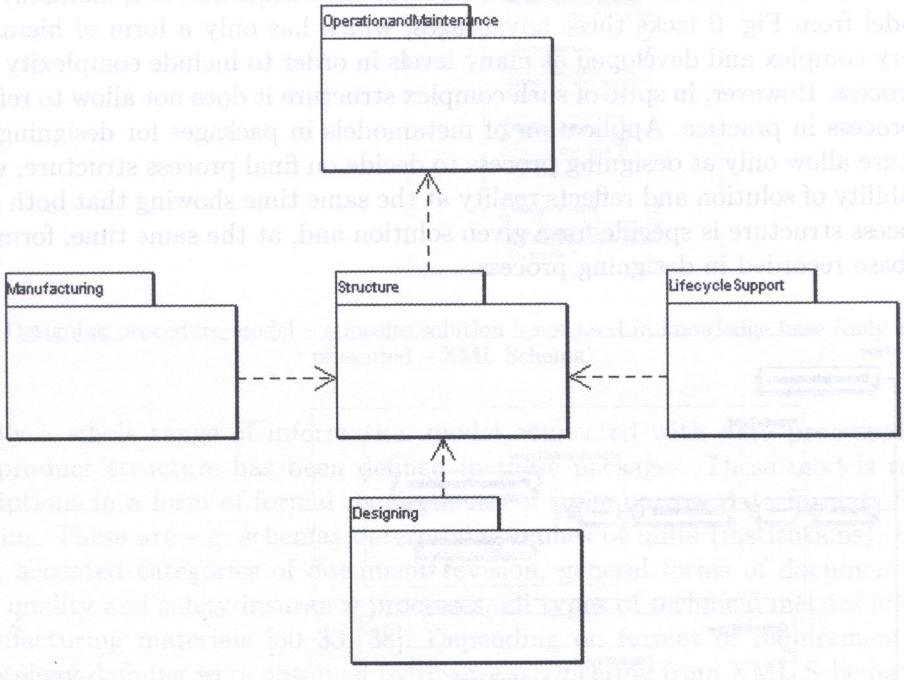

Fig. 6. Packages constituting equivalents of main scenario domains in methodology of creation [14, 15] of knowledge base and Namespaces in XML language syntax

The whole created ontology takes a form of mutually overlapping layers of net [16] where product structure defined in a form of a metamodel plays the most important role. Among available data models [2, 10], metamodel, which ensures generalization of a problem by means of forming not a structure for a given product but a general construction form of that structure allowing separate definition in the same structure metamodel of almost any product [13] (Fig. 7).

Dominant role in *Designing* package is played by designing process ontology and first of all by designing procedures. Designing metamodel (Fig. 8) has also been used for creating this model which compared with other designing models has some essential advantages. It permits defining the very structure of designing process in the next process, giving only a structural paradigm for creating

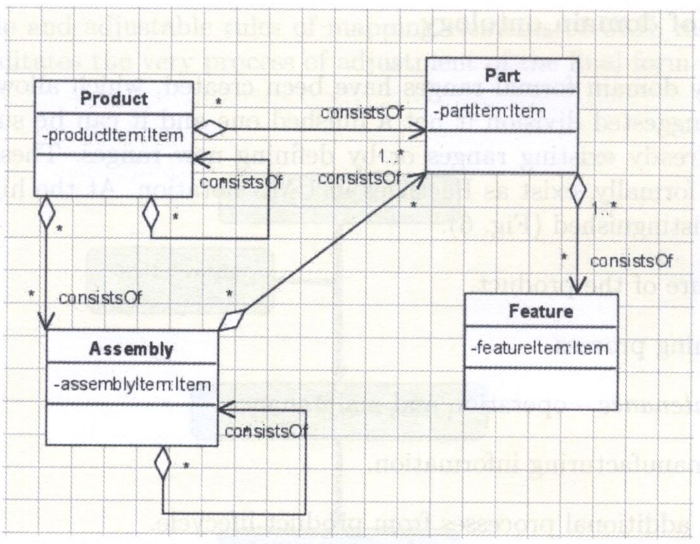

Fig. 7. A framework of product structure metamodel – example of UML model (ontology)

designing procedure. It allows creating any final structure in a sequence, net, hierarchy or any other form. The model from Fig. 9 lacks these advantages, which has only a form of hierarchy and the structure is very complex and developed at many levels in order to include complexity of structures of designing process. However, in spite of such complex structure it does not allow to reflect diversity of designing process in practice. Application of metamodels in packages for designing process and product structure allow only at designing process to decide on final process structure, which greatly improves scalability of solution and reflects reality at the same time showing that both product and, first of all, process structure is specific for a given solution and, at the same time, forms an element of knowledge base recorded in designing process.

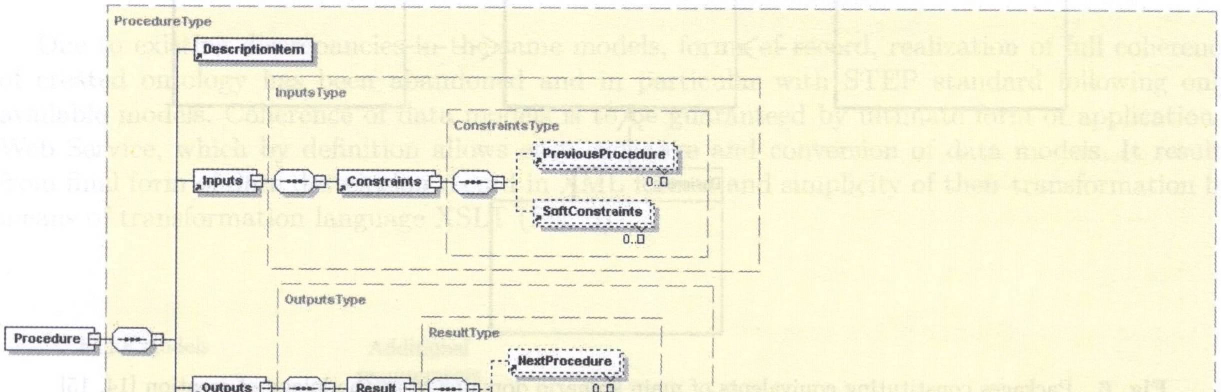

Fig. 8. Metamodel od designing procedure (XML Schema)

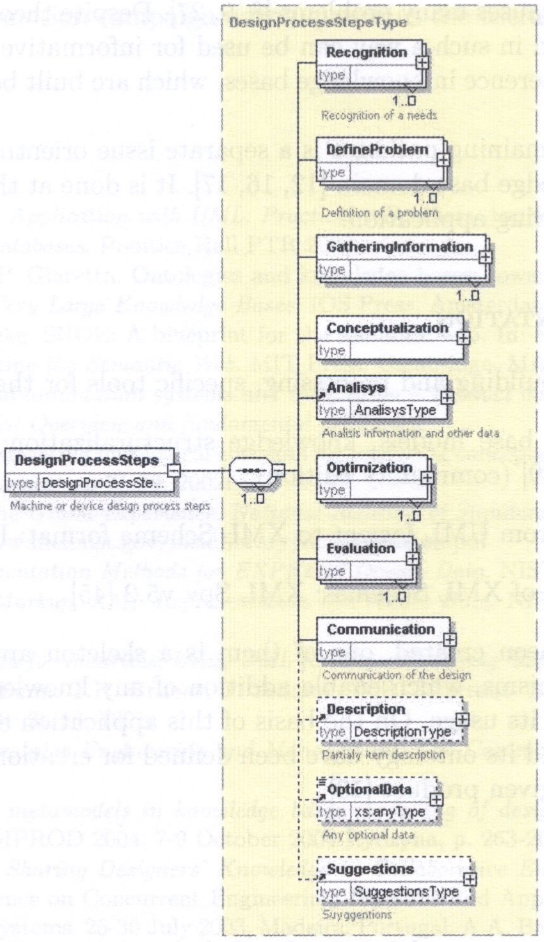

Fig. 9. Designing procedure model – opposite solution – not used in knowledge base (only root level presented – XML Schema)

Additionally a whole range of information model connected with data processed in designing process and product structure has been defined in these packages. These models reflect directly defined descriptions in a form of formal requirements of some narrow data formats from different, remote domains. These are e.g. schemas determining names of units (institutions), personal data, address data, accepted categories of document revision, general forms of document management and conduct, quality and safety insurance processes, all types of technical metrics and information for raw manufacturing materials [30–33, 38]. Depending on format of requirements description, models for ontology defining were obtained by reverse engineering from XML Schema to UML or in multistep conversion from EXPRESS to UML. In this case it is very beneficial to use models defined on the basis of STEP. Since there is no approved specification which would allow to transform model EXPRESS to UML, it is necessary to refer to repository of STEP schemas in XML format [38] and exff [37].The creation process of UML model would consist of a series of operations using different elements from STEP, exff and the following software:

- to generate the EXPRESS XML from EXPRESS text – Eurostep EXPRESS Parser (EEP) [26];
- to execute the XSLT transform of the EXPRESS XML into UML/XMI – Microsoft MSXSL [25];
- to create the compressed UML/XMI file for Poseidon CE [29] – a zip command line wrapper for WinZip [41];
- to read the UML model – Poseidon Community Edition [29].

Models' transformation causes many problems [8, 9, 27]. Despite theoretical possibilities of transformation, models generated in such a way can be used for informative purposes only, they do not assure the required data coherence in knowledge bases, which are built based on models transformed in this way.

Ontology structure of remaining packages is a separate issue orienting on given application scenario which specifies knowledge base domain [12, 16, 17]. It is done at the stage of scenario creation and constructing a given aiding application.

8. Software implementation

In the process of ontology building and processing, specific tools for that process have been used:

- Designing of knowledge base models, knowledge structuralization: ArgoUML v 0.12. [22] or Poseidon for UML 1.6 [29] (community edition).
- Models transformation from UML format to XML Schema format: Hypermodel v 1.2 [42].
- Verification and analysis of XML Schemas: XML Spy v5.2 [45].

Two applications have been created, one of them is a skeleton application [14, 15]. Skeleton application includes mechanisms, which enable addition of any knowledge domain, represented in a given way and scenario of its usage. On the basis of this application and earlier selected scenario [17] of operation, domain and its ontology have been defined for creation of knowledge base dealing with service processes of a given product [18].

9. Conclusions

The idea of STEP authors was to elaborate detailed description of domains connected with a product and its development at different phases for data processing. Unfortunately, at present form this standard fulfills its role only in a limited way. Formalized data description due to different than commonly used formalism and lack of formal conversion methods forms a big difficulty in widespread of this standard. Data models generated from STEP standard for the purpose of application creation in commonly used formats are not coherent. This standard is still being developed further and in the nearest future it should include solutions of these problems. Data models used in the standard can be successfully applied in ontology creation in knowledge bases where the assurance of coherence is obtained at the phase of data exchange between applications in a Web Services model. A clear advantage of models from STEP standard is that they allow ordering of ontologies of complex domains.

The presented way of ontology building, against possible ways of representing ontologies, is based on checked forms of representation of UML and XML. However, no specialized form of ontology representation has been used. The cause of that was the necessity of integration of ontology itself in the process of knowledge base creation. The used methodology allows easy application of special XML Profile defining syntax of XML Schema.

Furthermore, current problems connected with mapping rules of EXPRESS models to UML and XML do not allow to obtain coherent data models. Although the described process of transformation of EXPRESS models to UML and XML allows to generate such models, ways of mapping, which are not fully set or approved, of different representations do not assure data coherence. In building ontology it is recommended to use already elaborated schemas [5, 32, 33] which allow to reflect already defined and commonly accepted models.

STEP standard is one of the first ones which tries to implement data models described in a formal way to a wide usage, but the models themselves cover quite broad domains. Regardless the current problems with conversion of these data, standard implementation and usage forms a problem itself and requires many-year promotion campaign which questions the usefulness of elaboration of these requirements.

References

[1] D. Carlson. *Modeling XML Application with UML: Practical e-Business Application,* Addison-Wesley, 2001.

[2] M. Graves. *Design XML Databases.* Prentice Hall PTR 2002.

[3] N. Guarino, M. Cararara, P. Giaretta. Ontologies and knowledge bases: towards a terminological clarification. In: N. Mars eds. *Towards Very Large Knowledge Bases.* IOS Press. Amsterdam. pp. 25-32. 1995.

[4] J. Helfin, J. Hendler, S. Luke. SHOE: A blueprint for the semantic web. In: Fensel D., Hendler J., Lieberman H., Wahlster W. eds. *Spinning the Semantic Web.* MIT Press. Cambridge. MA. 2003.

[5] ISO 10303-1:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 1: Description methods: Overview and fundamental principles.*

[6] A. Kayed, R.M. Colomb. Extracting ontological concepts for tendering conceptual structures. *Data & Knowledge Engineering 40.* Elsevier Science. pp 71-89, 2002.

[7] S. Kemmerer ed. *STEP: The Grand Experience, National Institute of Standards and Technology,* NIST SP 939, July 1999 p. 185 http://www.mel.nist.gov/msidlibrary/doc/stepbook.pdf

[8] W.E. Kimber. *XML Representation Methods for EXPRESS Driven Data.* NIST 1999.

[9] J. Lubell. *From Model to Markup. XML Representation of Product Data.* NIST. Manufacturing Systems Integration Division. 2002.

[10] R.J. Muller. *Database Design for Smarties, Using UML for Data Modelling.* Morgan Kaufmann Publishing 1999.

[11] R. Neches, R.E. Fikes, T. Finin, T.R. Gruber, T. Senator, W.R. Swartout. *Enabling technology for knowledge sharing.* AI Magazine 12 (3): 36-56, 1991.

[12] A.Th. Schreiber et al. *Knowledge Engineering and Management.* The CommonKADS Methodology. The MIT Press 2000.

[13] W. Skarka. *Application of metamodels in knowledge bases modelling of designing process.* 4-th International Seminar and Workshop EDIPROD 2004. 7-9 October 2004 Rydzyna, p. 263-266.

[14] W. Skarka. *Collecting and Sharing Designers' Knowledge in Collaborative Environment.* Proceedings of 10-th ISPE International Conference on Concurrent Engineering: Research and Applications. Advanced Design, Production and Management Systems. 26-30 July 2003, Madeira, Portugal. A.A. Balkema Publishers Lisse s.265-273, 2003.

[15] W. Skarka. *Integration of product lifecycle knowledge in CAD.* e-Work and e-Business in Architecture, Engineering and Construction. ECPPM 2002 Portoż (Slovenia) 7-11 September 2002.

[16] W. Skarka. *Methodology for Building Knowledge Bases. Using UML and XML for Knowledge Bases Designing. Monograph* (in Polish) Wydawnictwo Politechniki Śląskiej, Gliwice 2005.

[17] W. Skarka. *Towards case scenarios in modelling knowledge base for designing process.* Proceedings of the Second International Computer Aided Design and Manufacturing CADAM 2004 Conference. Croatia 2004. Obsieger B. Zigo eds. Rjeka 2004.

[18] W. Skarka, G. Urbanek. *Web service for technical manuals.* Symposium on Methods of Artificial Intelligence AI-METH 2004, Gliwice 2004.

[19] M. Storga, D. Marjanović, N. Bojćetić. *Considerations on IT systems interoperability in product development.* Proceedings of TMCE 2004, April 12-16. 2004. Lausanne, Switzerland. Millpress. Rotterdam 2004.

[20] R. Studler, V.R. Benjamins, D. Fensel. *Knowledge engineering: principle and methods.* Data and Knowledge Engineering 25: 161-197, 1998.

[21] M. Ushold, R. Jasper. *A framework for understanding and classifying Ontology Application.* Proc IJCAI99 Workshop on Ontologies and Problem-Solving Methods. Stockholm, 1999.

[22] Website: ArgoUML http://ArgoUML.tigris.org

[23] Website: Contrasting Knowledge Representation in MOF, Topic Maps and RDF http://www.w3.org/2002/10/14-KR-MOF-XTM-RDF/#XmlTopicMaps

[24] Website: DAML.org http://www.daml.org

[25] Website: Download details: Command Line Transformation Utilyty (msxsl.exe) http://www.microsoft.com/download/details.aspx?familyId=2FB55371-C94E-4373-B0E9-DB4816552E41&displaylang=en

[26] Website: Eurostep – Information Solution for s Global Age http://www.eurostep.com

[27] Website: EXPRESS to UML Mapping http://exff.sourceforge.net/docs/e2u_mapping.html

[28] Website: Extensible Markup Language (XML): http://www.w3.org /XML/

[29] Website: Gentleware – Tools for developers – Home: http://www.gentleware.com

[30] Website: National Shipbuilding Research Program http://www.nsrp.org

[31] Website: Oasis: http://www.oasis-open.org/home/index.php

[32] Website: Oasis Customer Information Quality (CIQ) TC: http://www.oasis-open.org/committees/tc_home.php?wgabbrev=ciq
[33] Website: OASIS Produkt Life Cycle Suport (PLCS) TC http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=plcs
[34] Website: Knowledge Portal – Dictionaries (in Polish). http://portalwiedzy.onet.pl/polszczyzna.html
[35] Website: ProSTEP iViP Association – Homepage. http://www.prostep.org/
[36] Website: Resource Description framework/W3C Semantic Web activity. http://www.w3.org/RDF/
[37] Website: The "exPRESS for free" project http://exff.sourceforge.net
[38] Website: The STEP Module Repository http://stepmod.sourceforge.net
[39] Website: UML (Unified Modeling Language) http://www.uml.org
[40] Website: Web Ontology Language OWL. http://www.w3.org/2004/owl
[41] Website: WinZip Home Page http://www.winzip.com
[42] Website: XML Modeling: http://www.xmlmodeling.com
[43] Website: W3C World Wide Web Consortium. http://www.w3.org
[44] Website: XML Schema: http://www.w3.org/TR/NOTE-xml-schema-req
[45] Website: XML Spy: http://www.xmlspy.com
[46] K. Woestenenk. *Classification, taxonomy, ontology, what do we mean with it?* Proceedings of the 10-th ISPE International Conference on Concurrent Engineering: Research and Applications, 26-30 July 2003 Madeira, Portugal. In: Enhanced Interoperable Systems. Jardim-Goncalves R., Cha J., Steiger-Garcao A. ed. pp. 263-268.