EXPRESS to OWL morphism: making possible to enrich ISO10303 Modules

Carlos Agostinho[a,1], Moisés Dutra[b], Ricardo Jardim-Gonçalves[a], Parisa Ghodous[b], and Adolfo Steiger-Garção[c]

[a]Group for the Research in Interoperability of Systems (GRIS), UNINOVA, Portugal.
[b]Laboratory of Computer Graphics, Images and Information Systems (LIRIS), University of Lyon 1, France.
[c]Department of Electrotechnical Engineering of Fac. of Sciences and Technology of New University of Lisbon (DEE, FCT-UNL), Portugal.

Abstract. ISO10303 STEP has been acknowledged by the world's largest industrial companies, as the most important family of standards for the integration and exchange of product data under manufacturing domains. With the advent of globalization, smaller enterprises (SMEs) looking to level up with world-class competitors and raise their effectiveness are also realizing the importance of the usage of this kind of standards. However, to enable a model-based interoperability, STEP industrial standards, the Application Protocols (APs) follow a modular approach, i.e. they are composed by a set of generic purpose modules sharable by a number of different APs. This way, the core STEP reference models contain vague definitions that sometimes raise ambiguous interpretations. A possible solution to overcome this barrier would be to add further semantics to the concepts defined and enable STEP modules as ontologies, thus providing an alternative to traditional implementations. SMEs can benefit even more from this alternative, since OWL is currently a widespread technology, with abundant low cost supporting tools comparing to the ones dealing directly with STEP.

Keywords. Interoperability, ontology, transformation, content-representation language.

Introduction

Interoperability and standardization have been playing important roles in lowering costs related to production, sales and delivery processes, which permits to reduce final prices, and increase competitiveness. Enterprise and systems interoperability is frequently associated with the usage of several dedicated reference models,

[1] Group for the Research in Interoperability of Systems (GRIS), UNINOVA, Campus da Caparica, 2829-516 Caparica, Portugal; Tel: +351 21 2948365; Fax: +351 21 2957786; Email: ca@uninova.pt

covering many industrial areas and related application activities, from design phase to production and commercialization [1].

ISO10303, most commonly known as the Standard for the Exchange of Product Model Data (STEP), is one of the most important standards for representation of product information. However, despite the many success stories involving the large enterprises (i.e. from the aeronautics, ship building, automotive or aerospatiale sectors), where STEP enables estimated savings of $928 million per year, it still has some drawbacks [2]: STEP reference models are somewhat vague and contain definitions that can raise ambiguous interpretations among the industrial experts that have not been involved in the standardization process; and also, the use of languages that are unfamiliar to most application developers [1,4,5].

A solution to overcome the last problem would be to enable STEP industrial models, the Application Protocols (APs), in more user-friendly and supported technologies and standards, such as Extensible Markup Language (XML) [1,3-6]. Regarding the first drawback, an innovative approach would be to link STEP to the semantic web. It that were possible, it might be easier to reduce the misinterpretations, by associating sector specific semantics to each model.

In this paper we focus the harmonization of STEP with the Web Ontology Language, to cover both needs. OWL is an ontology language produced by the W3C Web Ontology Working Group. It is structured to be a major formalism for the design and dissemination of ontology information, particularly in the Semantic Web. OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans [7].

Because OWL is part of W3C's Semantic Web, the official exchange syntax for OWL is XML/RDF, a way of writing RDF in XML. However, since OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S (RDF Schema), it goes beyond these languages in its ability to represent machine interpretable content on the Web [7,8].

1 Model-based Interoperability: Building on Solid Knowledge

The increasing number of specialized and complementary software applications working for the industry, and specifically those covering inter-cross industrial areas, has driven industry in general, to look for standards for process and product data to support services, data exchange, and set up integration platforms to enable interoperability [9].

1.1 STEP Conceptual Models and the EXPRESS language

The standardization community is currently working hard to support the reuse, integration and extensions of already existent standards, stimulating recycling, and providing easier mechanisms to manage and understand the models. Modular approach is actually the newest activity related to the development of standards that responds to pointed needs [9,10].

STEP is nowadays one of the most important family of standards for the representation of product information. It contains more than forty APs that reflect the consolidated expertise of major industrial worldwide specialists working together for more than twenty years, covering the principal product data management areas for the main industries, e.g. oil and gas, automotive, aeronautics, aerospatiale. This kind of knowledge should not be wasted by the market, and gives STEP a distinct advantage over similar technologies and standards [1,3-6]. STEP is also one of the most innovative families of standards on the reusability sense. Application modules were recently introduced to the STEP architecture and are the key component to make its APs more interoperable, cheaper, easier to understand, manage, and quicker to develop [10].

However, the modular architecture in spite of promoting reusability raises the problem of abstracting too much the standards definitions because they stop being associated with any particular environment. Also, smaller industries like SMEs still don’t use STEP because of another problem: it is associated with technologies that lack tool support and require big initial investments [1,3-6]. The EXPRESS modelling language specified by STEP part 11 (ISO 10303-11) [11], is example of that. In spite of being a very powerful language, is not acquainted by most application developers and consequently is almost ignored by users outside the STEP community [1,3-6].

1.2 Model Morphisms (MoMo)

Another commom problem that is striking the communities working on interoperability, is the proliferation of terminology. This is a typical phenomenon that occurs when there are many different communities active on similar problems, but addressing them from different angles, with different backgrounds [12].

The InterOp network (www.interop-noe.org) intends to solve that issue, by introducing the MoMo concept, which is a terminology for model operations (i.e. mapping, merging, transformation, composition, or abstraction) independent from specific proposals. It uses the terms from consolidated mathematical areas, such as set theory, theory of functions and relations, and adapts them to the modelling context. This way, when applied to ICT purposes, a morphism details the relationship between two or more model specifications that may be described in different languages. A morphism can be classified as non-altering, if given two models (source and target), a mapping is created relating each element of the source with a correspondent element in the target, and the two models are left intact. Otherwise, the morphism will be classified as model altering, i.e. the source model is transformed applying some kind of function and generating a different output [1,5,12].

To describe unambiguously any tool that implements model morphisms, a reference ontology designated by MoMo ontology, was designed [13]. Using it, becomes possible to describe interoperability solutions related to model processing operations. Therefore, when properly instantiated, the ontology will provide a valuable knowledge-based for the MRS (MoMo recommendation system) to reason and make decisions and suggestions. Indeed, the MRS is able to assists any user in the resolution of mapping/transformation problems, by analyzing the ontology instances and recommending the most appropriate computational method(s) or tool(s) suitable for specific model morphism tasks [14].

2 Motivations for OWL

The Web Ontology Language can be the means to put way the enumerated problems that are common to formal industrial standard specifications like STEP.

OWL is used to define classes and properties as in RDFS, but in addition, it provides a rich set of constructs to create new class descriptions as logical combinations (intersections, unions, or complements) of other classes, or define value and cardinality restrictions on properties (e.g., a restriction on a class to have only one value for a particular property) [7].

It is a unique language since it is the first whose design is based on the Web architecture, i.e. it is open (non-proprietary); it uses URIs to unambiguously identify resources on the Web; it supports the linking of terms across ontologies making it possible to cross-reference and reuse information; and it has an XML syntax (RDF/XML) for easy data exchange. Semantically speaking, OWL is placed right above RDFS web stack layer (Figure 1).

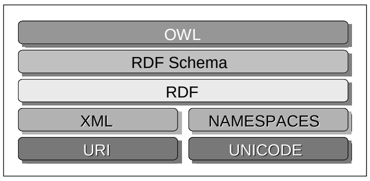


Figure 1 – Semantic Web Stack

One of the main benefits of OWL is the support for automated reasoning, and to this effect, it has a formal semantics based on Description Logics (DL). They are suitable for representing structured information about concepts, concept hierarchies and relationships between concepts. The decidability of the logic ensures that sound and complete DL reasoners can be built to check the consistency of an OWL ontology, i.e., verify whether there are any logical contradictions in the ontology axioms. Furthermore, reasoners can be used to derive inferences from the asserted information, e.g., infer whether a particular concept in an ontology is a subconcept of another, or whether a particular individual in an ontology belongs to a specific class [7,15].

3 A Proposal Mapping for EXPRESS and OWL

A first step to enable ISO 10303 industrial standards to a larger audience that can understand, reuse and implement them is to transform the EXPRESS models into OWL descriptions. The authors propose a mapping for the following EXPRESS statements (see Figure 2).

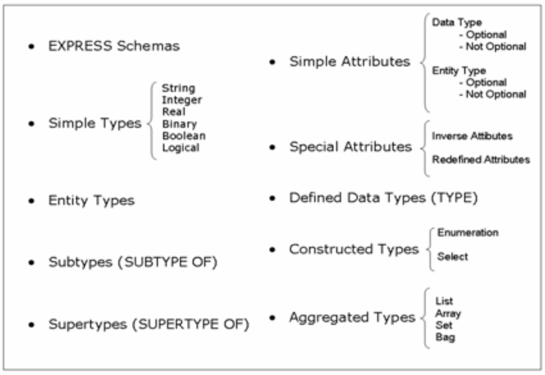


Figure 2 – EXPRESS statements mappeable to OWL

3.1 EXPRESS Schemas and Simple Types

Each OWL file represents an ontology. An OWL file header extends the RDF file header, by aggregating URIs to OWL vocabulary and to the ontology being described. In our approach, EXPRESS schemas were translated into OWL ontologies, by creating separated files to represent each one of them.

So, a typically OWL file representing an EXPRESS schema named *Fruit_schema* – which uses definitions from *Fruit_description* schema – should look like this:

```
<rdf:RDF
    xmlns="http://www.uninova.pt/ontology/Fruit_schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xml:base="http://www.uninova.pt/ontology/Fruit_schema">
   <owl:Ontology rdf:about="">
     <owl:versionInfo>1.0</owl:versionInfo>
     <rdfs:comment>Fruit_schema</rdfs:comment>
     <owl:imports rdf:resource="Fruit_description.owl"/>
   </owl:Ontology>
</rdf:RDF>
```

The conversion of EXPRESS simple types (*String, Integer, Real, Binary, Boolean, Logical*) was direct, as they have equivalents in OWL (in fact XSD types). For example, *String* type was mapped into *xsd:string* type.

3.2 Entity Types, Attributes, and Inheritance Relationships

EXPRESS entities are used to define concepts from the real world which have properties that characterize them. In the entity-relationship, they would be tables, but in OWL they are classes. By using this principle, we can map directly any entity as well as their subtypes and supertypes (profiting from OWL classes inheritance). The OWL data property is used to represent EXPRESS simple attributes, as well as the OWL object property represents named attributes. OWL cardinality is used according to the EXPRESS attribute's optional flag.

Some EXPRESS inherited attributes from a supertype entity, can be renamed or retyped according to the user's needs. These redefined attributes are mapped using OWL classes' specialization. Regarding EXPRESS inverse attributes, which are pointers to the relating entity, the OWL inverse property is available for the same purpose. Table 1 shows an example of EXPRESS entities mapped into OWL.

Table 1. Entities

EXPRESS	
ENTITY Fruit; description : OPTIONAL STRING; END_ENTITY;	ENTITY Tree SUBTYPE OF (Thing); root : Root; END_ENTITY;

OWL

```
<owl:Class rdf:about="#Fruit">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:minCardinality
 rdf:datatype=http://www.w3.org/2001/XMLSchema#int"> 0
   </owl:minCardinality>
    <owl:onProperty>
     <owl:DatatypeProperty rdf:ID="Fruit_description">
      <rdfs:domain rdf:resource="#Fruit"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
     </owl:DatatypeProperty>
    </owl:onProperty>
   </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Tree">
 <rdfs:subClassOf rdf:resource="#Thing"/>
 <rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality
 rdf:datatype="http://www.w3.org/2001/XMLSchema#int"> 1
    </owl:cardinality>
     <owl:onProperty>
       <owl:ObjectProperty rdf:ID="Tree_root">
         <rdfs:range rdf:resource="#Root"/>
       </owl:ObjectProperty>
     </owl:onProperty>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

3.3 Constructed Types

There are two kinds of constructed data types in EXPRESS: enumeration data types and select data types. The enumeration is a concept common to many other languages, and defines a set of names to be used in a domain. Regarding the select, it is a concept very characteristic of EXPRESS to define a data type that enables a choice among several named data types [11].

Enumeration and Select types were mapped through the use of OWL clauses *owl:oneOf* and *owl:unionOf*, respectively. In the Select case, it was also necessary to stand the oddity of each resulting class (see Table 2).

Table 2. Select type

EXPRESS

```
TYPE Citric_Fruit = SELECT (Orange, Lemon, Grapefruit);
END_TYPE;
```

OWL

```
<owl:Class rdf:about="#Orange">
  <owl:disjointWith rdf:resource="#Lemon"/>
  <owl:disjointWith rdf:resource="#Grapefruit"/>
</owl:Class>

<owl:Class rdf:about="#Lemon">
  <owl:disjointWith df:resource="#Orange"/>
  <owl:disjointWith df:resource="#Grapefruit"/>
</owl:Class>

<owl:Class rdf:about="#Grapefruit">
  <owl:disjointWith rdf:resource="#Orange"/>
  <owl:disjointWith rdf:resource="#Lemon"/>
</owl:Class>

<owl:Class rdf:ID="Citric_Fruit">
 <owl:equivalentClass>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Orange"/>
      <owl:Class rdf:about="#Lemon"/>
      <owl:Class rdf:about="#Grapefruit"/>
   </owl:unionOf>
  </owl:Class>
 </owl:equivalentClass>
</owl:Class>
```

3.4 Aggregated Types

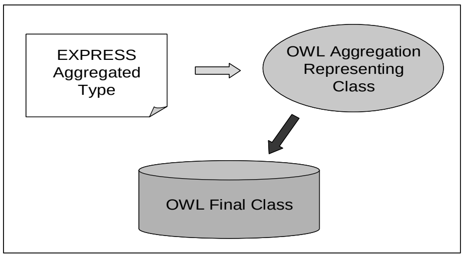
To map EXPRESS aggregated types was necessary to define an intermediate class, as there was no equivalent OWL structure to represent such types. Thus, EXPRESS *List*, *Array*, *Set* and *Bag* were first represented in an OWL metaclass, and the final class was set as subclass of this metaclass. With a new class we could easily manage properties and restrictions of each type, without changing its initial definition (see Table 3 and Figure 3).

3.5 EXPRESS Rules

The EXPRESS language contains a high level of expressiveness and uses constructs that are hard to map to other modeling languages. Among some of these constructs are rules, queries functions, and constraints to attribute values.

It was not possible to translate EXPRESS rules into OWL. Concerning the EXPRESS uniqueness rule, both languages have different ways to interpret this principle. While in EXPRESS a unique value is defined to an object and understood by all involved actors, there is no such possibility in OWL, where infinite URIs may be set to represent the same object. Moreover, as OWL is a declarative language, it is not possible to use it to represent functions statements. As well, the mapping of EXPRESS domain rules (WHERE clause) was also not possible.

Table 3. Aggregated types

EXPRESS	
TYPE Orchard = SET [1:?] OF Tree; END_TYPE;	 **Figure 3 – Aggregated types approach**

OWL

```
<owl:Class rdf:ID="OWL_Set_Tree">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="OWL_Set_belongTo_Tree"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#Tree"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#OWL_Set_belongTo_Tree"/>
      <owl:minCardinality rdf:datatype=
        "http://www.w3.org/2001/XMLSchema#int"> 1
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Orchard">
  <owl:sameAs rdf:resource="#OWL_Set_Tree"/>
</owl:Class>
```

4 UniSTEP-toolbox: EXP2OWL Morphism

To accomplish EXPRESS based model morphisms, a research prototype, i.e. the UniSTEP-toolbox, is being developed applying the principles of the OMG

MDA methodology [16]. UniSTEP relies on a framework for the interpretation of STEP models and their transformation into complementary technologies. This toolbox already includes standardized transformations to XML Schemas (XSD) and UML using its interchange format (XMI), but also relational databases, and JAVA [1,5]. The EXPRESS to OWL is the most recent morphism being implemented in the toolbox, to integrate STEP with the semantic web world.

In some cases, industry is already taking advantage of this approach of abstracting STEP to more common technologies. One example is the case of the furniture sector, mainly composed by SME's, which through the efforts of the funStep group (www.funstep.org/) and the usage of UniSTEP-toolbox, is now becoming aware, motivated and in some cases implementing the standard AP236 for the representation and exchange of furniture catalog and interior design data, using XML Schemas [17].

4.1 Formal Tool Specification

The EXP2OWL morphism of the UniSTEP-toolbox, is currently using a static mapping described on section 3 of this paper. However it is predicted that the user (i.e. the company implementing the STEP AP, or consultants for the STEP AP) can configure the tool to choose among certain mapping parameters and obtain outputs with different semantic granularity capability. This way, the tool output could be just the plain data model, or could be added with properties prepared to contain further semantics on the defined concepts. Of course, in the last case, those semantics would have to be added using a post-processor (not part of the morphism).

Adopting the mathematical notation the morphism implemented could be formally defined on the following way. Let:

a) *MOD* be the set of all models described by the EXPRESS language;
b) *MXSD* be the set of all XML models described using XSD;
c) *CONF* be the set of all possible configurations for the transformation morphism, and $CONF \subseteq MXSD$;
d) *MOWL* be the set of all OWL models described using OWL light;
e) *MOWLMap* be the set of all OWL models obtained from an EXPRESS model following the section 3 mapping, and $MOWLMap \subseteq MOWL$

This being, EXP2OWL is a function $\tau: MOD \times CONF \rightarrow MOWLMap$ where $\tau(A,C) = B$, where $A \in MOD$, $C \in CONF$, and $B \in MOWLMap$.

4.2 Using MoMo Ontology

To improve the efficiency of MRS, the more morphisms are classified, the better. Indeed, all the morphisms that are part of the UniSTEP-toolbox are classified under MoMo's reference ontology so that any user looking to work with STEP models but desires to use different technologies, could be advised to use the morphisms implemented in this toolbox.

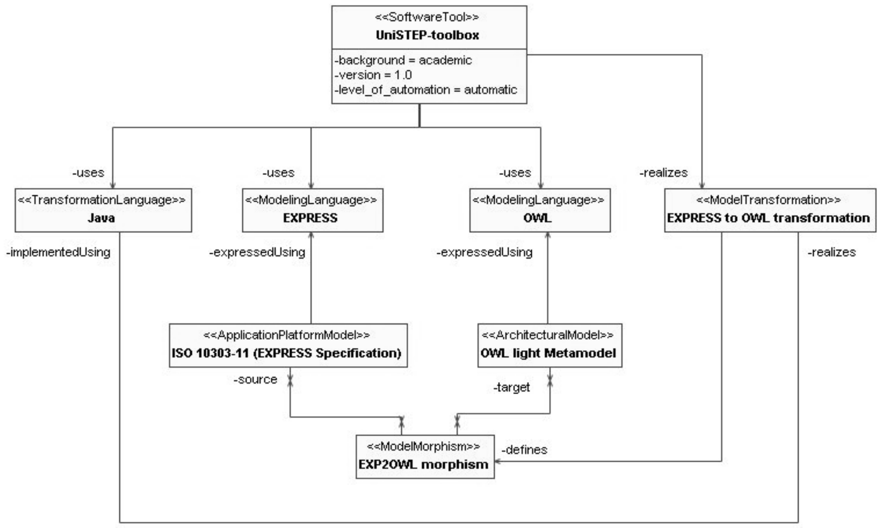


Figure 4 - Snapshot of the Instance of the MoMo Ontology for the EXP2OWL Morphism

The diagram from Figure 4 uses an UML notation to illustrate how the EXP2OWL is classified using the reference ontology for model morphisms. For the sake of simplicity, this figure does not reflect the entire set of elements of the ontology, especially some class properties. Having this classification performed, anyone (human or machine) should have only one interpretation of what the morphism is about, i.e. "UniSTEP is an automatic *SoftwareTool* with academic background that realizes an EXPRESS to OWL *ModelTransformation* implemented in the JAVA *TransformationLanguage* (is a restrictive classification for JAVA but expresses its role in this scenario). This transformation defines the *ModelMorphism* EXP2OWL which takes as input the source *ApplicationPlatformModel* represented using the EXPRESS *ModelingLanguage*, processes the *Architectural* part and generates a target model represented using the OWL *ModelingLanguage*".

5 Closing Remarks

With so many different modelling and implementation standards being used nowadays, STEP is one of the most distinguished regarding product data. To promote the reusability of its industrial standards, ISO adopted the modular approach for STEP to enable more efficient development, standardization, implementation and deployment. Compared with the classic STEP architecture, this emerging approach promises to bring major advantages for users and developers [9].

However, the modular standards may raise the problem of becoming quite hard to understand due to vague definitions not associated with any particular environment. Yet, another problems arises when the chosen product model is described using one particular technology (e.g. EXPRESS) and is required to be integrated with end-user systems that use totally different technologies with different degrees of expressiveness like XML or OWL.

The integration of the EXPRESS language with the Web Ontology Language can be the means to put way the enumerated STEP problems, since OWL provides a valuable link with the emerging field of Semantic Web which is gaining high relevance in the global market, and has XML syntax for easy data exchange using web-based systems. The Semantic Web has the aim of extending the current Web infrastructure in a way that the information is given a well defined meaning, enabling software agents and people to work in cooperation by sharing knowledge [18]. This way, if STEP standards are transformed to OWL, they could in the future, be easily complemented with links to semantic information contextualizing the scope of the defined concepts regarding the environment were they are applied.

Moreover, representing EXPRESS modules as ontologies enables the use of OWL reasoning, a very powerful way to check inconsistence and incoherence of information. This can lead us to a scenario where human users can, in an easier way, exchange and verify EXPRESS represented data. Such scenario can enhance the use of EXPRESS language, promoting its adoption by a large number of platform-independent and language-independent users. The morphism developed is also part of a collaborative design project, described in [19].

The different degrees of expressiveness of the referred languages impede a full binding (e.g. EXPRESS rules), thus originating a partial morphism. In this case, the morphism results in the loss of some information. This way if a user needs to transform an EXPRESS model into XML based languages, namely OWL, without losing much information, it probably should combine more than one technique and tool. This combination could be suggested in an automatic way by the MRS that reasons on knowledge-base provided by the MoMo reference ontology.

6 Acknowledgments

The authors would like to thank all the organizations supporting the international projects that resulted in the development of the UniSTEP-Toolbox. Namely, the funStep community (www.funstep.org) and its members that somehow contributed for the presentation of this work; Also, CEN/ISSS and ISO TC184/SC4 for the effort in developing industrial standards and cooperating with the presented research.

7 References

[1] Agostinho, C., Delgado, M., Steiger-Garção, A., and Jardim-Gonçalves, R., (2006), 'Enabling Adoption of STEP Standards Through the Use of Popular Technologies', in proceedings of 13th ISPE International Conference on Concurrent Engineering (CE 2006), Juan-Les-Pins, France, September 2006.

[2] White W. J., O'Connor A. C., Rowe B. R., (2004) Economic Impact of Inadequate Infrastructure for Supply Chain Integration. National Institute of Standards & Technology (NIST), Planning Report 04-2.

[3] Delgado, M., Agostinho, C., and Jardim-Gonçalves, R. (2006), 'Taking advantage of STEP, MDA, and SOA to push SMEs towards a Single European Information Space', in proceedings of eChallenges e2006, Barcelona, Spain, October 2006.
[4] Jardim-Gonçalves, R., Agostinho, C., Malo, P., and Steiger-Garção, A., (2005) AP236-XML: a framework for integration and harmonization of STEP Application Protocols. Proc. of IDETC/CIE'05, ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; Long Beach, California, USA, Sep 24- 28, 2005.
[5] Delgado M., Agostinho C., Malo P., Jardim-Gonçalves R., (2006) A framework for STEP-based harmonization of conceptual models. 3rd International IEEE Conference on Intelligent Systems (IEEE-IS'06), Wetsminster, Jul 4-6 , 2006.
[6] Lubell J., Peak R. S., Srinivasan V., and Waterbury S. C., (2004) STEP, XML, AND UML: COMPLEMENTARY TECHNOLOGIES. Proc. of DETC 2004, ASME 2004 Design Engineering - Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, Utah USA, Sep 28- Oct 2, 2004.
[7] OWL Web Ontology Language Overview. Available at: http://www.w3.org/TR/owl-features/. Accessed on: April 2007.
[8] Semantic Web, http://www.w3.org/2001/sw, Accessed on: April 2007.
[9] Jardim-Gonçalves, R., Olavo, R., Steiger-Garção, A., (2003) 'Modular Application Protocol for Advances in Interoperable Manufacturing Environments in SMEs' in proceedings of 10th ISPE International Conference on Concurrent Engineering (CE 2003), Madeira, Portugal, July 2003.
[10] Feeney, A. B., "The STEP modular architecture", Trans. ASME: J. Comput. Inform. Sci. Eng., vol. 2, pp. 132–135, Jun. 2002.
[11] ISO10303-11 (2003), Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.
[12] InterOP NOE consortium, (2005) Deliverable DTG3.2: TG MoMo Roadmap.
[13] D'Antonio F., Missikoff M., Bottoni P., Hahn A., Hausmann K., (2006) An ontology for describing model mapping/transformation tools and methodologies: the MoMo ontology. EMOI'06 - Open InterOP Workshop on Enterprise modelling and Ontologies for Interoperability, Luxemburg, June 2006.
[14] InterOP NOE consortium, (2006) Deliverable DTG3.3: Toolbox definition and Workshop report.
[15] Patel-Schneider, P. F. (2004), 'What is OWL (and why should I care)?', invited paper for the Ninth International Conference on the Principles of Knowledge Representation and Reasoning, Whistler, Canada, June 2004
[16] OMG Model Drivel Architectures (2006). www.omg.org/mda/
[17] ISO 10303-236 (2006), 'Industrial automation systems and integration — Product data representation and exchange — Application protocol: Furniture catalog and interior design'.
[18] Berners-Lee, T., Hendler, J. and Lassila O. (2001), 'The Semantic Web, a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities', Scientific American, May, 2001.
[19] Dutra M, Slimani K, Ghodous P (2007) *A Distributed and Synchronous Environment for Collaborative Work*, submitted to the Integrated Computer-Aided Engineering journal (ICAE), ISSN 1069-2509, vol. 14, IOS Press, Amsterdam, The Netherlands, 2007.