Meta-Modelling for Interoperability in Product Design

F.-L. Krause (1), U. Kaufmann
Fraunhofer-Institute for Production Systems and Design Technology, Berlin, Germany

Abstract
Interoperability of tools for product design relies on the utilization of common information models like ISO 10303 STEP. However STEP is based on the EXPRESS modelling language which is not supported by state-of-the-art software development tools. To reuse the huge repository of information models defined in the STEP Application Protocols transformations into other representations are needed. A generic and sound approach is provided by OMG’s Model Driven Architecture (MDA) and Meta Object Facility (MOF). This paper describes meta-modelling methods and tools for the coexistence and integration of EXPRESS with UML, the commonly used modelling language in modern software development environments.

Keywords:
Information, Modelling, Integration

1 INTRODUCTION
Integrating tools for a continuous product design process is a prerequisite for an effective development of innovative new products [1]. System integration and interoperability of tools for product design is tightly coupled with the existence and the evolution of standards. ISO 10303, also known as STEP (STandard for the Exchange of Product model data) [2], was primarily developed with the purpose of a vendor neutral exchange format of CAD data, but has evolved into a well accepted standard for the exchange of (meta-) data in all relevant areas of product design and manufacturing [3, 4].

STEP and the corresponding modelling language EXPRESS created a microcosm of software development and modelling tools which are used by experts in the domain of Product Lifecycle Management (PLM) systems. The need for integration with IT systems in other business areas, where different software development tools are in use, calls for interoperability of software systems between both worlds.

This integration issue can be considered mission critical for the whole STEP project and leads to the question how STEP can evolve and coexist with new emerging technologies. In this paper we show how the Model Driven Architecture (MDA) and Meta Object Facility (MOF) [5, 6] frameworks of the Object Management Group (OMG) can be applied to conciliate between the STEP product modelling world and the UML modelling world that is used in the software development domain.

2 META-MODELLING AND INTEROPERABILITY
Modelling is the process to provide formal descriptions of real-world elements through abstractions. Human languages may be used to describe abstractions but usually dedicated languages are used to provide more precise semantics of models. Meta-modelling is about the model of the models and is most commonly referred to the modelling language (Fig. 1).

EXPRESS [7] and UML [8, 9] are two widely used modelling languages. While EXPRESS is being in use since the late 80’s for the definition of STEP specifications, UML evolved in the mid 90’s as the de facto standard modelling language in the software development domain. Originally UML was used as a documentation language for large software systems. Now the potential of UML lies more in the support of Model Driven Development, a Software Engineering concept to put emphasis on the model of the system in development and to generate code for implementation more or less automatically.

OMG’s Model Driven Architecture (MDA) was designed to give a sound theoretic and methodological framework of code generation from UML models. But the capabilities of MDA are going beyond only supporting software engineering processes. Today MDA is also used to describe abstraction in other modelling domains like Business Process Modelling and Systems Modelling.

To describe abstraction MDA introduces the term ‘platform’. From a code generation perspective a platform is a specific programming language, like C++ or Java. A Platform Independent Model (PIM), usually in UML representation, in this regard abstracts from the implementation method – the platform, called the Platform Specific Model (PSM). For example, from a PLM Services interface specification (see 4.2), a C++, XML Schema or a Web Services implementation can be derived (see Fig. 1). The arrows in Fig. 1 have the semantic of ‘depends-on’ or ‘is-instance-of’. In other words, from a MDA viewpoint the C++ implementation is considered an ‘instance’ of the PLM Services specifications. The definition of an explicit mapping between the PIM and the PSM is a core element of MDA.
3 NEW APPROACH FOR THE MAPPING OF EXPRESS TO UML

3.1 Motivation

The general mapping of EXPRESS to UML is described in ISO 10303 Part 25: EXPRESS to XMI binding [11]. The scope of the Part 25 mapping was to define a one-way translation of EXPRESS constructs covering mostly the EXPRESS-G subset of the language. Therefore a number of EXPRESS constructs were excluded from mapping (see 3.3).

The nature of the Part 25 specification being a non-formal description of the mapping using natural language implies some ambiguities. Other weaknesses of this mapping are already identified in the specification. For example it suffers from deficiencies of UML 1 which was the existing specification of the time when Part 25 was finished.

Nevertheless Part 25 has to be considered the only normative specification of the EXPRESS to UML mapping which all implementations have to conform to. So far a number of implementations were developed, though the underlying transformation is a black box lacking flexibility and adaptability.

This was motivation to think about developing a, so to say, meta-model of the mapping. The necessary toolset is given by the maturing OMG standards supporting the MDA. The basic idea of the new approach is to utilize the capabilities of the Meta Object Facility (MOF) for the definition of the mapping between EXPRESS and UML (see Fig. 2). The newly developed QVT (Query View Transformations) language enables to define this mapping based on the meta-models of the EXPRESS and UML languages.

Advantages of this MDA-based approach for model transformations are:

- use a formalised transformation language used also in MDA tools
- transformation is no black box and hence adaptable (domain specific)
- Independency from model representation using XMI (depends only on meta-models)

The meta-model for the EXPRESS information modelling language was elaborated at NIST (National Institute of Standards and Technology) [12].

3.2 EXPRESS to UML transformation

In order to define a standards based way for the transformation of models between EXPRESS and UML we chose QVT as the underlying methodology. Unfortunately QVT is a very recent development and therefore no commercially available tools implement this standard for the time being.

We therefore had to look for a model transformation tool that comes as close as possible to the QVT standard. After studying different existing model transformation tools we finally selected the ATL (Atlas Transformation Language) that was developed at University of Nantes [13, 14, 15].

3.3 Comparison of EXPRESS and UML from a modelling point of view

EXPRESS and UML evolved from different modelling requirements. Hence their capabilities differ in a number of ways.

One of the main differences is that EXPRESS was primarily defined in a textual notation thus giving it a precise semantic. The graphical notation EXPRESS-G
was later developed as a subset of EXPRESS to give the users easier ways to create models. From a perceptual viewpoint EXPRESS-G is comparable to UML. This might explain the rationale of the Part 25 which mainly covers the mapping of artefacts which have a graphical representation. It is a known and often criticised fact that such graphical notation is ambiguous. As a consequence any mapping from EXPRESS to UML cannot be free of ambiguity.

<table>
<thead>
<tr>
<th>EXPRESS</th>
<th>UML 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Package within a UML Model</td>
</tr>
<tr>
<td>EntityType</td>
<td>Class</td>
</tr>
<tr>
<td>SUBTYPE OF</td>
<td>Generalization, multiple inheritance supported</td>
</tr>
<tr>
<td>single or SET ExpliciteAttribute of SimpleType</td>
<td>Attribute of corresp. UML Class with corresp. multiplicities</td>
</tr>
<tr>
<td>single or SET ExpliciteAttribute of EntityType</td>
<td>Association between corresp. UML Classes with corresp. multiplicities</td>
</tr>
<tr>
<td>DerivedAttribute of all kinds</td>
<td>same as ExplicitAttribute except &quot;/name&quot;</td>
</tr>
<tr>
<td>SelectType</td>
<td>abstract Class, generalization of all UML Classes for select items</td>
</tr>
<tr>
<td>EnumerationType</td>
<td>abstract Class with Stereotype &lt;&lt;enumeration&gt;&gt;</td>
</tr>
<tr>
<td>EnumerationItem</td>
<td>Attribute of Enumeration Class with Stereotype &lt;&lt;enumeration_item&gt;&gt;</td>
</tr>
<tr>
<td>SpecializedType</td>
<td>Class which specializes the underlying ConcreteType</td>
</tr>
<tr>
<td>SimpleTypes (String, Boolean, etc.)</td>
<td>DataTypes</td>
</tr>
</tbody>
</table>

Table 1: Mapping according to ISO 10303 Part 25

Table 1 shows an extract of the corresponding modelling artefacts in EXPRESS and UML as defined by ISO 10303 Part 25 [11]. Due to a lack of similar constructs in the UML notation a number of EXPRESS concepts are not mapped, although Part 25 identifies the OMG Object Constraint Language (OCL) as a suitable target:

- RULE declarations;
- domain rules in ENTITY or TYPE declarations;
- UNIQUE rules in ENTITY declarations;
- SUPERTYPE declarations other than ABSTRACT SUPERTYPE and complete ONEOF constraints;
- FUNCTION declarations;
- PROCEDURE declarations;
- CONSTANT declarations;
- OPTIONAL constraints;
- explicit attributes redeclared as derived attributes;
- remarks (comments).

Part 25 relies on UML 1.5 which had known shortcomings. UML 2 was adopted by OMG in 2003. It offers much potential for a more comprehensive and precise mapping of EXPRESS to UML. Especially the integration of OCL in the language definition and the MOF 2 based XMI representation of UML gives the basis to define the mapping of the so far unmapped constructs.

### 4 EXAMPLE AND USE CASE

#### 4.1 Simple example

![Fig. 3: EXPRESS-G representation of an entity product](image)

![Fig. 4: UML diagram of the example from Fig. 3](image)
The example shown here is a very simple case of an EXPRESS model that was translated to UML. The intent is to illustrate the connection between EXPRESS and UML constructs and to show that the transformation actually works. EXPRESS-G (see Fig. 3) is the graphical representation of the EXPRESS textual encoding which is used here for better illustration.

The resulting ATL translation for this example is shown in Fig. 4, represented as an UML diagram. It clearly demonstrates the preservation of all relations (SUBTYPE, associations), as well as the multiplicities of these relations. The dashed line in the EXPRESS-G diagram has the semantic of an OPTIONAL attribute which translates to an UML array of length [0..1], meaning that the attribute disappears in case of array length 0.

4.2 PLM Services specification as particular application

PLM Services is a MDA based specification under development with the objective of supporting interoperability between PLM (Product Lifecycle Management) systems. PLM Services V1.0 [16] is already a formally adopted OMG specification while PLM Services V2.0 is currently approaching its final status.

One important mandatory requirement that the PLM Services specification has to fulfill is the conformance to the STEP AP 214 [17]. STEP AP 214 is the application protocol for core data for automotive mechanical design processes and consists in the most recent version of 3550 pages and defines some hundred EXPRESS entities.

Conformance between PLM Services and AP 214 apparently demands a semantic equivalence of the EXPRESS models with UML models specified in the PLM Services standard. Consequently this calls for an accurate EXPRESS to UML transformation.

For the time being the UML models for the PLM Services specification were derived semi-automatic and then remodelled using a UML modelling tool. Besides the risk of errors due to human involvement this approach introduces modelling ambiguities which might cause incompatibilities in later implementations.

The transformation described in this paper is currently used to verify and to improve the semantic consistency of the underlying data models. Since this is an ongoing project results might be expected soon.

5 CONCLUSION

Maintaining the consistency between two standards as shown above is not just a static problem. Both ISO and OMG standards will evolve over time and synchronization of the two specifications will become an issue.

We see great potential to support this process by the method described in this paper. A manual maintenance of consistency is known to be fault-prone and therefore calls for a more automatic way.

Future work to extend the described methodology will include:

- Extensions of mapping to capture UML 2 capabilities
- Support for behavioural constructs of EXPRESS (Functions, Procedures, Algorithms)
- Reverse mapping EXPRESS to UML
- Consideration of other modelling languages as a target for the mapping, e.g. SysML

6 ACKNOWLEDGMENTS

The work described in this paper was partly supported by the ATHENA project, a European Commission funded project in the 6th Framework Programme (www.athena-ip.org). Moreover the authors would like to thank participants in the MEXICO initiative, especially NIST and Eurostep who contributed the EXPRESS meta-model and the EXPRESS parser. Results shown in this paper are based on work in an international collaboration of members of OMG’s Manufacturing Technology and Industrial Systems Domain Task Force (ManTIS DTF) and ISO’s TC184/SC4.

7 REFERENCES
