4

IMPLICIT HIERARCHICAL META-MODELING IN SEARCH OF FLEXIBLE INTER-OPERABILITY FOR MANUFACTURING AND BUSINESS SYSTEMS

Ricardo Jardim-Gonçalves and Adolfo Steiger-Garção
Dep. de Eng. Electrotécnica e Computadores da Fac. de Ciências e Tecnologia da Univ. Nova de Lisboa, UNINOVA., rg@uninova.pt and asg@uninova.pt

There are numerous proposals worldwide to represent data models and services for the main business and manufacturing activities. This paper suggests an architecture and methodology for the design and development of new integrated models, extending the use of existent standard-based protocols as a basis for the development of implicit models, avoiding "yet another model". These models are built on top of existent components, supported by a hierarchical architecture developed at a model's meta-level and offering several degrees of flexibility. The work results from the Research and Development done by the authors during the last years under the umbrella of a cluster of R&D international projects.

1. INTRODUCTION

When searching for integrated product and services life cycle, enterprises are facing a major problem regarding the explosion of the number of heterogeneous interfaces and data models the software applications need to handle (Ducroux, 1999) (Cofurn, 2001) (ATLAS, 1995) (e-Construct, 2000) (PDES Inc., 2002) (SUMMIT, 1998) (Jardim-Goncalves, 2001a). To have all software applications integrated and achieve compatibility in interfaces and data, it is required that each application develop one dedicated translator for each other application not compatible it would like to operate. This is a very complex situation considering the effort required to develop each translator and the unpredictable number of incompatible platforms that could exist (Camarinha-Matos, 1999) (Davalcu, 1999) (Poyet, 1999) (VEGA, 1999) (Umar, 1999) (Jardim-Goncalves, 2001b).

The scenario to seek out is the one where all applications could be easily integrated independently of the platform in use by each application as if all platforms were equal, interoperating in flexible and configurable enterprise environments. This could conduct to an open platform able to support full integration of systems, promoting the adequate use of the multiple existent and emerging standards like Application Protocols (APs) and Business Objects (BOs),

V. Mařík et al. (eds.), *Knowledge and Technology Integration in Production and Services*

providing an adequate environment for integrated modeling (Arsanjari, 1999) (Radeke, 1999) (Lazcano, 2000) (Nayak, 2001) (Chen, 2000) (Jardim-Goncalves, 2002a).

The manufacturing enterprises have identified this problem even bigger, regarding the large and varied number of product and business life cycle activities they have to support and integrate. In this scenario, a possibility could be to search for a unique standard model that covers the complete spectrum of an enterprise needs. However, this solution is not realistic, and a balanced approach needs to be found (West, 2001) (Alonso, 1999) (CECOM, 2001) (Clements, 1997).

Today, several APs and dedicated models have been developed to cover the main industrial application activities, from design to production and business. Most of these models were designed and developed using standard methodologies and techniques, and some of them are registered as International Standards, e.g., ISO10303 STEP APs (ISO10303, 2002) (SOAP, 2001). Others, although not developed directly under the umbrella of International Organizations for Standardization, were developed by international consortia and associations, like W3C, and due to its impact in the real world and large acceptance by the users, they are broadly in use and considered *de facto* standards ("standards and *de facto* standards" will be referenced hereinafter by "standards").

An example is ebXML (ebXML, 2001). ebXML is a modular suite of specifications that enables enterprises to conduct business over the Internet. Using ebXML, companies can have a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes. One of the technical foundations of ebXML is the Extensible Markup Language (XML) (XML, 2001) that allows parties to exchange structured data.

In spite of the large number of existent and emerging standards for enterprise data exchange, most of them have been developed using divergent methodologies and without concern to have their models interoperable with other standards, in the same or complementary scopes of usage (Jardim-Goncalves, 2000). Since most of these standards were developed in an isolated way, there is not any global plan to be interoperable with others. And to achieve it, it is needed to develop additional methodologies to support the integration of these models. Some international research projects are in course to help in this aim, as is the case of the European IST-2001-37368 project IDEAS: Interoperability Developments for Enterprise Application and Software - Roadmaps.

However, companies have to conduct habitually operations embracing horizontally several specialized domains, making necessary their applications to interoperate with others vertically specialized in each field. A challenge could be to reuse the existing standard models, finding and selecting reference models for each range of activities. Then, for the set of applications to be integrated and interoperable according to the scope of activities, to develop and implement an integrated model through harmonization and mapping based on the adoption and extension of the selected standard models in each application's platform (AP236, 2002).

This approach promotes the reuse of the existent application protocols and business objects, through a methodology that saves all the effort experts already spent when developing the standard models in reuse, admitting flexibility in the construction of new integrated models and avoid "yet another model".

This is a very important aim in face of the rapidly changing business and manufacturing requirements, and the scientific community researches looking for further proposals that could enable the immediate reuse of the available standardized models. The objective is to propose a framework for the implementation of an open business-oriented integrated platform, where Product Life Cycle activities and services can take place between trading parties using different platforms, in the same way as using the same platform (CECOM, 2001)(West, 2001)(Jardim-Goncalves, 1999).

This paper presents part of the Research and Development done by the authors during the last years under the umbrella of a cluster of international projects (i.e., ECOS, FSIG, Cofurn, funStep AP-DIS, prodAEC) resulted in a set of prototypes and in a framework to develop meta-models on top of standardized existing components (FSIG, 2002) (prodAEC, 2002). This proposal uses an inter-related meta-modeling mechanism as the basis for the design and development of new integrated components, appearing as an implicit model built on top of existent ones, and is part of the core of the first author's PhD thesis.

2. FRAMEWORK FOR META-STANDARD APs

Nowadays, a key question when searching for a model to be used as the support for an integration task is about the languages that describe and implement the model.

For instance, ISO10303 developed and has in development 38 APs, covering a large scope of the industrial needs. These APs are described following the STEP methodology (STEP-1, 1994), and its Application Schemas described using the EXPRESS language (STEP-11, 1998). Other standards have developed normative models for data exchange, e.g., VdDK, ebXML registries, OMG, WfMC, OAG, EDIFACT)

For data exchange based on these models it is expected to be used one of the standard's recommended Implementation Methods, as is the case of STEP's Neutral Format in Part 21 or XML in Part 28. However, the available tools to assist the implementation of such models are generally of very low functional and semantic level, and not so much spread in the market. STEP describes its Standard Data Access Interface (SDAI) in its part #22, specifying its functionalities in a general and neutral way independently of a programming language.

Also, the Document Object Model (DOM) is an API to access data represented in XML format (XML, 1998). This API understands the XML data described as a tree-based representation, and defines the mechanisms required to navigate across such tree in width and depth. They enable access and handle of its elements and attribute values as tree data nodes, allowing insert and delete of such nodes, and the conversion of the tree structure back into XML data format. These mechanisms provide a very flexible way of access and produce XML data format output, usually easier than simply writing or reading directly to a file in that format.

Nevertheless, there are in the market several tools for system's design and model development with a large acceptance by the users and software developers. Most of these tools have also facilities to automatically generate the data structures and interfaces for most of the popular programming and database languages, which are very convenient for implementation purposes. Examples are those from Rational Rose (www.rational.com) and Mega Suite (www.mega.com).

Facts:

- On one hand there is a huge investment developing models using standard-based methodologies.
- On the other hand, there is a technology very well accepted by the market, using methodologies like the Unified Modeling Language (UML) that, besides the modeling features provided, it also offers others like process design or system's deployment (OMG, 2001) (OMG, 2001a) (Starick, 1999) (DSTC Pty, 1999).

Question:

How to avail the large number of existent models described in languages like EXPRESS or XML, and reuse them and put them in the market in popular format like UML or any other?

One immediate answer could be to develop model translators to UML. But the core of the problem still persists, once UML models are represented in proprietary internal formats depending on the tools managing them, not existing an established neutral way to represent them.

2.1 XML Meta-Data Interchange (XMI)

The Object Management Group (OMG) released very recently a proposal named XML Metadata Interchange (XMI), intended to provide a general methodology for interchange of models, at first instance covering the OMG standards (e.g., MOF, UML, Corba) (OMG/XMI, 2001).

Today, XMI has been accepted universally as a standard for meta-model representation. Major groups for electronic data exchange, and most of the popular toolkits available in the market, have been adopting XMI as the standard for import/export of modeling information, supporting direct translation for major modeling technologies like UML, UMM and STEP (Starzyk, 1999).

To have mappers and translators between XMI and all major standard modeling languages, as is the case of EXPRESS for the STEP APs (STEP-25, 2001) or the several registered DTDs, would be an important achievement to assure reusability and acceptance of these models by the market.

The key issue in this case is to assure an accurate translation in terms of semantics and rules, what is not always simple to achieve. The mapping between modeling languages and semantics are not at all times realized through the identity operation, and transformation functions sometimes are complex to define (Parent, 2000)(Arsanji, 1999).

2.2 The STEP25 Tool

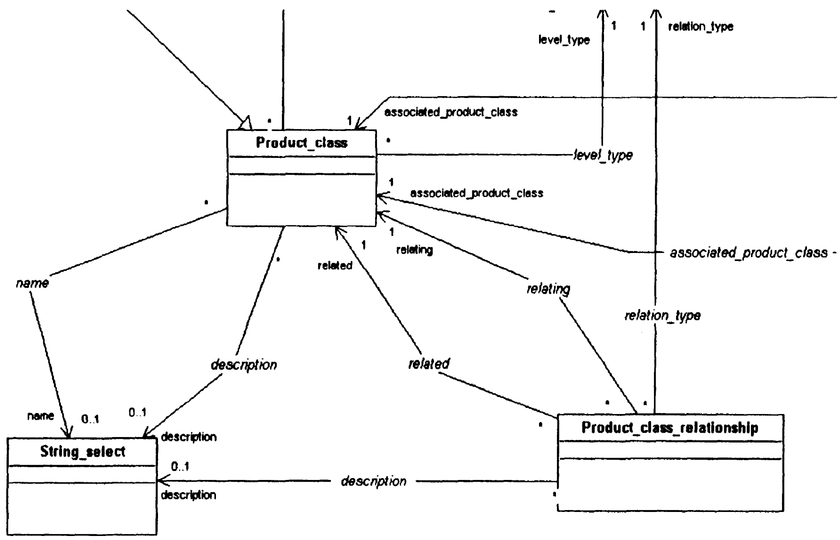
A major objective of the COFURN project (COFURN, 2001) is to demonstrate to industry the integration of several applications adopting the emerging standard ISO10303 AP236: Application protocol for furniture product and project decoration.

Its Application Reference Model is described in EXPRESS, and several software houses are in charge to develop interfaces to make their applications (e.g., CAD systems for decoration, electronic catalogue management systems, CAD for furniture design) interoperable among them using it. A public demonstration of this integration task using AP236 happens during the first week of July 2002 in Brussels.

Further than to prove the interoperability and automatic data exchange among these applications, the demonstration shows that the traditional effort required for the development of the translators can be significantly reduced when using standards for meta model representation associated to automatic code generators.

The new XMI language was adopted for the meta-model representation, and to achieve the aim as the proof of the concept, UNINOVA developed the STEP25 tool that translates EXPRESS-based model to XMI following the emerging ISO10303 Part25 directives (STEP-25, 2001)

This tool is a first that we know that implements and proofs this concept for EXPRESS to XMI binding, validating the Application Reference Model of AP236.



XMI:

```
(* ENTITY product_class; *)
<Foundation.Core.Class xmi.id="product_class.CLASS">
 <Foundation.Core.ModelElement.name>product_class</Foundation.Core.ModelElement.name>
 <Foundation.Core.ModelElement.visibility xmi.value="public"/>
 <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
 <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
 <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
 <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
 <Foundation.Core.Class.isActive xmi.value="false"/>
</Foundation.Core.Class>
(* END_ENTITY product_class; *)
```

EXPRESS:

```
ENTITY product_class;
 name : OPTIONAL string_select;
 id : undefined_object;
 description : OPTIONAL string_select;
 level_type : OPTIONAL undefined_object;
```

DTD:

```
<!ELEMENT Product_class EMPTY>
<!ATTLIST Product_class
    id ID #REQUIRED
    name IDREF #IMPLIED
    description IDREF #IMPLIED
```

version_id : OPTIONAL undefined_object; END_ENTITY; ENTITY product_class_relationship; relating : product_class; related : product_class; description : OPTIONAL string_select; relation_type : undefined_object; END_ENTITY;	level_type IDREF #REQUIRED version_id IDREF #IMPLIED> Product_class_relationship EMPTY> <!ATTLIST Product_class_relationship id ID #REQUIRED description IDREF #IMPLIED relation_type IDREF #REQUIRED relating IDREF #REQUIRED related IDREF #REQUIRED>

Figure 1 - Extract of EXPRESS to XMI mapping, and subsequent translation to UML and DTD.

With the ARM model described in XMI, it is used commercial Mega Suite platform to import such model to UML, and afterwards using the facilities of this platform to automatically generate code ready to assist in the implementation of the translators and repositories compatible with the reference model. Also the tool was tested with subsets of ISO10303 AP214, AP225 and ISO13584 part 20.

Figure 1 depicts a subset of the ARM AP236 and the respective meta-model representation in XMI resulting from the output of the execution of the developed STEP25 tool. Also it depicts the UML representation when imported the XMI representation in the MEGA Suite platform.

STEP25 is available for any one interested to use it. Authors will be very pleased to receive EXPRESS input files to help validate the tool.

3. ARCHITECTURE FOR INTER-MODELING

The integration of standard models can be executed by a platform for inter-modeling, based on standardized meta-level descriptions of the integrating model components and mapping rules. The authors propose the platform depicted in Figure2.

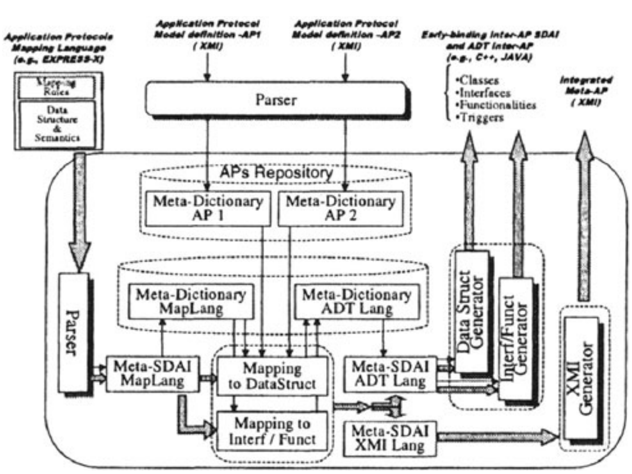


Figure 2 - Internal architecture of a platform for inter-modeling

The inputs for the platform are:

1) The models to be combined and integrated. These models should be described at a meta-level adopting in this case the XMI format. Those models not yet available at meta-level should be translated from its native modeling language (e.g. EXPRESS, XML) using for instance the framework proposed by (Jardim-Goncalves, 2001b). The translation process also includes a classification procedure of all model's components, to enable immediate search and reuse of the models' sub-schemas in the repository.
2) The mapping description to rule the integration, for example described in EXPRESS-X or XLST. This input describes the transformations needed to apply to a model component to be syntactically and semantically connected with the others. The general rule could be defined as:
   entityA.attribL = f(entitityV.attribK, entitityW.attribL,...)
   In the practical cases implemented, the use of the Identity operator was enough for most of the situations. Only a few required more complex transformations, as for instance the decomposition of an attribute in a set of them in the mapped model.

The outputs are:

1) The integrated model, described at meta-level using XMI, and resulting from the immediate reuse and extension of the selected model's components after the proper mapping.
2) A set of facilitators for implementation. These facilitators are libraries of Abstract Data Types (ADTs) generated automatically by the platform to assist in the implementation of the integrated model. These libraries can result available in many programming languages (e.g., C++, Java) and they are ready to be linked with the software applications. They will be a major part of the interfaces that enable the applications to adopt the new model (Jardim-Goncalves, 2001a), saving development effort to the implementers.

An analysis of the questionnaire filled by the Software Houses that participated in the development of the pilot demonstrators and used these facilitators, concluded that they saved about 60% of the implementation resources, if they did not use them.

3.1 Dynamics of the Platform

The first entry point of the platform are the input models (e.g., AP1 and AP2) in XMI format. Using a parser, the models are compiled to populate the Meta-dictionary according a meta-level structure in the APs repository, acting as a warehouse with all models' components classified and ready to be reused, extended and integrated.

This parser was developed based on the *ExParser* developed at the National Institute for Standardization and Technology (NIST) (www.nist.gov), with the inclusion of assertions in JAVA to produce the XMI translation.

In this architecture, the repositories for the meta-dictionary storage can be physically separated or be the same, using or not the same structure for dictionary data representation. However, for uniformity the interface to the repositories is normalized, through a unique Meta-SDAI's interface. In the implemented prototypes only one unique repository was adopted.

A meta-SDAI is a library of services that provides a set of pre-established functionalities needed for the translation at meta-level and repository access. Its existence in the architecture of the platform is of major importance to keep the platform flexible and independent of the modeling languages adopted. When substituting a meta-SDAI module by another, a new modeling or mapping language can be adopted by the platform using the same interface, without any further change.

The second platform's entry point is the mapping description. This input is also parsed and stored at a meta-level in the MapLang repository, to enable a neutral operation of the mapping information.

The platform outputs are generated by the execution of the mapping module, resulting from the interpretation of the mapping rules between the models, through the direct access to the models' meta-data repositories. To fit the requirements of the integrated meta-model, each selected component from the standard repositories could be reused, extended or reduced. The result is the new integrated model, together with the set of generated implementation facilitators.

For implementation of the new generated model, an early or late binding approach can be adopted. In the early binding case, the resultant integrated model is described standalone in XMI. In the late binding case, the integrated model is implicit. This means the mapper engine provides the interface for the integrated model, dynamically mapping the components' structure represented in XMI at different levels of abstraction, i.e., Semantics, Dynamics and Syntax. In this case, the mapper should be implemented using one of the technologies supported by the applications willing to adopt the new integrated model, offering the interface that virtualizes the integrated model to the implementer. The developments done were based on the early-binding approach.

The presented architecture is modular and open, making easy its adaptation to adopt any other modeling language, than EXPRESS, XML or XMI. Intending to translate to or from any other language, it means to consider new Meta-SDAIs, and plug them in the presented architecture.

3.2 Standard-Based Catalogue for Platform Setup and Model Description

To facilitate the search and decision support in the selection and usage of the models' components, these should be classified and described by standard catalogues (Figure 3). Moreover, the modular approach in the design of the proposed platform, results that a general system for inter-model integration can be dynamically adjusted using a plug-and-play mechanism, selecting the suitable modules (i.e., Meta-SDAIs and Mapping) from a catalogue, and link them to the platform according to the kind of integration required.

ISO13584 PLib (PLib, 1998) is a standard for representation of libraries of parts and catalogues, and was selected to support the platform for representation of catalogues of components and modules (i.e., Units of Functionality, Application Objects and Assertions, Integrated Resources, Data Access Interfaces, Object Business Data Types, etc.).

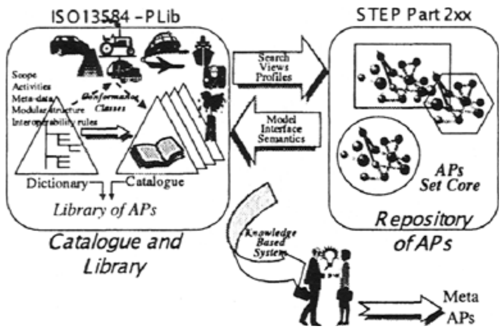


Figure 3– Framework for assisted management of inter-modeling platforms and meta-data

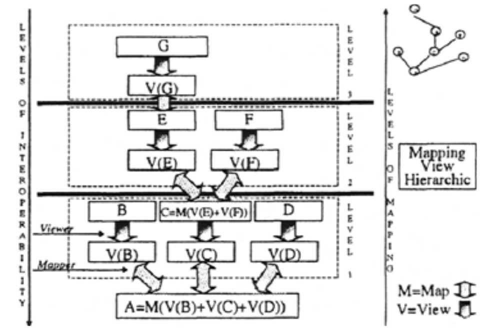


Figure 4 – Hierarchical meta-modeling in multi layer architecture

These libraries are defined jointly with a common ontological system, for classification and harmonization on the semantics of the classified elements (Sardet, 1997)(Fowler, 2000). It should support multi-language description in order to avoid semantically misunderstandings when using native language for search and use of the components to be included in the meta-protocol. With this possibility, anyone willing to develop an integrated model can search the available resources in a standard format catalogue, and select the best fit.

In COFURN project, it was developed an on-line dictionary using php technology on top of an Oracle RDMBS, where terms can be searched, inserted and translated on-line including related terms, synonyms, pictures and movies.

Additionally, a Knowledge Based System can offer intelligent support to the modeler for the selection of the better component according to search criteria and aims for reuse. In the developed case, mechanisms for synonyms and related terms are implemented, offering an integrated dynamic through references system.

4. HIERARCHICAL META-MODELING - APPROACH

For more complex cases, multi-layer architectures could be considered to implement the integrated model. This requirement conducts to the development of a hierarchical mapper, and multi-level implicit modeling. In this case it will be also needed to support an upper-level ontology structure, implementing the inter-relationships between the several ontologies used in the internal architecture of each meta-model component, as described in the previous section. Should this multi-standard integration comprises a multi-level hierarchical methodology, the complete integrated model will be materialized in the root of a resultant n-ary tree (figure 4).

Each node of the tree represents an already integrated model with a wider scope of coverage and interoperability than any of its children nodes, which could be another node or a standard-based model. One node can be used directly by the application, or reused as source for the construction of a new level of the integrated model. This methodology makes possible to construct in many levels the full integrated model ready to support the interoperability between all the application's activities, where the top level represents the complete integrated model, and each intermediate level is constructed using standard models (the atomic elements of the tree) or an already defined node. Each node is represented by a meta-model definition that describes the integrated model at its level. The integrated description results from the mapping procedure between all children nodes, also defined in a meta-model level. This example demonstrates the necessity to develop standards for meta-model representation, where the interchange and integration of models could be done in a neutral format. The expectative is to have available soon descriptions of APs and BOs in a standard meta-level format, and XMI has been identified as a very plausible candidate. While this is not globally achieved, it is necessary to use translators for the standard modeling languages.

Achieving a meta-modeling standardized description, universal catalogues, browsers and intelligent editing systems for development of integrated models can be developed. If these meta-model representations and catalogues become available through Internet using the presented framework, one can search using the web for existent models, or update and extend them. Furthermore, using this approach and in case of updates in the model's components, catalogues will be automatically updated reflecting promptly to the integrator.

The creation of a new layer in this hierarchical methodology should not be forced through a rigid methodological rule. Otherwise, it is up to the user to decide based on the available modules, those to select to link. For instance, during the development of AP236 it was identified the case of the link of a catalogue of furniture module with a room representation one, creating a new layer for project decoration. Afterwards, the shipbuilding industry would like to see project decorations included in the space of a ship. Therefore, a new layer could be created joining the previous joined one with the shipbuilding's.

Nevertheless, the documentation of the new integrated hierarchical model can be developed assisted by an intelligent system based on the existent information collected from the several components reused when constructing the new model. The developed methodology MetaDOC can be used for this purpose (Jardim-Goncalves, 2002b)

5. SUMMARY AND ACKNOWLEDGEMENT

Companies have been searching for flexible integrated environments to better manage their services and product life cycle, which their software applications could be easily integrated independently of the platform in use. Ideally, they would like to see open platforms based on a full-interoperable model covering the complete spectrum of their necessities. Because there are available today a large number of standard models developed to cover specialized domains of a company's needs, a global integrated model could be constructed based on the reuse, extension and assembly of these.

To contribute for this aim, this paper proposes a platform for inter-modeling based on standardized meta-level descriptions of the integrating model and mapping rules, and suggests the use of XMI for the meta-model representation. The major risks for the adoption of XMI identified in this context are twofold: 1) difficulties that can be find in the mapping rules for the translation from the input model to XMI, and 2) the degree of acceptance worldwide of XMI, that could be reduced on the emergence of a new standard for meta-model representation released by a more powerful group or organization.

To keep the platform flexible and independent of the modeling languages, the interface with the core of the platform's architecture is done through Meta-SDAIs, assisted by standard-based catalogues for platform's setup and model description. For implementation of the new generated model, an early or late binding approach can be adopted. In the early binding case, the resultant integrated model is described standalone in XMI. In the late binding case, the integrated model is implicit, handled by the mapper engine. For more complex cases, multi-layer architectures could be considered to implement the integrated model. The developed methodology MetaDOC can be used to document the new integrated model.

The authors would like to thank all the national and international organizations that supported the international projects that resulted in the development of the prototypes and framework presented in this paper, the European Commission, CEN/ISSS, Ministry of Industry of Portugal, Portugal/USA Foundation for Development, IPQ - Portuguese Standardisation Body, ISO TC184/SC4. Also, the authors express recognition for the project partners and our colleagues that work and contribute in the mentioned projects and in the development of ISO10303 (STEP) AP236.

6. REFERENCES

1. A. Lazcano et al. The wise approach to electronic commerce. International Journal of Computer Systems Science & Engineering, special issue on Flexible Workflow Technology Driving the Networked Economy, Vol. 15, No. 5, September 2000.
2. Alonso et al, "WISE: Business to Business E-Commerce", 9th International Workshop on Research Issues in Data Engineering - IT for Virtual Enterprises, RIDE-VE'99, IEEE Computer Society, pp 132-139, 1999
3. AP236, 2002, ISO10303 Part 236, Application Protocol: Furniture product and furniture project decoration data.
4. Arsanjari, Ali, 1999, Service provider: A domain patern and its business framework implementation, Proceeding of Pattern Languages of Programs (PLoP'99).
5. ATLAS, 1995, http://www.marchland.com/piebase/project/atlas.htm
6. Camarinha-Matos, L. and Afsarmanesh, H., 1999, Tendencies and general requirements for virtual enterprises, IFIP Working Conference on Infrastructures for virtual enterprises, Porto 1999, Chapter 2, Kluwer Academic Publishers, ISBN 0-7923-8639-6.
7. CECOM, 2001, http://www.cenorm.be/isss/Projects/c-ecom/default.htm
8. Chen, Q, Inter-enterprise collaborative business process management. Technical report, HP Labs Palo Alto, http://www.hpl.hp.com/techreports/2000/HPL-2000-107.pdf, 2000.
9. Clements, P., Standard support for the virtual enterprise, International Conference on Enterprise Integration Modeling Technology - ICEIMT'97, 1997, Torino, Italy. http://www.mel.nist.gov/workshop/iceimt97/pap-cle2/stdspt2.htm
10. COFURN, 2001, http://www.funstep.org/right/projects/cofFs.htm
11. Davulcu, H., Modeling and analysis of interactions in virtual enterprises. 9th International Workshop on Research Issues in Data Engineering – IT for Virtual Enterprises, pages 12–18. IEEE Computer Society, 1999.
12. DSTC Pty Ltd. Uml profile for enterprise distributed object computing (edoc), initial submission by dstc. Technical report, DSTC Pty Ltd, ftp://ftp.omg.org/pub/docs/ad/99-10-07.pdf, 1999.
13. Ducroux, F., The IT integration supporting the extended enterprise, 1999, pp. 137-145, ICE'99 - 5th International Conference on Concurrent Enterprising, The Hague, Netherlands, CCE-DMEOM, UK, ISBN 0-9519759-8-6.
14. ebXML. ebxml technical architecture specification v1.0.4. Technical report, www.ebXML.org, 2001.
15. e-Construct, 2000, http://www.econstruct.org
16. Fowler, Julian, 2000, "Co-operative use of STEP and PLib". http://www.nist.gov/sc4.
17. FSIG, 2002, funStep Interest Group, www.funstep.org
18. ISO10303, 2002, , www.tc184-sc4.org
19. Jardim-Gonçalves, R., et.al., 1999, Integrating manufacturing systems using ISO 10303 (STEP): An overview of UNINOVA projects, IJCAT, pp. 39-45, Vol.12, No 1,[2] ISSN 0-952-8091
20. Jardim-Gonçalves,R;Steiger-Garção,A.,2000,A framework for adoption of Standards for Data Exchange, 7th ISPE Int. Conf. on CE: Research and Applic.- CE'2000, pp.333-342, Technomic, ISBN 1-58716-033-1.
21. Jardim-Gonçalves, R., Steiger-Garção, A., 2001a, published in book: Agile Manufacturing: 21st Century Manufacturing Strategy, Chapter 48: "Putting the pieces together" using standards. Elsevier Science Publishers, pp. 735-757, ISBN: 0-08-043567-X.

22. Jardim-Gonçalves, R., Steiger-Garção, A., 2001b, Supporting Interoperability in Standard-based Environments, 8th ISPE Int. Conf. on CE: Research and Applic. CE'2001, ISBN: 0-9710461-0-7.
23. Jardim-Gonçalves, R. and Steiger-Garção, A., 2002a, Implicit multi-level modelling for integration and interoperability in flexible business environments, submitted to Communications of ACM, special issue on Enterprise Components
24. Jardim-Gonçalves, R., Steiger-Garção, A., 2002b, METAdoc: a framework for development of meta-protocol documents, Ricardo Jardim-Gonçalves and Adolfo Steiger-Garção, 6th WSEAS International Conference on SYSTEMS, Rethymna Beach, Rethymnon, accepted to be published.
25. Nagi, L., Design and Implementation of a Virtual Information System for Agile Manufacturing, IIE Transactions on Design and Manufacturing, special issue on Agile Manufacturing, 1997, Vol. 29(10), pp. 839-857
26. Nayak, N., et al, Virtual Enterprises - Building Blocks for Dynamic e-Business, In Proceedings of the workshop on information technology for virtual enterprises ITVE 2001, IEEE Computer Society, pp 80-87, 2001
27. OMG, 2001, Object Management Group, http://www.omg.org
28. OMG. "A UML profile for enterprise distributed object computing" Joint final submission. Document number ad/2001-06-09, June 2001
29. OMG/XMI, 2001a, Object Management Group/XML Meta Data Interchange, http://www.omg.org/technology/xml/index.htm
30. Parent, C. and Spacapietra, S., 2000, Issues and Approaches of Database Integration, Communications of ACM.
31. PDES, Inc., 2002, http://pdesinc.scra.org/
32. Plib, 1998, ISO TC184/SC4 DIS - ISO13584, Part1 - Parts Library: Overview and Fundamental Principles.
33. Poyet, P., A framework for distributed information management in the virtual enterprise: the VEGA project, The PRODNET communication infrastructure, IFIP Working Conference on Infrastructures for virtual enterprises, Porto 1999, Chapter 19, Kluwer Academic Publishers, ISBN 0-7923-8639-6
34. prodAEC, 2002, http://www.prodaec.net.
35. Radeke, E. et al. Distributed information management in virtual engineering enterprises by GEN. 9th International Workshop on Research Issues in Data Engineering - IT for Virtual Enterprises, RIDE-VE'99, pages 36–43. IEEE Computer Society, 1999.
36. Sardet, Eric; et.al, 1997, Formal Specification, Modelling and Exchange of Classes of Components according PLib. Global Network Engineering 97, pp 179- 201.
37. SOAP, 2000, http://www.mel.nist.gov/sc5/soap/
38. Starzyk, D., 1999, STEP and OMG Product Data Management specifications – A guide for decision makers, OMG Document mfg/99-10-04 and PDES Inc. Document MG001.04.00.
39. STEP-1, 1994, ISO10303 STEP Part1 - Overview and Fundamentals Principles
40. STEP-11, 1998, ISO TC184/SC4, IS - ISO 10303, Part11 – EXPRESS Language reference manual
41. STEP-25, 2001, ISO10303 STEP Part25: EXPRESS to XMI binding, www.nist.gov/sc4
42. SUMMIT, 1998, Project summary, http://exnorm.com/r&d/summit/pages/summary.html
43. Umar. A framework for analyzing virtual enterprise infrastructure. In Proceedings of the 9th International Workshop on Research Issues in Data Engineering - IT for Virtual Enterprises, RIDE-VE'99, pages 4–11. IEEE Computer Society, 1999.
44. VEGA, 1999, http://cic.cstb.fr/ILC/ecprojec/vega/home.htm
45. West, M., 2001, ISO/WD 18876-1 Integration of industrial data for exchange, access and sharing – Part1: Architecture overview and description, ISOTC184/SC4/WG10 N333.