



HAL
open science

Towards a Classification Framework for Interoperability of Enterprise Applications

Hervé Panetto

► **To cite this version:**

Hervé Panetto. Towards a Classification Framework for Interoperability of Enterprise Applications. International Journal of Computer Integrated Manufacturing, 2007, 20 (8), pp.727-740. 10.1080/09511920600996419 . hal-00116336

HAL Id: hal-00116336

<https://hal.science/hal-00116336>

Submitted on 25 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Classification Framework for Interoperability of Enterprise Applications

Hervé Panetto

*CRAN UMR 7039, Nancy-University, CNRS,
F 54506 Vandoeuvre les Nancy, France,
Phone: +33 383684426 – Fax: +33 383684459
Herve.Panetto@cran.uhp-nancy.fr*

Abstract: Software applications interoperability is a challenge for modern enterprises. It needs establishing standards and protocols for data exchange between different enterprise systems. Nevertheless, since there is no methodology for collecting data, exchanged information is generally incomplete. Business process modelling aims at specifying object flows and processes inside enterprise levels and among networked enterprises. Enterprise-control systems aim at driving and scheduling the manufacturing resources based on information coming from the whole enterprise. However, the increased complexity of these models does not help to ensure coherent relationships between its components. In this paper, the author will analyse how enterprise applications, models and standards used at different levels of the enterprise could be classified to come to a framework of many interoperability types.

Keywords: Enterprise integration, applications interoperability, integration standards, maturity models

1 INTRODUCTION

Many of today's enterprises are often faced with how to best manage their investment in existing system architectures but also address the need to change or evolve these architectures to meet ever-changing business demands. In many scenarios, enterprises encounter the challenging task of enabling coexistence between new and existing systems. Industry solutions today have implemented their software architectures on "building block" technology platforms provided by companies such as Sun Microsystems, IBM, and many others, each providing its own competitive advantage(s) within its respective domain. However, with the introduction of these new technology platforms comes the ever increasing complexity of extending these platforms. This arises to promote high degrees of interoperability with their respective competitors in order to provide a complete end-to-end business solution that meets the demands of the customer without increasing the costs associated with enabling platform interoperability. Standardisation initiatives, either supported by standardisation bodies such as ISO, IEC, or developed by industrial (ISA) or European projects have already tried to solve this issue. However, each focuses on one particular aspect of such interoperability without aligning their enterprise knowledge and skill for taking advantage of seamless cooperation. Interoperability problems may arise due to: different levels of the enterprises, different kinds of enterprise applications as well as due to the varying levels of abstractions of the considered models.

First interoperability definitions and issues with a brief state of the art in Enterprise interoperability models and standards are presented. Then, the primary purpose of this paper is to study the different maturity models in the literature and to propose, in section 3, an interoperability classification framework for classifying the interoperability process that arises. The classification depends on the degree of integration of the

enterprises, the kind of applications that have to interoperate and the abstract levels of the considered models that have to be exchanged between those applications. In order to put this framework in practice, sections 4 and 5 will present two initiatives dealing with interoperability at the business level and from the business to the manufacturing levels. Section 6 will discuss the classification of these initiatives on the proposed framework. Finally, the paper will conclude with some remarks and perspectives for ongoing research.

2 INTEROPERABILITY DEFINITIONS AND GENERAL ISSUES

Interoperability is typically defined as “the ability of two or more systems or components to exchange and use information” (IEEE, 1990).

The ISO 16100 (2002) standard defines the manufacturing software interoperability as “the ability to share and exchange information using common syntax and semantics to meet an application-specific functional relationship through the use of a common interface”.

More precisely *Interoperability* is the ability of different types of computers, networks, operating systems, and applications to work together effectively, without prior communication, in order to exchange information in a useful and meaningful manner. When trying to assess the understanding of an expression coming from a system in another system, there are several possible levels of interoperability (Euzenat, 2001):

- encoding: being able to segment the representation in characters;
- lexical: being able to segment the representation in words (or symbols);
- syntactic: being able to structure the representation in structured sentences (or formulas or assertions);
- semantic: being able to construct the propositional meaning of the representation;

- semiotic: being able to construct the pragmatic meaning of the representation (or its meaning in context).

This tiered structure is arguable in general; it is not as strict as it seems. It makes sense because each level cannot be achieved if the previous levels have not been completed. The encoding, lexical and syntactic levels are the most effective solutions, but not sufficient, to achieve a practical interoperability between computerized systems using existing technologies such as XML (eXtensible Mark-up Language) (W3C, 2004a) and its related applications (SOAP (Simple Object Access Protocol) (W3C, 2003), WSDL (Web Services Description Language) (W3C, 2004b), ebXML (Electronic Business XML Initiative) (OASIS, 2002), to name a few). In fact, current research work (Vernadat, 1996; UEMML, 2003; INTEROP, 2003; Panetto et al., 2004) deals with trying to enable a seamless data and model exchange at the semantic level. In that sense, standardisation initiatives (ISO 14528, 1999; IEC 62264, 2002; ISO 19440, 2004) then try to cope with this issue by defining generic constructs focusing on the domain concepts definitions. For more details on standardisation initiatives around integration and interoperability, the reader may refer to (Chen and Vernadat, 2002). The semiotic level is not currently taken into account in these initiatives because it requires complex processing more related to artificial intelligence domain.

Interoperability is a means to achieve integration (Chen and Vernadat, 2002). The difference between integration and interoperability has been further clarified in ISO 14528 (1999) (Concepts and rules for enterprise models). This standard considers that models could be related in three ways: (1) *integration* when there exists a standard or pivotal format to represent these models; (2) *unification*: when there exists a common meta-level structure establishing semantic equivalence between these models; and (3) *federation* when each model exists per se but mapping between concepts could be done at an ontology level to formalise the interoperability semantics.

Integration is generally considered to go beyond mere interoperability to involve some degree of functional dependence. While interoperable systems can function independently, an integrated system loses significant functionality if the flow of services is interrupted. An integrated family of systems must, of necessity, be interoperable, but interoperable systems need not be integrated. Integration also deals with organisational issues, in possibly a less formalised manner due to dealing with people, but integration is much more difficult to solve, while interoperability is more of a technical issue.

Compatibility is something less than interoperability. It means that systems/units do not interfere with each other's functioning. But it does not imply the ability to exchange services. Interoperable systems are by necessity compatible, but the converse is not necessarily true. To realize the power of networking through robust information exchange, one must go beyond compatibility.

In sum, interoperability lies in the middle of an "Integration Continuum" between compatibility and full integration. It is important to distinguish between these fundamentally different concepts of compatibility, interoperability, and integration, since failure to do so, sometimes confuses the debate over how to achieve them. While compatibility is clearly a minimum requirement, the degree of interoperability/integration desired in a joint family of systems or units is driven by the underlying operational level of those systems.

Then, classifying interoperability problems may help in understanding the degree of development needed to solve, at least partially, these problems. The next section will propose such a classification framework based on maturity models and definitions of classes of interoperability.

3 A TENTATIVE CLASSIFICATION OF THE INTEROPERABILITY PROCESS

The problems of enterprise applications interoperability can be defined according to various points of view and perspectives. These aspects correspond to modelling frameworks, with, as a common point, an implicit or explicit perspective of evolution according to a linear scale: the more an application is interoperable with another and thus higher in a value scale, the more it relates to a high level of abstraction of the models and their semantics. For this reason, an interoperability development process is often classified in so-called "levels of interoperability" in the literature:

A widely recognized model for information systems interoperability is, 'Levels of Information Systems Interoperability' (LISI) (C4ISR, 98). LISI focuses on the increasing levels of sophistication of systems interoperability. It defines the following five levels of interoperability relating the kinds of systems involved in the interoperability process:

Level 0 – Isolated systems: Interoperability at this level consists of the manual extraction and integration of data from multiple systems. This is sometimes called "sneaker-net".

Level 1 – Connected interoperability in a peer-to-peer environment: This relies on electronic links with some form of simple electronic exchange of data. Simple, homogeneous data types, such as voice, text email, and graphics (e.g., Graphic Interface Format files) are shared. There is little capacity to fuse information.

Level 2 – Functional interoperability in a distributed environment: Systems reside on local area networks that allow data to be passed from system to system. This level provides for increasingly complex media exchanges. Logical data models are shared across systems. Data generally contains heterogeneous information from many simple formats fused together (e.g., images with annotations).

Level 3 – Domain based interoperability in an integrated environment. Systems are connected via wide area networks. Information is exchanged between independent

applications using shared domain-based data models. This level enables common business rules and processes as well as direct database-to-database interactions. It also supports group collaboration on fused information.

Level 4 – Enterprise based interoperability in a universal environment: Systems are capable of using a global information space across multiple domains. Multiple users can access complex data simultaneously. Data and applications are fully shared and distributed. Advanced forms of collaboration are possible. Data have a common interpretation regardless of format.

Within a level, LISI identifies additional factors that influence the ability of systems to interoperate. These factors comprise four attributes: Procedures, Applications, Infrastructure, and Data (PAID). PAID provides a method for defining the set of characteristics required for exchanging information and services at each level. It defines a process that leads to interoperability profiles and other products. Scenarios depict the possible uses of LISI in different circumstances throughout the system life cycle.

LISI focuses on technical interoperability and the complexity of interoperations between systems. The model does not address the environmental and organizational issues that contribute to the construction and maintenance of interoperable systems (e.g., shared processes for defining interoperability requirements and maintaining interoperability across versions).

Acknowledging this limitation, Clark and Jones (1999) proposed the Organizational Interoperability Maturity model (OIM), which extends the LISI model into the more abstract layers of command and control support. Five levels of organizational maturity, describing the ability to interoperate, are defined. These include:

Level 0 - Independent - The Level 0 interoperability describes the interaction between independent organisations. These are organisations that would normally work without

any interaction other than that provided by personal contact. They are likely to be organisations that do not normally share common goals or purpose but that may be required to interoperate in some scenario that has no precedent. Essentially the arrangements are unplanned and unanticipated.

Level 1 - Ad hoc - At this level of interoperability only very limited organisational frameworks are in place which could support ad hoc arrangements. There will be some guidelines to describe how interoperability will occur but essentially the specific arrangements are still unplanned. There will be some overarching shared goal but individual organisation aspirations will take precedence and the organisations remain entirely distinct.

Level 2 - Collaborative - The collaborative organisational interoperability level is where recognised frameworks are in place to support interoperability and shared goals are recognised and roles and responsibilities are allocated as part of on-going responsibilities; however the organisations are still distinct.

Level 3 - Integrated - The integrated level of organisational interoperability is one where there are shared value systems and shared goals, a common understanding and a preparedness to interoperate; for example, a detailed doctrine is in place and there is significant experience in using it. The frameworks are in place and practised, however there are still residual attachments to a home organisation.

Level 4 - Unified - A unified organisation is one in which the organisational goals, value systems, command structure/style, and knowledge bases are shared across the system. The organisation is interoperating on a continuing basis. This is really the ideal level where there is no impediment in the organisational frameworks to full and complete interoperation

Beyond this organisational interoperability, the type of content of the exchange flows is also an issue. To cope with it, the NATO C3 Technical Architecture (NC3TA) Reference Model for Interoperability (NATO, 2003) focuses on technical

interoperability and establishes interoperability degrees and sub-degrees. Four degrees of interoperability are defined as follows:

Degree 1 - Unstructured Data Exchange: exchange of human-interpretable unstructured data such as the text found in operational estimates, analyses and papers.

Degree 2 - Structured Data Exchange: exchange of human-interpretable structured data intended for manual and/or automated handling, but requires manual compilation, receipt and/or message dispatch.

Degree 3 - Seamless Sharing of Data: automated sharing of data amongst systems based on a common exchange model.

Degree 4 - Seamless Sharing of Information: universal interpretation of information through data processing based on cooperating applications.

The degrees are intended to categorize how operational effectiveness could be enhanced by structuring and automating the exchange and interpretation of data.

Moreover, at a conceptual level, Tolk (2003) has developed the Levels of Conceptual Interoperability (LCIM) Model that addresses levels of conceptual interoperability that go beyond technical models like LISI. The model is intended to be a bridge between conceptual design and technical design. The focus lies in the data to be interchanged and the interface documentation that is available. The layers of the LCIM model include:

Level 0 - System specific data: black box components with no interoperability or shared data

Level 1 - Documented data: shared protocols between systems with data accessible via interfaces

Level 2 - Aligned static data: a common reference model with the meaning of data unambiguously described. Systems are black boxes with standard interfaces. However, even with a common reference model, the same data can be interpreted

differently in different systems.

Level 3 - Aligned dynamic data: Use of data is defined using software engineering languages like the Unified Modeling Language (UML, 2005). This allows visibility into how data is managed in the system. But even systems with the same interfaces and data can have different assumptions and expectations about the data.

Level 4 - Harmonized data: Non-obvious semantic connections are made apparent via a documented conceptual model underlying components. This goes beyond Level 3 because the assumptions concerning the data are made apparent.

As LCIM points out, in order to achieve the highest levels of interoperability, the assumptions underlying how systems interpret data must be made transparent. Tolk (year) observes that the model has been developed for the simulation domain but the basic premises apply to many complex sets of interoperating systems.

Applications interoperability is not only a technical problem (as stated by LISI or LCIM) but also deals with organisational issues (OIM). These aspects of interoperability are coherent with the definitions proposed by the European Interoperability Framework (EIF, 2004), which considers three aspects of interoperability:

Organisational Interoperability: This aspect of interoperability is concerned with defining business goals, modelling business processes and bringing about the collaboration of administrations that wish to exchange information and may have different internal structures and processes. Moreover, organisational interoperability aims at addressing the requirements of the user community by making services available, easily identifiable, accessible and user-oriented.

Semantic Interoperability: This aspect of interoperability is concerned with ensuring that the precise meaning of exchanged information is understandable by any other application that was not initially developed for this purpose. Semantic interoperability enables systems to combine received information with other information resources and

to process it in a meaningful manner. Semantic interoperability is therefore a prerequisite for the front-end multilingual delivery of services to the user.

Technical Interoperability: This aspect of interoperability covers the technical issues of linking computer systems and services. It includes key aspects such as open interfaces, interconnection services, data integration and middleware, data presentation and exchange, accessibility and security services.

Table 1 below shows the overlap and alignment between the various maturity models.

Table 1: The maturity models

EIF	Organisational				
	Semantic				
	Technical				
LISI	0 – Isolated	1 – Connected	2 – Functional Distributed	3 – Domain Integrated	4 – Enterprise Universal
OIM	0 – Independent	1 – Ad-hoc	2 – Collaborated	3 – Integrated	4 – Unified
LCIM	0 – System specific	1 – Documented	2 – Aligned static	3 – Aligned Dynamic	4 – Harmonised
NATO	1 – Unstructured data	2 – Structured data	3 – Seamless data sharing	4 – Seamless information sharing	

Complementary to the previous classifications of applications one has to distinguish the abstraction level of the applications models and the kind of interoperability that may arise.

Model abstraction levels have already been widely studied and have resulted to the so-called MDA (Model-driven Architecture) (Mellor, *et al.*, 2004), specified in the frame of OMG (Object Management group)¹. MDA defines three levels of models:

CIM – Computation Independent Model: A CIM is a model of a system that shows the system in the environment in which it will operate, and thus it helps in presenting exactly what the system is expected to do. It is useful, not only as an aid to understanding a problem, but also as a source of a shared vocabulary for use in other models. In an MDA specification of a system, CIM requirements should be traceable to the PIM and PSM constructs that implement them, and vice versa.

¹ OMG – <http://www.omg.org>

PIM – Platform Independent Model: A PIM might consist of enterprise, information and computational viewpoint specifications. (The structure of this information model might be quite different from the structure of an information viewpoint model in a computation independent model of the same system.)

PSM – Platform Specific Model: A PSM is a view of a system from the platform specific viewpoint. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform.

Applications interoperability problems may occur when exchanging models at the same abstraction level (CIM, PIM or PSM). That process is called “*horizontal interoperability*”. Similarly, when exchanging models from different abstraction levels, that process is called “*vertical interoperability*”. In both cases, the exchange process from one application to another involves models transformations (syntactic) and semantic alignment (also called concept mapping) (Doan, *et al.*, 2003; Noy and Musen, 2003).

Interoperability developments involving, mainly (but not only) technologies (or standards) to solve model syntactic transformations may be classified as ‘Model-driven interoperability (MDI)’ while developments where semantic alignment is the main issue are classified as the so-called ‘Semantic-driven interoperability (SDI)’. Of course, both share problems of language syntax and semantic. Many research works currently relate to the first class (MDI), not because they are simple, but because they should be able to develop new methods and technologies to practically solve the problem. The second class (SDI) also interests many scientists but it involves many theories and postulates that are, at least currently, difficult to prove (Rahm and Bernstein, 2001).

In the context of interoperation between enterprise business and manufacturing applications such as ERP (Enterprise Resource Planning), MES (Manufacturing Execution System), SCM (Supply Chain Management), to name only a few, the interoperability problem also faces the decision level related by a horizon and a period

(Doumeingts, *et al.*, 1998) of the models embedded in such applications. Taking into account, then, this time-frame dependence of the models, one may define two other classes of interoperability:

Diachronic² interoperability – are issues when applications interoperate over the time by exchanging models referring to different views of the same product. In this case, models have compatible semantics but need to be syntactically transformed before being exchanged. This allows streamlining model management and creating a true information management system. The issues come from disparate data repositories that should communicate through a platform-independent, vendor-neutral communications architecture based on open standards.

Synchronic³ interoperability – are issues where applications exchange models defined by compatible languages (the same syntax) but with different semantics, in a synchronous way.

Figure 1 describes the multidimensional issues, and thus, the complexity, of an interoperability problem and will be used, in the last section, to classify some examples of interoperability developments.

An application interoperability process has then to take into account:

- The degree of integration of the considered enterprises (referring to some maturity models such as LISI, LCIM, OIM,
- The degree of abstraction of the considered models (referring to the MDA),
- The decision levels of the considered applications.

² From the Greek, dia-, through, across, and chronos, time. Diachronic systems focus on processes changing over time

³ From the Greek, syn-, together, and chronos, time. Synchronic systems focus on processes occurring or existing at the same time

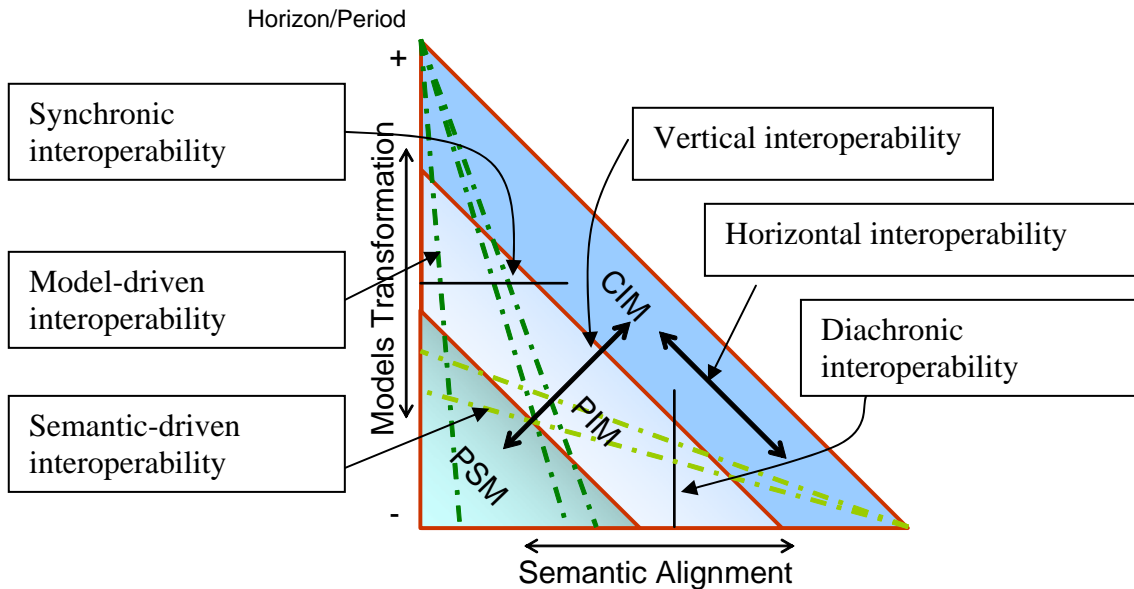


Figure 1: Interoperability classification framework

To show an example of interoperability problem and its possible classification in the proposed interoperability classification framework, the next sections will present two initiatives. The first is related to, the business level and uses UEML (Unified Enterprise Modelling Language) that has been developed during a European project (UEML, 2003) in order to provide a pivotal language for exchanging enterprise models between applications and software. The second initiative is related to IEC 62264 (2002), a standard developed to facilitate the interoperability between business applications and manufacturing control-systems, hence providing definitions of data units exchanged between a subset of functional activities that comprise the various enterprise processes. The next two sections present a short analysis of these two initiatives.

4 UNIFIED ENTERPRISE MODELLING LANGUAGE

In this section UEML (Berio, *et al.*, 2002; Panetto, *et al.*, 2004) is presented. The meta-model of the language is represented in Figure 2.

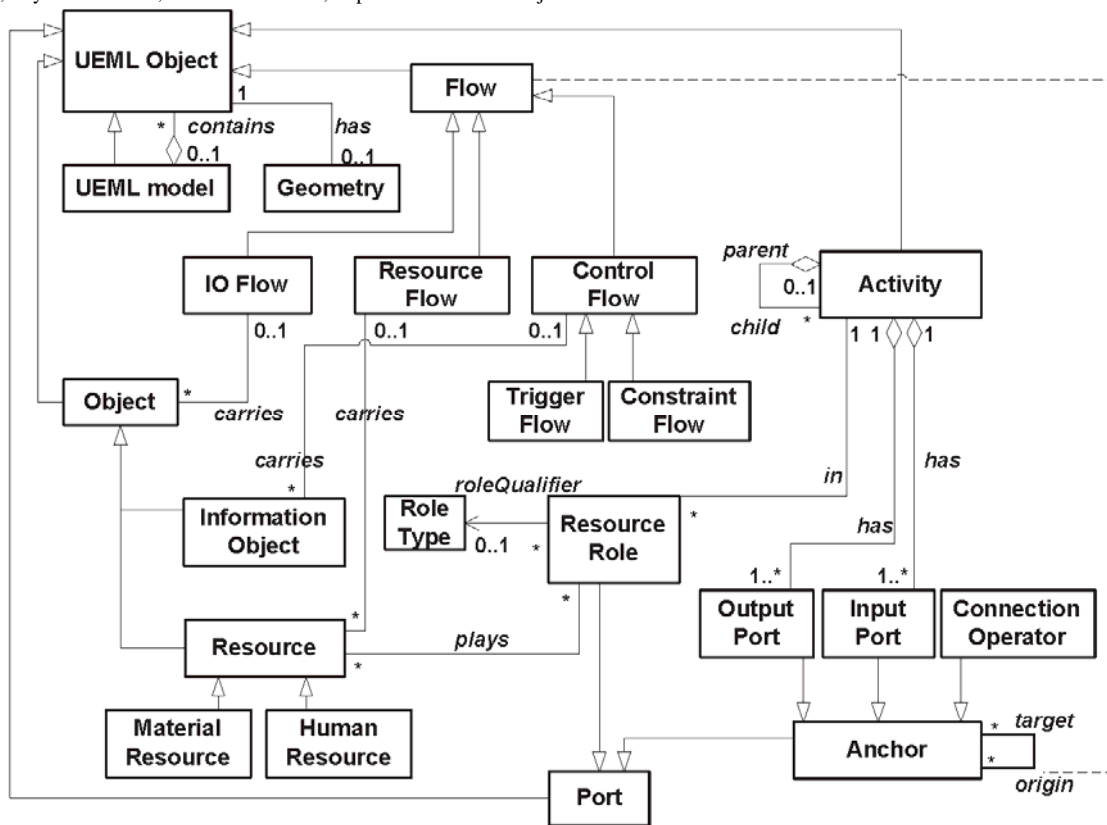


Figure 2: Part of the UEML meta-model

The UEML, as defined in (Berio, *et al.*, 2002) has been developed to facilitate the integration of different enterprise modelling languages and to demonstrate the feasibility of applying a bottom up approach for doing so. UEML is under development and the current version, UEML 1.0, is to be considered as an intermediate result.

The UEML project⁴ was set up in an attempt to contribute to the solving of the problems of multiple Enterprise Modelling Languages. The long term objective of UEML is the definition of a core language called the Unified Enterprise Modelling Language, which would serve as an Interlingua between enterprise modelling tools and applications that need enterprise models as part of their configurations. This language:

- Provides the business community with a common visual, template based language to be used on top of most commercial enterprise modelling and workflow software tools;

⁴ UEML IST-TN 2001 34229, www.ueml.org

- Provides standardised mechanisms for sharing and exchanging enterprise models among projects, overcoming tool dependencies;
- Supports the implementation of open and evolutionary enterprise model repositories to leverage enterprise knowledge engineering services and capabilities.

The UEML comprises a set of constructs and their interrelationships, which defines an ontology of the domain of business enterprise processes (Berio, *et al.*, 2002):

An *Activity* represents a generic description of a part of enterprise behaviour which produces outputs from a set of inputs. An *Activity* may be decomposed into other activities and an activity may require one or several *Resource Roles* played by *Resources* for its completion. A *Resource* may be a *Material Resource* or a *Human Resource*.

An *Activity* has at least one *Input Port* and at least one *Output Port*, where flows representing inputs or outputs of the activity are connected.

A *Resource* is used by an *Activity*. This may be specified in two ways:

- 1) Through the definition of a role (i.e. *Role Type*) which a resource plays in an *Activity*. This method is used when the origin of the resource is not explicitly given; and
- 2) Through a flow, connected to the *Input Port* of the *Activity* carrying the resource. This method is used when the origin of the resource is explicitly given or if the resource to be used is the result of some decision, grouping or decomposition of some other resource(s) through a *Connection Operator*.

An *Object* is anything that can be attached to a *Flow*. In other words, it is anything that may be needed or produced by an *Activity*. It can be an *Information Object* or a *Resource*.

A *Flow* represents the flowing of an object from an origin to a target. The origin and target of a flow are called *Anchor* and can be an *Input Port*, an *Output Port* or a *Connection Operator*.

A *Flow* is an *IO Flow*, a *Resource Flow* or a *Control Flow*.

- An *IO Flow* represents the flowing of an *Object* between two Activities. If the *Object* is an input object, then the *IO Flow* is connected to an *Input Port* of an *Activity* and this means that the *Object* is necessary for the execution of the *Activity*. The *Object* can possibly be consumed or modified by the *Activity*. If the *Object* is an output object of an *Activity* the *IO Flow* is connected to the *Output Port* of the activity indicating that the object has been produced by the *Activity*.
- A *Resource Flow* represents the flowing of a *Resource* between two Activities. The flow then connects an *Output Port* of an *Activity* that produces it and an *Input Port* of the *Activity* that requires it.
- A *Control Flow* connects two Activities and represents either:
 - 1) a precedence relationship between Activities (a *Control Flow* carrying no *Object*);
 - 2) a triggering of an *Activity* (a *Trigger Flow*, which carries an *Information Object* that triggers a second activity after the completion of a first one);
 - 3) a constraining of an *Activity* (a *Constraint Flow* carrying a constraining *Information Object*. For instance, this could be a description of a procedure to be followed when executing the activity).

A *Connection Operator* represents the grouping or splitting (*Join* and *Split*) of flows between activities. A *Connection Operator* of the type “*Join*” is a target of at least two Flows and is the origin of exactly one *Flow*. A *Connection Operator* of type “*Split*” is origin of at least two Flows and is the target of exactly one *Flow*. An attribute (not represented in the figure) specifies whether the *Connector* is relating an “AND”, “OR” and “XOR”, indicating parallelism, choice and synchronisation.

The semantics attached to these constructs realises a semantic interoperability between applications dealing with business process. These applications are related to the models relating the business needs of the enterprise, thus UEML participates to partially solve issues on the horizontal interoperability (CIM), as explained in section 2.

In order to cope with the lower levels of interoperability (syntactic, lexical), an implementation of this model has been done by translating the UML Class Diagram (UML, 2005) model into an XML Schema Definition (W3C, 2001) which defines the operational structure of an XML file that is exchanged between applications.

In section 6, the author will discuss the possible classification of interoperability processes involving the UEML.

While UEML is dedicated to the interoperability at the business level of the enterprise, the next section will present a standard defined for the interoperability between business and manufacturing applications.

5 IEC 62264 STANDARD FOR ENTERPRISE-CONTROL SYSTEMS INTEGRATION

In this section the IEC 62264 standard (IEC 62264, 2002), mainly its first part named “Models and Terminology”, is presented.

The IEC 62264 set of standards is extending the ANSI/ISA S95 (ANSI/ISA, 2000) specifications, that defines an information exchange framework to facilitate the integration of business applications and manufacturing control applications, within an enterprise. It is composed by six different parts designed for defining the interfaces between enterprise activities and control activities. Among all its parts, part 1 describes the relevant functions within an enterprise and within the control domain of an enterprise, stating which objects are normally exchanged between these domains.

Figure 3 depicts the different levels of a functional hierarchy model: business planning and logistics, manufacturing operations and control, and batch, continuous, or discrete control.

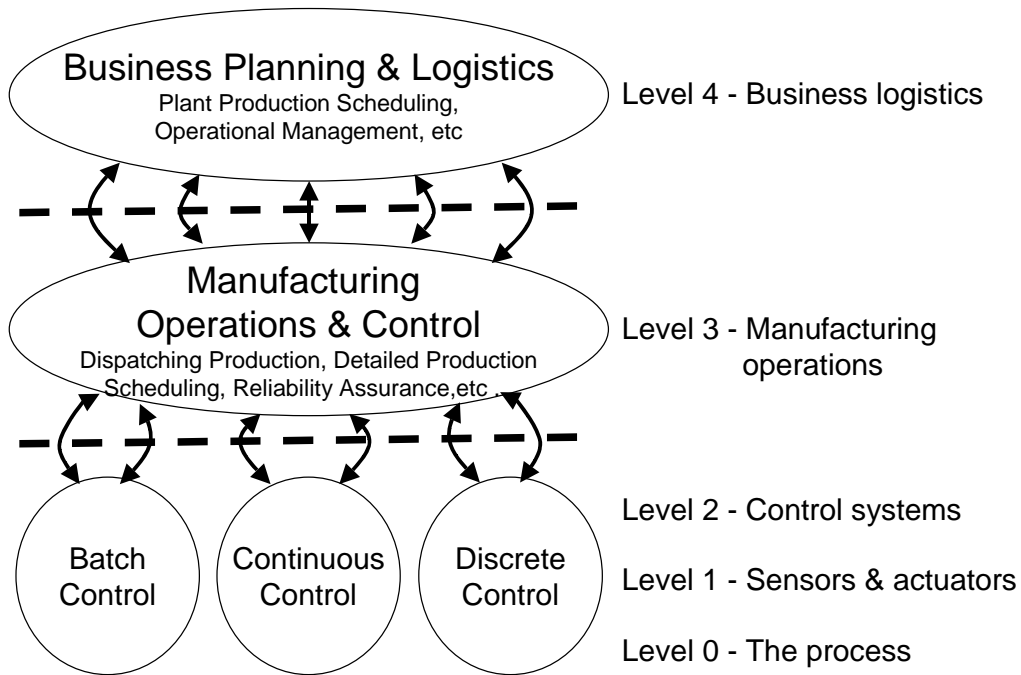


Figure 3: Functional hierarchy as defined in IEC 62264

The model shows the hierarchical levels at which decisions are made. The interface addressed in the standard is between Level 4 and Level 3 of the hierarchy model. This is generally the interface between plant production scheduling and operation management and plant floor coordination.

Levels 2, 1, and 0 present the cell or line supervision functions, operations functions, and process control functions, not addressed by this standard. Level 0 indicates the process, usually the manufacturing or production process. Level 1 indicates manual sensing, sensors, and actuators used to monitor and manipulate the process. Level 2 indicates the control activities, either manual or automated, that keeps the process stable or under control. There are several different models for the functions at these levels based on the actual production strategy used.

The key aspects for integrating the business applications at Level 4 and the manufacturing operations and control applications at Level 2 (and below) are the

information structures and exchanges managed by Level 3 activities, applications, processes, resources, and functions. Examples of Level 3 activities include the management of various manufacturing operations, such as: production, maintenance, product quality testing, and material handling.

The standard defines a set of models that describe all concepts for enterprise-control integration. Each model concerns a particular view of the integration problem. Those models show increasing detail level and are operational models or resource models. Even if all models specified in the standard are important for trying to answer the vertical interoperability issues of an enterprise, the paper will focus on analysing one of them: the material model (Figure 4).

The Material model is a resource model that defines the actual materials, material definitions, and information about classes of material definitions. Material information includes the inventory of raw, finished, and intermediate materials. *Material classes* are defined to organise materials. A *material definition* is a means to describe goods with similar characteristics for purposes of scheduling and planning.

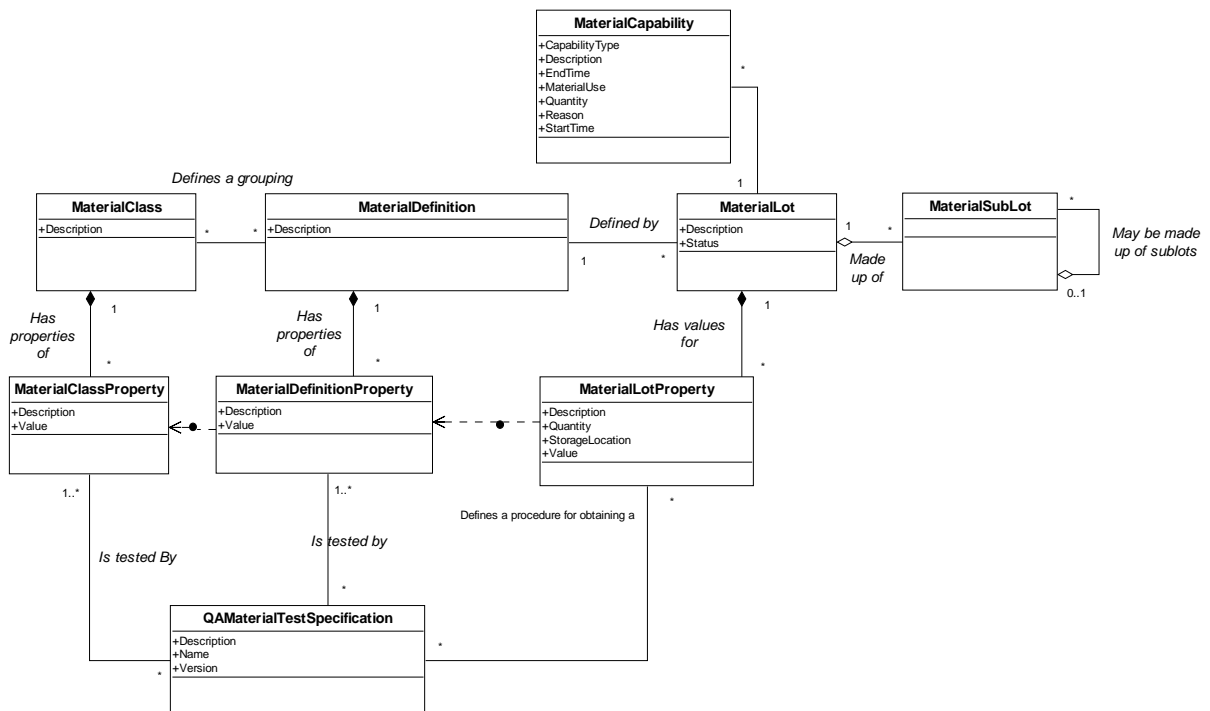


Figure 4: The IEC 62264 Material model

A *material class* is a means for defining sets of material definitions. A material class may be further characterised through zero or more material class properties. The *material class properties* usually define the nominal or standard values for the material. A *material definition property* does not have to match a material class property. A *material lot* uniquely identifies a specific amount of material, as defined by its *Material Definition*. It defines specific instances, where *Material Lot Properties* have specific values, regarding a particular Material Definition and its *Material Definition Properties*. Material lots and Material sublots may be used for traceability when they contain unique identification. A material lot may be stored as a separate identifiable quantity. Each separate identifiable quantity of the material is identified in a *material sublot* object. The semantics attached to these constructs realises a semantic interoperability between business applications (level 4) and manufacturing operations (level 3). The standard participates to partially solve issues with the vertical interoperability, as explained in section 2. In order to cope with the lower levels of interoperability (syntactic, lexical), an implementation of the part 1 of the standard has been realised by the WBF (World Batch Forum⁵) where a working group has delivered a set of XML Schema Definitions called B2MML (Business to Manufacturing Mark-up Language) that are used to encode well-formed XML files (Figure 5).

```
<Material>
  <MaterialLot>
    <ID> W89 </ID>
    <Description> A lot of material </Description>
    <MaterialDefinitionID> WXE908 </MaterialDefinitionID>
    <Location> Tank 1 </Location>
    <Quantity
      UnitOfMeasure = "KL" > 4500
    </Quantity>
    <MaterialLotProperty>
      <ID> dateTimeProduction </ID>
      <Value> 2001-01-06T00:14:23+11:30 </Value>
    </MaterialLotProperty>
    <MaterialLotProperty>
      <ID> Quality Status </ID>
      <Value> Good </Value>
    </MaterialLotProperty>
  </MaterialLot>
</Material>
```

Figure 5: Extract of the XML file corresponding to a Material Lot instance

⁵ <http://www.wbf.org>

The next section will now discuss the classification, on the proposed framework, of interoperability developments applying the UEML, the ISO 62264 and models exchanged between both.

6 EXAMPLES OF INTEROPERABILITY CLASSIFICATIONS

6.1 UEML

UEML is an exchange format, but could be considered as an enterprise modelling language, that is able to specify enterprise processes in its environment in which it will operate, and presents exactly what the enterprise is expected to do. As such, UEML is a CIM, as defined in the MDA. Using UEML is thus classified as “horizontal interoperability”. As it has been shown, UEML deals with the exchange of mainly business process models between applications at the business level of the enterprise (level 4 of the functional hierarchy defined by IEC 62264 (Figure 3)). Interoperability between applications using UEML as a pivotal language needs, at some extent, both model transformation and semantic alignment. This, of course, depends of the expressiveness and the inner semantic of the considered enterprise models. Figure 6 represents graphically the interoperability process using UEML by a segment, located in the CIM lane, at a high level regarding the time frame of the involved applications (Business applications), which length depends on the “semantics distance” between the two applications considered in this interoperability process. Of course, formally valuating this “semantic distance” is an open issue involving many researchers (Roddick, *et al.*, 2003 ; Patwardhan, *et al.*, 2003 ; Rodríguez and Egenhofer, 2003), but this issue is not the objective of this paper.

Referring to the maturity models presented in Section 3, UEML, by defining some semantics on concepts related to the enterprise business domain, enables enterprise systems to be classified at Level 2 and partially Level 3 of LISI, OIM and LCIM (Table 2).

Table 2: UEML facilitating enterprise systems interoperability

EIF	UEML				Organisational
	Technical				
LISI	0 – Isolated	1 – Connected	2 – Functional Distributed	3 – Domain Integrated	4 – Enterprise Universal
OIM	0 – Independent	1 – Ad-hoc	2 – Collaborated	3 – Integrated	4 – Unified
LCIM	0 – System specific	1 – Documented	2 – Aligned static	3 – Aligned Dynamic	4 – Harmonised
NATO	1 – Unstructured data	2 – Structured data	3 – Seamless data sharing	4 – Seamless information sharing	
		IEC 62264			
		UEML <> IEC 62264			

6.2 IEC 62264

The IEC 62264 standard, at least its part 1, consists of the information exchange specifications between business applications such as ERP (Enterprise Resource Planning) and manufacturing control systems such as MES (Manufacturing Execution Systems). The model defines the interfaces between enterprise activities and control activities. As such, the IEC 62264 is a PIM, as defined in the MDA. However, the interface relates only to some model transformation without any change of semantic regarding the exchanged information. Thus, it participates to a so-called “diachronic interoperability”. Figure 6 shows the location of the segment related to the standard in the proposed interoperability classification framework.

Referring to the maturity models presented in Section 3, the IEC 62264, by defining mainly the structured data to be exchanged between business and manufacturing applications facilitates enterprise systems to be classified at Level 1 of LISI, OIM and LCIM (Table 2) but is limited to the technical aspect as defined in EIF.

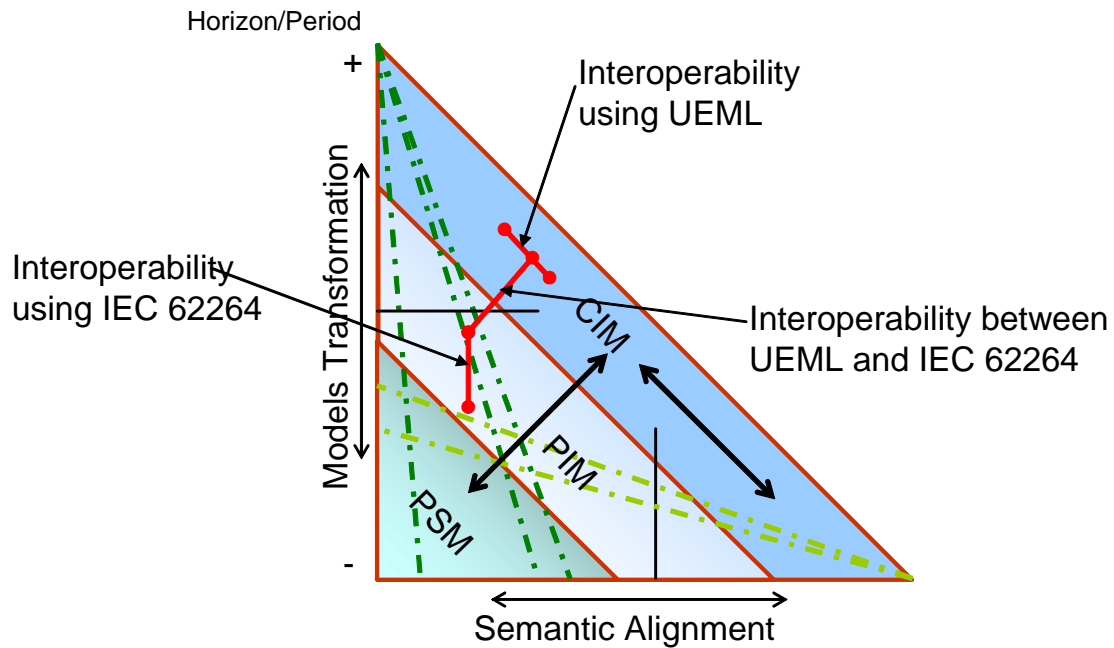


Figure 6: Classification of some interoperability processes

Previous work (Morel, *et al.*, 2003) put forward that information and beyond, a form of technical intelligence embedded into the manufacturing systems components and inside the products themselves, are playing a prominent role as the pivotal technologies that make it possible to address agile Business to Manufacturing issues. This is emphasised by the need to enable interoperability and cooperation between existing manufacturing systems (Nguyen and Vernadat, 1994), and by facilitating model exchange between enterprise platforms and systems. In manufacturing environments, and more particularly in the enterprise-control systems context, before information exchange, models concerning all resources, products, goods and services have to be collected at the shop floor level. In order to cope with sharing partial enterprise models, and not only exchanging data from the business level of the enterprise to the manufacturing level, current work (Baina, *et al.*, 2005) is defining modelling constructs that are able to facilitate interoperability between UEML and the IEC 62264. This interoperability process is typically a vertical interoperability crossing the boundaries from CIM to PIM (Figure 6). It needs both models transformations and semantic alignment of modelling concepts. Referring to the maturity models presented in Section 3, this interoperability

process is encompassing the initial models and then is facilitating enterprise systems to be classified at Level 1, 2, and partially Level 3 of LISI, OIM and LCIM (Table 2).

7 CONCLUSIONS

Enterprise integration has made great progress during last decade. This progress is due to increased cooperation and collaboration needs in enterprise circles. Interoperability issues are of many kinds and many maturity models have been defined to classify them. The author has proposed an interoperability classification framework that may help determine the degree of development to be done for solving interoperability issues. This proposal has been applied to business process interoperability using the UEMML exchange format and business to manufacturing interoperability using the IEC 62294 standard and mainly the B2MML implementation (Figure 7). These applications have been classified in the proposed framework to evaluate the effort needed to put them into practice. But many other interoperability developments have to be further classified in the same way to evaluate the applicability of our proposal. However, many researchers agree that one of the major bottlenecks in interoperability is semantic integration by mapping discovery. There are simply too many ontologies available and they are too huge to enable manual definition of correspondences as the primary source of mapping discovery. Further work will be to formalise such mappings in order to evaluate, at least partially, the semantic distance between concepts used by different modelling languages in order to push enterprise applications to be better integrated, even unified. This work is taking place within a joint research activity in the INTEROP NoE (INTEROP NoE, 2003).

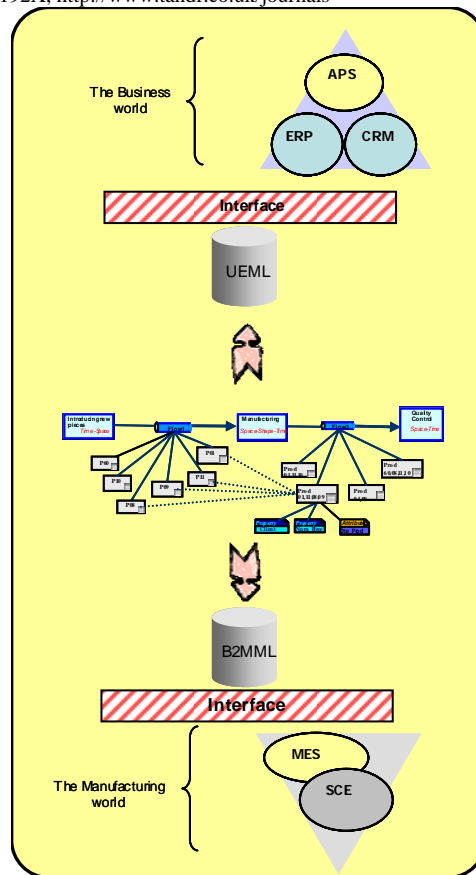


Figure 7: Snapshot of the application study using UEML and B2MML

8 REFERENCES

ANSI/ISA S95, (2000) *Enterprise-control system integration*, ANSI/ISA 95.00.01 Part

1: Models and Terminology; ANSI/ISA 95.00.02 Part 2: Object Model Attributes

ISA, Raleigh, North Carolina, U.S.A.

Bařna S., Panetto H., and Morel G. (2005), Holon-oriented b2m process modelling

approach for applications interoperability in manufacturing systems environment,

Proceedings of the IFAC World Congress, July 4-7, Prague, Czech Republic

Berio G., et al. (2002). *D3.1: Requirements analysis: initial core constructs and*

architecture, UEML TN IST 2001 34229. <http://www.ueml.org>

C4ISR Architectures Working Group report (1998). *Levels of Information Systems*

Interoperability (LISI), DoD, February 1998, Washington, DC

Panetto H. (2007). Towards a Classification Framework for Interoperability of Enterprise Applications. *International Journal of CIM*, Taylor & Francis, ISSN: 0951-192X, <http://www.tandf.co.uk/journals>

Chen D. and Vernadat F.B. (2002). Enterprise Interoperability: a standardisation view.

IFIP International Conference on Enterprise Integration and Modelling Technology

(ICEIMT'02), pp. 273-282, Kluwer Academics Publisher, Valencia, Spain, 24-26

April 2002, ISBN 1-4020-7277-5

Clark, T. and R. Jones (1999). Organisational Interoperability Maturity Model for C2,

In Proceedings of the Command And Control Research And Technology Symposium

(CCRTS), June 29, July 1, Newport, RI, USA

EIF (2004), *European Interoperability Framework for pan-European eGovernment*

Services, Interoperable Delivery of European eGovernment Services to public

Administrations, Businesses and Citizens (IDABC), November, Luxembourg

Euzenat J., Towards a principled approach to semantic interoperability, *CEUR*

Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing,

Seattle, USA, August 4-5, 2001, ISSN 1613-0073, Vol. 47., 19-25,

<http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-47/>

IEC 62264 (2002). IEC 62264-1:2002. *Enterprise-control system integration, Part 1.*

Models and terminology, ISO/IEC, Geneva

IEEE STD 610.12 (1990). *Standard Glossary of Software Engineering Terminology*,

IEEE, May, ISBN: 155937067X

INTEROP NoE (2003). Interoperability Research for Networked Enterprises

Applications and Software. *Technical annex (restricted)* IST FP6 Network of

Excellence IST 2003-508011, November, <http://www.interop-noe.org>

ISO 14528 (1999). *Industrial Automation Systems – Concepts and rules for Enterprise*

Models, TC 184/SC5/WG1, Geneva, Switzerland

ISO 16100 (2002). *Manufacturing Software Capability Profiling for interoperability*,

Part 1: Framework, TC 184/SC5/WG4, Geneva, Switzerland, ICS 25.040.01

ISO EN DIS 19440 (2004). *Enterprise integration - Constructs of enterprise modelling*,

Draft version, TC 184/SC5/WG1, Geneva, Switzerland

- Panetto H. (2007). Towards a Classification Framework for Interoperability of Enterprise Applications. *International Journal of CIM*, Taylor & Francis, ISSN: 0951-192X, <http://www.tandf.co.uk/journals>
- Doan A., Madhavan J., Dhamankar R., Domingos P., and Halevy A. (2003). Learning to match ontologies on the Semantic Web. *The VLDB Journal — The International Journal on Very Large Data Bases archive*. **12/4**, 303-319, November, ISSN: 1066-8888. Springer-Verlag New York
- Doumeingts G., Vallespir B. and Chen D. (1998). Decision modelling GRAI grid, 321-346 Chapter in: P. Bernus, K. Mertins, G. Schmidt (Eds.) *Handbook on architecture for Information Systems*, Springer-Verlag.
- Mellor S.J., Kendall S., Uhl A. and Weise D. (2004). *Model Driven Architecture*, Addison-Wesley Pub Co, March, ISBN: 0201788918.
- Morel G., Panetto H., Zaremba M.B. and Mayer F. (2003). Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues. *IFAC Annual Reviews in Control*. **27/2**:199-209, December
- NATO Allied Data Publication 34 (ADatP-34) (2003). *NATO C3 Technical Architecture (NC3TA)*, Version 4.0
- Nguyen G.T and Vernadat F. (1994). Cooperative information systems in integrated manufacturing environments. *Proceedings of the 2nd International Conference on Cooperative Information Systems*. Toronto, Canada. May.
- Noy N. F. and Musen M. A. (2003). The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 983–1024, **59(6)**.
- OASIS, (2002). *Organization for the Advancement of Structured Information Standards*. ebXML specification, February, <http://www.oasis-open.org>
- Panetto H., Berio G., Benali K., Boudjlida N. and Petit M. (2004). A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. *Proceedings of IFAC INCOM Symposium*, April 7th-9th, Salvador de Bahia, Brazil
- Patwardhan S., Banerjee S., and Pedersen T. (2003). Using Measures of Semantic Relatedness for Word Sense Disambiguation. *Proceedings of Computational*

- Panetto H. (2007). Towards a Classification Framework for Interoperability of Enterprise Applications. *International Journal of CIM*, Taylor & Francis, ISSN: 0951-192X, <http://www.tandf.co.uk/journals>
- Linguistics and Intelligent Text Processing: 4th International Conference, CICLing*
2003, Mexico City, Mexico, February 16-22, 2003. 241-257. LNCS **2588**, Springer-Verlag, ISSN: 0302-9743
- Rahm E. and Bernstein P. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*. 334-350, **10**
- Roddick J. F., Hornsby K., and de Vries D. (2003). A unifying semantic distance model for determining the similarity of attribute values. *Proceedings of the twenty-sixth Australasian conference on Computer science: research and practice in information technology*, Adelaide, Australia, ACM International Conference Proceeding Series, **35**, 111-118
- Rodríguez A., and Egenhofer M. (2003). Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering*. **15/2**, 442-456, 2003.
- Tolk, Andreas. "Beyond Technical Interoperability – Introducing a Reference Model for Measures of Merit for Coalition Interoperability." *In Proceedings of the 8th International Command and Control Research and Technology Symposium (ICCRTS)*, Washington, DC, June 17-19, 2003. Washington DC: Command and Control Research Program (CCRP), 2003.
- UEML. (2003). *Unified Enterprise Modelling Language (UEML) Thematic Network*. IST-2001-34229, <http://www.ueml.org>.
- UML (2005). Unified Modeling Language. UML 2.0 Superstructure, v2.0 formal 05/07/04. OMG
- Vernadat F.B. (1996). *Enterprise modelling and integration: principles and applications*. Chapman & Hall.
- W3C (2001). *XML Schema Definition Language 1.0*, W3C specification, March, <http://www.w3c.org>

Panetto H. (2007). Towards a Classification Framework for Interoperability of Enterprise Applications. *International Journal of CIM*, Taylor & Francis, ISSN: 0951-192X, <http://www.tandf.co.uk/journals>
W3C (2003). *Simple Object Access Protocol (SOAP) 1.2*, W3C specification, February,
<http://www.w3c.org>

W3C (2004a), *XML, Extensible Mark-up Language*, W3C XML 1.1 recommendation,
February, <http://www.w3c.org>

W3C (2004b). *Web Services Description Language (WSDL) 2.0*, W3C specification.
August, <http://www.w3c.org>

ACCRONYMS

B2M	Business to Manufacturing
B2MML	Business to Manufacturing Markup Language
CIM	Computation Independent Model
ebXML	Electronic Business XML
EIF	European Interoperability Framework
ERP	Enterprise Resource Planning
IEC	International Electrotechnical Commission
ISA	Instrument Society of America
ISO	International Organisation for Standardisation
LCIM	Levels of Conceptual Interoperability
LISI	Levels of Information Systems Interoperability
MDA	Model-Driven Architecture
MES	Manufacturing Execution System
OIM	Organisational Maturity Model
PAID	Procedures, Applications, Infrastructure and Data
PIM	Platform Independent Model
PSM	Platform Specific Model
SCM	Supply Chain Management
SOAP	Simple Object Access Protocol
UEML	Unified Enterprise Modelling Language
UML	Unified Modeling Language
WSDL	Web Service Description Language
XML	eXtensible Markup Language

LIST OF TABLES

Table 1: The maturity models.....	11
Table 2: UEML facilitating enterprise systems interoperability	23

LIST OF FIGURES

Figure 1: Interoperability classification framework	14
Figure 2: Part of the UEML meta-model	15
Figure 3: Functional hierarchy as defined in IEC 62264.....	19
Figure 4: The IEC 62264 Material model	20

Figure 5: Extract of the XML file corresponding to a Material Lot instance.....	21
Figure 6: Classification of some interoperability processes.....	24
Figure 7: Snapshot of the application study using UEML and B2MML	26