AN ENTERPRISE MODEL OF INTEROPERABILITY

Lawrence E. Whitman[1], Danny Santanu[1], and Hervé Panetto[2]

[1]*Industrial & Manufacturing Engineering Department, Wichita State University, USA*
{Larry.Whitman, dxsantanu}@wichita.edu
[2]*Research Centre for Automatic Control, University Henri Poincaré Nancy I, France*
Herve.Panetto@cran.uhp-nancy.fr

Abstract: Enterprise integration (EI) is the re-engineering of business processes and information systems to improve teamwork and coordination across organizational boundaries, thereby increasing the effectiveness of the enterprise as a whole. EI is enabled by interoperating enterprise models. This paper presents an IDEF0 model specifying an approach for interoperating dissimilar enterprise models. Enterprise models are transformed by the activity, "interoperate enterprise model" into interoperable enterprise models. This activity is decomposed into the following five functions: *Define meta-models*, *Validate the meta-model*, *Investigate and define the relations between the enterprise modelling languages*, and *Validate unified meta-model*. These functions are described in detail in this paper.

Keywords: Enterprise integration, Enterprise modelling, Architectures.

1. INTRODUCTION

In order to face global competition and fluctuating market conditions, companies require the management of change (Vernadat, 1996). To simplify managing the change, Enterprise Integration (EI) plays an important role. The process of integrating an enterprise is simplified by using a model. An enterprise model is defined as “a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals, and constraints of a business, government, or other enterprise” (Fox and Gruninger, 1998). An enterprise modelling language is required to create an enterprise model.

There are many enterprise modelling languages and tools available and each modelling language has different characteristics. “This intensive production of tools has led to a Tower of Babel situation in which the many tools, while offering powerful and distinct functionalities, are unable to interoperate and can hardly or not at all communicate and exchange models” (Panetto, *et al.*, 2004b). Each enterprise modelling language and tool has its own characteristics, features and suitability for each type of industry. It is impossible to force industries to only utilize one type of enterprise modelling language and one enterprise modelling tool. Diversity of enterprise models may create obstacles in achieving the goal of enterprise integration. To respond to these obstacles, interoperability of enterprise modelling languages is needed. Interoperability focuses on communication between applications (Vernadat, 1996). Even though there are many different enterprise modelling languages, it would be productive if they can communicate with each other. In addition, interoperability leads to a standard for different languages. One example of interoperability commonly accepted language is called the Unified Enterprise Modelling Language (UEML) (Berio, *et al.*, 2002; Panetto, *et al.*, 2004a). UEML is not another enterprise modelling language, but it is a meta-model that supports interoperability. The issue of interoperability addressed in this paper is the challenging goal of enterprise integration. This paper presents an enterprise model specifying an approach for interoperability.

2. INTEROPERABILITY

The word interoperability has many wide uses. The IEEE (1990) definition of interoperability is, “the ability of two or more systems or components to exchange information and to use the information that has been exchanged.” Interoperability has been implemented differently in different domains. Healthcare systems are working towards interoperability to process and manage all patient services in efficient ways (Ray and Vargas, 2003). Geographic information systems need interoperability to enable users to receive combined results from dissimilar geographic information systems (Gupta, *et al.*, 1999). European policymakers have long recognized the importance of interoperability in the e-Government context (Muller, 2005).

Enterprise Integration consists of connecting and making interoperable all functional areas of an organization. Interoperability will improve organization's synergy in achieving its mission and vision in effective and efficient manner (Molina, *et al.*, 2004).

The term interoperability is increasingly used in enterprise engineering and its related standardization activities (Chen and Vernadat, 2002). Interoperability is defined as the "ability of two or more systems or components to exchange information and to use the information that has been exchanged" (delaHostria, 2002). To achieve interoperability, the systems need to interoperate their data, resources and business processes with semantics defined in a business context regardless of different languages, data formats, interfaces, executions platforms, communication protocols or message formats (Tsagkani, 2005). Interoperability is not only about transferring information but also performing an operation on behalf of another system. This is commonly called "interoperation".

One of the main obstacles to interoperability arises from the fact that the systems that support the functions in many enterprises were created independently, and do not share the same semantics for the terminology of their process models. Interoperability requires data stored in software systems on one machine to be sent and interpreted by another software system on another machine and for different purposes. To make these happen, standards on message format and transfer are needed. Today, Internet technology seems to be the most likely to support intra and inter-enterprise interoperability.

2.1 Types of interoperability.

To achieve interoperability of an enterprise, four types are needed to achieve. The first type is technical interoperability, which then continues with semantic and pragmatic interoperability.

Technical interoperability means transporting messages from one application to another. With today's technologies, technical interoperability is not the issue anymore. Current applications have full support in technical interoperability. Many applications have been able to understand each other. For example, a file from one application can be translated and read with a different application.

Syntactic interoperability is the second type. Before performing data exchange, the systems must agree on the format for data exchange. The Extensible Markup Language (XML) has solved this issue. XML has addressed the syntactic interoperability issue (Lilleng, 2005).

Semantic interoperability means understanding the content of messages and models in the same way by senders and receivers. Semantic interoperability refers to a system's ability for exchanging information inside organizations with heterogeneous information and-or between organizations without having to do tailoring to make this possible (Lilleng, 2005).

Finally, pragmatic interoperability captures the willingness of partners for the actions necessary for the collaboration. This willingness to participate involves both capability of performing a requested action, and policies dictating whether the potential action is preferable for the enterprise to be involved in collaboration (Tsagkani, 2005).

3. INTEROPERABILITY MODEL

To make interoperability easier to understand, this paper presents an IDEF0 model of interoperability. The IDEF0 model is prescriptive (this means that it communicates all activities which are required for performing interoperability).

This paper uses the IDEF (Integration DEFinition) modelling language to present the interoperability method. IDEF was developed by the U.S. Air Force's Integrated Computer Aided Manufacturing (ICAM) project in the late 1980's. There are many different IDEF methods. Each method is useful for describing a particular perspective of an enterprise. The specific modelling method used in this paper is IDEF0. This method best facilitates a structured approach to system model development and review and the creation of a corresponding discrete-event simulation of the system. IDEF0 uses a subordinate principle of abstraction called decomposition, which is the breaking down of each box (activity) into more detail in a continuous manner until the greatest level of detail is achieved (Marca and McGowan 1988).

There are five elements in the IDEF0 functional model as shown in Figure 1. The activity (or function) is represented by the boxes; inputs are represented by the arrows flowing into the left hand side of an activity box; outputs are represented by arrows flowing out the right hand side of an activity box; the arrows flowing into the top portion of the box represent constraints or controls on the activities; and the final element represented by arrows flowing into the bottom of the activity box are the mechanisms that carry out the activity (Marca and McGowan 1988; Mayer, 1992).

The purpose of the IDEF0 model proposed in this paper is to assist in interoperating enterprise models. The viewpoint of this model is that of enterprise modellers.

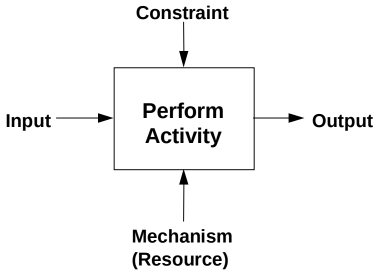

Fig. 1. IDEF0 Nomenclature.

3.1 A-0 - Interoperate enterprise model.

Figure 2 shows the A-0 diagram which bounds the model. This diagram represents that isolated enterprise models, which are not able to share knowledge among different enterprise models are transformed by the activity, "interoperate enterprise model" into interoperable enterprise models, constrained by feedback from enterprise modelers. This activity is enabled by UML (Unified Modeling Language) for mapping the meta-model from each isolated enterprise model. In addition, real-world case studies are used to define the common constructs.

Test cases and enterprise model documentation are the source of information about the isolated enterprise models. Those sources will be useful in finding the meta-model of each enterprise models. In addition these sources will be used for validating activity in order to ensure there are no misleading information in defining meta-model.

Constructs are part of the meta-model that from each enterprise model, which are the generic object or template structure which models a basic concept (Panetto, 2002). To be able in interoperating between enterprise model, finding relation between constructs of enterprise models is necessary.

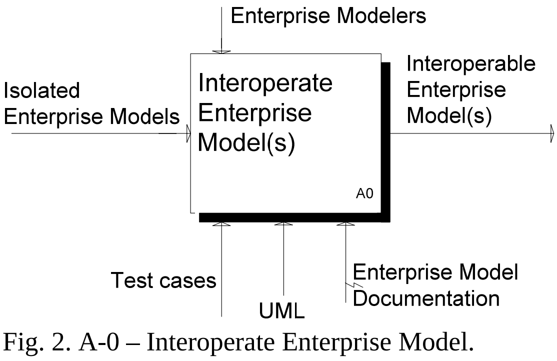

Fig. 2. A-0 – Interoperate Enterprise Model.

3.2 A0 - Interoperate enterprise model.

Figure 3 shows the decomposition of the A-0 diagram called the A0 diagram. The activity shown in the A-0 diagram is decomposed into the following five functions: *Define meta-models*, *Validate the meta-model*, *Investigate and define the relations between the enterprise modelling languages*, and *Validate unified meta-model*. These functions are described in detail in this section.

To be able to interoperate isolated enterprise models, a meta-model from each isolated model must be defined. Interoperating an enterprise model requires a schema from each enterprise model. The schema in this case is the meta-model from the candidate enterprise model (Berio, *et al.*, 2002). In some cases, obtaining a meta-model requires reverse engineering of the language or an elaboration by hand based on the literature describing the particular enterprise modeling language (Petit, *et al.*, 2002). Elaboration by hand means whoever will interoperate the isolated meta-model might need combination information from different source than manually combined during this activity. The literature that describes the enterprise model is the ontology of enterprise model. The meta-model is described by using meta-modeling language (Petit, *et al.*, 2002). Compared to other meta-modelling languages, UML is most often used for defining meta-models of languages (Berio and Petit 2003). The UML diagram will consist of several classes of meta-models. The owner from each isolated model will ensure there is no misleading information. This is all completed during the activity, *Define meta-models.*

After defining the meta-model from each enterprise modelling language, this meta-model must be validated. To be successful in validating the meta-model, validation through interaction with the owners of the model is required. The ontology from each enterprise model is used for reference in the validation activity. The outcome of this activity is whether the meta-model is either valid or not valid. If the meta-model is not valid, then redefinition of meta-model is required.

For the definition of common enterprise meta-model, establishing correspondences among the classes is defined in the different meta-models of the isolated enterprise models. After defining the meta-model from each isolated enterprise model, investigation of the correspondences among model elements is completed. These correspondences are then generalized if the model elements have common semantics. “The semantics comparison must in this case make sure that the sets of process behaviors described by both models are corresponding” (Berio, *et al.*, 2002). After finding the common semantics between enterprise models a unified meta-model is produced. The unified meta-model is mapped by using UML. This is all completed during the activity, *Investigate and define the relations between the enterprise modelling languages.*

The common meta-models must now be validated. The validation of a unified meta-model is the last activity before the isolated enterprise models can interoperate. The unified meta-model combines all meta-models that have a common meaning between the different isolated models. The validation can be done through interaction with the owners of the enterprise models. This interaction will be useful to maintain the value from each model to achieve the goal of interoperable enterprise models. Real-world case studies are also required in the validation activity. "Case studies have to be carried out in which

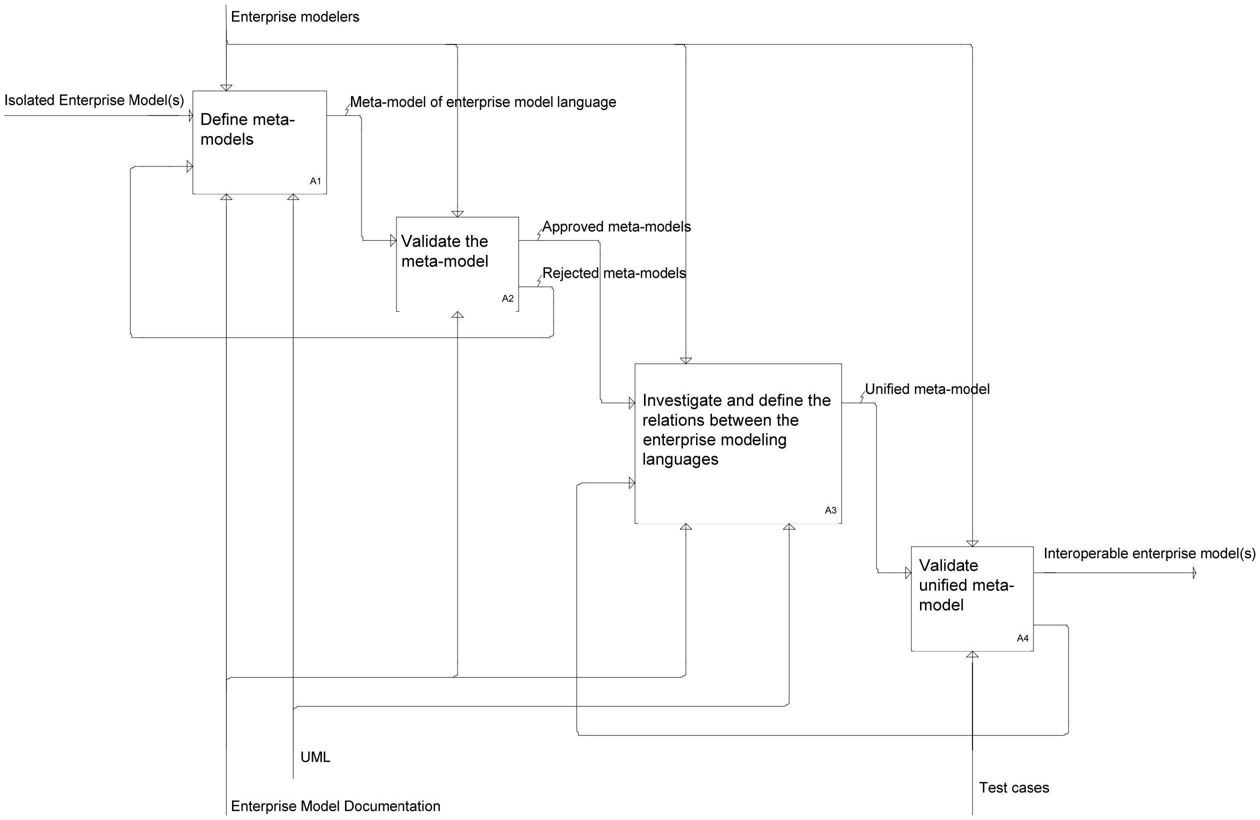

Fig. 3. A0 – Decomposition of Interoperate Enterprise Model.

a single reality is modelled with these different languages. The correspondences among the obtained models have to be established by comparing the semantics of the obtained model elements in terms of the states of real world elements or behaviours they represent" (Petit, 2002, 7). The outcome is either the unified meta-model is valid or not. If the unified model is invalid, the investigation of common correspondence between models is required again. If the unified meta-model is valid, the unified meta-model shows the interoperability between at least two isolated enterprise model. This is all completed during the activity, Validate unified meta-model.

Several activities are decomposed selectively. The complete model, with the full text, glossary, and additional decompositions, may be found at: http://enteng.wichita.edu/idefmodels/interoperability/

4. SUMMARY

This paper provided an approach for interoperating dissimilar enterprise models. The IDEF0 activity model presented transforms *isolated enterprise models* into *interoperable enterprise models*. The IDEF0 diagram containing the top level function is further decomposed into a child diagram which has four functions: *Define meta-models*, *Validate the meta-model*, *Investigate and define the relations between the enterprise modelling languages*, and *Validate unified meta-model*.

REFERENCES

Berio G. (2002). D3.1: Requirements analysis: initial core constructs and architecture, UEML TN IST 2001 34229. http://www.ueml.org.

Berio, G. and M. Petit (2003). Enterprise Modelling and the UML: (sometimes) a conflict without a case. In *Proc. of Concurrent Engineering Conference*, July 26-30, Madeira Island, Portugal.

Chen, D., and F. B. Vernadat (2002). Enterprise interoperability: A standardization view. In *Proc. of the IFIP International Conference on Enterprise Integration and Modelling Technology (ICEIMT'02)*, Valencia, Spain, April 24-26.

delaHostria, E. (2002). Interoperability of standards to support application integration. In *Proc. of the IFIP International Conference on Enterprise Integration and Modelling Technology (ICEIMT'02)*, Valencia, Spain, April 24-26.

Fox, M.S., and M. Gruninger (1998). Enterprise modeling. *AI Magazine*, Fall 1998.

Fox, M.S., M. Gruninger and Y. Zhan (1994). Enterprise engineering: An information systems perspective. In *Proc. of the 3rd Industrial Engineering Research Conference*, Atlanta GA.

Gupta, A., R. Marciano, I. Zaslavsky, and C. Baru (1999). Integrating GIS and imagery through XML-Based information mediation. In *Integrated Spatial Databases: Digital Images and GIS, Lecture Notes in Computer Science*, **1737**.

IEEE (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.

Lilleng, J. (2005). Towards semantic interoperability. *Proc. of the IFIP/ACM SIGAPP INTEROP-ESA Conference*, Geneva, Switzerland, February 23-25.

Molina, A., D. Chen, H. Panetto, F. Vernadat, F., and L. Whitman (2004). Enterprise integration and networking: issues, trends and vision. In *Proc. of the IFIP International Conference on Enterprise Integration and Modeling Technology (ICEIMT'04)*, Knowledge Sharing in the Integrated Enterprise Interoperability - Strategies for the Enterprise Architect, **183**, Toronto, Canada, October 9-11.

Muller, B. (2005) eGovernment, interoperability and innovation. In *Proc. of the IFIP/ACM SIGAPP INTEROP-ESA conference*, Geneva, Switzerland, February 23-25.

Panetto H. (2002). UML semantics representation of enterprise modelling constructs. In *Proc. of the IFIP International Conference on Enterprise Integration and Modelling Technology (ICEIMT'02)*, pp. 381-387, Valencia, Spain, April 24-26.

Panetto H., G. Berio, K. Benali, N. Boudjlida and M. Petit (2004a). A unified enterprise modelling language for enhanced interoperability of enterprise models. In *Proc. of IFAC INCOM 2004 Symposium*, Bahia, Brazil, April 5-7.

Panetto, H., L. Whitman and K. Chatha (2004b). Ontology for enterprise interoperability in the domain of enterprise business modelling. In *Proc. of the IFAC TC 5.3 International workshop on Enterprise Interoperability, Integration and Networking (EI2N'2005)*, Geneva, Switzerland, February 22, pp. 103-114.

Petit M, et al. (2002). *D1.1: State of the Art in Enterprise Modelling*, UEML TN IST 2001 34229, www.ueml.org

Ray, P., and B. Vargas (2003). Interoperability of hospital information systems. In *Proc. of the 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry*, Santa Monica, CA.

Tsagkani, C. (2005) Inter-Organizational Collaboration on the Process Layer. In *Proc. of the IFIP/ACM SIGAPP INTEROP-ESA Conference*, Geneva, Switzerland, February 23-25.

Vernadat, F.B. (1996). E*nterprise Modeling and Integration: Principles and Applications*. London: Chapman & Hall.

Vernadat, F.B. (2000). UEML: towards a unified enterprise modeling language. *International Journal of Production Research*, *40*, 4309-4321.

Institute of Electrical and Electronics Engineers. New York, NY: 1990