A review of the methodologies, techniques and suggestions proposed for enabling knowledge sharing and reuse in computer integrated manufacturing environments and other enterprises

February 1998

Panayiotis Periorellis, Albert Bokma

School of Computing and Information Systems, University of Sunderland,
St. Peter's Campus, Sunderland, SR6 0DD

+44 191 515 3641

Email :cs0ppe@isis.sunderland.ac.uk

Abstract
The term enterprise modelling, synonymous with enterprise engineering, refers to methodologies developed for modelling activities, states, time, and cost within an enterprise architecture. They serve as a vehicle for evaluating and modelling activities resources etc. CIM - OSA (Computer Integrated Manufacturing Open Systems Architecture) is a methodology for modelling computer integrated environments, and its major objective is the appropriate integration of enterprise operations by means of efficient information exchange within the enterprise. PERA is another methodology for developing models of computer integrated manufacturing environments. The department of industrial engineering in Toronto proposed the development of ontologies as a vehicle for enterprise integration. The paper reviews the work carried out by various researchers and computing departments on the area of enterprise modelling and points out other modelling problems related to enterprise integration.

Keywords : Enterprise modelling, PERA, CIM-OSA, TOVE, Ontology, Ontolingua

Introduction

Most of the articles written on enterprise modelling deal with the complexity of planning and scheduling enterprise activities. As according to Fox S. [1996] planning is determined by the degree to which activities contend for resources, most emphasis has been put into creating a model of the availability of resources. The purpose these models serve is to enable reasoning on the various facets of resources such as their amounts capacity and availability.

Michael Gruniger [1994] presented a different approach to enterprise modelling. He divided an enterprise into four major components: its *common-sense, advisors, visualisation* and information agents. *Common-sense* was the most important model as it provided a set of representations of the enterprise knowledge. This representation includes processes, activities, time, quality and cost. His argument for creating such a model was that the various perspectives that exist in an enterprise such as efficiency quality and cost must be represented and managed in a well defined way.

The concerns we're having is that most of the approaches lead to information systems that support enterprises as if they are autonomous entities. Enterprises are open systems with relationships and links to other systems, making them dependent upon other systems' functions.

We should develop a method to model these links and networks of enterprises and assess the impact they have on their structure and behaviour. The fact that makes this task difficult, is that you cannot create a general definition for these links as they depend upon the nature of the enterprises related. The most essential factor that needs to be taken into account, is the strength of the link which is usually determined by political factors, such as the power of the enterprise who initiates the link, or degree of dependency between the two parties. The result will be a network of enterprises. The model should also attempt to tackle the problem of assessing the impact -a link being created or deleted- has on the rest of the network, in terms of what the alterations are, and what will the position of the enterprises be after the link has either being broken or created. Ideas and suggestions for developing such a model have been expressed in Periorellis[1997].

Enterprise Modelling

The term *Enterprise Modelling* is mainly related to the modelling of the activities within an enterprise. The aim is usually to create a general model of all the activities, cost, time and resources that take place between the different departments of an enterprise. Many of those systems were developed in isolation and were only concerned with the activities of the department which they represented. The problems, that seemed to appear over time, was the lack of information exchange capabilities between departments and the consistency problems which appears, when the same data were represented by different names or when the same data structures had different representations in different departments. Various commercial methods attempt to tackle these problems either by introducing new programming concepts for representing and sharing data, or modelling departmental activities and their relationships.

Categories of Methods

The methods I came across during my research can be divided into two major categories. The ones that were concerned with the actual knowledge sharing procedures and the ones that emphasised on the modelling of the departmental activities in terms of costs, time, resources and labour. Ontologies and the *TOVE* methodology described below, aimed to create a protocol of communication between the enterprises' department, where knowledge bases were represented using various knowledge representation techniques. *CIM-OSA* and *PERA* both emphasise on the modelling of a computer integrated manufacturing environment.

Ontologies for Enterprise modelling

The department of industrial engineering of the *University of Toronto* attempted to tackle the problem by creating an *Ontology* for enterprises. *Ontology* is a fairly new concept and in order to understand the work that was carried out by the Toronto team it is better to explain the meaning of the term first. The problem with knowledge sharing and re-use can be easily understood if one thinks of the requirements necessary to start a conversion or a dialogue with someone else. They both have to agree on the communication protocols prior the commencement of the dialogue and obey the rules set by these protocols too. Language would be the major issue. Establishing a common language as the basis of a dialogue will ensure that a two directional conversation could begin were the two parties can exchange ideas, derive solution and finally solve problems. Another issue here of major importance, is the terminology the two people will use. Terminology refers to these special and unique symbols or words that people may use when they exchange information. Symbols or keywords can be used to mean different things or represent different data when we deal with a conversation between two people or with information between two departments in the same enterprise. Whenever this happens during an information exchange procedure may lead to misunderstandings or inconsistencies as it was mentioned above. Terminology agreement will ensure that the two parties are referring to the same things when they are using symbols, commands or special keywords. Imagine two people trying to communicate when they both speak different languages and they have no knowledge of a common language which can be used as the basis to exchange information. At this point the parties are isolated and the knowledge they possess cannot be re -used because of the language barrier. The same applies to departments that have no means of exchange knowledge. The problem would be solved by using a translator, in the first case, who would be able to *'listen'* to one speaker, translate the information from the speaker's language to his own and then translate the information from his own language to the second party's language. The diagram illustrates the process of translation.

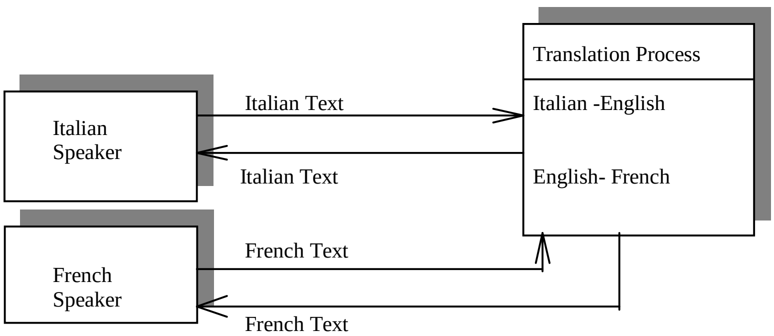

Figure 1: The translation process

Although this process may seem straight forward, in computing terms is a lot more complicated. Knowledge bases are developed using deferent methodologies or techniques, that lead to different structures. Knowledge is therefore represented is different ways and the functions or algorithms that actually process this knowledge is derived from these structures. In order to share some domain of common interest between knowledge bases we use *ontologies*. As M.Ushold [1996] described:

> ‘An ontology necessarily entails some sort of world view with respect to a given domain. The world view is often conceived as a set of concepts (e.g. entities, attributes, processes), their definition and their inter-relationships; this is referred to as conceptualisation, which may be explicit -existing in one’s head- or embodied in a piece of software.’

An ontology includes a vocabulary of the terms and some specification of their meaning. The ways these vocabulary’s are created varies. Some are highly informal ;expressed loosely unnatural language. Others semi-informal, are expressed in restricted and structured natural language form using symbols and other notations. There are also special languages developed specifically for the development of ontologies. *Ontolingua* is such an example which was created for the development of ontologies in enterprise modelling. The diagram shows how two departments with different knowledge structures communicating, being assisted by an *ontology*.

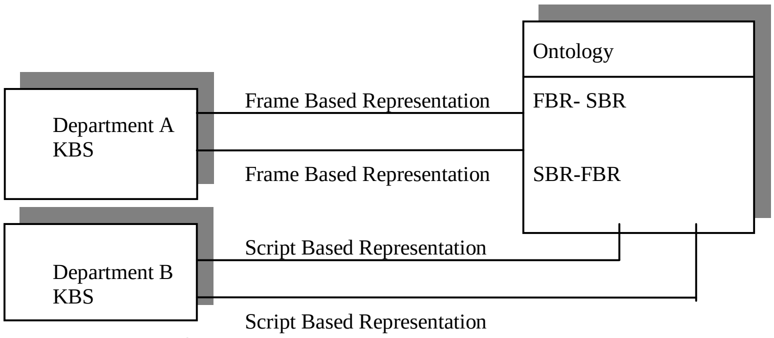

Figure B: Ontology for knowledge sharing

The purpose of every ontology related to enterprise modelling is to support integration within the boundaries of the enterprise by making available a common knowledge representation maximising the communication potential and on the other hand minimising ambiguity and misunderstandings.

TOVE methodology

TOVE (Toronto Virtual Enterprise) is such an ontology aiming to create a common and shared terminology in an enterprise, defines the meaning of each term (semantics) in an easy to understand order, implements the semantics as a set of axioms and finally defines a set of symbols for depicting a term constructed in a graphical form. According to Mark S. Fox [1995] all of the attempts to create a general enterprise model for knowledge representation fail to produce a set of evaluation criteria so that knowledge representation models can be evaluated against. Some of those criteria suggested on the paper are ***Generality, Competence, Efficiency, Perspicuity, Transformability, Extensibility, Granularity*** and finally ***Scalability.*** The first two criteria examine to what degree is the representation shared and how well problem solving is supported. The third and fourth criteria examines whether the representation supports efficient reasoning and whether it is easily understood by the users. The transformability and extensibility criteria make sure that the representation can be transformed into another, more appropriate for a particular decision problem, and can also be extended to encompass new concepts. Granularity ensures that the representation supports reasoning at every

level of abstraction and detail and finally scalability examines whether the representation scale to support large applications. The Philosophy behind *TOVE* is that enterprises are action oriented and therefore the ability to represent action lies at the heart of all enterprises. *TOVE* developers approached the issues raised by the evaluation criteria by defining a generic level representation of activities, time, causality and constraints. In *TOVE* actions are represented by the combination of an activity and its corresponding state. Activities are the basic transformational action primitive with which processes and operations can be represented. *Enabling state* defines what has to be true prior the implementation of an activity and *caused state* defines what has to be true after an activity had been performed. An activity may at any point carry a status, which may be either *dormant*; the activity is idle, *enabled*; the activity is executing *completed*; the activity is finished or *suspended*; the activity has been forced to idle state. The *state* of an activity is another term and indicates what has to be true for an activity to be performed.
These attributes along with some others represent the terminology of the *TOVE model.* The definition of the terminology is in the form of first order logic and implemented in Prolog. An English description of some of these definitions are:

- *An activity can be executed if its enabling state is enabled.*
- *A resource is physically divisible if it has at least one sub-component.*
- *A resource is reusable if it is temporarily divisible in its role in an activity.*

Methodologies for the modeling of C.I.M. environments
There appeared to be two major methodologies for enterprise modelling, I came across during my research. Both emphasised on computer integrated manufacturing modelling. One was developed as an endeavour in enterprise modelling for a CIM (computer integrated manufacturing) factory by the Purdue Laboratory for Applied Industrial Control at Purdue University West Lafayette. The methodology *(PERA)* was developed in order to establish a basis for the treatment of human-implemented functions in a CIM enterprise.

CIM - OSA *(Computer Integrated Manufacturing Open Systems Architecture)* is the other methodology and its major objective is the appropriate integration of enterprise operations by means of efficient information exchange within the enterprise with the help of information technology. The Open Systems Architecture defines an integrated methodology to support all the phases of a CIM life cycle from requirements specification , through system design , implementation operation and maintenance. According to K.D. Tham [Enterprise Integration Laboratory, University of Toronto]

> 'CIM is a new manufacturing paradigm which has been developed over the last decade and has been recognised to be of strategic importance for the European Industries'

Enterprise Modelling Using PERA
The methodology suggests to define a general task representation of the information system tasks the manufacturing tasks and the human based tasks. The methodology was developed in order to assist in the modeling process of computer integrated manufacturing enterprises and its success lies in its ability to develop an overall view of all three categorized tasks as well as the interdependencies amongst them. The functional descriptions of the tasks are divided between the information stream and the manufacturing stream. The first is initiated by the planning ,scheduling ,control and data

management requirements of the enterprise whereas the manufacturing stream is initiated by the physical production requirements of the enterprise. The final version of the models present the enterprise from two different viewpoints; these are the functional view and manufacturing or physical view.

The extent of automation line shows the actual degree of automation carried out or planned in the CIM System of the CIM Business Entity. The location of the extent of automation line has economic, social, customs, laws & directives, union rules, and technological factors in its determination. The extent of automation line may be considered sandwiched between humanizability and automatability lines. The automatability line shows the absolute extent of technology in its capability of actually automating the tasks and functions of the CIM system of the CIM Business Entity. It is limited by the fact that many tasks and functions require human innovation, etc. and cannot be entirely automated with current available technology perhaps. The humanizability line shows the extent to which humans can be used to actually implement the tasks and functions of the CIM system of the CIM Business Entity. It is limited by the human capabilities in speed of response, and human powers of comprehension, vision, strength, etc.

The information architecture of the implemented view basically lends control features like human decision making, human monitoring of information systems to support the enterprise mission. On the other hand, the manufacturing architecture of the implemented view provides production for the fulfillment of the enterprise mission.

Enterprise modelling using CIM-OSA

CIM-OSA's development intended to model the world of the manufacturing enterprises. It does in fact provide an architecture for describing the world of manufacturing enterprises. The modelling process is concentrated on three distinct levels which are the *requirements definition* the *design specification* and the *implementation design*. Using the modelling methodology proposed by *CIM-OSA* a manufacturing enterprise can create a clear view of its requirements.

> 'By ensuring that the physical implementation model is directly procesable by the information technology (IT) components of the system, control of the operation of the CIM systems at run time may be achieved in congruence with the specified behaviour of the enterprise.'[Tham D.K.]

Basic Concepts of CIM-OSA

CIM-OSA does not provide a standard architecture which can be used by all the manufacturing enterprises. Instead it provides a reference architecture from which particular architectures can be derived. In order to select a particular architecture *CIM-OSA* employs a number of structural concepts and architectural principles. According to Klatch the purposes they serve are :

1. to create a modelling framework of the CIM enterprise wholly or partially which distinctly segregates the WHAT (or model of required enterprise functionalities and behaviour) from the HOW (or model of an actual enterprise system implementation) by means of the HOW TO (or model of optimized enterprise system design); and

2. to derive a particular implementation model of the enterprise which is active during the operation of the enterprise system, and is the basis for the computer-controlled execution of the modelled business processes and enterprise activities, thereby providing true computer-integrated manufacturing [Jorysz and Vernadat, 8].

The Functional View

In order to model the functional view of an enterprise as well as *PERA* the information view has to be modelled first. The functional view modelling process will help those who carry out the modelling process to establish a model of the functionality and behaviour of the enterprise in terms of domains, domain processes, enterprise activities and business processes. The model of the functional view describes the structure, content, behaviour and functionality of the entire enterprise. Recall here that *CIM-OSA* is a methodology for establishing the requirements of three levels within a manufacturing enterprise which are definition, design and implementation. The functional model specifies *What* is required in terms of structure, content, behaviour and control, *How* these requirements will be implemented as well as the *Actual* implementation.

The Information View

CIM-OSA employs its own knowledge representation techniques in order to capture the semantics of information in the information view. The technique is divided into four major concepts such as *Generalization, Aggregation, Particularisation or Classification and Generalised Relationships.* Each one of these concepts is described below by Jorysz and Vernadat.

1. Generalisation which enables an individual object type to be thought of as a more generic object type so that type-subtype relationship definitions are established between objects or entities to propagate the ISA hierarchy [Brachman, 3]. An enterprise object can be involved in one or more ISA links as a child of one or more higher-level enterprise objects so that this enterprise objects can inherit properties of two or more super-objects through the phenomena termed multiple property inheritance.

2. Aggregation refers to an abstraction mechanism in which an enterprise object type is regarded as a conjunctive collection of sub-component objects. aggregation defines one-to-many or many-to-many associations between enterprise objects types. This is known as the PARTOF link between enterprise objects [ESPRIT, 6].

3. Particularisation or Classification refers to the abstraction mechanism linking an enterprise object to an enterprise object type. The objects being modelled that share common properties are gathered into classes. The objects of the class are unique in the class and in CIM-OSA, this is known as the MEMBER OF relationship between the objects and their class. Particularisation defines a one-to-one relationship between an enterprise object and its type [ESPRIT, 6].

4. Generalised Relationships refer to all other user-defined relationships between enterprise objects and are referred to as the LINKED TO relationship in the Information View of CIM-OSA [Jorysz and Vernadat].

The Resource View

The resource view, as the name suggests contains all the relevant information about the equipment, hardware software and people. It takes a hierarchical approach in obtaining this information starting of the most abstract task and the resources that are needed it to perform it. The task is then broken down to different levels creating a tree of tasks. Each branch has a number of people allocated on to it and a number of resources associated with it too.

The Organisational View

The organisational view consists of all the relevant information on the responsibilities within the enterprise and allows to gather and structure the different responsibilities for functions, information and resources. Like the functional view it emphasis on the same three levels. At the requirements definition level the view identifies and defines all the responsibilities that must be known within the enterprise for the human decision making process. Responsibilities may be defined for assets i.e. resources capitol etc. or they may be for operational entities such as business processes, products and services. At the design specification level, the responsibilities for assets and operational entities have to be organised in an optimised and balanced manner in order to satisfy the decision making needs of the enterprise. Finally in the implementation level the organisational view described the responsibilities for configurations of the real physical equipment that realises the enterprise operations.

Assessment of methods and further considerations

Ontologies and the TOVE model propose a structured approach to enterprise integration. The success of the approach lies in the quality of the ontology and how well can it support communication within an enterprise. The construction of ontologies is not an easy task. It requires good knowledge of the enterprises' operations as well as the resources, time and cost. The definition of terminology have to be consistent, otherwise the communication protocol –the ontology provides- would produce ambiguous results. Consistency and obedience to the definitions and symbols has to be forced so that the ontology enables knowledge representations being communicated between departments.

On the other hand methods for computer integrated manufacturing environments also require detailed knowledge of the activities of an enterprise. In order to develop a *functional view* of an enterprise using *CIM-OSA,* the analyst must have good knowledge of the domains, domains processes, enterprise activities as well as business processes. Furthermore in order to develop the model up to the *information view* level additional information related directly to each task, resources, time and labour is necessary.

The main objective of this view, is to create a model, of the relationships between enterprises, with the ability to predict their impact on enterprises' operations, either during creation or deletion. The methodologies and techniques proposed by other researchers on the subject, emphasise on the internal operations of an enterprise and show how these links between departments affect their operations. This review however, is concerned with external links; those between enterprises themselves. Using the *CIM-OSA* concepts, we're only concerned with the *information view* of the network of the enterprises as this information is the key for assessing the impact these links have on enterprises and on the network as a whole.

Ontologies could be used in case this information was known or easy to find. However when we refer to links between enterprises as opposed to links between departments within an enterprise, we really refer to confidential information. Most of the major enterprises would be reluctant to share their information with analysts who try to create a model of network dependencies. This wouldn't be the case within an enterprise hence common terminology and definitions can be produced for the sake of communication. Therefore, we need to consider further modelling techniques for the modelling of the external factors that seem to affect an enterprises operations; and these of course are the relationships established between enterprises themselves.

Conclusion

The paper presented a general view of the research currently undergoing in the area of enterprise modeling. Enterprise modeling of course, is a broad area and receives attention from many academic disciplines. The area of particular interest in this review is the communication procedures between the departments of an enterprise or enterprises themselves. The way these *links* -either internal or external- are perceived has a great impact on the enterprises operations. Having established a clear picture of the methodologies, techniques and suggestions for enabling communication between departments within an enterprise, we’ve examined during this review, the possible ways of modeling the links between enterprises themselves and how they affect the decision making process.

References

1. Michael Gruniger and Javier A. Pinto 1996 / A theory of complex actions for enterprise modelling. Department of Industrial engineering. Toronto

2. Mark Fox, John F. Chionglo, Fadi G. Fadel 1996/ A common sense model of the enterprise. Department of Industrial engineering. Toronto

3. M.Gruninger M.Fox 1996 / The logic of enterprise modelling. Department of Industrial engineering. Toronto

4. M.Gruninger M.Fox 1994 / Enterprise modelling. Department of Industrial engineering. Toronto

5. Panayiotis Periorellis, Albert Bokma, Peter Smith 1997/ An Object Oriented approach for the modelling of enterprise networks. Tools97 2nd International conference proceedings on object oriented experiences

6. K. Donald Tham 1995 / CIM-OSA enterprise modelling . Enterprise integration laboratory, University of Toronto.

7. K. Donald Tham 1995 / PERA methodology . Enterprise integration laboratory, University of Toronto.