COMMON ENTERPRISE MODELLING FRAMEWORK FOR DISTRIBUTED ORGANISATIONS

Giuseppe Berio[1], Kai Mertins[2], Frank Walter Jaekel[2]

[1] *Dipartimento di Informatica, Università di Torino, C.so Svizzera 185, 10149 Torino, Italy*
berio@di.unito.it
[2] *Division: Corporate Management, Fraunhofer IPK-Berlin, Pascal Str. 9-8, 12045 Berlin, Germany*
{kai.mertins, frank-walter.jaekel} @ipk.fhg.de

Abstract: The mission of **Common Enterprise Modelling Framework for Distributed Organisations (CEMF-DO)** is to provide, within distributed organisations, a common framework in which modelling, simulation, analysis and management of models for designing interoperability solutions are explored and integrated. The paper presents activities related to the work on a unify enterprise modelling language (UEML) and on the management and synchronisation of enterprise models (SDDEM) performed as one work package within the Network of Excellence INTEROP (508011).

Keywords: Business process engineering, Distributed models, Distributed simulation, Information systems, Model management, Process models, Synchronization

1. INTRODUCTION

Distributed organisations are characterised by their autonomy ranging from decisional (control) processes to more operational processes. Concerning this kind of organisations, enterprise engineering and enterprise integration are both confronted to the concept of *interoperability*. In fact, despite of the many definitions of interoperability, from narrower to broader scopes, (Vernadat 2002; Chen and Vernadat 2002), one of the underlying and common key points to these definitions is related to the fact that distributed organisations (as distribute systems) are composed of various interrelated enterprises (inter-interoperability) and/or various units (intra-interoperability), which evolve in an autonomous manner. Therefore, the major question is how to allow enterprises part of a distributed organisation to evolve *autonomously* (high *heterogeneity* and *independence*) while reducing the overhead of their integration due to the required (continuous) alignment of IT infrastructures, IT systems and internal organisations to new situations.

The paper describes the Common Enterprise Modelling Framework for Distributed Organisations (CEMF-DO) as it has been defined in the INTEROP Network of Excellence project, funded by the European Commissions (http://www.interop-noe.org/). The aim of CEMF-DO is to provide the enterprises with information about how their distributed organisation should be defined in order to reduce as much as possible the overhead due to their integration, as explained above. In other words, to make distributed organisations more interoperable.

The CEMF-DO deals with two levels of the problems:

- a common language which is called UEML (Unified Enterprise Modelling Language);
- the **s**ynchronisation and management of **d**ifferent **d**istributed **e**nterprise **m**odels (SDDEM) provides enterprise model approaches for interoperability.

The framework is mainly related to enterprise models. Its aim is related to understand how a distributed organisation is working. In this sense, CEMF-DO is less related to the integration between IT systems and infrastructures. However, enterprise models are usually a pre-requisite to build meaningful integration of IT systems and infrastructures. The paper provides examples of possible application. Specifically, these examples are real world case studies dealing with supply chain.

2. THE FRAMEWORK

The CEMF-DO mission is to provide, within distributed organisations, a common framework in which modelling, simulation, analysis, management for designing interoperability solutions are explored and integrated.

Distributed organisations are characterised by distributed modelling and distributed heterogeneous models. Therefore, these distinct models and distributed modelling need clear procedures, standards and easy ways to adapt, build, connect, manage and operate models in heterogeneous, distributed simulation and execution environments, and compositional approaches to analysis of models.

The framework will be based on a common language for enterprise modelling. More specifically, the meta-model resulting from 5th FP UEML Project (Unified Enterprise Modelling Language, IST - 2001 – 34229) will be maintained and adjusted with respect to other INTEROP activities.

The expected result is a framework containing:

- A widely accepted Unified Enterprise Modelling Language (UEML) as the base for modelling with its reusable and traceable "Strategy for UEML Extension",
- Procedures for model management, model interconnection and model synchronisation,
- Showcases illustrating the use and benefits of the framework as well as gaps for further development,
- Easy and goal-oriented access to the information within the framework by a interactive role based representation.

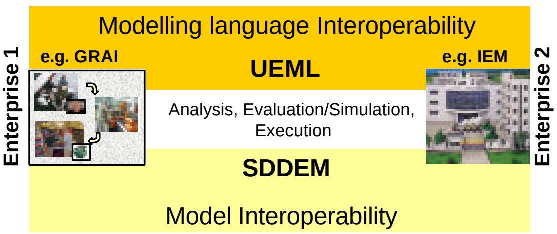

Fig. 1. CEMF-DO structure

The CEMF-DO presented in the paper focuses on two main axis "common enterprise modelling language" and "synchronisation and management of distributed enterprise models" (Fig. 1). These axis go across three key topics for interoperability:

- Enterprise modelling,
- Software architectures and enabling software technologies,
- Ontologies.

This includes respectively business, knowledge, IT infrastructures and systems and "semantic" aspects of interoperability. Experiments and showcases will be developed in the project. They will illustrate the relations between this axis and the benefits of the framework.

2.1 Modelling Language (for) Interoperability

The objective within this axis is to deliver an *intermediate language* for supporting distributed heterogeneous enterprise models and enterprise modelling. The starting points are three of the major outcomes of the 5th FP - UEML Project (Berio, *et al.*, 2003; Berio, *et al.*, 2004; Panetto, *et al.*, 2004):

- The UEML 1.0 meta-model represented in UML 1.5 (www.uml.org),
- The "Strategy for UEML" and,
- The set of collected requirements (Knothe *et al.*, 2003).

The current organisation within the axis is based on the following three main kinds of activity:

- Requirements,
- Languages,
- Approaches.

Requirements activities should continuously assess and adjust the UEML goals and provide the detailed objectives of the UEML (for instance, in term of application domains). The *current goals of UEML* is defined as:

- To provide an **intermediate language (also called *exchange format* or *inter-lingua*)** which supports integration, composition and transformation of information (i.e. models) represented in **different (enterprise) modelling languages which is the base for exchange information between distinct (enterprise) modelling tools**,
- To enable the achievement (through some techniques) of the ***global consistency*** between the various entreprise models (which means that one model does not contradict another model or there is way to justify a model in term of another one).

In this aim, UEML should support two kinds of operation on enterprise models:

- Manipulations of models (i.e. integration, transformation and composition),
- Verification and validation of models.

However, according to the UEML perspective, these operations should be put in practice by using some external specific techniques (see Section 2.2 below, on model interoperability).

Requirements are very important because constitutes the preferred mean for making the influence on UEML of Researchers, Enterprises and other relevant Entities a reality. This influence. Allows to state the major problems to be solved (or partially solved), or barriers in application concerning enterprise models and modelling:

- Conveys new ideas about modelling the enterprises which can be capitalised,
- Collects experiences about relevant existing researches/tools/languages.

Due to complexity and to openness of the community potentially involved, requirements cannot directly be used to build a UEML. In fact, requirements bring together distinct visions, distinct granularity, distinct knowledge and experience: while interesting, they are difficult to be used as they are. Therefore, UEML should mainly be built according to the existing languages and tools. Requirements can better used for making the resulting UEML:

- *More complete*, without the need of analysing all the languages and tools that exist today and in the future,
- *More shared*, i.e. making the UEML more consensus driven.

The activities around languages are for continuously collecting information about existing languages. However, knowledge about tools is also required because some tools do not use an explicit language: in this sense, the language underlying the tool should be reengineered. This important set of activities guarantees:

- The relationship with existing initiatives and products concerning at some extent the enterprise modelling,
- The reuse of the knowledge about existing languages and tools.

Another relevant set of activities related to languages is about criteria for deciding when a language is relevant to UEML (i.e., in principle, to be taken into account for building a UEML). These activities make UEML:

- *More consensus driven*, through the explicit set of criteria, and
- *More consistent* in its content, because not relevant languages would not be taken into account.

In fact, due to the many initiatives (Enterprise Architectures (e.g. Open Group, 2000), new IT concepts (e.g. web services) and new IT Infrastructures (e.g. workflow management systems)) related, at some extent, to languages, an important aspect is to position these languages in the CEMF-DO. In fact, these languages probably address distinct phases in the (classical) *lifecycle* (i.e. requirement, design, implementation). In this sense, we may identify three possible kinds of influence of existing languages on UEML:

- *Forward influence*,
- *Horizontal influence*,
- *Backward influence*.

Forward influence: The relationships between the languages for representing *business models* (which mainly describe the economic system underlying the distributed organisation and its economic value (e.g. REA (McCarthy, 1982)) and, for instance, languages for *operational models* describing, for instance, real enterprise processes (already included in UEML1.0). Specifically, the forward influence concerns the link allowing to understand if business models are correctly realised and followed in the working enterprise environment.

Horizontal influence: This is the most important influence. Specifically, it addresses languages already recognised as and used for enterprise modelling (for instance, GRAI (Doumeingts, *et al.*, 1998).

Backward influence: The relationships from *languages for IT infrastructures and systems*. Specifically, the backward influence concerns the information required for making the IT infrastructures and systems models.

One of the basic argument which makes models and modelling poorly accepted in practical work environments, is related to the fact that *models quickly become useless*: they are not used in the daily working environment, they are specifically built and need to be specifically maintained. This results in additional costs. Therefore the backward influence can be perceived in two ways:

- If IT infrastructures and systems are built according to some principles, compatible principles should be for the enterprise models (based on analogous observations stated in software engineering and information systems engineering; modelling does not change in its basic principles!),
- *Enhanced IT infrastructures* may support the enactment or direct usage of enterprise models (eventually, enhanced IT infrastructures turn models into programs and modelling into programming).

Finally, the UEML should be defined in a suitable way. This is mainly due to three points:

- Any UEML is a "*living entity*" and probably needs to continuously evolve,
- There are probably distinct ways to use outcomes from requirements and languages activities for making a UEML,
- The UEML language itself should be well built according to some *general criteria*.

For these reasons, activities around the approaches should provide:

- How to define an UEML by using the outcomes of activities related to languages and requirements;
- A model of how to define a UEML, to guarantee the required "strategy for evolving UEML".

For instance, an ontology can firstly be defined or selected (i.e. understanding the basic phenomena that need to be addressed and distinguished in the language). Alternatively, it is possible to define an Enterprise Reference Architecture, then focusing on the language: in this case, the specific ERA serves as a base for clarifying the context, then for analysing specific phenomena to be represented. Meta-models and meta-modelling is another alternative approach to provide a language, mainly focusing on the *abstract syntax*.

2.2 Distributed Models (for) Interoperability

The need of taking into account different models is the result of slightly different objectives or different requirements for enterprise models. The synchronisation and management of these different distributed enterprise models (SDDEM) is required under several aspects (e.g. consistence of information modelled in different models, responsibilities for modelling and model maintenance, security of model information, extraction of knowledge represented in the model, configuration of IT systems across organisations, flexible adaptation to different enterprise networks, etc.). The models have to be selectable and translatable. The coordination and operation of the models has to be easy. Clear procedures, standards and easy ways to adapt, build, connect and operate models in a heterogeneous, distributed environment should be provided.

The connection of different and distinct enterprise models is not limited to "technical" or modelling language problems because, for instance, the same "enterprise process" modelled with the same modelling language under the same objectives and requirements may differ whenever modelled by distinct persons. Clear procedures stating how to manage and synchronise such models will help to increase the acceptance and use of enterprise models. It will also address the interoperation between enterprise models. The SDDEM approach should develop and establish a enterprise model concept(s) for interoperability taken into account ontologies, modelling templates, modelling procedures as well as the configuration of IT platforms.

Enterprise modelling is usually clearly target orientated (e.g. network analysis and optimisation, simulation, IT system implementation or configuration, quality management, etc.). Therefore, the information covered by the model are different and the enterprise model concept for interoperability has to be taken into account the different needs required by the different targets. Nevertheless, general concepts can be proposed like modelling templates, semantic of terms, natural language translation mechanism, modelling procedures etc.. This concepts can be specialised to target specific concepts. For example in one case study described below templates based on Supply Chain Operational Reference Model (SCOR) are used to model the companies within the supply chain. The first results of the work on SDDEM are

- Definition of the relations between UEML and SDDEM and of the needs of UEML within SDDEM,
- A framework of approaches around SDDEM,
- A show case of benefits and gaps in this area.

Interoperability issues addressed in the approach:

- Enterprise modelling to support networked enterprises in the context of interoperability,
- Case studies of enterprise model support for collaborative enterprises,
- Management of distributed enterprise models,
- Templates and ontologies for the coordination of distributed enterprise models,
- Synchronisation and interoperability of distributed enterprise models,
- Architectures to coordinate and synchronise different enterprise models within and between enterprises,
- Evaluation and analysis of enterprise models between companies,
- Execution of distributed enterprise models within different tools (e.g. distributed simulation),
- Reusable and interoperable building blocks to archive executable distributed analysis scenarios,
- Procedures for enterprise modelling involving more companies to improve the interconnection of the companies and their interoperability,
- Monitoring of networked enterprises based on enterprise models.

3. CASES STUDIES

The cases studies illustrate the use of enterprise modelling across company borders and the need of enterprise model concept(s) for interoperability as well as the need of common basis modelling languages. The first case study is an actual running EU IST project 507601 SPIDER-WIN. (www.spider-win.de) It applies enterprise modelling within single enterprises and along three different supply chains to archive the targets of the project. The approach requires the combination of different single enterprise models as well as the derivation of a general model from three regional supply chain models. Based on the model approach a software system will be implemented and configured. This system will support the order management related interoperability between the companies within the supply chain. In a second step the idea is to use the model to support the maintenance of the IT system based on configuration features.

The modelling within SPIDER-WIN starts from scratch. Therefore it was possible to select one unique modelling language for the whole approach. In a opposite situation of already existing models in heterogeneous modelling languages in the different enterprises the problem would be much more complicated. This situation could be arise for example if an enterprise participates in different supply chain networks. Here UEML would be help to harmonise the different languages. However also the work on the different models needs concepts to archive compliant model structures which can be managed, combined and analysed. In SPIDER-WIN some mechanism are introduced like templates, semantics of terms, definition of interface objects as well as modelling procedures.

The second approach deals with interoperability of different enterprise simulation models (Rabe and Jaekel, 2000; Rabe and Jaekel, 2004). This approach uses modelling on different levels. The first level is the definition of simulation templates which includes simulation models. The template description includes the definition of possible interfaces with other simulation templates (Mertins, *et al.*, 2000). The second level is the definition of simulation scenarios based on the predefined templates. The third level is the execution of the different models as federates in a running simulation scenario. The simulation scenario is configured by the model of the second level. This concept requires clear definitions of the interfaces between the different scenarios as well as a clear semantic. It also requires suitable modelling languages and tools. Finally it needs a configurable execution architecture which allows the synchronisation of the federates and the data exchange between the federates (simulation models).

3.1 First Case: Supply Chain Analysis Across Companies & Regions

In the SPIDER-WIN project different persons work on different models in different companies within different European regions under the same requirements. Therefore concepts are developed to archive compliant model structures as well as a common understanding of the model structures.

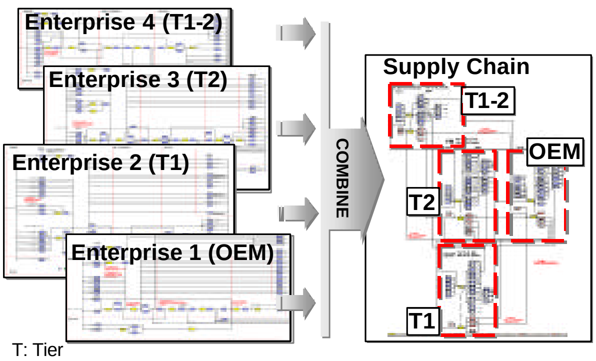

Fig. 2: Model approach for interoperability

The modelling approach starts from scratch therefore one unique modelling language and modelling tool was introduced (the IEM method and the modelling tool MO²GO (Mertins and Jochem, 1999)). Independent from the modelling language it can be arise

- different interpretations of the model,
- different structuring of the model,
- differences in the modelled scope and content,
- different naming of the same concepts,
- different understanding of the modelling targets,
- different interface descriptions.

The flexibility of the enterprise modelling methods is usually a benefit because of less restrictions for the application domain. However, in a high distributed modelling approach this flexibility could be a danger because of not compliant models. The main problems are related to different European regions, languages, cultures, supply chains, industries and therefore to different business processes.

Modelling templates has been introduced based on SCOR to increase the comparability of the single enterprise models. The templates define some kind of ontology (terms, semantic of the terms and structure information) in order to support that all information and requirements detected can be systematically documented within one consistent overall model (Fig. 2). This general model is generated in English, for exchange and dissemination reasons. However the local national models can be in the native languages. The model allows to switch between the native language and English. The study has the following achievements:

- Common understanding of objects within the interfaces of the models,
- Clear interface descriptions between the models using unique names,
- Similar interpretation of the processes related to external interfaces of the enterprise model,
- Guideline to collect similar data within the different models,
- Support for the development of supply chain models,
- Distributed development of different enterprise models and analysis of a structure including all these models in order to realise a suitable organisational and software support (services).

The modelling approach within SPIDER-WIN also shows limitations and gaps:

- The approach does not solve problems of different cultures. This is still a hard job to coordinate the different ways of thinking expressed by the models.
- Human issues are also not addressed but identified. Still, the models reflect the background and private targets of the modeller.
- The approach does not take into account security issues. It is identified that some categories of information are exchanged but not addressed in the model, because of trust or other security issues.
- It is still a hard job to manage the extensions of the templates and guidelines in order to have a consistent and synchronous structure of models.
- It is quite hard to get detailed information about performance indicators and detailed process structures from the companies.
- During the modelling phases different levels of detail are identified between the three supply chains.
- The interview guideline has to be adjustable to the experiences of the modeller and to the modelling task.

3.2 Second Case: Federated Supply Chain Simulation

The approach focuses on distributed and federated simulation between different enterprises along the supply chain network. It includes scenario modelling on an enterprise business level and configuration of runtime scenarios based on the model.

A design methodology of data exchange across different federates including graphical notations, XML notation for the configuration of an execution platform based on the graphical design and of the execution of the federates within a federated scenario is part of the concept. This is already illustrated by a prototype developed in the MISSION project (EP 29 656) (http://www.ims-mission.de/). The prototype also includes a communication platform based on the High Level Architecture (HLA) (IEEE 1516-2000, 2000; Kuhl, *et al*., 1999). The execution platform concept consists of 4 layers: communication platform, generic adapter, tool interface, and a tool executed as a federate. The generic adapter hides the communication platform and the tool interface allows the configuration of the data exchange. This concept is the basis for the integration of different tools and models into federations. Ontology aspects are addressed in the description of "exchange objects". "Exchange objects" are predefined objects for the data exchange between different simulation models and tools.

The limitation and gaps of this case study are similar to the first one. In addition missing standardisation is the main limitation of the approach. Standards are needed for:

- Exchange Object Structure: Description of the objects exchanged within a distributed simulation environment,
- Federate Configuration File format: Reduces reprogramming effort for simulation tool interfaces in an HLA runtime environment. Different approaches have to be synchronised in this field,
- Simulation Parameters: The MISSION approach identifies missing harmonisation in the definition of simulation parameters,
- Ownership Mechanism: Incompatible approaches to adapt the HLA ownership mechanism for industrial usage,
- Generic Adapters: Interfaces to adapters should be standardised in order to reduce the investment for distributed simulation environments.

4. CONCLUSION

The case studies illustrate that some approaches already exist for the management and synchronisation of distributed enterprise models. However, this approaches are heterogeneous and sometimes incompatible. In the actual situation a solution will be just created under the special modelling requirements. This requires more work on general concepts and standards for the enterprise model approaches for interoperability. The case studies illustrate also that SDDEM covers more than the "synchronisation" of models. The model management is also included in this approach.

In general UEML is important to easier achieve standards for SDDEM and to allow the coordination of models across different modelling languages.

REFERENCES

Berio, G., et al. (2003). Deliverable 3.1. The UEML Project, www.ueml.org.

Berio, G., V. Anaya, A. Ortiz (2004). Supporting Enterprise Integration Through a Unified Enterprise Modelling Language. In *Proceedings of EMOI Workshop*, joint with CaiSE04, Riga, Latvia.

Chen, D., F.B. Vernadat (2002). Enterprise Interoperability: A Standardisation View. In *Proceedings of ICEIMT02*, Kluwer, Valencia, Spain.

Doumeingts, G., B. Vallespir and D. Chen (1998). Decision modelling GRAI grid, Chapter in: P. Bernus, K. Mertins, G. Schmidt (Eds.) *Handbook on architecture for Information Systems*, Springer-Verlag.

IEEE 1516-2000 (2000). Standard for Modeling and Simulation (M&S) High Level Architecture (HLA).

Knothe, T., et al. (2003). Deliverable 2.3. The UEML Project, www.ueml.org.

Kuhl, F., R. Weatherly, J. Dahmann (1999). *Creating Computer Simulation Systems. An Introduction to the High Level Architecture*, Prentice-Hall.

McCarthy, W. E. (1982). *The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review*, **Vol. LVII, No. 3**, July.

Mertins, K., R. Jochem (1999). *Quality-Oriented Design of Business Processes*. Kluwer Academic Publishers, Boston.

Mertins, K., M. Rabe, F.-W. Jäkel (2000). Neutral Template Libraries for Efficient Distributed Simulation within a Manufacturing System Engineering Platform. In: *Winter Simulation Conference*, pp. 1549-1557, Orlando (USA).

Open Group, (2000). *TOGAF: The Open Group Architecture Framework*, Document n. 1910, Version 6.

Panetto, H., G. Berio, K. Benali, N. Boudjlida, M. Petit (2004). A Unified Enterprise Modelling Language for Enhanced Interoperability of Enterprise Models. In *Proceedings of 11th IFAC Symposium on Information Control Problems in Manufacturing*, Salvador, Brasil.

Rabe, M., F.-W. Jäkel (2000). Simulation for Globally Distributed Enterprises. In: *12th European Simulation Symposium (ESS)*, pp. 322-327, Hamburg, 2000.

Rabe, M., F.-W. Jäkel (2004). Distributed Simulation and Supply Chains. In *International Conference on Competitive Manufacturing (COMA'04)*, pp. 437-442, Stellenbosch, South Africa, 2004.

Vernadat, F.B. (2002). Enterprise Modelling and Integration; From Fact Modelling to Enterprise Interoperability. In *Proceedings of ICEIMT02*, Valencia, Kluwer, Spain.