



GRAI-ICE Model Driven Interoperability Architecture for Developing Interoperable ESA

Lanshun Nie, Xiaofei Xu, David Chen, Gregory Zacharewicz, Dechen Zhan

► To cite this version:

Lanshun Nie, Xiaofei Xu, David Chen, Gregory Zacharewicz, Dechen Zhan. GRAI-ICE Model Driven Interoperability Architecture for Developing Interoperable ESA. The international conference on Interoperability for Enterprise Software and Applications, Apr 2010, Coventry, United Kingdom. pp.111-122, 10.1007/978-1-84996-257-5_11 . hal-00479808

HAL Id: hal-00479808

<https://hal.science/hal-00479808>

Submitted on 4 Sep 2018

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.

GRAI-ICE Model Driven Interoperability Architecture for Developing Interoperable ESA

Lanshun Nie¹, Xiaofei Xu¹, David Chen², Gregory Zacharewicz², Dechen Zhan¹

¹ School of Computer Science and Technology, Harbin Institute of Technology, 150001 Harbin, China.

² IMS/LAPS/GRAI, University Bordeaux 1, 351 Cours de la Liberation, 33405 Talence Cedex, France.

Abstract. This paper presents GRAI-ICE Model Driven Interoperability Architecture (MDI) which is developed based on MDA (Model Driven Architecture) of OMG and some initial works performed in INTEROP NoE. This MDI architecture aims at supporting the development of changeable on-demand and interoperable ESA (Enterprise Software Application). The architecture defined five modelling levels, i.e., Top CIM, Bottom CIM, Object oriented PIM, Pattern oriented PSM, and Component and configuration oriented CODE. This paper presents in detail core concepts and rational of each modeling level. An application example in nuclear equipment industry is outlined.

Keywords: MDI, MDA, ESA, Interoperability, Architecture, Model transformation

1. Introduction

Today, with the increasing pressure from competitive market, enterprises particularly SMEs are obliged to work in network in order to meet the changeable and customized demand. ESA (Enterprise Software Application) supporting the business should be adaptable and flexible to the business change. In such networking approach and with the development of the globalization of the economy, enterprises have to cooperate with partners such as material suppliers, service suppliers, distributor, retailer and public administration to provide high-quality products and services to customers. In order to cooperate, it is required for enterprises to have capabilities of interoperability in the level of both business and IT/software systems, i.e. ESA. Changeable on-demand and interoperable ESA brings great challenge to software architecture, implementation, deployment, in a word, software development [1, 2]. This paper demonstrates how MDI (Model Driven Interoperability) contribute to these developments.

1.1. Problems and Research Objectives

Code is the core artefact in traditional software development process. This process is usually very long, and the cost is usually very high. It is not rapid and adaptative enough to develop changeable on-demand interoperable ESA. Since years there is a shift from code-based software development to model-based software development. The most known approach is MDA (Model Driven Architecture)[3]. In the objective of tackling interoperability at each level of life cycle development, MDI (Model Driven Interoperability) architecture has been introduced by generalising the steps of MDA. It deals with interoperability between ESAs during their respective developments. In this new methodology, conceptual models of enterprise interoperability are elaborated to facilitate the interoperability at the coding level. Software development starts from business level where a Computing Independent Model (CIM) is elaborated and then transformed into executable ESA system by automatic/semi-automatic transformation and code generation tools.

There are still some research questions due to the fact that MDA, and by heritage, MDI only defines the framework and some guiding rules, but not detailed method to follow, nor modelling languages. For example enterprise modelling techniques are not well considered at the CIM level. Should the CIM level be further divided in order to model both business and software requirements? The interoperability among adjacent levels is also not sufficiently studied.

The research objective is to address the problems mentioned above through the elaboration of a Model Driven Interoperability Architecture (entitled GRAI-ICE MDI) taking into account the MDA and some initial works performed in INTEROP NoE. This MDI architecture aims at defining both the modelling levels of the architecture and model transformation method between adjacent levels. The proposed GRAI-ICE MDI will support the development of interoperable, configurable and executable next generation ESA.

1.2. Background and Related Works

Model Driven Interoperability (MDI) architecture [2, 3] aims at decreasing the complexity and difficulty of interoperable software system development by decomposing complex, abstract system into multiple levels. The basic ideas are: (1) decoupling the requirement specification and function specification from implementation specification in specific platform; (2) establishing computation independent model (CIM) which describes business requirement, platform independent model (PIM) which describes design of software, and platform specific model (PSM) which describes implementation mechanism of system; (3) transformation and mapping from upper level model to lower level model, refining the model and automatically generating software from models [4,5].

It has been recommended for modelling using languages and technologies such as UML, MOF and CWM, and transforming models using technologies such as meta-model and pattern [6, 7]. However, MDI only defined some guiding rules, but not the details such as concrete modeling elements, modeling methodology and model transformation method [8]. Consequently, it is necessary to select or refine modeling method(s) and organize them consistently.

Researchers have studied some most known existing modeling methodology, language and tools such as CIMOSA [9], GRAI [10] and IDEF[11]. UML (Unified Modeling Language) is the most widely used modeling language for software specification [11]. The former is suitable for business level, i.e. CIM level. The latter is suitable for software oriented modelling, i.e. PIM and PSM levels. However, there still exists a big semantic gap between them and no existing approach shows how to integrate them to a consistent architecture. Therefore, the question is still open on how to select, from the existing business modeling methods and software modeling methods, adequate formalisms, to refine, combine and integrate them into MDI architecture.

Motivated by the requirement of developing changeable on-demand and interoperable ESA, and based on some existing modeling languages, a Model Driven Interoperability architecture, entitled GRAI-ICE MDI architecture, is proposed. GRAI refers to GRAI methodology and is used at CIM level to describe business user oriented requirements. ICE means Interoperable Configurable and Executable. This approach is based on some initial works performed in INTEROP-NoE and further developed within the frame of INTEROP VLab through a research collaboration between University Bordeaux 1 (IMS/LAPS) and Harbin Institute of Technology (HIT) and also the company GFL. It aims at defining concrete modelling elements (concepts) and relations among them at all levels of MDI architecture so that the model of ESA can be described precisely and consistently by these elements. This architecture supports model transformation top-down (CIM→PIM→PSM→CODE), and from code to system deployment.

Also performed in INTEROP NoE, R. Grangel etc. proposed a similar MDI framework. They broke down CIM level into two sublevels, i.e. top-level CIM and bottom-level CIM. Besides this, their research work focus on transformation from top-level CIM model to bottom-level CIM model, with an example of transforming GRAI extended actigrams into UML activity diagrams[12,13].

1.3. Purpose and Structure of the Paper

The purpose of the paper is to present the preliminary research results on Model Driven Interoperability architecture for developing Interoperable ESA. In this paper focus is on the set of modelling methods and languages defined in MDI. The related transformation methods will be presented in a forthcoming paper.

After the introduction given in section 1, the global GRAI-ICE MDI architecture is presented and explained in section 2. Then, section 3 to section 5 will explain the problems, target of modelling, core concepts and relations, as well as modelling methods used at the three main levels (CIM, PIM, and PSM). Section 6 presents the last level of the architecture (ESA component and system level) and outlines the principle of model transformation and supporting tools. An application case is discussed in section 7. Finally section 8 and 9 respectively conclude the paper and discuss on future perspectives.

2. Overview of GRAI-ICE MDI Architecture

GRAI-ICE MDI follows the basic structure of MDA proposed by OMG. It includes four modelling levels and one system level. It also includes transformation methods between these levels. The global view of GRAI-ICE MDI architecture is shown Fig.1. Note that CIM level is divided into two sub-levels: Top CIM (business oriented CIM) and Bottom CIM (ESA oriented CIM).

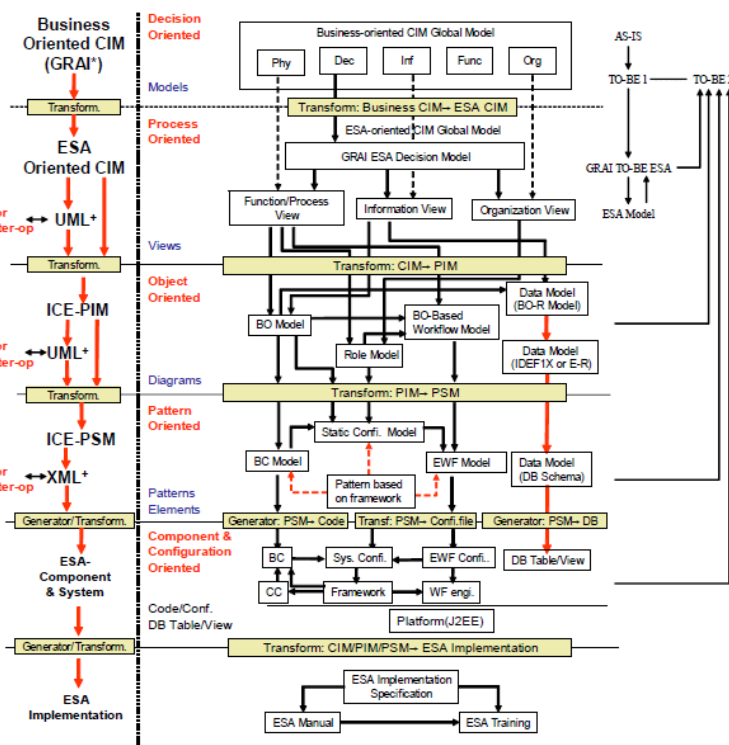


Fig. 1. GRAI-ICE MDI architecture

Top CIM, which is business oriented and based on GRAI methodology, is a set of models focusing on the description of decision making process, business process and physical product flow process. Bottom CIM, which is ESA oriented, extracts from Top CIM and describes more in detail business requirements which must be supported by ESA, i.e. function/process, information, organization and decision making structure.

ICE-PIM is concerned with the object-oriented design of ESA. It focuses on identifying and describing business objects (BO). It includes BO model, BO-Based workflow model, role model, and data model.

ICE-PSM, which describes the implementation of ESA, is pattern oriented and XML based. It includes business component (BC) model, static configuration model, executable workflow model and relational data model.

ESA system is assembled by business components. It provides service to users by parsing the executable workflow model and scheduling function of business components supported by workflow engine.

3. TOP CIM and Bottom CIM

There are two main modelling objectives at CIM level. On the one hand, CIM supports business process modeling, analysis, and optimization, with business people as the stakeholder. On the other hand, CIM also supports the modeling of ESA requirements, with business people and software developer as the stakeholders. Consequently the CIM level of GRAI-ICE MDI is further divided in two sub-levels: Top CIM (business oriented) and Bottom CIM (ESA oriented).

Business oriented CIM aims at modeling three kinds of process (or flow). The first is physical process and physical product flow, which supports the analysis and optimization of the time, cost, quality and needed services. The second is business process and document flow, which supports managing, controlling and recording physical process. The third is decision process and order/plan flow which aims at controlling and coordinating physical process and resources. Besides, it is also necessary to describe enterprise organization and decision-making structure.

The core modeling concepts/elements of business oriented CIM are: (1) Physical Product, (2) Information, (3) Function/Action, (4) Resource/Facility, (5) Organization, (6) QV (Quality-Quantity-Value points): the point or the location where input, output or support should be measured, verified, recorded, controlled, processed, or reported, (7) Connection Sign. The models at business oriented CIM level are composed by these elements and their relations. The meta-model of business oriented CIM is shown Fig. 2

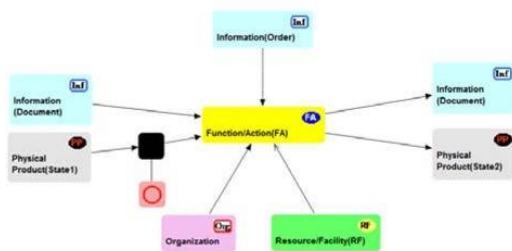


Fig. 2. Metamodel of business oriented CIM

All elements can be further classified into sub-elements. For example, the element “Function/Action” is classified into “function”, “business action”, “decision action” and “physical action”. The element “Information” is classified into

“document”, “order”, “event” and “report”. The view of CIM model can be exactly described by the combination of these elements and sub-class of them. GRAI-ICE CIM model is composed of three kinds of process diagram and some UML class diagram for information, organization, resource, etc.

Physical process model describes the transformation and state transition of physical product in manufacturing. It includes six kinds of basic elements, i.e. product, action, order, resource, organization and QOV. **Business process** controlling/recording physical process describes the transformation and state transition of documents, such as receipts, orders, reports. It includes four kinds of basic elements, i.e. information/documents, order, action, and organization. **Decision process** model is based on GRID model of GRAI methodology. It has two dimensions. One is the decision level, for example strategic, tactics, operational levels. The other is the functions. The decision process model describes both the decision making process and the results which coordinate various functions and control other decision making processes.

Information entities and the relations can be identified by analyzing the information flow embedded in business process to build information model. Also, organization model is also needed to construct at this level. Relation inside each kind of elements, such as information, organization, function, product, and resource, is described using UML class diagram.

Analysis and simulation can be performed on business oriented CIM (AS-IS) of existing system. Then, by optimization and re-engineering, the TO-BE CIM can be obtained

Business requirement for ESA is analyzed on the basis of TO-BE CIM. The result is ESA oriented CIM which focus on software requirement definition. For example, the physical process is not concerned since ESA does not control this process directly. We refine models extracted from TO-BE CIM and distinguish those activities (processes and information) which will be processed by machine or human with human-machine interaction. Due to the limited space, the specification of ESA oriented CIM is not described in detail in this paper.

4. ICE-PIM

ICE-PIM is based on object-oriented approach, UML and workflow technology. The core concept BO is proposed for integrating both the process oriented model and object oriented software implementation. ICE-PIM is the bridge connecting business model and software implementation.

At this level, information entities in the business process of CIM are abstracted and formalized as Business Objects (BO) describing the data structure, data dependency and relations. From the perspective of users, BO is an information entity with its state transitions and operations. From the perspective of software implementation, BO is an integrated object with unique identity, a lifecycle, several data sets and several operation sets. For example, the abstraction of a purchasing order is a BO. It provides a good basis for component-based software development. As mentioned above, CIM is more process oriented, in addition to

BO, it is necessary to transform business process of CIM into workflow model. In other words, PIM mainly consists of business objects model and workflow model.

The meta-model of ICE-PIM is shown in Fig. 3.

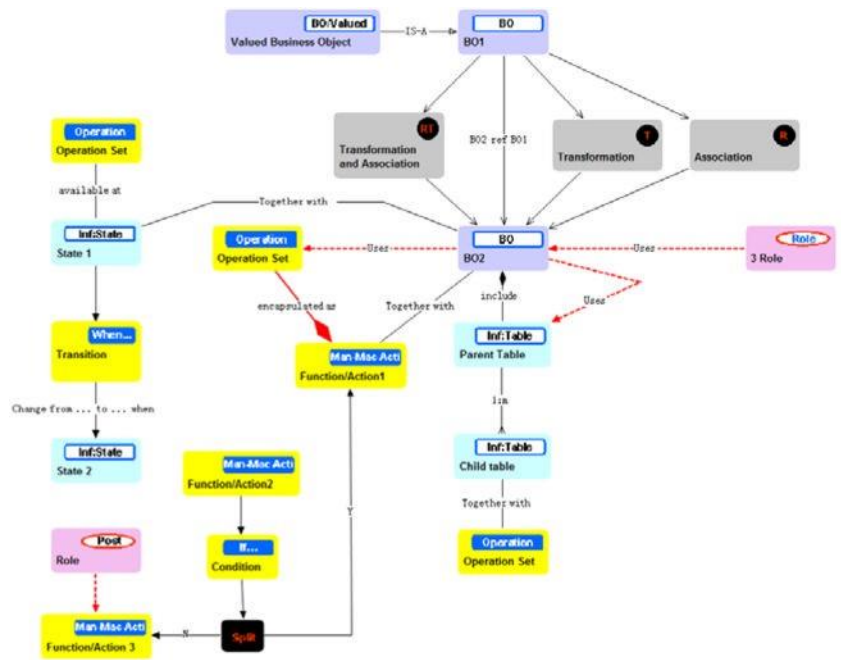


Fig. 3. Metamodel of GRAI-ICE PIM model

Based on the meta-model, the composition of PIM model is as following.

Business object model describes BOs and relationship of BOs. BO integrated diagram describes the detail of BO in four dimensions. (1) BO Data diagram describes the relations among data sets in IDEF1x. (2) BO class diagram describes the *DPR* (controlling relation between operation and data set), the relations among *OP* (operation of BO) and activity *AT* (activity set of BO). (3) BO state diagram describes the state *SS*, transition *ST*, and the properties associated with lifecycle. (4) BO use case diagram describes the relation among roles and operations. Relations among several BOs are described by BO-R model in two levels. The first level, i.e. BO level, describes the data dependency relations among BOs, such as association and integration. The second level, i.e. entity-relation level, considers the data set inside BO as entity and then describes the relations among them.

Workflow model describes the sequence of activities of one BO of multiple

BOs. The business object model is the core of PIM level. Workflow model assembles BOs into process supporting business function. Data, operation and state of BO provide input, output and trigger event for workflow execution. Data model describes the data schema of all BOs and provides global business view for users.

Role model defines the access to data and operation. ICE-PIM provides all essential design facets of ESA.

5. ICE-PSM

On one hand, ICE-PSM follows and inherents the design information from ICE- PIM level models and transforms them into platform specific implementation model. On the other hand, it is the basis for code generation. Software pattern and platform specification are two cornerstones for ICE-PSM.

Business Component (BC) is a largest grained unit of software development, and is a realization body of Business Object (BO) which is the finest grained operation unit of business system. BC is an implementation and running pattern for business components and is the core modeling elements in ICE-PSM. It is based on software pattern and XML, and can be transformed to code by code generator. It is a link between business object model and ESA system. Because of limited space, meta-model of BC is omitted.

The executable workflow model consists of workflow model of PIM level and implementation details based on schedule pattern. It is described by the language supported by workflow engine (e.g. business process modelling engine, state transition event driven engine...). The static configurable model (SCM) is extracted from the relation between business object and role model.

Pattern based ICE-PSM formulizes the software implementation model, simplifies the development of complex ESA and supports industrialized component development. The use of XML provides standard data source for code generation, automatic system configuration and deployment, and interoperability.

6. Generation of Code: ESA Components and System

In this level, PSM will be transformed into business components and ESA system. There are three phases for business component. The first is stem business component in software development phase. The properties related with roles and individuals are still in parameter. The second is reused business component in deployment phase. Variable properties of BC are finalized for special role/user. So, the BC in this state is role-related component. The third is individualized business component in running phase. The components show different interfaces and behaviors according to the configuration and requirement of users. By doing so, the requirement of flexibility is distributed in several phases and the difficulty of development is decreased.

The ESA development is based on business components. Firstly, the business component models are transformed into stem business components by code generator. Then, they are assembled by workflow, association mechanism and extend mechanism into complete business process. The workflow engine will manage the execution by invoking the different implement BO in the right order. Verification of correctness can be done by running simulation using the workflow

engine before execution. After that, they are configured and deployed based on the information of PSM.

7. An Application: Purchase Management

The proposed architecture, methods, and associated tools and framework have been applied in ESA (Enterprise Resource Planning system) development in several Chinese enterprises. The time of development and deployment is shortened in more than 40%. The application in one of them, i.e. SHANDONG Nuclear Equipment Co. LTD, China is outlined below.

Based on the GRAI-ICE MDI architecture, a set of tools have been developed. Firstly, a MOF, XML and Eclipse based extendable modeling tool supports defining modeling elements on demand and is used in modelling. A running framework has also been developed with a workflow engine inside for ESA. Some patterns in ESA have been extracted and formulized for business objects. XML and pattern based PSM model for BO is built. Based on pattern based PSM, we developed a PSM modeling tool and code generation tool, which support software developer for modeling PSM quickly, generate BC code, configure them and deploy them into the running framework.

Fig. 4 (a) gives a segment of business process model at Top CIM level in purchase management domain. Fig. 4 (b) shows the BO-relation model at PIM level in the same domain.

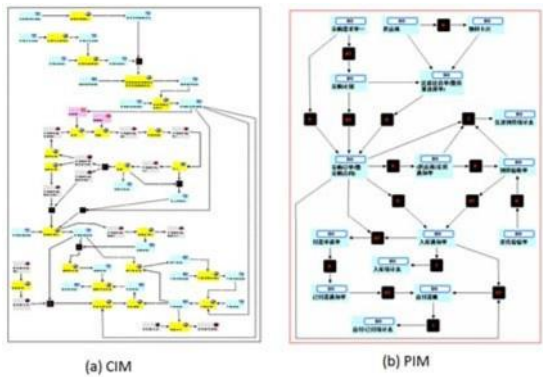


Fig. 4. Example of CIM model and PIM model in purchase management domain

Fig. 5 (a) shows a segment of XML file, which is the PSM of purchase order BO. Fig. 5 (b) shows the running component of purchase order, which is generated from PSM model.



Fig. 5. Example of PSM model and component of purchase order BO

8. Conclusions

Changeability on-demand and interoperability are critical characteristics for ESA in today dynamical economical environment. A concrete and operable model driven interoperability architecture entitled GRAI-ICE MDI is proposed for developing ESA. It integrates both business modeling methods and software modeling methods, defines proper levels and transformation between models elaborated at these levels. The vertical transformation is developed straightforward along the thread “information entity→business object→business component model→stem business component→reused business component”. Dynamic component/service assembly based on workflow technology and variable business component gives good support to ESA assembly, configuration and deployment.

The originality of the proposed approach is as following: (1) A nearly complete implementation of MDA/MDI is given in GRAI-ICE MDI, which appropriately integrates enterprise modelling, software modelling and software architecture. (2) The proposed architecture is interoperable, configurable and executable. (3) CIM is divided into business oriented CIM and ESA oriented CIM to capture both business and software requirements. (4) The proposed architecture and associated tools have been successfully applied in changeable on-demand and interoperable ESA development. Last but not least, the time of development and deploy has been greatly shortened.

9. Perspectives

Considering the technical trend such as service computing and SaaS (Software as a Service), the GRAI-ICE MDI architecture should be adapted to service-oriented model-driven interoperable architecture. Business components will evolve to service components and ESA will evolve to service components composition.

More and more components and services are available on Internet. In order to incorporate, manage and reuse them in GRAI-ICE MDI architecture, it should not only be top-down, i.e. transformation and generation from business to IT, but also bottom-up, i.e. discovery, assembly and composition from existing IT elements to meet new business interoperability requirement[14]. Also, validation of the models by simulation can be envisaged.

For now, GRAI-ICE MDI architecture focuses more on function requirement of ESA. However, non-function requirements such as confidentiality, security, timing, are crucial for run-time system. So, how to extend the proposed architecture to modeling non-function requirement and transform them into ESA is another important issue to consider in the future.

10. References

- [1] C Pautasso, T Heinis, G Alonso. Autonomic resource provisioning for software business processes. *Information and Software Technology*, 49(1): 65-80(2007)
- [2] J.P. Bourey, R. Grangel, G. Doumeingts and A.J. Berre, "Report on Model Driven Interoperability", Deliverable DTG2.3, INTEROP NoE, , April 2007, pp. 91
- [3] OMG. MDA guide version 1.0.1. http://www.omg.com/.12th June 2003
- [4] J. Bezivin, O. Gerbe. Towards a precise definition of the OMG/MDA Framework. *Automated Software Engineering.(ASE 2001)*. Proceedings. 16th Annual International Conference on 26-29 Nov. 2001:273 - 280
- [5] T. Meservy, K. D. Fenstermacher. Transforming software development: An MDA road map. *Computer*, 38(9): 52-58(2005)
- [6] J. Bezivin. From object composition to model transformation with the MDA. *Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems*, Santa Barbara, August 350 (2001)
- [7] R. Lemesle. Transformation rules base on meta- modeling. *Enterprise Distributed Object Com- puting Workshop. EDOC'98*. Proceedings. Second International,3-5 Nov. 113 – 122(1998)
- [8] W. Tan, L. Ma, J. Li, Z. Xiao. Application MDA in a conception design environment. *First International Multi-Symposiums on Computer and Computational Sciences, IMSCCS'06: 702-704(2006)*
- [9] K. Kosanke. CIMOSA--overview and status. *Computers in Industry*, 101-109(1995)
- [10] D. Chen, B. Vallespir and G. Doumeingts. GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology. *Computers in Industry*, 387-194(1997)
- [11] D. Chen, G. Doumeingts, F. Vernadat. Architectures for Enterprise Integration and Interoperability: Past, Present and Future. *International Journal of Computers In Industry*, 59(5), May 2008.
- [12] R. Grangel, R. Ben Salem, J. P. Bourey, N. Daclin, Y. Ducq. Transforming GRAI extended actigrams into UML activity diagrams: a first step to Model Driven Interoperability, I-ESA 2007, Funchal (Madeira Island), Portugal, March 28– 30, 2007.
- [13] R. Grangel, R. Chalmeta, C. Campos, R. Sommar, J. P. Bourey. A proposal for goal modelling using a UML profile. *I-ESA 2008*, Berlin, Germany, March 26–28, 2008.
- [14] G. Zacharewicz, D. Chen; B. Vallespir. Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability, I-ESA China 2009, Beijing, 22-23 April 2009.