Interoperability in Enterprises Networks: A Characterisation and Analysis Approach

Matthieu Roque, Vincent Chapurlat

Internal Research Report / Rapport Interne de Recherche n° RR/09-02

Interoperability in Enterprises Networks: A Characterisation and Analysis Approach

M.Roque, V.Chapurlat

Ecole Nationale Supérieure Mines Alès -- 6 avenue de Clavière, 30319 Alès cedex
LGI2P - Laboratoire de Génie Informatique et d'Ingénierie de Production, Parc scientifique Georges Besse, F30035 Nîmes

Interoperability in Enterprises Networks: A Characterisation and Analysis Approach

M.Roque, V.Chapurlat

LGI2P- Laboratoire de Génie Informatique et d'Ingénierie de Production
Site EERIE de L'Ecole des Mines d'Alès, Parc Scientifique Georges Besse
30035 Nîmes cedex 1 – France – tel. (+33) (0)466 387 066
{matthieu.roque, vincent.chapurlat}@ema.fr

Abstract. Interoperability problems which can occur during the collaboration between several enterprises can endanger this collaboration. Consequently, an enterprise manager needs tools in order to analyse rapidly the enterprise network and then become able to anticipate interoperability problems. Thus, the proposed approach in this paper is based on the specification of properties, representing interoperability requirements, and their analysis on enterprise models. Due to the conceptual limits of existing modelling languages, formalising these requirements and intending to translate them under the form of properties need to add conceptual enrichments to these languages. Finally, the analysis of the properties on enriched enterprise models, by formal checking techniques, aims to provide tools allowing to reasoning on enterprise models in order to detect interoperability problems, from an anticipative manner.

1 Introduction

The interoperability concept is started from a pure software problem in the middle of 90's where it is defined as "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*" [1]. Since a decade, although some efforts have been made to develop enterprise interoperability concepts, especially in Europe under various projects from FP5 and FP6, there is still no an overall satisfactory solution on interoperability. For example, [2] defines interoperability as "*the ability of a system or a product to work with other systems or products without special effort from the customer or user*". Interoperability is then analysed by considering simultaneously different levels of detail of the pointed out enterprise (business, process, service and data) and three kinds of barriers (conceptual, technological and organisational) i.e. three kinds of 'incompatibility' or 'mismatch' obstructing sharing and exchanging data. According to this, the classification of some related works and solutions for interoperability issues become possible. A lot of research and development works have been done concerning the conceptual barrier such as UEML [3] or PSL [4]. The goal is then to provide solutions to solve syntax and semantic problems. In the same way [5] proposes and interesting approach in order to design a Mediation Information System dedicated to deal with exchanged data, shared services and collaborative processes. This approach covers the technological and organisational barrier but considering only information system point of view. Other approaches are focused on the definition of maturity interoperability models. Let us cite for example LCIM [6], IEC 62390 [7], LISI [8], OIM [9] or EIMM [10] in order to evaluate the level of maturity of enterprises concerning their abilities to collaborate with other enterprises. However, they do not propose tools to measure and to evaluate interoperability itself. To solve this problem [11] proposes three kinds of enterprise interoperability measurements: interoperability potentiality, interoperability compatibility and interoperability performance.
Finally, all the works presented below do not provide a relevant solution in order to detect interoperability problems from an anticipative manner taking into account the different enterprise objects and their relationships within a network in which various enterprises must work together. Moreover, they do not allow identifying, in a formal way, what are the causes of interoperability problems. Furthermore, the analysis of organisational interoperability which depend from the structure, the available resources (human, material and application), their abilities and capabilities, the different processes, etc. are not really developed.
The research work presented in this paper aims to provide concepts and formal supports for reasoning on enterprise models in order to formalise and to detect interoperability problems as proposed in another domain by [12]. The paper is structured respecting the proposed approach shown in Fig. 1:

- Enterprise and Interoperability modelling: formalisation of concepts, existing modelling language conceptual enrichment and property modelling.
- Enterprise model re-writing: from enriched model of collaborative process to formal model allowing reasoning mechanisms.
- Checking technique and mechanisms: proving properties in order to check the interoperability requirements.

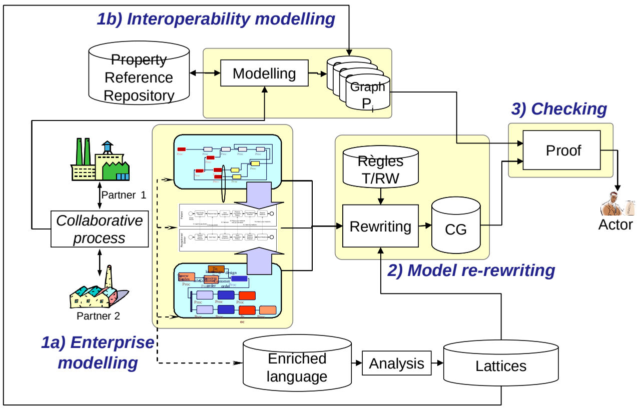


Fig. 1. Proposed approach synthesis

2 Enterprise and Interoperability Modelling

2.1 Definition of Interoperability in CARIONER Context

Interoperability is defined in the CARIONER project as a crucial requirement having to be verified by systems when being in relationship (cooperation, collaboration, exchange) with other systems in order to assume a common mission. In this case, considered systems are enterprises or parts of enterprises which have to interact in a collaborative and common process with other enterprises or parts of enterprises in order, for example, to design a new product, to produce and integrate different part of a given product, etc. Formalising what kind of relationships can exist in this area allows us to define more precisely this interoperability requirement. First, the relationship may be punctual or may exist during more or less long periods. Second, all along the relationship life cycle, systems must being able to:

- Continue to fulfil their own missions achieving their own objectives and their own finality respecting the common mission,
- Remain independent of other systems and thus able to resume its autonomy when relationship will stop. In another words, the relationship must be totally reversible,

Third, this relationship should be beneficial to the considered systems. In other words, even if a local deterioration of systems running conditions can be generally accepted during the collaboration, it does not generate excessive loss of:

- Performance in terms of cost, quality and delay;
- Stability in terms of ability to solve problems inducing efficiency loss when they occurs (excess capacity, reaction time, failure, error of misinterpretations or transmission, etc.);
- Integrity in terms of ability to stay efficient even in case of modification of the structure of the enterprise.

Last, any relationship between two systems induces a modification of the exchanged object(s). These modification may concern time, space and form of objects. This modification may differ taking into account environment and context of systems at the time of relationship.

Then, interoperability is a multiform requirement of each system involved in the relationship. It characterizes the systems' ability to communicate, to exchange and to work together staying autonomous and efficient, during a more or less long time. Of course, this ability depends from the relationship nature and duration but depends also of the characteristics (behaviour, function and structure) of each part of the considered systems in relation. So, when designing a new system, here a new network of enterprises, it is necessary to assume that the systems, here enterprises, and the required relationships between them being created and managed during the collaborative process check this requirement. This has to be done by analysing network model(s) and particularly concerned enterprises parts models in order to anticipate the problems. A solution is to detect possible effects of non interoperability and to deduce from an anticipative manner its causes. So, this can be done by:

- Quantitatively or qualitatively measuring the interoperability level through indicators, and/or,
- Establishing and proving the absence/presence of such problems by using different techniques: test, simulation, model analysis, and formal proof.

2.2 Formalisation of Interoperability Requirement

The formalisation of interoperability requirement uses the formalisation of the different abilities that the concerned enterprises or enterprise parts must have and of the relationship itself.
In the next, any enterprise, process, activity of an enterprise in relation with another one will be simply considered as a processor inspired by [13]. A processor is a point where an object carried by a flow (concretising the relationship) is processed i.e. transformed. Indeed, one or several object's characteristics (time – duration, delay, ... –, space – position, speed, acceleration,... – or form – geometry, colour, ... –) change during a processor execution under the action of entities considered as resources of the processor and respecting some constraints and rules.
In enterprise modelling domain [14], a processor can be an activity, a service, a process, an enterprise or an enterprise network. Indeed, any of these processors use resources (human actor, organisational unit, machine, tool, or software application) as means necessary to transform the inputs into outputs.
According to the system modelling framework called SAGACE [15], three types of relationships between two processors can be considered: transaction, coupling and interaction. Each relationship induces a set of requirements (functional and not functional [12] in order to assume that concerned processors are interoperable when it is needed. All these requirements have then to be checked in order to detect and to avoid interoperability problems. So, enterprise parts (processor, resources, flows, etc.) and interoperability requirements must be modelled.
The next part intents to formalise the interoperability requirement corresponding to relationship typology.

– **Transaction** is the basic relationship and only focuses on the flow of exchanged objects between two processors (supplier to customer) (Fig. 2). The flow can carry material, energy, financial, information or human objects. The customer processor can use this flow as an input to process or as a resource which support its execution.

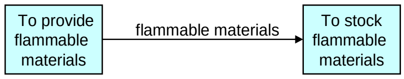


Fig. 2. Transaction relationship example

Thus, according to the proposed definition of the interoperability and whatever may be the nature of objects carried by the flow:

- The customer processor has to be able to take into account its inputs or resource flows.
- The supplier processor must be able to provide in time, space and form the required objects.

These expectations must consider simultaneously different attributes of the objects or of the processors. These attributes may characterize:

- Time dimension of the object such as synchronisation rules, possible delays, etc.
- Space dimension of the object i.e. object attributes related to the position of the object, its speed, etc., and
- Form dimension of the object i.e. taking into account object attributes related to the geometrical form of the object, its colour, its temperature, etc.

Transaction concerning objects of nature information induce, for example, the well known problems of the syntactic and semantic (form) of the exchanged information. It can also be related to the organisational aspects (time, form and/or space) i.e. the rules indicating how the different entities in the enterprise are structured and organised in order to fulfil the processor mission. For example, "is the actor in charge of a given processor must dispose of the required and updated information (about environment context, other processors and abilities for controlling the processor execution)?". Last, in the case of the transaction illustrated in the (Fig. 2), the "inflammable material" needs a specific kind of stock (form). Moreover, human resources involved in the processor have to be authorised to manipulate inflammable material (have the good skills) and have to be available.

– **Coupling** represents a reciprocal influence of a processor P1 named then controller processor to another processor P2 named operating processor: the controller processor P1 controls or constraints the execution of the operating processor P2 which have to provide reporting information and data to P1 as a feedback loop. This relationship corresponds typically to the link between decision and operating systems in system theory. The interoperability requirements are then, in addition to the ones of the transaction (based on form, state and time attributes), more related to the objectives or constraints provided by the controller processor to the operating processor. Thus, in the case of the (Fig. 3), it is necessary that the "production objectives" are clearly defined and well understood by all the resources involved in the "Reach production objectives" processor. Moreover, these production objectives have to be reachable in order to not induced interoperability problems between the two processors. Concerning the feedback loop, the requirements are the same ones of the transaction.

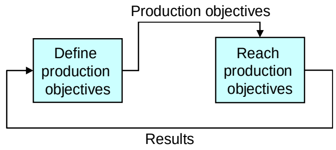


Fig. 3. Coupling relationship example

– **Interaction** (Fig. 4) represents an influence of a processor to another processor requiring an intermediate processor which plays the role of interface between the two processors.

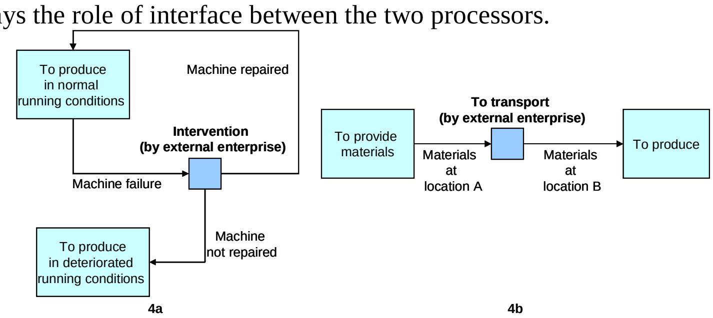


Fig. 4. Interaction relationship examples

This interface remains required because some change of one or more attributes of time, space and / or form of the object carried by the flow cannot be done by one of the two connected processors. However the interface processor cannot be controlled by one of the two processors. For example, the interface processor can be a service provided by external entity and the enterprise cannot intervene during its execution. An interaction is then defined as a 3-uple {Event, Processor, Condition} where:

- Event: event from which occurrence is required to execute the intermediate processor corresponding to an interruption of the normal running of the processor e.g. a machine failure (Fig. 4a) or simply the end of the processor (Fig. 4b),
- Processor: description of the intermediate processor as an input/output function,
- Condition: condition under which the processor has fulfilled its mission. In case of the condition is not valid, the processor cannot provide its output and this can generates a hazard which can produce or not another interaction.

So, interaction can have a stochastic behaviour taking into account the event occurrence, the condition validity but also of external constraints.

In this case interoperability requirements focus essentially on the intermediate processor. Indeed:

- Neither of the two processors can have an influence on the behaviour of the intermediate processor. For example, in the case of the Fig. 4b, the duration of the “To transport” processor is related to external constraints (as traffic jam, freight operators strike, etc.) and moreover this processor belongs to an external enterprise. The requirements consist then to prove that the processors are simultaneously aware about the possible risks associated to the fluctuations of interface processor behaviour and able to adapt their own behaviour, structure or functioning modes in order to anticipate these risks occurrences. In other words, are the processors able to find alternative in case of dysfunction of interface processor? For example, a requirement for interoperability for the processor “to provide materials” is to have an alternative solution i.e. defined scenarios which will be executed when the event occur, in order to provide the materials in time.
- The relationships between processors and this interface processor can be considered as a kind of Transaction relationship. So, the interoperability requirements concerning transaction have then to be checked to detect other interoperability problems.

2.3 From Interoperability Formalisation to Property

A property is defined by [16] as *a requirement or a characteristic that models of systems must check*. A property is then modelled by using a causal relation between a cause or condition and its effect(s) or conclusion. The causal relation can be of different types (logical – implication, equivalence -, temporal constrained or not, corresponding to influence or emergence) and is constrained in order to take into account different hypothesis. From an informal manner, a property example can be a simple implication between a cause described as “if an activity uses a resource” and an effect described as “the resource has to be available”. In order to formalise the interoperability requirements, a formal notation of property named CREDI [16] is then used. It describes formally the cause, effect and relation concepts. In CARIONER project, these ones take into account the different modelling entities defined in the proposed modelling framework presented after in the paper. Each interoperability requirement is then described as a set of informal properties by experts from the domain by using natural language such as “If a processor1 provide a material object to a processor2 then processor2 have to have a stoking area which sufficient volume to accept object”, “The mission processor is clearly expressed” or “The true email address of the activity responsible has to be known by all activities which want to send an email to this activity”. To use the checking technique proposed in the next part, these ‘informal’ properties have to be written into a formal language. Conceptual Graphs have been chosen. Conceptual graphs [17], [18] come from the Graph Theory for formal knowledge modelling and analysis. A conceptual graph is a finite, connected, directed bipartite graph. It is defined as a graph with only two kinds of nodes: the concepts and the relations. Then, each property is translated into a conceptual graph by using interpretation mechanisms (considering the concepts and relations extracted from the modelling language which are described later in this paper). These ones are now under development and use the tool COGITANT [19]. The result is a set of conceptual graphs. However, due to the conceptual limits of existing modelling languages, formalising these requirements and intending to translate them under the form of properties induces to adapt i.e. to enrich these modelling languages.

2.4 Impact on Enterprise Modelling Languages: Conceptual Enrichment

Enterprise modelling domain provides several approaches, frameworks and modelling languages which are synthesised in [20]. The analysis of this SoA shows different thinks:

- The multi-view paradigm is required such as proposed in GERAM [21] or CIMOSA [22] in order to allow actors to describe under several viewpoints an enterprise part or a network: information, processes, resources, and organisation.
- Existing modelling languages are not able to provide mechanisms or concepts for interoperability representation.

So, the work is based on a system framework inspired by the SAGACE approach as proposed by [23] (not presented here) and a set of conceptual enrichments of one of these modelling languages on which CARIONER project will focus and called BPMN (Business Process Modelling Notation) [24]. The primary goal of BPMN is to provide annotation that is readily understandable by all business users. Thus, it creates a standardized bridge for the gap between the business process design and process implementation. According to [5], a BPMN model allows to represent mainly the behaviour of a system. It is not possible to describe the organisation aspects with a sufficient level of detail. Even if some concepts of BPMN allow to represent a part of the organisation structure, the resources involved in a process are difficult to represent. For example, in BPMN the resources are only software resources which are supposed to be qualified and available and in order to analyse interoperability problems we cannot assume that it is always true. Moreover, BPMN does not have concepts allowing to represent the finality, the mission and the objectives of a system. Thus, to formalise interoperability requirements, we need to add conceptual enrichment to the BPMN meta-model. The meta-model of a modelling language (enriched BPMN language), allowing to represent a model of the enterprise network to interoperability analysis issue, is presented in Appendix. It has been implemented by using Graphical Modelling Framework of the Eclipse Platform [25]. This allows first to provide a modelling tool which is used in order to represent the network and enterprise models, second to develop the property proof mechanisms presented in the next part.

3 Enterprise Model Re-Writing

The approach proposes to re-write enterprise models based with our enriched BPMN language in others models based on a formal language. The objective is to obtain models without sense ambiguity in order to check formal properties describing interoperability requirements. Thus, the enriched enterprise model is translated into Conceptual Graphs by using formal rules not completely described in this paper.

The re-writing procedure starts from the meta-model of the enriched BPMN language, established in UML. This UML diagram is analysed and formalised in order to provide all the needed concepts and relations of the Conceptual Graph. All concepts are obtained by considering all the modelling entities which will be used in the checking task. Thus, each class of the meta-model (but also its attributes) is translated into concepts. Then, the relations are obtained by translating each association between classes into a relation between concepts. Then, the defined concepts and relations (described in hierarchical structures called concepts and relations lattices) allow transforming the enterprise network model build with the enriched BPMN language into a conceptual graph. To do this transformation, each marker (which refers to specific instances of concepts) has to be extracted from the model in order to produce a unique conceptual graph G. Thus, G gathers all the knowledge described in the model.

4 Checking Technique and Mechanisms

The checking technique is inspired by [26] and [27] which use analysis mechanisms allowed by conceptual graphs. These analysis mechanisms are:

– **Projection**: This involves comparing the obtained conceptual graph coming from the translation of the model with another one translating the property. If the projection fails, then the modelled property cannot be verified and the causes are highlighted.
– **Constraint**: a property describes what the links and/or constraints are between facts. In this case, the property is translated on a positive or negative conceptual graph constraint. A positive constraint between two facts A and B must be interpreted as: “If A is true, then B must also be true”. Conversely, a negative constraint must be interpreted as: “If A is true then B must be false” (if B is true, A must be true or false).
– **Dynamic and static rules**: A property is directly modelled as a property composed of a cause and an effect. If the graph corresponding to the causes match with a part of the conceptual graph translating the system models, then the effect must be checked in the same way.

The Fig. 5 illustrates an example of a property proof by using the projection mechanism.

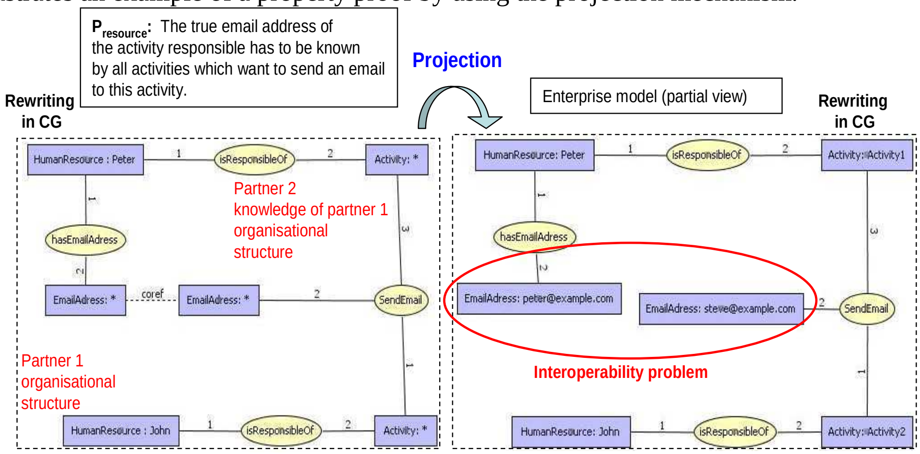


Fig. 5. Example of use of conceptual graph

5 Conclusion and Perspectives

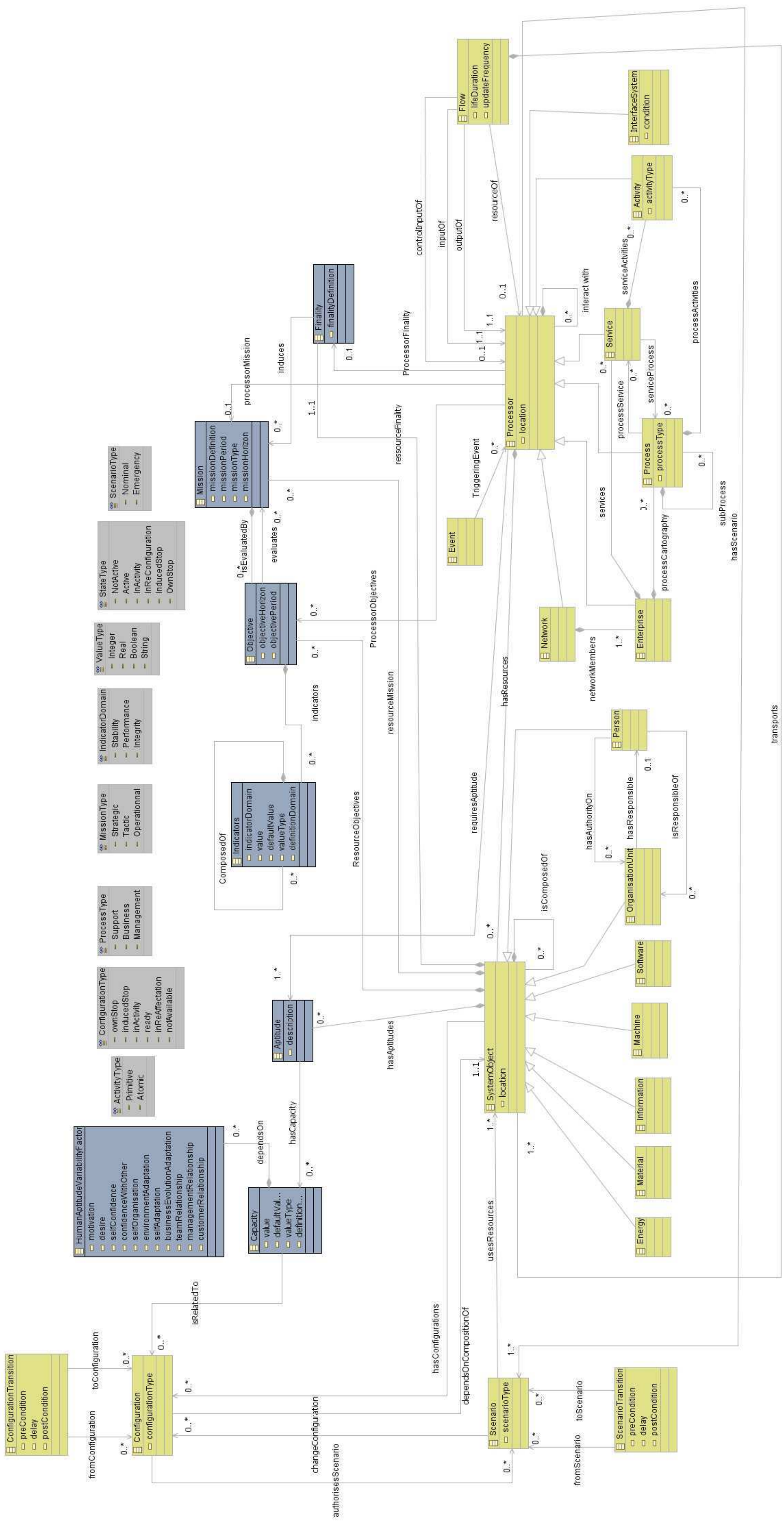
This article presents the first results of the CARIONER project. A formal model of interoperability requirement in the enterprises networks context is introduced. A set of modelling and formal proof mechanisms is then described in order to analyse from a static point of view the network model. The main perspectives of this work are the following. First, a reference properties data base and rewriting mechanisms have to be developed in order to help actors in charge of the network to analyse more rapidly the network and then become able to anticipate interoperability problems. Second, rewriting mechanisms from network model and properties date base have to be implanted in order to be interfaced with model checkers such as UPAAL. Third, other works in progress intents to make the gap between network model and an enriched multi agents system allowing to simulate the behaviour of the different parts of the enterprises involved in the collaborative process. The goal these two last works is then to assume dynamic properties can be then checked.

References

1. IEEE, A compilation of IEEE standard computer glossaries: standard computer dictionary, New York, (1990)
2. INTEROP, Enterprise Interoperability-Framework and knowledge corpus - Final report, INTEROP NoE, FP6 – Contract n° 508011, Deliverable DI.3, May 21st (2007)
3. Berio G., UEML 2.0, Deliverable 5.1, INTEROP project UE-IST-508011 (2005)
4. Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., Lee, J., The Process Specification Language (PSL): Overview and Version 1.0 Specification, NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD (2000).
5. Bénaben F., Touzi J., Rajsiri V., Truptil S., Lorré J.P., Pingaud H., Mediation Information System Design in a Collaborative SOA Context through a MDD Approach. MDISIS'08: Model Driven Interoperability for Sustainable Information System. Montpellier, France (2008)

6. Tolk A., Muguira J.A., The Levels of Conceptual Interoperability Model, Fall Simulation Interoperability Worshop (2003)
7. IEC, Common Automation Device: Profile guideline, TC 65: Industrial Process Measurement and Control (2005)
8. C4ISR, Levels of Information Systems Interoperability (LISI), Architecture Working Group ,United States of America, Department of Defence (1998)
9. Clark T., Jones R., Organisational Interoperability Maturity Model for C2, Australian Department of Defence (1999)
10. ATHENA, Framework for the establishment and management methodology, Integrated Project ATHENA, deliverable A1.4 (2005)
11. Daclin N., Contribution au développement d'une méthodologie pour l'interopérabilité des entreprises (in French), PhD Thesis, University Bordeaux1, December, (2007)
12. Aloui S., Chapurlat V., Penalva J.-M., Linking interoperability and risk assessment: A methodological approach for socio-technical systems, Proceedings of INCOM2006, 12th IFAC Symposium on Information Control Problems in Manufacturing, Information control: a complex challenge for the 21st century, A. Dolgui, G. Morel and C. Pereira Eds., ISBN: 978-0-08-044654-7, Saint Etienne, France, hal-00354778 (2006)
13. Le Moigne, J.L.: La modélisation des systèmes complexes, Paris, Bordas, Dunot (1990)
14. Vernadat, F.B.: Enterprise Modelling and Integration: Principles and Applications. Chapmann & Hall (1996)
15. Penalva, J-M., La modélisation par les systèmes en situation complexe (in French), PhD thesis. Paris Sud university (1997)
16. Lamine E., Définition d'un modèle de propriété et proposition d'un langage de spécification associé : LUSP (in French), Ph.D Thesis University Montpellier II (2001)
17. Sowa J.F, Conceptual structures: information processing in mind and machine, Addison-Wesley Longman Publishing Co., Inc., New York (U.S.A.) (1984)
18. Chein M., Mugnier M-L., Conceptual graphs: fundamental notions, Revue d'intelligence artificielle, vol.6, n°4 (1992)
19. Cogitant, CoGITaNT Version 5.1.92: Reference Manual (http://cogitant.sourceforge.net)
20. Petit, M. et al., Enterprise Modelling State of the Art. UEML Thematic Network.Contract n°: IST–2001–34229. Work Package 1 (2002)
21. GERAM, Generalised Enterprise Reference Architecture and Methodology, Version 1.6.1, IFIP–IFAC Task Force on Architectures for Enterprise Integration (1999)
22. AMICE CIMOSA : Open Architecture for CIM. Berlin. Springer Verlag (1993)
23. Aloui S., Contribution à la modélisation et l'analyse du risque dans une organisation de santé au moyen d'une approche système (in French), Ph.D Thesis, Ecole des Mines de Paris (2007)
24. BPMN, Business Process Modeling Notation, V1.1. http://www.bpmn.org/
25. GMF, Graphical Modelling Framework, http://www.eclipse.org/modeling/gmf/
26. Kamsu-Foguem B., Modélisation et Vérification des propriétés de systèmes complexes: application aux processus d'entreprise (in French), Ph.D. Thesis University Montpellier II (2004)
27. Chapurlat V. and Aloui S., How to detect risks with a formal approach? From property specification to risk emergence, in proceedings of Modeling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS), Paphos, Cyprus (2006)

Appendix



Flow
lifeDuration
updateFrequency
InterfaceSystem
condition
Activity
activityType
Finality
finalityDefinition
Processor
location
Service
Process
processType
Mission
missionDefinition
missionPeriod
missionType
missionHorizon
Objective
objectiveHorizon
objectivePeriod
Indicators
indicatorDomain
value
defaultValue
valueType
definitionDomain
Event
Network
Enterprise
Person
OrganisationUnit
Software
Machine
Information
Material
Energy
SystemObject
Aptitude
description
Capacity
HumanAptitudeVariabilityFactor
motivation
desire
selfConfidence
confidenceWithOther
selfOrganisation
environmentAdaptation
selfAdaptation
businessEvolutionAdaptation
teamRelationship
managementRelationship
customerRelationship
ScenarioType
Nominal
Emergency
StateType
NotActive
Active
inActivity
inReConfiguration
InducedStop
OwnStop
ValueType
Integer
Real
Boolean
String
IndicatorDomain
Stability
Performance
Integrity
MissionType
Strategic
Tactic
Operationnal
ProcessType
Support
Business
Management
ConfigurationType
ownStop
inducedStop
inActivity
ready
inReAffectation
notAvailable
ActivityType
Primitive
Atomic
ConfigurationTransition
preCondition
delay
postCondition
Configuration
configurationType
Scenario
scenarioType
ScenarioTransition
preCondition
delay
postCondition
controlInputOf
inputOf
outputOf
resourceOf
interact with
serviceActivities
processActivities
processorMission
induces
ProcessorFinality
ressourceFinality
TriggeringEvent
processService
serviceProcess
services
subProcess
hasScenario
processCartography
isEvaluatedBy
evaluates
ProcessorObjectives
resourceMission
hasResources
networkMembers
transports
indicators
ComposedOf
ResourceObjectives
requiresAptitude
hasAuthorityOn
hasResponsible
isResponsibleOf
isComposedOf
hasAptitudes
dependsOn
hasCapacity
isRelatedTo
hasConfigurations
dependsOnCompositionOf
usesResources
toConfiguration
fromConfiguration
changeConfiguration
authorisesScenario
toScenario
fromScenario