



HAL
open science

Developing interoperability in collaborative process: an Anticipative Effects-Driven Approach

Sihem Mallek, Nicolas Daclin, Vincent Chapurlat

► **To cite this version:**

Sihem Mallek, Nicolas Daclin, Vincent Chapurlat. Developing interoperability in collaborative process: an Anticipative Effects-Driven Approach. Advances in production Systems Management, APMS 2009, Sep 2009, Bordeaux, France. hal-00588818

HAL Id: hal-00588818

<https://hal.science/hal-00588818>

Submitted on 20 Jun 2023

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.

Developing interoperability in collaborative process: an Anticipative Effects-Driven Approach

Sihem Mallek, Nicolas Daclin, Vincent Chapurlat

Laboratoire de Génie Informatique et d'Ingénierie de Production - LGI2P - site EERIE de l'Ecole des Mines d'Alès, Parc Scientifique Georges Besse, F30035 Nîmes cedex 5, France
Email : Sihem.Mallek@ema.fr, Nicolas.Daclin@ema.fr, Vincent.Chapurlat@ema.fr

Abstract: *This paper aims at presenting an anticipative effects-driven approach to help enterprises to enhance their ability to interoperate within collaborative processes before their implementation. This research focuses on the verification of interoperability requirements. Interoperability characterizes the aptitude to exchange and to share flow of any kinds. Thus in a collaborative context, it takes an important place for enterprise that want to improve their competitiveness in a globalized environment. This research aims to anticipate the possible effects of non-interoperability, to evaluate their impact on performance(s) or risky situations to be managed and to develop corrective actions.*

Keywords: *Collaborative process; interoperability; interoperability requirements; verification*

1. Introduction

Nowadays, the notion of collaboration covers many scopes such as industrial activities management system, health care system, crisis management system... In the frame of collaborative processes, most of services (resources, organizations, actors ...) are involved to accomplish a mission and to reach given objectives. In such context, the services have to make collaboration successful; they must guarantee their coordination and synchronization, but also their availability, their capacities and abilities to interact all along collaborative process. This must be done from an autonomous manner and without any disturbance, loss of performance or risk for the enterprises and process. From the last years, the improvement of interoperability is became a major issue to ensure the respect of the points previously mentioned and furthermore to ensure the perennality of the collaboration. Indeed, interoperability is defined (IDEAS, 2003) as: *“the ability of interaction between enterprise applications at three enterprise levels: data, application and process”*. As a consequence, the here presented research is related to the development of concepts, mechanisms of reasoning and methodology in order to characterize, to detect and to anticipate the effects of the non-interoperability in collaborative processes *via* the verification of interoperability requirements.

This article is structured as follows. Section 2 introduces and argues the problematic, the needs covered by this research work and the proposed methodology composed of different steps. These different steps are presented and developed successively in section 3, section 4 and section 5. Finally section 6 outlines the perspectives of this research.

2. Problematic and position

Collaboration between enterprises connotes a more durable and pervasive relationship for commitment to a common mission (Kvan, 2000). Such collaboration can take place between

services of different enterprises working together and may be deployed throughout a collaborative process. A collaborative process can be defined as “process whose activities may belong to different organization” (Aubert *et al.*, 2002). This kind of process evokes the classical notion of collaboration *i.e.* the interaction between actors during a work. This work is done with some particularities of distribution of responsibilities, synchronicity of actors, simultaneity or parallelism of activities *etc...* In the industrial context, each interaction may be materialized by a set of flows from different nature: knowledge, data, skills, but also energy, material, product or finance. A collaborative process highlights a complex context due to multi domain, multi disciplinary, multi tools and multi actors, even geographically distant. In summary, collaborative process represents today a new vision of the organization aiming to support a new working approach and inducing to develop new skills and new means for each actor in enterprise but whose results are closely linked to the ability of interoperability of actors and above all, the enterprises themselves, their internal organization, resources, skills and behaviors.

The complexity of services or enterprises involved in process, the complexity induced by the required interactions between these services, including fluctuant market environment, have to be understood and analyzed in order to detect if any dysfunction mode, improper resources profiles, risks, delays, loss of performance during the collaboration. As a consequence, the here proposed research work aims to help these actors to detect interoperability problems and cover the four followings needs:

- **This analyze must be guided.** This requires to define clearly the interoperability problematic between partners, and to establish a set of formal and valid interoperability requirements.

- **This analyze must be done by focusing on collaborative process models** allowing to reason and finally to anticipate the interoperability problems before its runtime. This induces the use of modeling languages and framework. However, these ones have to be enriched in order to take into consideration the interoperability problematic.

- **This analyze must be automated so far as possible.** This induces to develop and to apply reasoning mechanisms allowing to verify that requirements are fulfilled all along the collaborative process.

- **This analysis must be done in an anticipative way** in order to reduce risky situations, loss of performance...

Thus, this research work aims to develop an Anticipative Effects-Driven Approach (AEDA) to perform the four needs identified and defined previously in order to anticipate the possible effects (desired, undesired, good, harmful) of non-interoperability and to develop corrective actions. The expected results of the implementation of the anticipative effects-driven approach are to allow actors:

- To formulate and formalize interoperability needs: the result is a set of requirements to guarantee interoperability. These requirements must be satisfied by the collaborative process. As a consequence, this set of requirements has to be formalized into properties with formal language to make verification possible.

- To model the collaborative process taking into accounts these interoperability requirements.

- To prove the collaborative process checks the different interoperability requirements in order to detect potential lack of interoperability. Formal verification techniques (model checkers) and simulation are used.

The principles of the approach are presented in the following figure.

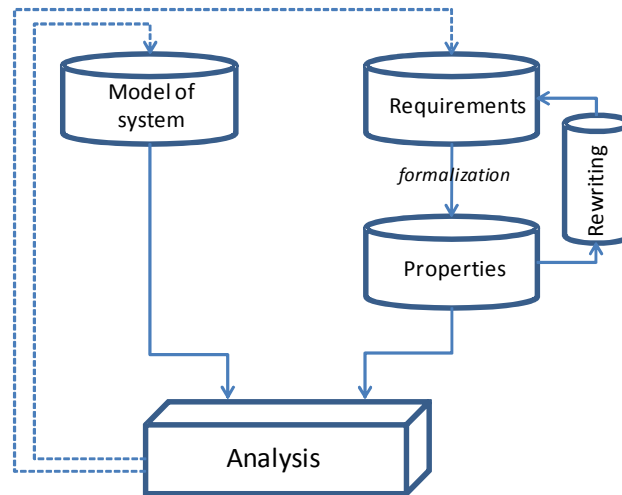


Figure 1: The principles of the anticipative effects-driven approach

This paper focuses on the conceptual principle of AEDA, especially on the formulation and the formalization of interoperability requirements. The collaborative process modeling approach and the checking techniques will be shortly introduced.

In the following sections the different steps of the proposed methodology are presented. The next section presents how interoperability requirements are first conceptualized and formalized. Then, the collaborative process model and the interoperability analysis are presented successively in section 4 and section 5.

3. Interoperability requirements

To identify and to structure interoperability aspects, different interoperability frameworks have been developed. Following the same approach, interoperability requirements can be (1) clearly identified according to the interoperability aspects that have to be considered and, (2) consistently structured by their positioning into interoperability framework.

a) Interoperability framework

As soon as different services or enterprises parts are interoperable, they are able to collaborate more efficiently. Indeed, Interoperability is a prerequisite for efficient collaboration. In order to make interoperability possible, several aspects have to be taking into consideration (*e.g.* interoperability problems, interoperability approach...). To structure these different aspects, interoperability frameworks have been developed in different domains (EIF, 2004; e-Government Unit, 2005). In the field of enterprise interoperability, the framework developed in (Chen *et al.*, 2007) proposes to consider two fundamental interoperability aspects such as “interoperability barriers” and “interoperability concerns”.

Interoperability barriers mean an incompatibility or mismatch which obstructs the sharing and exchanging of flow. There are three main kinds of barriers: conceptual, technological and

organizational. Conceptual barriers are concerned with the syntax and semantic incompatibilities of flow to be exchanged. Technological barriers are concerned with the use of computer or ICT (Information and Communication Technology) to communicate and exchange flow. Finally, organizational barriers are concerned with the incompatibilities of organization structure and management techniques implemented.

Furthermore, interoperability concerns represent different levels of interaction between two parts. These include four levels: data, service, process and business. Data refers to operate together different data models and the use of different query languages. Service is concerned with identifying, composing and operating together various applications and also functions of the company or of the networked enterprises. Process aims to make various processes work together. Finally, business refers to work in harmonized way at the levels of organization.

The following figure represents the enterprise interoperability framework with these two basics dimensions such as “interoperability concerns” and “interoperability barriers”.

barriers \ concerns	CONCEPTUAL	TECHNOLOGICAL	ORGANISATIONAL
BUSINESS			
PROCESS			
SERVICE			
DATA			

Figure 2: The Enterprise Interoperability Framework

Several works, based on this framework to address problems of interoperability, have been developed from the last years. For instance, in (Daclin *et al.*, 2008) a matrix of compatibility based on this framework was proposed to measure compatibility and to inform enterprises on the attendance (or not) of interoperability problems. However, interoperability requirements were not formalized and hardly exploitable.

b) Interoperability requirements

The goal is to obtain a set of interoperability requirements those services or enterprise parts have to satisfy. Requirement can be defined as “*a statement that specifies a function, ability or a characteristic that a product or a system must satisfy in a given context*” (Scukanec *et al.*, 2008).

Works related to the formulation of interoperability requirements were proposed in (Blanc *et al.*, 2007) as rules of interoperability. Five rules of interoperability were identified. However, these rules are generic and are not consistently related to the interoperability framework (in terms of interoperability concerns and interoperability barriers), thus they are hard to integrate and to use within a formal verification of interoperability requirements.

Furthermore, formalizing interoperability requirements needs classically to focus on compatibility of services. However, it is to note that “*two interoperable services are compatible but two compatible services are not necessarily interoperable*” (Kasunic *et al.*, 2004). As a consequence, requirements related to the compatibility can be considered as not insufficient especially because of their static aspect. They remain independent of time and

from the behavior of each service or from the environment in which the services have to be interconnected. Thus, the ability to interact depends not only of the compatibility but also of the behavior of services and the moving environment. They influence how and when the interaction can be efficient. This statement leads to consider a second sort of requirements called “interoperation requirements” and related to the dynamic aspect of the interaction. In summary interoperability requirements can be decomposed in two types:

Interoperability requirements = Compatibility requirements and Interoperation requirements

Thus, these two types of interoperability requirements are formulated according to the interoperability framework previously presented. For example, at the service level, interoperability deals with (1) the interoperation (mean of transport, environment ...) and (2) the compatibility between the service provider (S₁) and service demander (S₂) as shown in figure 3. The lack of interoperability can occur from the two services or from the interoperation itself.

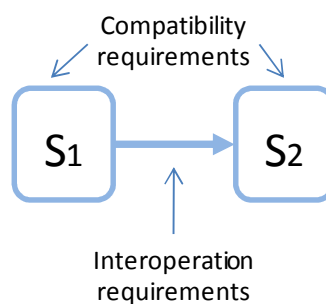


Figure 3: The decomposition of the interoperability requirements between two services.

In order to better structure, to represent and to detail interoperability requirements the concept of causal tree can be developed. Precisely, a causal tree allows to:

- Represent, structure and formalize the requirement from an unambiguous manner and;
- To authorize reasoning mechanisms to be applied in order to verify these requirements by the use of techniques such as verification and simulation.

The here proposed causal tree is based on the interoperability framework’s dimensions presented previously. Thus, the first level of the tree represents the different interoperability concerns (business, process, service and data). The second level represents both compatibility requirements and interoperation requirements. The third level allows differentiating interoperability barriers (conceptual, technological and organizational). Finally, the next levels allow to represent the different requirements which can induce a loss of interoperability if one of them is disrespected. The path followed in the causal tree allows to target exactly the problem of interoperability as shown in figure 4.

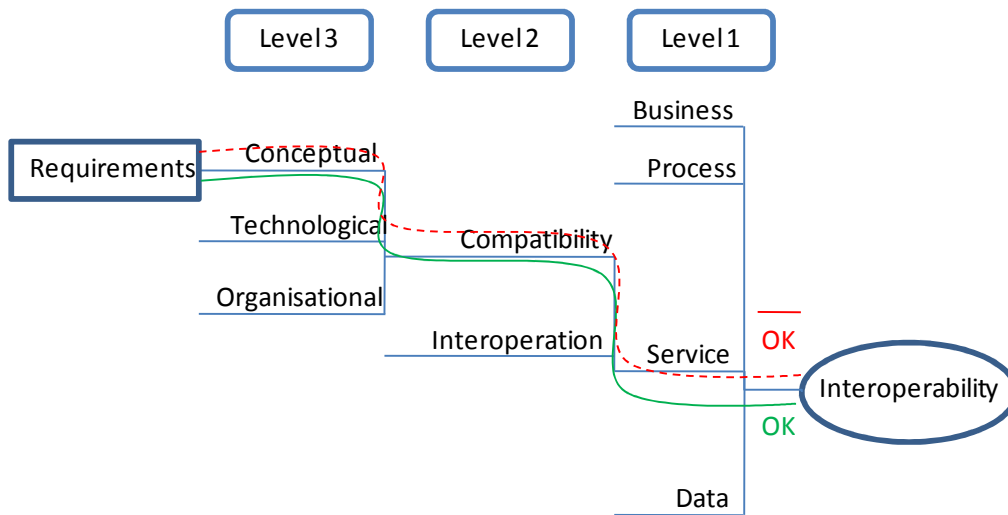


Figure 4: The Interoperability requirements causal tree

To get an idea of these requirements, some examples for compatibility requirements related to the interoperability barriers in the level of service are presented.

In the field of conceptual interoperability, one requirement may be formulated as “*Two services which aim to interoperate efficiently must have a clear definition of semantic of their name*” to avoid ambiguities and confusion.

In the field of technological interoperability, one requirement can be expressed such as “*Two services which aim to interoperate together must dispose of compatible tools*” to assume information exchange.

Last, in the field of organizational interoperability, one of requirement can be formulated as “*Two services which aim to interoperate together must have responsibility defined*” to know who is responsible for doing what.

All the interoperability requirements presented above are deployed in the interoperability causal tree given in figure 5.

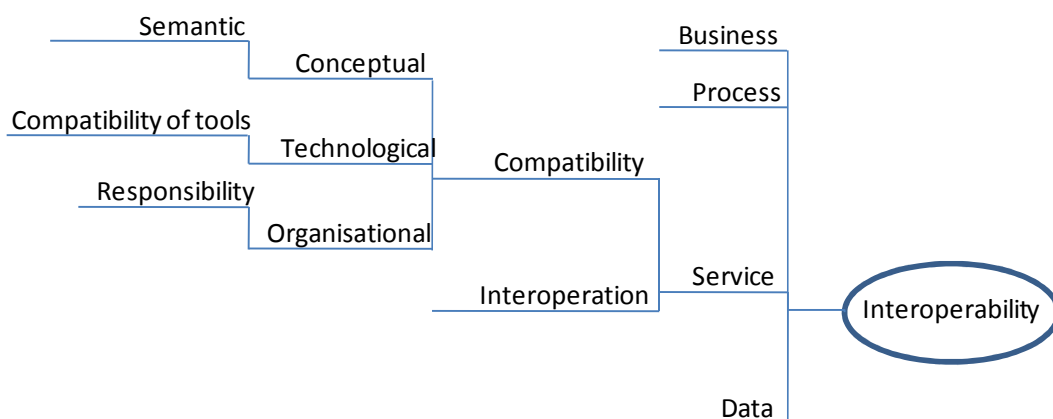


Figure 5 : Some interoperability requirements formulation in the interoperability causal tree

To check these requirements, they must be formalized into properties. Different ways to formalize requirements are proposed in the literature. For instance, (Evrot *et al.*, 2008) has

developed mechanisms allowing to refine needs into requirements then properties and to allocate properties to components. The refinement makes the requirement model more and more formal and, by the addition of successive details, the allocation identifies components that realize a specific requirement. These mechanisms help, thereafter, to formalize interoperability requirements into properties in order to facilitate their verification. The truthfulness of these properties are verified into a model of the collaborative process.

4. Collaborative process model

The collaborative process model allows to describe with a systemic approach, the behavior, the functions and the organization of the system. The goal is to obtain a correct model which can be exploitable in order to extract the obstacles of interoperability.

The complexity of a collaborative process requires the use of a systemic approach. The modeling of collaborative process with a systemic approach provides (Le Moigne, 1977):

- Finalities and objectives of the system,
- Components of the system : elements which compose the system and the relationships between them,
- Limits of the system : identification of elements which do not belong to the system,
- Interactions of the system with its environment : types of relationships and influences able to lead to modifications of the system,
- Functions that the system provides,
- Dynamics of evolution.

The analysis of the respect or the disrespect of the interoperability requirements will be performed according to this collaborative process model. Thus, this collaborative process model has to take in consideration the interoperability problematic dimension (interoperability barriers, interoperability concerns) by the enrichment of modeling language used.

5. Interoperability requirements analysis: proof and simulation

Requirements must be checked on the model *i.e.* for each service or enterprise part involved in the collaborative process. Formal verification techniques based on model checking and simulation are used.

Verification aims to answer the question: "Do we build the model correctly?" The goal is classically to prove consistency of the model and to ensure proper use of modeling (Chapurlat, 2007). In our case, verification techniques are used to check the interoperability requirements on the collaborative process model.

To verify interoperability requirements, there are two techniques that depend of the types of interoperability requirements (compatibility requirements, interoperation requirements). The first technique allows to check the static requirements *i.e.* the compatibility requirements. It is based on the use of Conceptual Graphs as proposed in (Roque *et al.*, 2009). The second

technique allows us checking the dynamic requirements *i.e.* the interoperation requirements. In this case, model checking is used on the behavioral model of the process. Verification with models checkers (Uppaal¹, SMV², Kronos³...) (Behrmann *et al.*, 2004; McMillan, 1992; Yovine, 1997) requires two phases. The first phase consists to describe or to generate the behavioral model of the process. The second phase consists to reformulate the interoperation requirements described in the causal tree under the form of properties by using formal language such as temporal logic (Schnoebelen, 2002) or UPSL (Unified Properties Specification language) (Lamine, 2001). Then, the principle of a model checker is to verify exhaustively properties by using temporized and eventually constrained automata describing the behavior of the pointed out system. A model checker responds with true or false if a property is satisfied or not and, in the second case, gives a counter example.

Moreover, simulation can be used as a perspective in this research to help detection of failure, risky situations or even some events occurrence effect on the process behavior. In this way, it allows to evaluate the impact of some potential lack of interoperability. These are the reasons for which the research proposes to use simulation and formal proof in order to check interoperability requirements. If interoperability requirements are not satisfied, corrective actions will be developed to tackle the obstacles of an effective collaboration.

6. Conclusion

In collaborative context, the development of interoperability is became a crucial question for enterprises that want to gain competitiveness in a globalized environment. This paper has presented the foundation of an approach that allows enterprises to detect and to anticipate the effects on non-interoperability between them. This approach is based on the formal verification of interoperability properties that enterprises have to respect in order to interoperate efficiently and furthermore, to improve their collaboration.

At this stage, the here presented research focuses on the development of a complete referential of interoperability requirements. Future is concerned by the transformation of these requirements into formal properties and their analysis *via* model checking and simulation tools.

7. References

Aubert B., Dussart A., *Système d'Information Inter-Organisationnel*, Rapport Bourgogne, CIRANO, Mars 2002 (In French).

Behrmann G., David A., Larsen K. G., *A tutorial on Uppaal*, Department of Computer Science, Aalborg University, Denmark, 2004.

Blanc S., Ducq Y., Vallespir B., *A Graph based approach for interoperability evaluation*, Interoperability for Enterprise Software and applications – IESA 2007, Madeira, Portugal, 2007.

Chapurlat V., *Vérification et validation de modèles de systèmes complexes : application à la modélisation d'entreprise*, HdR, Université Montpellier 2, France, Mars 2007 (In French).

¹ Available online at: <http://www.uppaal.com/>

² Available online at: <http://www.kenmcmil.com/smv.html>

³ Available online at: <http://www-verimag.imag.fr/~tripakis/openkronos.html>

- Chen D., Dassisti M., Elvesaeter B.,** *Enterprise interoperability Framework and knowledge corpus – advanced report*, INTEROP Deliverable DI.3, 2007.
- Daclin N., Chen D., Vallespir B.,** *Methodology for enterprise interoperability*, 17th IFAC World Congress (IFAC'08), Seoul, Korea, 6-11 July 2008.
- e-Government Unit,** *e-Government Interoperability Framework*, Version 6.1, 2005.
- EIF,** *European Interoperability Framework For Pan-European eGovernment Services*. European Commission, version 1.0, European Communities Ed., 2004.
- Evrot D., Pétin J-F., Morel G.,** *Combining SysML and formal methods for safety requirements verification*, Insight Journal of INCOSE 11, pp 21-22, 2008.
- IDEAS,** *State of the Art – Introduction*, IDEAS Project - Deliverable D1.1, June 2003. Available online at: <http://www.interop-vlab.eu/>.
- Kasunic M., Anderson W.,** *Measuring systems interoperability: challenges and opportunities*, Software engineering measurement and analysis initiative, Technical note CMU/SEI-2004-TN-003, 2004.
- Kvan T.,** « *Collaborative design: what is it?* », Automation in Construction, Volume 9, 2000.
- Lamine E.,** *Définition d'un modèle de propriété et proposition d'un langage de spécification associé : LUSP*. PhD thesis, Université Montpellier 2, France, 2001 (In French).
- Le Moigne J.-L.,** *La théorie du système générale – théorie de la modélisation*, Presse Universitaire de France, 1977 (In French).
- McMillan K. L.,** *Symbolic Model Checking: An approach to the state explosion problem*, PhD thesis, Carnegie Mellon University, USA, May 1992.
- Roque M., Chapurlat V.,** *Interoperability in collaborative processes: Requirements characterisation and proof approach*, PRO-VE'09, 10th IFIP Working Conference on VIRTUAL ENTERPRISES, Thessaloniki, Greece, 7-9 October 2009.
- Schnoebelen Ph.,** *The Complexity of Temporal Logic Model Checking*, Advances in Modal Logic, Volume 4, pp 1-44, 2002.
- Scukanec S. J., Van Gaasbeek J. R.,** *A Day in the Life of a Verification Requirement*, U.S. Air Force T&E Days, Los Angeles, California, 5-7 February 2008.
- Yovine S.,** *Kronos: A verification tool for real-time system*, Springer international journal of Software Tools for technology, October 1997.