HAL
open science

Capturing and structuring interoperability requirements: a framework for interoperability requirements

Nicolas Daclin, Sihem Mallek

To cite this version:

Nicolas Daclin, Sihem Mallek. Capturing and structuring interoperability requirements: a framework for interoperability requirements. Interoperability for Enterprise Software and Applications (I-ESA'14), Mar 2014, Albi, France. 10.1007/978-3-319-04948-9_20 . hal-01927951

HAL Id: hal-01927951

https://hal.science/hal-01927951v1

Submitted on 5 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.

Capturing and structuring interoperability requirements: a framework for interoperability requirements

N. Daclin[1], S. Mallek[2]

[1] LGI2P – Laboratoire de Génie Informatique et d'Ingénierie de Production – Ecole Nationale Supérieure des Mines d'Alès – Parc scientifique G. Besse, 30035 Nîmes cedex 1, France, nicolas.daclin@mines-ales.fr

[2] LGEI – Laboratoire de Génie de l'Environnement Industriel et des Risques Industriels Naturels – Ecole Nationale Supérieure des Mines d'Alès – 7 rue J. Renard, 30319 Alès cedex, France, sihem.mallek@mines-ales.fr

Abstract. The main objective of this communication is to discuss and present a framework for interoperability requirements. More precisely, the here presented research focuses on the dimensions to consider in order to capture and structure interoperability requirements in a precise way for parnters that want, further, to verify the truthfulness of these requirements. First, the concept of interoperability is presented according its main characteristics and the need to express it as a requirement. Then, the dimensions of the framework are highlighted and related in order to be suitable and usable. Finally, some interoperability requirements and their positionning in the framework for interoperability requirements are introduced to show the interest of such approach.

Keywords: Interoperability, requirements engineering, interoperability requirements, framework for interoperability requirements

1.1 Introduction

The "*ability of enterprises and entities within those enterprises to communicate and to interact effectively*"[1] -in other words interoperability - has become, over the past years, a major issue and a key factor of success for collaborating enterprises. As a consequence, numerous research works was proposed to characterize [2], implement [3][4] and improve [5][6] this aspect of a partnership. Although, these works are fully adapted to develop enterprise interoperability, few of those take an interest in the definition and the structuration of interoperability requirements that partners can verify during partnership. Nevertheless,

requirements engineering is an important aspect, offering the possibility for partners to dispose of a clear repository of interoperability requirements that guide them in their characterization and their use for verification in order to know the possible defects of a partnership in term of interoperability. In this context, it is interesting to propose a framework allowing to guide the definition, the capture and the structuring of interoperability requirements in a good way *i.e.* respecting approaches of requirements engineering proposed in other fields (*e.g* System Engineering [7]). Indeed, disposing, in the end, of a set of interoperability requirements clearly defined and structured has to enable that selected solutions are perfectly eligible to satisfy requirements and further, to assess their satisfaction throughout a partnership lifecycle. The here presented research, focuses on the development of a framework in order to guide partners in their interoperability requirements engineering according to basic dimensions related to the concept of interoperability and analysis levels.

This paper is structured as follow. After this brief introduction, the problematic and expected result of this research work is presented in section 1.2. A survey related to this research is given and discussed in section 1.3. Section 1.4 presents the proposed framework for interoperability requirements regarding its dimensions, and their relationship as well as the way to exploit the framework. The final section presents the conclusions and the prospects of this research.

1.2 Problematic and expected results

There are two main ways to make out when talking about requirements engineering practice [8]. On the one hand, the requirements engineering includes the elicitation, the writing, the refinement… of requirements to verify. On the other hand, the requirements management includes the support for user access, the versioning, the change, the traceability… of requirements. Although different in nature, these two aspects are complementary and are a crucial phase for stakeholders that want to obtain a system (*e.g.* technical, product, service, organization…) that meet their expectations initially expressed. Indeed, these requirements will have to be verified (*e.g.* analysis, inspection, test, simulation…), all along their life cycle from the engineering phase to the integration phase *via* the production phase, in order to show that a given system satisfies what is required and to avoid problems (*e.g.* drift from the original cost and quality, delay, cancellation in worst cases…). In the same manner, some requirements will have to be verified, as well, during the operational phase of the system in order to ensure and to uphold the satisfaction of these requirements until the retirement (or dismantling, recycling) of the system.

Although this step is a key step for the success of a project and, conversely, a factor of project failure [9], this aspect remains too often and unfortunately neglected by actors (stakeholders, acquirer…) due to several points such as [10]:

- Actors do not know what to do (*e.g.* they do not know how to write requirements, they do not understand the process);

- Actors do not understand why requirements are necessary (*e.g.* they do not understand the impact, they think requirements are just obviousnesses, they think it is just a document);
- Actors prefere rather do something else (*e.g.* they do not have time, they think the review process will highlight errors/omissions).

Beyond the classical functionnal requirements and in the limited frame of interoperability, this one is clearly identified as a non functional requirement (NFR) that have to be considered all along system's life cycle (from the engineering for its definition and also, after the initial use for its satisfaction) [11] [12]. Once more, even if interoperability is identified as a NFR which can have a strong impact on the functioning of a system and furthermore, its quality of service, it falls often in the category of the so called "forgotten '-illities" [13].

Thus, according to the statements mentionned hereinbefore, capturing, defining and structuring requirements - in our case interoperability requirements - is a lock for partners that want to implement interoperability in their partnership and further, to make sure that interoperabilty is properly reached while partnership is existing. Two major issues have to be considered for those that want to dispose of requirements. On the one hand, they have to face up with the expression of requirements. Authors of requirements dispose of languages, each one with their strengths and weaknesses from natural language (readability, richness, extensibility *vs.* low consistency and precision) to formal language (strong precision and consistency *vs.* low readability and constructability). They dispose also of practical rules guiding and allowing to write, review and share requirements in a proper way (we can mention here the well know acronyms M.U.S.T[1] and S.M.A.R.T[2] to ensure that a requirement is well expressed). This is not the purpose of the paper and reader may wish to refer to numerous works related to this field for more details [14]. On the other hand, requirements baseline is not sufficient to manage them and use it. They have to be grouped within a repository namely a "*set of requirements*". The main purpose of this set is to perfectly structure and precise the baseline of requirements in order to dispose of requirements that are easier (1) traceable in order to link a requirement to its origin and trace it throughout its lifecycle (*e.g.* defined, verified, allocated, satisfied…), (2) modifiable/removable/addable, (3) usable to implement relevant solutions that meet expectations and, (4) identifiable. Thus, this repository has to allow to manage properly the identified requirements.

Furthermore, the use of a clear set of requirements can guide and help stakeholders to elicit other requirements from a well-defined basis rather than with a jumbled list of requirements. In this way, a set of requirements has to possess some characteristics such as [15]:

---

[1] M.U.S.T is a mnemonic device that stands for Measurable, Useful, Simple and Traceable.
[2] S.M.A.R.T is a mnemonic device that Specific, Measurable, Achievable, Relevant and Traceable.

- *Complete*. The set of requirements includes complete definition of expectations. This means that the set includes all stakeholders' requirements (no more, no less).
- *Consistent*. The set of requirements is a consistent expression of expectations. This means there are no conflicts between requirements leading to an empty solution space.
- *Feasible*. The set of requirements is a feasible expression of expectations. This means that all requirements can be satisfied simultaneously.
- *Bounded*. The set of requirements takes place within a defined scope. The set clearly communicates expectations. Furthermore, the set is used by actors in charge of the implementation of solutions. Thus, requirements that are irrelevant with the scope have to be removed.
- *Structured*. The set of requirements is entirely understandable and assimilable, without undue cognitive loading for a given reader. Furthermore, each requirement is placed in a context in such a way that it is easily identifiable.

Thus, disposing of a framework of interoperability requirements helps actor to better capture and to structure their interoperability requirements in order to facilitate their implementation and their evaluation during partnership lifecycle. In this case, this framework has to consider several dimensions related to interoperability concept. First, the framework has to consider the concept of interoperability. This means that it is required to define precisely what interoperability is and, by extension, what interoperability is not. This aspect focuses mainly on the definition of the characteristics of interoperability during a partnership. Indeed, depending of the phase of the partnership (*i.e.* when interoperability is susceptible to occur and to be implemented within a partnership) interoperability can have more or less properties that define it. Then, framework for interoperability requirements has to consider the main problems of interoperability. Indeed, a requirement finds its origin in an expectation, a lack or dissatisfaction. Hence, the framework has to include this aspect in order to show which problem interoperability requirement has to satisfy. Finally, the framework for interoperability requirements has also to consider what "*perspectives*" can be impacted by an interoperability requirement. The objective is to identify precisely what criteria are concerned by interoperability in order to show the importance of the satisfaction of a given (set of) requirement(s).

All these dimensions have to be consistently related in order to offer a framework for interoperability requirements that respect the characteristics - of a set of requirements - as presented previously. This framework has to ensure completeness of interoperability requirements. With such framework actors are able to improve completeness by having a global view of interoperability requirements. Then, it has to ensure the consistence of requirements as well as the feasibility. The fact to dispose of a framework allows (1) to identify easier possible conflict between requirements and (2) to select easier the requirements to satisfy and to avoid requirements that are difficult (or even impossible) to perform. Finally, it has to be bounded and structured. The here proposed framework focuses on interoperability requirements. In this way, all requirements that are not relevant

to this field have to be removed, and have to be easily identifiable in agreements with all dimensions of the framework.

1.3 Interoperabilityand requirements: a quick scan

Numerous works dealing with interoperability, in different application domains (*e.g.* medicine, military, computer science), have been developed in recent years. As far as enterprise interoperability is concerned, these works endeavored to (1) clearly define and structure interoperability [2], (2) develop methods/tools to evaluate interoperability [5] [6] and (3), propose methodolgies [3][4] for its implementation trough existing and identified solutions. Some of these works are directly, or can be related to interoperability requirements aspects.

Interoperability frameworks focus on the structuration of the concept of interoperability by the consideration and the arrangement of its different aspects (problems, approach…). Thus, the major part of existing frameworks [16] [17] highlights that interoperability problems are related to conceptual, organizational and technological issues. Other frameworks go further and integrate more dimensions in order to capitalize interoperability solutions [18]. In that sense, frameworks for interoperability take, more often, an interest in the "*how*" (*i.e.* solution) rather than in the "*what*" (*i.e.* what we have to do?). Although, interoperability frameworks seem separated to the interoperability requirements consideration, they allow to structure perfectly interoperability and its aspects have to be fully integrated into a framework which aims to facilitate the capture and the structuring of interoperability requirements.

Methods to evaluate interoperability, whether in term of maturity [19] [20] or operational interoperability [6] [21], underlying the notion of requirement. Indeed, regarding maturity evaluation, each layer represents key points that enterprise has to satisfy in order to evolve throughout maturity levels. In that sense, each level describes, in natural language, what is requested to reach a given level of interoperability. These descriptions are no more no less expected conditions in order to guarantee the satisfaction of a given level. However, as mentioned just above, the notion of requirement is implicit and not formalized, but these models represent a strong contribution and can be used to express and elicit requirements. In the same manner operational measurement tools define and give equations in order to evaluate interoperability. The effective results of this measurement are often compared to expected results. Once more, expected results repesent, in some ways, requirements in term of operationnal interoperability.

Fully related to interoperability requirements aspects, [22] proposes a definition and a characterisation of fundamental interoperability requirements. Nevertheless, these requirements take place at high level of abstraction (first level of requirements, neither derived requirements nor refined requirements) and do not consider their structuration. In the same way, the last version of TOGAF [23] highlights the necessity to define clearly interoperability requirements. This necessity is characterized by guidelines that have to help to define and establish interoperability requirements. Besides, let us note [24] which proposes a set of

interoperability requirements (related to Gas and Electricity Smart Metering Systems) respecting a defined and common format to express them.

Regarding the consideration of interoperability requirements structuration we can mention [25] which proposes interoperability requirements and proposes a model as a formalized graph including 3 dimensions related to interoperability. Additionally, this work proposes some interoperability requirements that can be verified. This work is relevant but more oriented towards the verification of requirements thanks to formal techniques and does not consider the impact of interoperability on partnership and on a given partner. We can also note the interoperability framework proposed in [18] which includes an additional dimension (interoperability engineering phase) that consider requirements definition. However, this dimension is no more developed and none explanations are given on the way to exploit this aspect. Hence, it can be interesting to extend this dimension in order to fully consider interoperability requirements.

It emerges two points of interest from these considerations. First, the necessity to dispose of a clear definition and expression of interoperability requirements. Even if interoperability requirements can be perceived, they are not well expressed *i.e.* following a strict guideline to write requirements in a good way and that really exploitable. Second, the necessity to clearly structure each of identified interoperability requirements according to (1) the problems of interoperability, (2) the definition of interoperability and (3) the different "*perspectives*" that can be impacted by interoperability. If numerous interoperability requirements can be extract from existing works (beyond the correctness of their expression), the first point to consider is to dispose of a set of requirements that respect defined characteristics in order to provide feasible solutions that satisfy stakeholders' expectations.

1.4 Framework for interoperability requirements

1.4.1 Fundamental dimensions

The framework for interoperability requirements organizes and structures requirements to guide the selection of the best solution of interoperability but also to verify their satisfaction continually during a partnership. In other words, it has to make available a set of interoperability requirements that can be easily identifiable, verifiable and traceable. As a consequence, the framework has to consider:

- The problems of interoperability to which partners face up;
- The characteristics impacted by the non-satisfaction of requirements;
- The moment when interoperability occurs during a partnership.

At the end, the framework has to provide adapted requirements for each dimension in order to verify their satisfaction according to stakeholder's expectations. The proposed framework for interoperability requirements defines three basic dimensions such as: abstraction, analysis and interoperability levels. *Abstraction*

levels represent the categories of interoperability that can be developed in enterprise and thereby the problems of interoperability. A requirement comes from dissatisfaction, a lack, thus, the definition of requirement belongs to the definition of problem space (that a solution will have to satisfy further). Hence, it is required to consider the identified problems of interoperability in the framework. *Analysis levels* represent the characteristics that are impacted by the implementation of interoperability. The satisfaction of a given requirement can impact criteria either on partnership or on a given partner. Hence, requirements have to be defined according to these identified criteria in order to highlight what is expected in term of interoperability and, vis-à-vis of the partner and partnership. *Interoperability lifecycle levels* represent the requirements of interoperability according to what is expected in term of interoperability and all along a partnership. This means that requirements, to verify, evolve and are not necessarily the same as we go along the evolution of partnership. For instance, requirements at the beginning can be related to the interfacing while requirements at the end can be related to the unplugging in a proper way and with a limited impact on partner.

The dimension of *abstraction levels* takes up the classical categorisation of interoperability problems. For the record, these categories are defined such as [1]:

– **Conceptual**. This category is related to syntactic and semantic problems of interoperability. It can also integrate the expressivity problem.

– **Organisational**. This category is related to the definition of organisation structure, management techniques and policies within partners. This dimension can be refined, taking into consideration the responsibilities, authorities decision-making processes, policies, organisational processes and regulatory.

– **Technological**. This category is related to the problems inherent to the use of Information and Communication Technologies. Beyond the logical aspects this category can also concern physical aspects (*e.g.* compatibility problem of physical interfaces).

The dimension of *analysis levels* identifies the main characteristics that can be impacted by the implementation of interoperability namely, performance, stability and integrity.

– **Performance**. It refers to the ability of a system to reach its objectives. In this case, requirements positioned on this characteristic are about the expected performances in term of interoperability but also about the impact of interoperability on partner.

– **Stability**. It refers to the ability of a system to maintain its reliability and to get used to its environment.

– **Integrity**. It refers to the ability of a system to stay coherent and to be able to ensure its functions (or to get back to a functioning mode which is known) in case of modifications (*e.g.* loss of resource).

The dimension of *interoperability lifecycle levels* is related to the decomposition of interoperability in agreements with partnership evolution. Interoperability requirements can evolve during a partnership, depending if it is the beginning (*e.g.* connection of partner), the operational phase (*e.g.* exchange, sharing...) or the end (*e.g.* dismantling). Thus, each phase has to be clearly defined in term of interoperability. To do that, this dimension is concerned by compatibility, interoperation, autonomy and reversibility.

– **Compatibility**. It is related to the beginning of collaboration. Requirements positioned on this group are related to the interfacing aspect of the collaboration (logical as well as physical).

– **Interoperation**. It is related to the effective aspect of collaboration. In this case, requirements are related to the operational aspects of interoperability (*e.g.* performance).

– **Autonomy**. It is related to the effective aspect of collaboration as well. Requirements are related to the fact that partners are interoperable while retaining their operational thinking.

– **Reversibility**. It is related to the end of collaboration (*i.e.* dismantling). In this group requirements are related to the achievement, by a partner, of original objectives, functions… after a partnership and despite adaptations or changes.

The following figure presents the framework for interoperability requirements and the space requirements are positioned according its dimensions.

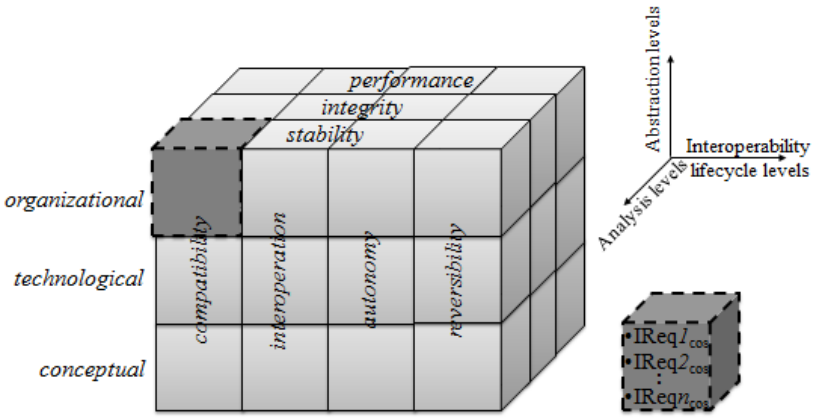

Fig. 1.1. Basic dimension of the framework for interoperability requirements

1.4.2 Additional dimensions

In order to better structure interoperability requirements, the framework for interoperability can be extended and consider further dimensions. At this stage, one other dimension is identified such as "*partnership/parnter*". As briefly addressed in previous sections, interoperability requirements can impact either the partner ship itself or partners themselves. For instance, the requirement "*the duration to connect application is less than x time units*" will impact the partnership and not necessarely a given partner. This requirement is positionned in the framework at the compatibility, performance and technological levels. Conversly, the requirement "*function f, performed by resource r involved in partnership, is even though performed"* will impact directly a partner (see Fig. 1.2 for its positionning in the framework). It is important to consider this aspect to know what is impacted by the requirements to find/adapt solutions or untighten requirements. Another dimension could be related to the verification means of requirements. In that sense,

verification methods and means must be defined to guide user in the verification phase. Finally, Fig 1.2 shows some examples of interoperability requirements within the framework.

	Compatibility	Interoperation	Autonomy	Reversibility
Performance	$Req_{partnership}$: "*resource r required by service has the required aptitude*" Verif: conceptual graphs			
Stability		$Req_{partnership}$: "*resource r performs expected services despite perturbing environment*" Verif: test		
Integrity			$Req_{partner}$: "*function f, performed by resource r involved in partnership, is even though performed*" Verif: expertise	$Req_{partner}$: "*function f is performed despite the loss of its resource r*" Verif: expertise

Fig. 1.2. Interoperability requirements for the organizational level

Other dimensions can be defined and added by users for a specific purpose or usage of the framework. However, the basic dimensions previously presented will have to be keeped since they represent the fundamental aspects of the interoperability requirements.

1.5 Conclusion and prospects

In a colloborative context, having interoperability requirements can allow to verify that a selected interoperability solutions will meet original expectations and further to ensure that interoperability is always reached. Nonetheless, do not possess a clear framework - to manage requirements - is to take the risk of not meeting certain interoperability expectations (non related requirement, conflict, omission, no traceability…). This paper has presented a first framework to better structure interoperability requirements according fundamental aspects to consider when interoperability is implemented. Moreover, this framework can also used as a guide to elicite other interoperability requirements. Futur works are related to the positioning of existing interoperability requirements which can be present in the literrature or coming directly from stakeholders. Furthermore, this framework will be also technically implemented with works performed in [25] to allow the selection and the verification of requirements.

References

[1] ISO/DIS 11345-1: Advanced automation technologies and their applications. Part 1: Framework for enterprise interoperability, 2009

[2] Chen D., Doumeingts G., Vernadat F. Architectures for enterprise integration and interoperability: Past, present and future. Computers in industry, vol. 59, issue 7, pp. 647-659, 2008.
[3] Daclin N., Chen D., Vallespir B. Methodology for enterprise interoperability. 17th IFAC World Congress, Seoul, South Korea, July 2008.
[4] Bourey J.P., Grangel R., Doumeingts G., Berre A. J. Report on Model Driven Interoperability, INTEROP Network of Excellence IS 508011, deliverable DTG 2.3, 2007.
[5] C4ISR Architecture Working Group (AWG). Levels of Information Systems Interoperability (LISI), USA Department of Defense, 1998.
[6] Ford C.T. Interoperability measurement, PhD Thesis, Department of the Air Force Air University, Air Force Institute of Technology, 2008.
[7] INCOSE. Systems Engineering Handbook – A guide for system life cycle processes and activities – v.3.2.1, INCOSE-TP-2003-002-03.2.1, January 2011.
[8] Favaro J., De Koning H.S., Schreiner R., Olive X. Next generation requirements engineering, 22th Annual INCOSE International Symposium, Rome, Italy, 2012.
[9] The Standish Group. The chaos report. The Standish Group International, Inc, 1995.
[10] Bochmann G.V. Writing better Requirements. University of Ottawa, slide only, 2010.
[11] Ross A. M. Defining and using the new "ilities", Systems Engineering Advancement Research Initiative (SEARi) working paper series, 2008.
[12] ATHENA Integrated Project. Advanced Technologies for Interoperability of Heterogeneous Enterprises Networks and their Applications, Integrated Project description of work, 2003.
[13] Willis J. Systems Engineering and the forgotten '-Illities, 14th Annual Systems Engineering conference, San Diego, CA, the United States of America, 2011.
[14] ISO/IEC/IEEE. Systems and software engineering – Life cycle processes – Requirements engineering, ISO/IEC/IEEE 29148:2011, 2011.
[15] INCOSE. Guide for writing requirements, v1.0, Requirements WG-INCOSE, INCOSE-TP-2010-006-01, 2012.
[16] IDEAS. Interoperability Development for Enterprise Application and Software, IST – 2001 – 37368, IDEAS Project Deliverables (WP1-WP7), Public reports, 2003.
[17] ATHENA Integrated Project. Requirement for interoperability framework, product-based and process-based interoperability infrastructures, interoperability life-cycle services, ATHENA deliverable A4.1, 2005.
[18] INTEROP. Enterprise Interoperability-Framework and knowledge corpus-Advanced report, INTEROP Deliverable DI.2, DI (Domain Interoperability), Network Of Excellence IS 508011, 2006.
[19] ATHENA Integrated project. Framework for the Establishment and Management Methodology, ATHENA Integrated Project, Deliverable DA1.4, 2005.
[20] C4ISR. Levels of Information Systems Interoperability (LISI), USA Department of Defense, Architecture Working Group (AWG), 1998.
[21] Kasunic M., Anderson W. Measuring systems interoperability: challenges and opportunities, Software engineering measurement and analysis initiative, Technical note CMU/SEI – 2004 – TN – 003, 2004.
[22] Daclin N. Embedding interoperability in System of System: characterization and definition of fundamental requirements, 13th IFIP Working Conference on Virtual Enterprises (PRO-VE'12), Bournemouth, United Kingdom, 2012.
[23] The Open Group Architecture Framework. TOGAF version 9, chapter 29 Interoperability requirements, 2011.
[24] Harrison S., Brogden J. Interoperability requirements, Supplier Requirement for SMART Metering (SRSM) project report, 2007.

[25] Mallek S., Daclin N., Chapurlat V. The application of interoperability requirement specification and verification to collaborative processes in industry, Computer in Industry, vol.63, issue 7, pp. 643-658, 2012.