HAL open science

Validation and Verification of Interoperability Requirements

Mamadou Samba Camara, Rémy Dupas, Yves Ducq

To cite this version:

Mamadou Samba Camara, Rémy Dupas, Yves Ducq. Validation and Verification of Interoperability Requirements. 6th International IFIP Working Conference on Enterprise Interoperability (IWEI), May 2015, Nîmes, France. pp.39-52, 10.1007/978-3-662-47157-9_4 . hal-01438406

HAL Id: hal-01438406

https://inria.hal.science/hal-01438406v1

Submitted on 17 Jan 2017

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.

Distributed under a Creative Commons Attribution 4.0 International License

Validation and Verification of interoperability requirements

Mamadou Samba Camara[1], Rémy Dupas[2], and Yves Ducq[2]

[1] ESP, Université Cheikh Anta Diop, Génie Informatique,
BP : 5085 dakar-fann,Dakar, Senegal,
mamadou.camara@ucad.edu.sn
[2] Université de Bordeaux IMS, UMR 5218,
F-33400 Talence, France.

Abstract. The research objective of this work is to develop an approach to reach inter-enterprise interoperability and to test its achievement using practices from the software engineering process. Four fundamental activities are identified in the software process: software specification, software development, software validation and software evolution [1]. In this work, the interoperability requirements specification is based on measurable and non-measurable quality characteristics. It is also demonstrated that the improvement proposed in software specification activity will have positive impact on the software development activity. For the validation activity, the definition of a interoperability testing sub-process is made through a two-step decomposition: one step to verify measurable requirements and another to validate non-measurable ones.

Key words: Enterprise Interoperability, Software Specification, Software Validation, Service Oriented Architectures, Business Process Modeling

1 Introduction

Interoperability is defined by the ALCTS[1] [2] as the ability of two or more systems or components to exchange information and use the exchanged information without special effort by either system. Improving interoperability depends largely on the implementation of a Collaborative Information System(CIS) by means of a software engineering process. In this work, the interoperability requirements specification is based on measurable and non-measurable quality characteristics. It is also demonstrated that the improvement proposed in software specification activity will have positive impact on the software development activity. For the validation activity, the definition of a interoperability testing sub-process is made. The next three sections presents respectively the literature reviews of requirements specification, interoperability in relation to the software process and business process performance. The two following sections are dedicated to the research design and the application respectively.

[1] The Association for Library Collections and Technical Services

2 Literature of software requirements verification and validation

After the validation activity, the software system is delivered to the customer and is installed and put into practical use. When the validation activity reveals problems, this means that the system is not good enough for use, then further development is required to fix the identified problems [1]. Software system requirements are often classified as functional and non-functional requirements [1]. Software validation or, more generally, verification and validation is intended to show that a system both conforms to its specification and that it meets the expectations of the system customer [3, 1]. Except for small programs, systems could not be tested as a single, monolithic unit. The testing process is made up of three stages, Development testing, System testing and Acceptance testing, in which system components are tested then the integrated system is tested and finally, the system is tested on customer's data [1]. The aim of verification is to check that the software meets its stated functional and non functional requirements [3, 1, 4]. The aim of validation is to ensure that the software meets the customer's expectations (i.e. expectations of the organization that commissioned the system)[3, 1]. Properties expressed as quantitative measures can be naturally verified [5]. Properties that refer to subjective feeling can be difficult to verify and are a natural target for validation [5].

3 Interoperability literature review

A review of the literature conducted to analyze how the activities of software specification, software development and software validation [1] were carried out in the interoperability domain.

3.1 Interoperability requirements

Approaches used to represent interoperability requirements, in the literature can be summarized as following with their limitations:

1. the maturity models [6, 7, 8]: repetition, ambiguity, imprecision and incoherence because the needs are expressed in natural language [9].
2. formal representation of interoperability requirements [9, 10]: the implementation of this approach may experiment scalability problems [11].
3. interoperability requirements as problems
   a) requirements are specified by the mean of collaboration processes models: the requirements are disconnected from existing systems and also to the interoperability problems to solve
   b) the interoperability matrix [12, 13]: do not provide a structured set of requirements [14].

3.2 Collaborative Information System (CIS) architecture

A CIS aims at supporting "Information Systems (IS) Interoperability", that is to say, to satisfy requirements such as data conversion, application sharing and process management [15]. The CIS should then include two different parts: connectors to be plugged into partners information systems and the concrete entity managing the collaboration: an intermediate information system. Several research works were conducted in order to find logical and technical solutions for the CIS. Approaches proposed in these works can be categorized in two groups: Model-Driven Interoperability (MDI) approaches [16, 17] and Business Process Lifecycle (BPL) based approaches [18]. It can be noticed that, in all approaches for CIS development, the proposed platforms are based on Service Oriented Architectures (SOA) [19]. Requirements specification proposed in the approaches for the development of CIS does not provide sufficient information to describe interoperability problems and then facilitate the software development activity.

3.3 SOA testing

Since our problem is related to the verification and the validation of interoperability, we will focus primarily on testing approaches and techniques for service-based systems. For the testing of SOA applications, [20] identified four distinct testing layers: Unit testing, Service testing, Integration testing and System testing. [21, 22] advocated for the realization of system testing at process level. The first limitation of this work [21], is the fact that the business process performance metrics used in the test process are not defined. In our opinion, although interoperability implementation is generally based on SOA, the testing approaches proposed for service-based systems are not suitable for testing inter-enterprise interoperability achievement. Indeed, these approaches [22, 21, 20] do not reference interoperability problems, which makes them useless for verifying the elimination of the latter.

4 Business process performance indicators calculation from event logs

The Business Process represents a chain of interrelated activities that normally must be connected with the customer requirements [23]. Process models are built in the design phase and are later used to implement the information system. Process performance measurement is an important aspect of Business Process Management (BPM). In this research, the considered process PIs are the average elapsed time, average cost and percentage of failure at the process level and represent the aggregation of the PIs for the activities [24, 25]. This choice relies on the assumption that these characteristics encompass all other types of dynamic properties of business processes [11].

An "event log" is defined as "a chronological record of computer system activities" which are saved to a file on the system [26]. Any information system using transactional systems will provide workflow information in some form and to some extent, such as tasks available, their events and the details of these events like starting/ending timestamp [27, 28]. The main data source that can be used for the calculation process PIs in the verification of interoperability requirements are the integrated event logs which are widely used in the fied of cross-organizational process mining. Cross-organizational workflow is usually distributed on different servers owned by different partners or different organizations [29, 30]. Cross-organizational workfows are enabled by web-services technology [31]. A Framework for workflow integration based on process mining is proposed in [30]. The integrated log results from the integration between the running logs of two or more organizations. In the running log collected from each organization, an event record is an 8-tuples containing Case(Id), Activity (Name), Start time, end time, Required resources, Released resources, Messages received, Messages sent. The data integrated between the running logs contains an additional column labeled "Organization".

5 Research design and hypothesis

The present research work aims at developing an approach to reach inter-enterprise interoperability and to test its achievement using practices from the software engineering process. In the approach proposed in [32], interoperability requirements are specified by representing interoperability problems directly on business process models. The testing sub-process defined in this work [32] was limited to the validation of interoperability requirements but did not enable the verification because of the nature of these requirements. Four fundamental activities are identified in software process: software specification, software development, software validation and software evolution [1]. For each of the first three activities, our approach proposes to define a sub-goal and determines how to achieve it:

1. **Software specification**. Interoperability requirements specification ensures certain characteristics: requirements are structured (i.e. understandable and identifiable) and testable.
2. **Software development**. The structured nature of interoperability requirements will ease the logical and technical architecture definition.
3. **Software validation**. The testability of interoperability requirements facilitates the definition of a method that enables testing the level of achievement of interoperability improvement. The objective will be to prove that the implementation of the CIS has improved the interoperability up to the desired level.

We propose a framework for the definition of a Performance Measurement System (PMS) (Fig. 1) made up, in its complete version, of cost, time and quality PIs for all the processes impacted by the implementation of the CIS. To

have a comprehensive verification process, interoperability measures defined at process level will be used. Indeed, as it is explained in [11], defining measures at activity level will produce too numerous indicators and increase the PMS complexity. The framework should be understood as road-map where each state represents additional process PIs on which the verification step of the interoperability testing sub-process (section 5.3) is applied using approaches defined in process mining. We assume that the PMS is constructed through a series of states where S_i denotes the set of indicators in the $state_i$

- the difference $S_i - S_{i-1}$ represents additional indicators defined in order to take into account new concerns defined in S_i. Therefor S_{i-1} is always included in S_i
- the directed arcs from S_{i-1} to S_i represents the influence of S_{i-1} to $S_i - S_{i-1}$ that is also a part of S_i

The four states as well as their influence relationships are defined as following:

1. State1. The first state concerns time PIs for process instances excluding those containing loops. A loop causes a task to be executed multiple times for a given case [33, 28]. According to [34], if the maximum number of loops is high then the number of loops can greatly differs from instance to instance. In other words, the presence of loops will results in variability in the time and other process PIs between different instances of a given process. In our opinion, the origin of this variability is the random nature of reasons why process or task can fail.
2. State2. In the second state, the new concern is the measurement of loops in business processes. Traditional workflow systems define loops for repeating parts of the process. Tasks or sets of tasks are iterated for multiple reasons:
   - they have not yielded an expected result [35].
   - the workfow management system is configured to act automatically (i.e. pass the case back to the last performer) after a long passivity of performers [31]. Therefore, we can conclude that the time PIs of the state1 influence the failure (or loop) PIs added in state2.
3. State3. The process PIs added in state3 are related to time but the difference with the ones defined in state1 is the fact that they also measure the variability of time caused by loops. The group of indicators in state3 are influenced by the indicators defined in state2 because, according to [36], the quality of the process expressed in the number of failures or loops is an indicator for necessary rework. More generally, it is safe to say that quality characteristics of business process influence time (and cost).
4. State4. The new process PIs included in state4 are related to cost. These measures receive impact of time and failure.

The rest of this paper represents the first state of the framework, the scope of the verification step is then limited to process PIs of $state_1$ (i.e. time process PIs). Process mining techniques enable to automatically mines upper bounds for different key performance indicators (like waiting times, execution times etc) of

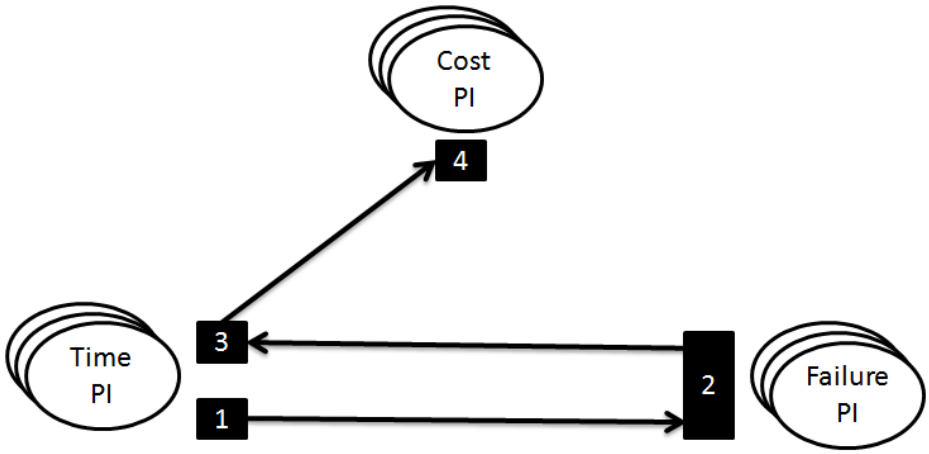

Fig. 1. Framework for the definition of the PMS

a process by taking into account both the timestamps of tasks in a log and the overall structure of the process model given as input [37, 33, 38].

5.1 Improving the interoperability requirements specification

The interoperability requirements specification proposed in this work results in the definition of two categories of requirements:

1. **Measurable interoperability requirements**. This category concerns measurable interoperability requirements that can be verified in the testing activity. Indeed, according to [1], non-functional requirements should be written quantitatively, whenever possible, so that they can be objectively tested. The measurable interoperability requirements are metrics representing the desired interoperability level for each business process as a result of the implementation of the CIS. These measurable requirements are related three process PIs considered as sufficient to measure interoperability by [11]: the average elapsed time, average cost and percentage of failure. As explained in the beginning of section 5, this study uses only the average elapsed time and will measure this indicator for all business processes impacted by the implementation of the CIS.
2. **Non-measurable interoperability requirements**.This category contains interoperability requirements that can be validated in the testing activity but

not verified because of to their non-measurable nature. For this category, the requirements specification consists in representing directly interoperability problems in business process models, mainly in "As-is" ones. The adopted representation is based on a principle that consists of distinguishing between business activities and Non-Value-Added (NVA) activities, mainly inspired by the work done in [11]. Business activities create value in a business process. The NVA activities are defined as the components of business processes that represent efforts between partners to achieve interoperability in information exchange. According to [39], Non-value-adding work, creates no value for customer but is required in order to get the value-adding work done. One the most important principles of Business Process Reengineering (BPR) is the value-focus principle. The value-focus principle states that [40], NVA activities must be targeted for elimination in order to save time and/or money. Interoperability problems are depicted in BPMN (Business Process Model and Notation)[41] business process models. The stereotypes [42], generally used for the UML language, will be used to differentiate NVA to business activities in the BPMN process models. BPMN is more suitable for modeling collaborative processes because it helps situate the boundaries of the collaborating companies using pools. The proposed representation technique enables to overcome limitation presented in the section 3.1, since the requirements will relate interoperability problems to target elements such as people, organization units and material resources which are clearly identifiable in the collaborating enterprises.

Interoperability requirements are part of non-functional requirements. Both measurable and non-measurable requirements defined in this research work are considered as non-functional requirements because of their relation to interoperability. Indeed, according to [1], non-functional requirements arise through user needs, because of budget constraints, organizational policies or the need for interoperability with other software or hardware systems.

5.2 Ease architecture definition

Software development is the activity where the software is designed and programmed. The requirements specification results are inputs of the design and implementation processes [1]. In order to develop solution for interoperability problems, the interoperability matrix utilizes the concepts of solution space [12, 13]. The solution space is composed of the three dimensions of the INTEROP framework. The cross of an interoperability barrier, an interoperability concern and an interoperability approach includes the set of solutions to breakdown a same interoperability barrier for a same concern and using a same approach. In order to determine correctly the solution (third dimension), there must be sufficient information to describe interoperability problems (two first dimensions). The requirements representation in process models contains the following information: tasks where interoperability problems arise and the resources (human

and non-human) involved in the interoperability problems. This set of information will facilitate the design process and then improve the software development activity.

5.3 Adapt validation activity

In the SE process, the validation activity can be considered as a test process that can be divided into a set of test sub-processes defined to perform a specific test level (e.g. system testing, acceptance testing) or test type (e.g. usability testing, performance testing) within the context of an overall test process for a test project [43]. A test type is a group of testing activities that are focused on specific quality characteristics [43]. In the validation activity of software process, the improvements will consist on considering interoperability as a quality characteristic and developing the interoperability test type. On the basis of the recommendations given in the literature of service-based systems testing, the interoperability testing sub-process will be executed at system testing level using business process models (section 3.3). The interoperability testing sub-process is divided into two steps: Verification and Validation of interoperability requirements.

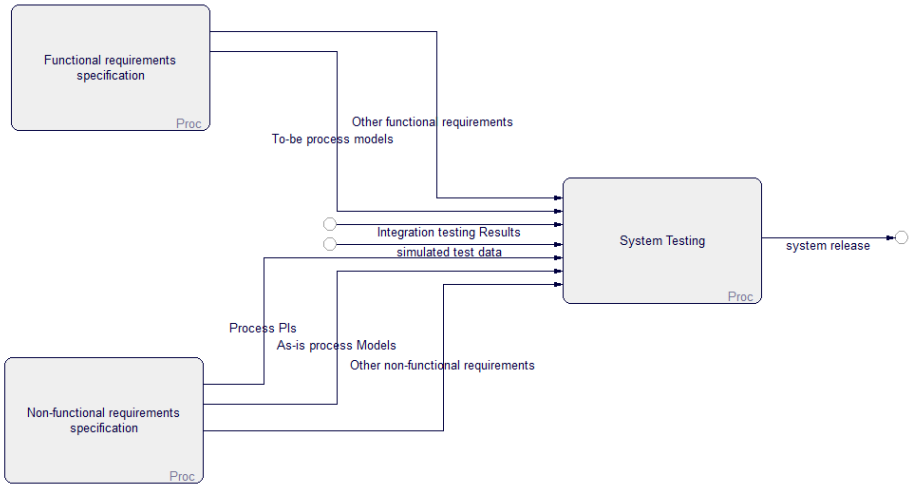

Fig. 2. The interoperability testing sub-process

Verification of interoperability requirements The first step of the interoperability testing sub-process is the verification of measurable interoperability requirements. Each business process in the scope of the project must be executed several times to calculate its average elapsed time. As mentioned in section 4, information needed for the computation of this process PI average elapsed time are the probability related to the different path and the average elapsed time of the activities in the process model. In the context of inter-enterprise collaboration these information can be obtained using the cross-organizational workflow model and the integrated event log. Applying the Framework for workflow integration based on process mining described in section 4 will help to get the cross-organizational workflow model and the integrated event log. The events log analysis enable calculating process PIs that represent measurable interoperability requirements. The goal of this step is to verify if the level of interoperabilty is improved at the level expected from the implementation of the CIS as determined in the software specification activity. When the percentage of interoperability improvement is judged unsatisfactory then the validation step is launched in order to identify, understand and possibly resolve interoperability problems.

Validation of interoperability requirements The second step of the interoperability testing sub-process is the validation of non-measurable interoperability requirements. The main input of this step (Fig. 2) is the "As-is" business process models which contain the interoperability requirements specification (section 5.1). The validation is carried out by executing each process in order to verify if all the NVA activities it contains in its "As-is" version are effectively eliminated by the implementation of the CIS. The execution of a business process may reveal the presence of NVA activities. In this situation, the "As-is" business process model gives sufficient information about the interoperability problems related to the concerned NVA activities. The information will be used to fix the interoperability problems.

6 Application

The illustrative example used to demonstrate the applicability of the methodology involves a supply chain in which an interoperability investment is used to improve the quality of the collaboration. The partners involved in this collaboration are a customer (an e-commerce company), a stockist (a warehouse owner), a customs agent and the customs administration. The goal of the interoperability investment is to allow the customers to be quickly connected at a low cost and with flexibility to their partners and to the customs administration using an interoperable information system.

6.1 Interoperability requirements specification

The specification of both measurable and non-measurable interoperability requirements is illustrated through a goods entry (collaboration) process in which

all four partners concerned by the investment participate. The goods entry process begins when the customer places an order and terminates when the stockist updates the material accounting and informs the customer that it can begin to distribute and sell the goods. For this process, the specification of the non-mesurable requirements consist in representing NVA activities in the "As-is" version of the model using stereotypes (Fig. 3).

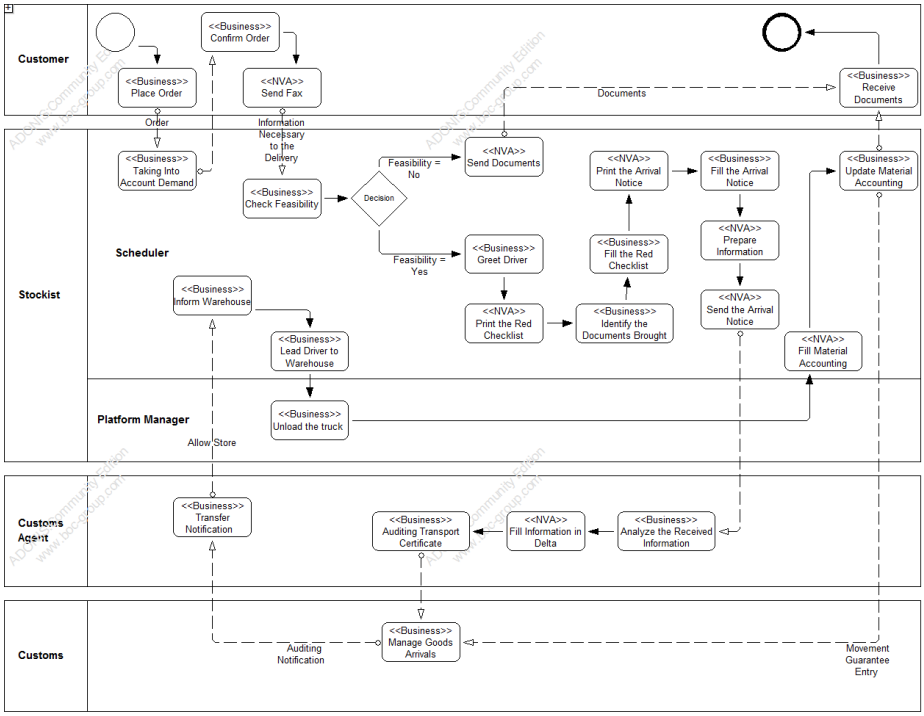

Fig. 3. Goods entry process- "As-is" version

The specification of the measurable interoperability requirements for the goods entry process consist in calculating the percentage of improvement in the average elapsed time PI between "As-is" and "To-be" situations. The partners in the illustrative example provide the average elapsed time for each activity in the goods entry business process, which allows the calculation of the average elapsed time process PI of 183 min for the "As-is" situation. The "To-be"

process models essentially depends on the number of identified barriers that the solution is intended to remove. In the the illustrative example, the SOA-based solution was expected to remove all of the identified barriers. Therefore, to arrive at the "To-be" process model, the NVA activities are simply removed from the "As-is" model (Fig. 3). Simulation of the "To-be" process models can be used to estimate the average elapsed time process PI which has a value of 152 min. The value of the metrics representing this interoperability requirement is -17, meaning that the decrease of average elapsed time between the "As-is" and "To-be" situations for the goods entry process is expected to be of 17 percent.

6.2 Software Architecture definition and implementation

Consider the following activities of the goods entry process: "Transfer notification"(1), "Inform warehouse" (2), "Lead driver to warehouse"(3), "Unload the truck"(4), "Fill Material Accounting"(5), "Update Material Accounting"(6). All these activities are business ones except "Fill Material Accounting" which is a NVA activity representing a non-measurable interoperability requirement and it is hypothesized that an interoperability barrier existing somewhere in the process. Identifying software systems supporting the activities connected to the NVA activity will help in determining technological barriers between(e.g. 1 is executed using Delta wich is non-inter-operable with Sage used for 5 and 6. Understanding why these systems are not inter-operable will facilitate finding solutions. Web services are an industry effort to provide platform-independent SOA using interoperable interface descriptions, protocols, and data communication [44]. In the software development activity, the decision was made to use web services architecture to implement the CIS. The main reason is that web services architecture can support the business activities while removing NVA activities identified in all the "As-is" business process models obtained in the requirements specification. The software development activity was realized using .NET Web Services implemented using the Microsoft .NET platform [45].

6.3 Interoperability testing sub-process application

The interoperability testing sub-process is executed in the system testing phase after the testing of the functional requirements. The functional requirements testing allowed verifying that the implemented CIS supports the business process as described in the "To-be" models. For the goods entry process, the verification of interoperability requirements shown a reduction of 20 percent in average elapsed time, a value greater than the threshold set in the requirements specification. The validation of interoperability requirements conducted reveals that all the NVA activities identified in the "As-is" goods entry process models were removed in the implementation of the CIS. The results of verifcation and validation steps allow us to assert that the interoperability problems in the goods entry process were fully eliminated.

7 Conclusion

This work was aimed to improve software specification, development and validation activities of software process in projects related to interoperability. The interoperability requirements specification is based on measurable and non-measurable quality characteristics. Non-measurable requirements are defined as interoperability problems directly depicted in business process models using the concept of NVA activity. The average elapsed time process PI is used for the development of measurable interoperability requirements. It has been then demonstrated that the proposed form of interoperability requirement specification can positively impact the software development and the software validation activities. For the validation activity, the definition of a interoperability testing subprocess is made through a two-step decomposition: one step to verify measurable requirements and another to validate non-measurable ones. A perspective of this work will consist in extending the verification step to the remaining process PIs defined in [11] : average cost and percentage of failure.

References

1. Sommerville, I.: Software Engineering. Addison-Wesley. (2011)
2. ALCTS: Description and access task force on metadata (2010)
3. Boehm, B.W.: Software engineering: Rd trends and defense needs. In Wegner, P., ed.: Research Directions in Software Technology. MIT Press (1979)
4. Chapurlat, V., Braesch, C.: Verification, validation, qualification and certification of enterprise models: Statements and opportunities. Computers in Industry (2008)
5. Baresi, L., Pezze, M.: An introduction to software testing. Electronic Notes in Theoretical Computer Science **148**(1) (2006) 89111
6. Tolk, A., Muguira, J.: The levels of conceptual interoperability model (2003)
7. Morris, E., Levine, L., Meyers, C., Place, P., Plakosh, D.: System of systems interoperability (sosi): Final report. Technical report, Carnegie Mellon University (2004)
8. Kingston, G., Fewell, S., Richer, W.: An organizational interoperability agility model. Technical report, DSTO Fern Hill,Department of Dfence,Canberra ACT 2600,Australia (2005)
9. Mallek, S., Daclin, N., Chapurlat, V.: The application of interoperability requirement specification and verification to collaborative processes in industry. Computers in Industry **63**(7) (2012) 643658
10. Cornu, C., Chapurlat, V., Quiot, J.M., Irigoin, F.: Customizable interoperability assessment methodology to support technical processes deployment in large companies. Annual Reviews in Control **36**(2) (2012) 300–308
11. Camara, M., Ducq, Y., Dupas, R.: A methodology for the evaluation of interoperability improvements in inter-enterprises collaboration based on causal performance measurement models. International Journal of Computer Integrated Manufacturing 27(2) (2014)
12. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for enterprise integration and interoperability:past, present and future. Computers in Industry **59**(7) (2008)

13. Ducq, Y., Chen, D., Doumeingts, G.: A contribution of system theory to sustainable enterprise interoperability science base. Computers in Industry **63**(8) (2012) 844857
14. INCOSE: Guide for writing requirements. Technical report, INCOSE (2012)
15. Touzi, J., Lorr, J.P., Bnaben, F., Pingaud, H.: Interoperability through model-based generation: The case of the collaborative information system (cis). In Gonalves, R.J., Mller, J.P., Mertins, K., Zelm, M., eds.: Enterprise Interoperability. New Challenges and Approaches. Springer, London (2007)
16. Chungoora, N., Young, R.I., Gunendran, G., Palmer, C., Usman, Z., Anjum, N.A., Cutting-Decelle, A.F., Harding, J.A., Case, K.: A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. Computers in Industry **64**(4) (2013) 392–401
17. Khadka, R., Sapkota, B., Pires, L.F., Sinderen, M.v., Jansen, S.: Model-driven approach to enterprise interoperability at the technical service level. Computers in Industry **64**(8) (2013) 951965
18. Franco, R.D., De Juan-Marn, R., Rodrguez-Merino, C., Martnez, J.L.: Colnet platform: Resilient collaborative networks through interoperability. In Poler, R., Doumeingts, G., Katzy, B., Chalmeta, R., eds.: Enterprise Interoperability V: Shaping Enterprise. Springer, London (2012)
19. Pessoa, R.M., Silva, E., Sinderen, M.v., Quartel, D.A.C., Pires, L.F.: Enterprise interoperability with soa: a survey of service composition approaches. In: Enterprise Distributed Object Computing Conference Workshops, IEEE (2008)
20. Wieczorek, S., Stefanescu, A.: Improving testing of enterprise systems by model-based testing on graphical user interfaces. In: 17th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, IEEE (2010)
21. Lee, Y.: Event-driven soa test framework based on bpa-simulation. In: First International Conference on 'Networked Digital Technologies(NDT 2009), IEEE (2009)
22. Vieira, M., Leduc, J., Hasling, B., Subramanyan, R., Kazmeier, J.: Automation of gui testing using a model-driven approach. In: the 2006 international workshop on Automation of software test, ACM (2006)
23. Doumeingts, G., Ducq, Y., Vallespir, B., Kleinhans, S.: Production management and enterprise modelling. Computers in Industry 42(2-3) (2000)
24. Yaxiong, T., Zhen, X., Huibin, X.: Bpm exception monitoring based on process knowledge. In: Proceedings of IEEE Conference on Cybernetics and Intelligent Systems (CIS), Singapore , 28-30 June 2010, IEEE Xplore Digital Library (2010)
25. IBM: Websphere business modeler: Process cost analysis (2014)
26. Jans, M., Alles, M., Vasarhelyi, M.: The case for process mining in auditing: Sources of value added and areas of application. International Journal of Accounting Information Systems 14(1) (2013)
27. Liu, Y., Zhang, H., Li, C., Jianxin Jiao, R.: Workflow simulation for operational decision support using event graph through process mining. IDecision Support Systems in Industry 52(3) (2012)
28. van der Aalst, W., Dongen, B.F.v. Herbst, J., Maruster, L. Schimm, G.W.A.: Workflow mining: a survey of issues and approaches. Data & Knowledge Engineering 47(2) (2003)
29. Van Der Aalst, W.M.P.: Configurable services in the cloud: supporting variability while enabling cross-organizational process mining. In: OTM'10 Proceedings of the 2010 international conference on On the move to meaningful internet systems
30. Zeng, Q., Sun, h.X., Duan, Hua Liu, C., Wang, H.: Cross-organizational collaborative workflow mining from a multi-source log. Decision Support Systems (2013)

31. van der Aalst, W.M.P., Reijers, H.A., Weijters, A., van Dongen, B., Alves de Medeiros, A., Song, M., Verbeek, H.: Business process mining: An industrial application. Information Systems 32(5) (2007)
32. : Interoperability improvement in inter-enterprises collaboration: A software engineering approach. In: Enterprise Interoperability VI: Interoperability for Agility, Resilience and Plasticity of Collaborations. Springer (2014)
33. van der Aalst, W. M. P. Weijters, A.: Process mining: a research agenda. Computers in Industry 53(3) (2004)
34. Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.C.: Business process intelligence. Computers in Industry 53(3) (2004)
35. Gunther, C.: Process Mining in Flexible Environments. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands (2009)
36. Muehlen, M.z., Rosemann, M.: Workflow-based process monitoring and controlling-technical and organizational issues. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 4-7 Jan. 2000, IEEE Xplore Digital Library (2000)
37. : Business Process Intelligence. In: Handbook of Research on Business Process Modeling. IGI Global (2009)
38. Rozinat, A., Mans, R., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Information Systems 34(3) (2009)
39. Hammer, M.: Beyond Reengineering. HarperBusiness (1996)
40. Hammer, M., Champy, J.: Reengineering the corporation: A manifesto for business revolution. New York: HarperBusiness (1993)
41. OMG: Business process model and notation (bpmn). version 2.0 (2011)
42. Frankel, D.S.: Model Driven Architecture: Applying MDA to Enterprise. Computing. Wiley (2003)
43. ISO/IEC/IEEE: 29119-2-software and systems engineering. software testing. (2013)
44. Chappell, D.: Enterprise Service Bus. O'Reilly Media (2004)
45. Ferrara, A., MacDonald, M.: Programming .NET Web Services. O'Reilly Media, Inc (2002)