



Qualitative Evaluation of Manufacturing Software Units Interoperability using ISO 25000 Quality Model

H Basson, M Bouneffa, M Matsuda, A. Ahmad, D Chung, E Arai

► To cite this version:

H Basson, M Bouneffa, M Matsuda, A. Ahmad, D Chung, et al.. Qualitative Evaluation of Manufacturing Software Units Interoperability using ISO 25000 Quality Model. I-EISA 2016, 8th International Conference INTEROPERABILITY FOR ENTERPRISE SYSTEMS AND APPLICATIONS, 2016, Guimarães, Portugal. hal-01894103

HAL Id: hal-01894103

<https://hal.science/hal-01894103>

Submitted on 12 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Qualitative Evaluation of Manufacturing Software Units Interoperability using ISO 25000 Quality Model

H. Basson¹, M. Bouneffa¹, M. Matsuda², A. Ahmad¹, D. Chung³ and E. Arai⁴

¹ LISIC, University of Lille, ULCO, EILCO, France

² Kanagawa Institute of Technology, Japan

³ Rockwell Automation, Cleveland, USA

⁴ Osaka University, Osaka, Japan

Abstract. In the context of development and evolution of heterogeneous industrial systems, the quality of interoperability among manufacturing software units is a major concern. Qualitative characteristics of the interoperability such as reliability, functionality, performance, and security are necessary for the assurance of high quality of the developed manufacturing applications. The implemented interoperability among various software units of manufacturing applications often have various quality criteria. Therefore, an adequate qualitative modeling is required to satisfy the qualitative evaluation of the interoperability. This paper presents the use of quality model developed in respect of quality standard ISO 25000 series as a means to permit the comprehensive evaluation of interoperability of the applications developed regarding the ISO 16100 standard series.

Keywords: Manufacturing Software Units, Quality of Interoperability, ISO 25000 series, Quality Model, ISO 16100 series.

1.1 Introduction

The functional and qualitative characteristics of interoperability cover a wide range of possible capabilities according to the considered types of system, their description level, and their organizations. In the manufacturing domain, along with the evolution of software applications in various activities domains, the interoperability has become a major critical aspect influencing the quality of working systems. Various definitions have been proposed for the interoperability, as illustration, Kurt Kosanke cited 22 different definitions extracted from the literature on the topic [1]. For its specified domain, an important work has been

realized by the US DoD to propose a schema called LISI (Levels of Information Systems Interoperability) [2] where the interoperability was defined as "*The ability of systems, units or forces to provide services and accept services from other systems*". The IEEE definition designates the interoperability as the "*ability of two or more systems or components to exchange information and to use the exchanged information*".

The present work considers mainly the manufacturing applications for which the constituting manufacturing software units may require interoperability facilities among units for the application development and use. More precisely, the interoperability mechanisms such as data exchange, messages communication, services calls, etc. may need to focus on the quality of interoperability and its functioning between concerned manufacturing application units.

The interoperability multiplicity of levels inevitably encourages the standardization activities [3]. Effectively, the governing bodies have always encouraged and supported the standardization to insure an integration of multiple heterogeneous systems to operate as a coherent whole.

In this paper, we focus on the standards, dedicated to interoperability of manufacturing software units and the standards that can be used for the qualitative evaluation. The major goal is to use adequate recognized standards to examine the proposed practices for system integration through interoperability and to evaluate its quality using ISO recommended practices. Primarily, our work considers the ISO 16100 series standard [4] to propose an approach and reference model to assure better qualitative level of interoperability solutions to reach better integration of manufacturing systems. The present work considers the association of these applications with the qualitative characteristics of the ISO/IEC 25000 standard [5] providing one of the most exhaustive models for software quality characteristics. The major goal remains to enhance the competency of application designer using the ISO 16100 series approach while considering the quality of interoperability of working application software units.

The rest of the paper is organized as follows: the section 2 is dedicated to describe the ISO 16100 series. The corresponding approach advocates the use of standardized manufacturing software units to build integrated manufacturing applications. The section 3 presents the ISO/IEC 25000 series software quality standards. In section 4, we briefly show how to associate, the quality characteristics as designed in reference to ISO/IEC 25000, with the manufacturing software units developed according to the ISO 16100. The section 5, describes the prototype scheme of the proposed approach where the implementation is based on the linked data and ontologies. The conclusion is given in section 6.

1.2 The interoperability of manufacturing applications: approach as per ISO 16100 standard

The ISO 16100 standard series defined an approach to insure the interoperability of software applications that can be used in manufacturing domain. The central element of this approach consists of the implementation of a repository or distributed catalogue of software units called MSUs (Manufacturing Software

Units). The MSUs are used by the vendors and the application developers of the manufacturing systems to construct their applications. These MSUs are indexed by profiles so that they could be located according to a search criteria. These criteria describes the functionalities assured by the MSUs and other quality criteria such as reliability, performance, etc. Obviously, the MSUs are distributed on multiple vendors that can be geographically apart. It is therefore necessary to adopt a standardized vocabulary to elaborate the indexation profiles of these MSUs. In this regard, the ISO 16100 standard adopts the following approach:

- Define a data dictionary, called MDD (Model Data Definition), which can be considered as an ontology regrouping the most significant elements of a given domain. These elements represent the processes, activities, and the resources that can be used in a given domain.
- Elaborate the taxonomy of the regrouped capability elements in a hierarchical structure of capability classes. A capability class encapsulates activities or functions i.e. “elaborate a fabrication sequence” to be performed. It must also take into account certain qualitative criteria such as response time, optimal use of resource use, etc. to be assured. In its definition the capability class is linked to the MDD elements.

The profile of a software unit in this regard is an instantiation of one or more capabilities classes, which might be serialized in XML format, etc. A more detailed description of ISO 16100 can be found in [6].

Table 1.1. Structure of a “Software Unit Profile”.

Common Part
Template ID
Capability Class Name and Reference CCS
Software Unit ID
Vendor Name
Version Number & History
Computing Facilities Required
Processor
OperatingSystem&Options
Language
RuntimeMemory
DiskSpace
MultiUserSupport
RemoteAccess
AddOns&PlugIns
Measured Performance of the Unit
ElapsedTime
NumberOfTransactionsPerUnitTime
Reliability Data of the Unit
UsageHistory
.....
Support Policy
Price Data
Capability Class Reference Dictionary Name

Number Of Profile Attributes
.....
Specific Part for Capability Class
Reference MDM Name
MDD Description Format
MDD Description
Set Of MDD Objects
List Of MDD Objects
Time Ordered MDD Objects
Event Ordered MDD Objects
Interoperability MDD Objects
List Of Capability Class Attributes
List Of Capability Class Methods
List Of Capability Class Resources
.....

The Table 1.1 shows the structure of a profile in terms of capabilities. This structure represents a template that a profile may instantiate with the associated values of each element. Different elements involved in a profile describe also the MSU identification, used language, and its vendor, etc. The various profile indications on MSU structure and some quality criteria are related to capability classes that are implemented by the MSU along with its associated constraints. We can also remark that the quality characteristics in regard to the interoperability (intrinsic to the MSU) can only be invoked across a list of objects or entities of the MDD. We can assume that such objects may represent a software artifact like a data bus, a database connection, or a web service, etc. For instance, it is difficult to establish the associated quality of interoperability not only within a software unit, but also in the constituted application. In the following section, we present our approach that emphasizes the use of quality standard ISO 25000 to elaborate the guidelines for the evaluation of interoperability of MSUs developed according to the ISO 16100 series.

1.3 Adaptation of ISO 25000 Quality Model for Interoperability Validation of MSUs

The interoperability validation process requires to examine the correspondance between the specified qualitative requirements of interoperability between MSUs and its relevant implementation.

A major aim of our work is to provide guidelines and to illustrate the series of ISO 25000 quality standards addressing all aspects related to: Quality requirements specification, Quality model, Product Quality, Quality evaluation, and Quality measurement. For each quality aspect, a set of ISO standards has been elaborated where a set of standards constitutes a “Division” aiming at a specific aspect related to system or software quality [5].

The ISO 25000 series of standards have been supported with the framework, SQuaRE (Software product Quality Requirements and Evaluation). The objective of SQuaRE is to help the practice of main concepts and models provided by ISO 25000. SQuaRE is composed of five divisions intended to cover the wide range of aspects related to System and Software Quality Specification, as well as its modeling, assessment, and measurement (Fig. 1.1).

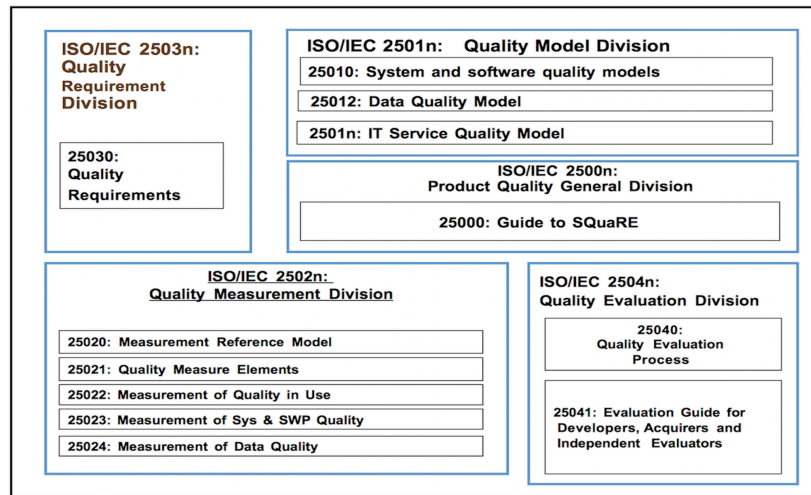


Fig. 1.1. Global Architecture of ISO 25000 SQuaRE

The ISO norms of SQuaRE make distinction between three major types of quality (as shown in Fig. 1.2) reflecting the perception of the quality by the developers and users. The major quality types are the quality in use, the external quality, and the internal quality. SQuaRE proposes to examine the conformity of the each quality type between expected specified quality of system or software and the effective quality of implemented system or software, in order to validate the developed interoperability of manufacturing application. At each hierarchical quality level a quality model is built in terms of quality characteristics, sub-characteristics, and properties.

For this goal, ISO/IEC 25010 proposes a generic quality model along with the corresponding characteristics and sub-characteristics for the concerned major quality levels i.e. internal quality, external quality, and quality in use. These quality models are to be instantiated and adapted in respect to the interoperability of the MSUs of the manufacturing application under consideration. The generic quality model, to be instantiated, proposes six characteristics and 27 sub-characteristics. These indicate the links of dependency between each individual characteristic and the corresponding sub-characteristics. The six major quality characteristics are adapted to interoperability, as shown in Table 1.2. SQuaRE proposes the breaking-down of the characteristics into sub-characteristics, which is adapted to the interoperability quality, as stated below:

The quality engineer of the current application defines each individual sub-characteristic. It specifies its contextual meaning and the way to assess the extent

to which each characteristic is met. Moreover, in a given application, sub-characteristics may have different priorities. The priority of each individual sub-characteristic depends on the hierarchical scale established between sub-characteristics according to the specific functional and qualitative priorities of the current manufacturing application.

Table 1.2 Characteristics and Sub-Characteristics of SQuaRE schema adopted for the quality of interoperability between MSUs.

Quality Characteristics of MSU interoperability	Description	Quality Sub-Characteristics of MSUs Interoperability
Functionality	The capability of the MSUs to provide functions and interoperability which meet stated and implied needs when the MSUs are used under specified conditions.	Suitability Accuracy Security
Reliability	The capability of the MSUs to provide functions and interoperability which meet stated and implied needs when the MSUs are used under specified conditions.	Testability Fault Tolerance Recoverability
Usability	The capability of the MSUs interoperability to be understood, used, and appreciated by the developer.	Understandability Learnability Testability
Efficiency	The capability of the MSUs interoperability to provide the required performance relative to the amount of resources used, under stated conditions.	Time Behavior Resource Utilization
Maintainability	The capability of the MSUs interoperability to be modified.	Adaptability Analyzability Changeability Testability
Portability	The capability of the MSUs interoperability to maintain its behavior when hosted on a different hosting environment.	Adaptability Installability Replaceability

To each individual sub-characteristic the quality engineer adapts a set of metrics addressed to quantitatively assess the extent to which a sub-characteristic is

met in reference to the expected values decided at the phase of qualitative requirements specifications.

The Interoperability quality evaluation is based on the three quality tree-like hierarchical structure instantiated by the quality engineer for the quality in use, external quality, and internal quality. Fig. 1.2 shows the procession of the interoperability validation as indicated by SQuaRE schema.

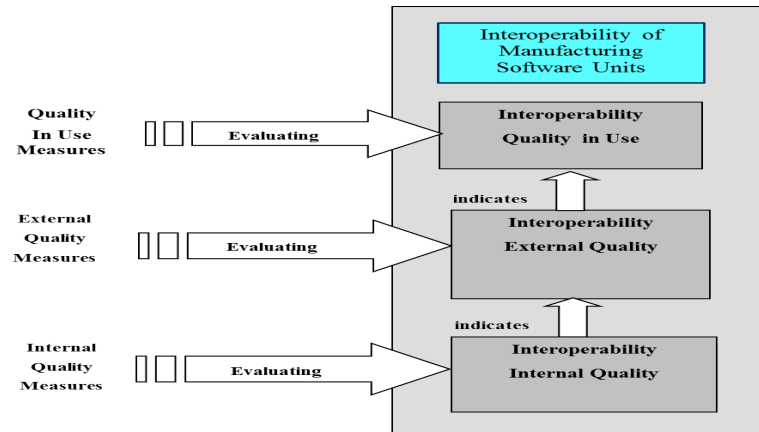


Fig 1.2. Evaluation of interoperability quality

For the selection of measures for the characteristics of interoperability, the Quality Measurement Division defined in ISO/IEC 25020 covers the mathematical definitions and guidelines for practical measurements of internal quality, external quality, and quality in use. In addition, it includes the definitions for the basic primitive measurements, and is able to formulate the all possible measures. By applying the guidelines of Quality Measurement Division, a set of basic and derived measures is to be defined to select metrics for evaluation of the internal quality, external quality, and the quality in use.

1.4 Classification of quality evaluation metrics

We propose to adapt a layered model [7] for the qualitative evaluation of interoperability of MSUs. As shown in Fig. 1.3, it is comprised of a core and two layers. The core is called Basic Metrics Set (BMS). BMS is composed of elements where each basic metric can not be calculated using other defined metrics. Each metric examines the extent to which a quality characteristic (or sub-characteristic) of interoperability is reached. It can be observed that the base metrics are insufficient for evaluating diverse interoperability qualities. Two layers of metrics can be used to serve the purpose: the set of Widely-Used Metrics (WUM) and the set of User-Defined Metrics (UDM). The WUM is comprised of the proposed metrics in the literature and widely used in the industry. An element of WUM is a function parameter of either one or more MSUs for interoperability evaluation, or one or more metrics belonging to the set BMS.

The Table 1.3 summarizes basic metrics frequently used to measure software systems interoperability. Similarly, examples of mostly used metrics and user defined metrics are listed in Table 1.3.

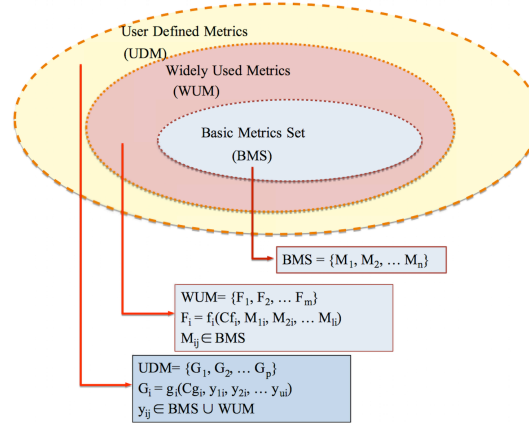


Fig 1.3. Classification of Interoperability quality metrics

Table 1.3. Some Basic Metrics (BMS)

Metric	Semantic	Formula
FUN_INDE	The number of input data elements of MSU_i (the i th MSU of the current application).	$FUN_INDE(MSU_i)$
FUN_OUTDE	The number of output data elements of MSU_i .	$FUN_OUTDE(MSU_i)$
MSU_PORT_IN	The number of ports of message reception of MSU_i .	$FUN_PORT_IN(MSU_i)$
MSU_PORT_OUT	The number of ports of message sending of the MSU_i .	$MSU_PORT_OUT(MSU_i)$
NB_MinR_MSU	Lowest number of messages to be received by an MSU_i .	$NB_MinR_MSU(MSU_i)$
MSU_XMLR	Number of XML files read by an MSU_i .	$MSU_XMLR(MSU_i)$
MSU_XMLW	Number of XML files written by an MSU_i .	$MSU_XMLW(MSU_i)$

1.5 Prototyping Issues

In order to validate the proposed approach of evaluating the qualitative characteristics of MSU interoperability, a prototype tool is under development to validate the proposed approach based on ISO 16100 and further extended in

reference to ISO 25000 model. The general architecture of the framework and its components are described in Fig. 1.4. The architecture represents the modeling of interoperability quality. The implementation strategy considers that the vendors of MSUs publish their profile on Internet in form of linked data. It allows the construction of a large collection of MSUs, which can be accessed *via* the semantic web tools, such as the SPARQL queries[8]. The adapted implementation approach considers mainly two constituents:

- A knowledge base which is composed of two major parts:
 - (1) A database of linked data which is a distributed collection of RDF (Resource Data Framework) [9] files, representing the template classes of capability, their instances, which in fact are the profiles of MSU and the inter-linking qualitative characteristics.
 - (2) A set of three ontologies, specified by the OWL (Web Ontology Language) [10] and which explicitly describe the semantics of each capability class, the Model Data Dictionary (MDD) and the modeling of the interoperability quality. Indeed, the profile templates are the RDF files, where the elements or tags have the semantics defined by the ontologies of capability classes. These are referred to the entities defined by the MDD. Similarly, the semantics of the qualitative characteristics are defined by ontologies of quality.

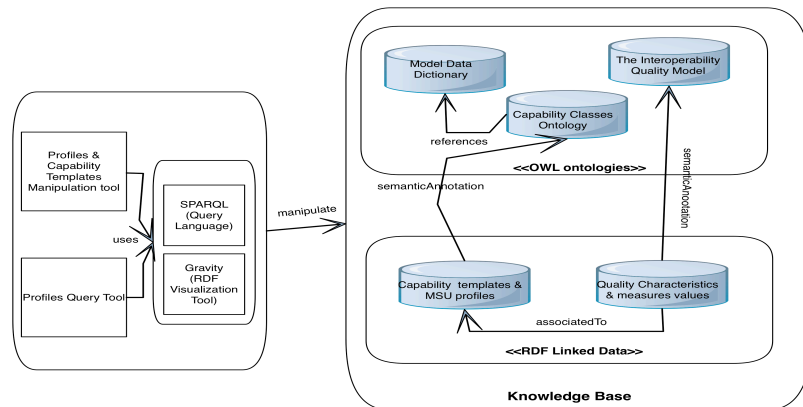


Fig. 1.4. Global Architecture of the Framework

- Profiles and templates which allow the manipulation (creation, modification, and deletion) of the profiles and templates, along with a tool to search the MSU profiles. These two tools use the SPARQL language to query the RDF files and OWL. We also use Gravity¹ that allows a graphical exploitation of the linked data, in graph-oriented forms.

The MSUs vendor may use a profile or template manipulation tool to find the best template instance, in order to create a profile for its MSU. In case if the

¹<http://semweb.salzburgresearch.at/apps/rdf-gravity/>

template exists already, the vendor may proceed to instantiate it with the provision of respective specific data.

1.6 Conclusion and Perspectives

For an adequate modeling of the qualitative evaluation of the interoperability among the Software Units of a manufacturing application, we adapted and used a quality model developed in respect of quality standard ISO 25000 series. The used model is exhaustive, therefore, it requires a good experience in software quality evaluation. The proposed use of metrics examine the extent of each interoperability characteristic or sub characteristic. Metrics are structured into layers for better traceability of adapted quality measurements. The work aims at permitting the developers to search the MSUs based not only on the functional capabilities characteristics but also on qualitative estimation of MSU interoperability characteristics. The proposed work is directly related to the activities of ISO-TC 184/SC 5/WG4 aiming at the development of ISO 16300 series with the perspective of integrating the work on Data Quality in ISO 8000, elaborated by ISO-TC 184/SC4.

References

- [1] Kosanke, K. ISO Standards for Interoperability: a Comparison. In D. Konstantas, J.-P. Bourrières, M. Léonard & N. Boudjlida (Eds.), *Interoperability of Enterprise Software and Applications* (pp 55-64). 2007, London: Springer-Verlag.
- [2] Kasunic, M., Anderson, W. *Assuring Systems Interoperability: Challenges and Opportunities*: Carnegie Mellon University, 2004.
- [3] Folmer Erwin and Verhoosel Jack. *State of the art on semantic IS standardization, interoperability & quality*. UT, CTIT, TNO en NOiV. ISBN 9789090260303, 2011.
- [4] Michiko Matsuda, Qian Wang. *Software Interoperability Tools: Standardized Capability-Profiling Methodology ISO16100*. EAI2N 2010: 140-151. 2010.
- [5] *Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data Quality model*. 2008.
- [6] ISO/IEC 1600-N(1..5) *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability* — 2009.
- [7] A Ahmad, H. Basson, L. Deruelle, M. Bouneffa. "Towards an integrated quality-oriented modeling approach for software evolution control", in *Software Technology and Engineering (ICSTE)*, 2010 2nd International Conference on , vol.2, , pp.V2-320-V2-324, San Juan, Puerto Rico, USA 3-5 Oct. 2010.
- [8] The SPARQL 1.1 Query Language, W3C Recommendation, <http://www.w3.org/TR/sparql11-query>. 2011.
- [9] RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, <http://www.w3.org/TR/rdf11-concepts/>. 2014
- [10] OWL Web Ontology Language Overview, W3C Recommendation, <http://www.w3.org/TR/owl-features>. 2013.