FOR IMPROVEMENT OF MANUFACTURING SOFTWARE UNITS REUSE

[1]H. BASSON, [2]M. BOUNEFFA, [3]M. MATSUDA, [4]A. AHMAD, [5]D. CHUNG, [6]E. ARAI

[1,2,4]LISIC University of Littoral, France
[3]Kanagawa Institute of Technology, Japan
[5]Rockwell Automation Cleveland, USA
[6]Osaka University, Japan
E-mail: [1]henri.basson@univ-littoral.fr

Abstract - With the growing trend in the development of distributed digitalized manufacturing systems, the interoperability of Manufacturing Software Units (MSU) is becoming more critical for manufacturing system productivity, reliability, and security. In order to assure better development and evolution of working manufacturing units, specification of required interoperability mechanisms should be made since the earliest phases of manufacturing system development. The specified interoperability mechanisms are then to be designed, implemented, and tested before the phase of operating production of integrated manufacturing system. In a given manufacturing domain, for the automation system integration, ISO 16100 series of standards propose a profiling model to classify, store, and reuse MSU in various cases of manufacturing system development and evolution. This paper starts from ISO 16100 MSU profiling, to associate two extended descriptions aiming at better use of ISO 16100 series.

Index Terms - ISO 16100 16300 25000 series, Manufacturing Software Units, Software Units Interoperability, Software Unit Profiling.

I. INTRODUCTION

With the accelerating trend towards smart manufacturing systems, manufacturing activities are increasingly digitalized using a growing set of Manufacturing Software Units (MSU). The MSUs are destined to choose the most adequate behavior of parametrized manufacturing devices corresponding to the adopted execution plan of manufacturing processes. Their objectives of controlling the parallel execution flows of several manufacturing processes and underlying hardware devices, the adopted MSU interoperability mechanisms accomplish an important task to assure a satisfying behavior of the interacting manufacturing units. IS0 16100 series propose a methodology for manufacturing system integration where “manufacturing software requirements shall be expressed in terms of software unit capability profiles. The profiling of a software unit involves the generation of a concise statement of manufacturing capabilities enabled by the software unit in terms of the functions performed, the interfaces provided, and the protocols supported as required by the target manufacturing capability” [ISO 16100- 2].

The profiling of MSU according to ISO 16100 approach constitutes a very important work for MSU reuse. It helps, within a giving manufacturing domain, to choose a set of software units from already built repository where MSU are classified according to ISO 16100 profile to be reused by the manufacturing systems developers.
Nevertheless, some profile parameters, adopted and represented in ISO 16100 series, are generally mentioned such as the corresponding adapted instantiation work is totally left to the manufacturing developers. In this context, an extended description of interoperability parameters with associated indications about the instantiation of interoperability parameters can improve the use of ISO 16100 MSU profiling.
In this paper we present the proposed ISO 16100 series approach for Manufacturing Software Units profiling aiming at creating a very large database dedicated to store a wide range of MSUs classified according to their profiles in order to reuse them to simulate or develop an evolving manufacturing system.

II. MSU PROFILING ACCORDING TO ISO 16100 APPROACH

The ISO 16100 standard series defined an approach to insure the profiling and classification of MSUs such as to be reused for development or evolution of software applications as a crucial part of manufacturing application operating in a manufacturing domain. The major goal of this approach is to design and to implement of a repository or distributed catalogue of software units called MSUs (Manufacturing Software Units) classified according to a set of MSUs description parameters. The MSUs can be used by manufacturing domain actors, the vendors, the manufacturing systems developers in the context of manufacturing software development or evolution. These MSUs are indexed and classified by profiles so that they could be identified and located according to a set of search criteria starting by the major functionalities which can be accomplished by the MSUs [2][3]. It also includes technical information such as performance and required capabilities of hosting operating environment. The MSUs concerns mainly various manufacturing development actors (architects, designers, developers, buyers, etc.) that it becomes necessary to adopt a common standardized vocabulary to elaborate some sort of indexation profiles corresponding to a potential high number of MSUs, the ISO 16100 standard adopts the following approach phases:

- Define a data dictionary, called MDD (Model Data Definition), which can be considered as a sort of 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 manufacturing domain.
- Elaborate the taxonomy of the regrouped manufacturing capability elements in a hierarchical structure of capability classes. A capability class may encapsulate the actions; e.g. an activity, "realize a specific fabrication action" and should also fulfill some constraints or qualitative criteria (response time for instance) to be respected. Such a capability class, in its definition is linked to the MDD elements [4].

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 [5].

Common Part
Template ID
Capability Class Name and Reference CCS
Software Unit ID
Vendor Name
Version Number & History
Computing Facilities Required
Processor
Operating System & Options
Language
Runtime Memory
Disk Space
Multi User Support
Remote Access
Add Ons & PlugIns
Measured Performance of the Unit
Elapsed Time
Number Of Transactions Per Unit Time
Reliability Data of the Unit
Usage History
…………….
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
…………………………………….

Table 1. Structure of a "Software Unit Profile".

The Table 1. indicates the structure of a profile in terms of capabilities which constitute a template that an instantiation creates a profile associated to chosen value(s) to each element. Different elements belonging to a profile describe the identification of a MSU and the identity of its vendor(s). The profile various indications specify, MSU structure, some quality criteria related to classes in terms of capabilities that are implemented by the MSU, and also the associated constraints, etc. The quality characteristics in regard to the interoperability can only be invoked across a list of objects or entities of the MDD.

III. EXTENSION I: MSU INTEROPERABILITY SPECIFICATION

As first extension, to the proposed ISO 16100 Profile, we propose to specify precisely, to make it explicit in MSU profile, the various possibilities of interoperability in reference to an adopted interoperability taxonomy. The indications about Interoperability MDD Objects such a data bus, a database connection, or a web service, etc. are essential for various use cases use of MSUs.

In order to identify the type of required interoperability mechanisms which are necessary to integrate an MSU, an interoperability typology development is proposed. A small part of possible built interoperability is given here as example, although not exhaustive, but it would illustrate the importance for adopting some interoperability typology which can be referenced in MSUs profile. The development and use of interoperability typology structured by levels or layers, according to concerned components granularity and associated possible interactivities, was already proposed in other contexts of applications development. The MSUs reuse context requires also a detailed interoperability mechanism specified

according to an adopted typology to be referenced in various use cases which may occur at the development and evolution phases of manufacturing systems.
The specification of interoperability required mechanisms starts by the list of couples of interoperating units inside the whole application architecture [8]. The type of interoperability relationship is to be specified in reference to an interoperability typology. A small part of non-detailed typology is given below using a gradual tree-like enumeration of some interoperability, which may occur between linked couple of software units (or components). These can be illustrated as follows:

<MSU interoperability relationship> =>
<Inter-MSUs control relationship> |
<Inter-MSUs service exchange> |
<Inter-MSUs synchronization> |
<Inter-MSUs Message based communication> |
<Inter-MSUs Event based communication> |
<MSU Data interoperability mechanisms>

<MSU Data interoperability mechanisms> =>
<Interoperability via shared variables>
|
<Interoperability via shared simple files>
|
<Interoperability via shared XML files>
|
<Interoperability via shared specific tool-dependent files> |
<Interoperability via shared JSON files>
|
<Interoperability via shared Tabular files>
|
<Interoperability via shared Database files>

<MSU Message exchange based interoperability>=>
<One direction connection initiation> |
<Request-Response interoperability> |
<Bi-directional asynchronous message flow> |
<Bi-directional synchronous message flow> |
<One direction notification> etc.

To each interoperability mechanism, mentioned in the concerned MSU profile, a set of quality characteristics and associated metrics are to be specified according to the prioritized quality characteristics of the current developed manufacturing application [8].

IV. EXTENSION II: MSU DETAILED QUALITY SPECIFICATION

The ISO 25000 series of norms has 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).

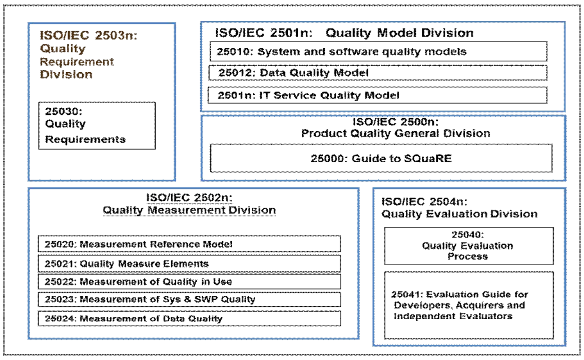

Fig. 1. Global Architecture of ISO 25000 SQuaRE.

The ISO norms of SQuaRE make distinction between three major types of quality (Fig. 1) 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 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.

The interoperability Qualitative specification process starts at the application design phase to concern each interoperability mechanism considered as necessary to realize the required operational exchange and sharing relationships between manufacturing software units. Depending on the qualitative priorities specified for the whole application level in terms of characteristics such as reliability, performance, efficiency, security, etc. these required characteristics influences the choice of interoperability mechanisms to be designed and later implemented in a way to meet the specified qualitative priorities of the current applications. In the case of hard real-time manufacturing application, for instance, the corresponding qualitative constraints in terms of efficiency and corresponding sub-characteristics such as response time i.e., are highly targeted. In such a case, the choice among different possibilities of interoperability mechanisms, which can be functionally equivalent, is oriented toward the adoption of interoperability mechanism which is considered as comparatively the best in term of response time. It is not the case when the performance in terms of ratio between the adopted interoperability mechanisms and associated interoperability services assured for a satisfying functioning of inter-activities between operating MSUs.

Quality Characteristics of MSU	Description	Quality Sub-Characteristics of MSUs
Functionality	The capability of the MSUs to provide functions 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 which meet stated and implied needs when the MSUs are used under specified conditions.	Testability Fault Tolerance Recoverability
Usability	The facility with which of the MSUs can be understood, used, and appreciated by the developer.	Understandability Learnability Testability
Efficiency	The capability of the MSUs to assure the required performance relative to the amount of resources used, under stated conditions.	Time Behavior Resource Utilization
Maintainability	The facility with which the MSUs can 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

Table 2. Characteristics and Sub-Characteristics of SQuaRE schema adopted for the MSUs quality

CONCLUDING REMARKS

In the context of rapidly developed and evolving software manufacturing applications, the MSUs reuse approach as proposed by an important work of ISO series standards is pertinent for rapid development using an optimal set of resources. Nevertheless, the MSU profile specification proposed to better matching between required MSUs and repository provided MSUs can be better performed through a detailed specification of interoperability mechanisms associated to MSU to be reused and integrated with other operating existing software units of the developed or evolving manufacturing application. The invoked interoperability specification requires a detailed taxonomy or ontology of interoperability mechanisms to be used as shared reference between required MSU and provided MSU. The associated interoperability qualitative specification is another important aspect invoked in the present paper regarding the particular qualitative priorities differences which may occur between developed or evolving manufacturing applications.

REFERENCES

[1] ISO 16300-3:2017, Automation systems and integration — Interoperability of capability units for manufacturing application solutions — Part 3: Verification and validation of interoperability among capability units, available at https://www.iso.org/obp/ui/#iso:std:iso: 16300:-3:ed-1:v1:en
[2] ISO/IEC 1600-[N=1..5] Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — 2009.
[3] Michiko Matsuda, Qian Wang. Software Interoperability Tools: Standardized Capability-Profiling Methodology ISO16100. EAI2N 2010: 140-151. 2010.
[4] ISO/TR 18161:2013, Automation systems and integration -- Applications integration approach using information exchange requirements modelling and software capability profiling. Publication date: 2013-07, available at https://www.iso.org/standard/60899.html
[5] H. Basson, M. Matsuda, M. Bouneffa, A. Ahmad, D Chung and E. Arai. "Qualitative Evaluation of Manufacturing Software Units Interoperability Using ISO 25000 Quality Model." Enterprise Interoperability VII. Springer International Publishing, 2016. 199-209.
[6] Adeel Ahmad, Henri Basson, and Mourad Bouneffa, Analyzing and modeling the structural and qualitative interdependencies of software evolution, 7th International Workshop on Computer Science and Engineering, workshop of, The 5th International Conference on Information Technology and Science, 25-27 June 2017, Beijing, China.
[7] H. Basson, M. Matsuda, and D Chung. "A contribution to Software interoperability of software application". I-ESA, INTEROP-VLAB, European Virtual Laboratory for Enterprise Interoperability. Albi, France, 24-28, March 2014.
[8] H. Basson, M. Bouneffa, M. Matsuda, A. Ahmad, D Chung, and E. Arai. I-ESA, 9th Conference on Interoperability for Enterprise Systems and Applications. Berlin, Germany, 19-23, March 2018.

★★★