

Modeling of manufacturing software catalogue for interoperability in manufacturing systems

Sangkeun YOO*, Hyunjeong LEE*, Yong-woon KIM*, Hoon CHOI**

*Protocol Engineering Center (PEC)/ETRI (Electronics and Telecommunications Research Institute),
218 Gajeong-ro, Yuseong-gu, Daejeon, KOREA

** Department of Computer Engineering/CNU (Chungnam National University),
99 Daehak-ro, Yuseong-gu, Daejeon, Korea

lobbi@etri.re.kr, hjlee294@etri.re.kr, gkim@etri.re.kr, hchoi@cnu.ac.kr

Abstract— This paper proposes a way to describe the capability of software of manufacturing systems that can be mapped to the functional requirements of target manufacturing application for interoperability. For this objective, this paper specifies how to develop and manage software catalogue in terms of capability properties, and defines mapping rules from capability profiles to software catalogue.

Keywords— manufacturing software, software cataloguing, interoperability, smart manufacturing

I. INTRODUCTION

In relation with software in manufacturing systems, user and suppliers of manufacturing software may require interoperability of manufacturing software for integrating, substituting, validating and verifying software components. Therefore, development of manufacturing software involves defining software capability in terms of its functions provided in the way of representing the software capability, how and what it can do.

International Standard, ISO 16100 series [1] has been developed with the aim of providing a standardized method to describe capabilities of manufacturing software as capability profile. Software capability profiling is the basis for providing interoperability of manufacturing software.

Based on ISO 16100 series, this paper proposes how to describe the capability of manufacturing software that can be mapped to the requirements of target manufacturing application by user and suppliers of manufacturing software for interoperability. Also, a class diagram of software capability and procedures of cataloguing will be given based on class diagram of capability defined in ISO 16100 series.

II. INTEROPERABILITY OF MANUFACTURING SOFTWARE

Similar to typical software systems, there are requirements of user and providers of manufacturing software as follows.

Users may require integrating a manufacturing application system by combining capabilities of a set of software components provided by various sources, substituting another software component to provide an equivalent capability required by the manufacturing system, and integrating the

capability of manufacturing software from one resource system platform to another platform.

Also, manufacturing software providers may require representing the set of capabilities provided by manufacturing software, verifying software component capability as a part of a required manufacturing software capability, and cataloguing a manufacturing software in terms of its capability for interoperability in manufacturing systems to support wide distribution.

III. OVERVIEW OF CAPABILITY CATALOGUING

This clause briefs capability cataloguing which is described in International Standards [1][2][3][4][5][6][7].

A. Overview of ISO 22745

ISO 22745[5] deals with an open technical dictionary (OTD) which is a dictionary of concepts (metadata) for describing items including dictionary representation, identification of concepts, dictionary query interface, maintenance, identification guide, query, master data representation, etc.

OTD allows information to be unambiguously defined and exchanged between multi-national partners without loss of meaning, supports synchronization of databases with minimum data mapping requirements, multiple classifications and multiple languages. Based on these functions, each items described in OTD can be exchanged and displayed between different entities (organizations) without any loss of meaning.

Also, ISO 22745 [5] defines identification guide representation which are requirements stating which data is needed for an item. With information stored in OTD for an item, identification guide expedites automatic data analysis based on understanding of what data is required without any human intervention. Requirements for the manufacturing applications can be described using identification guide.

OTD for manufacturing software capability description serves as a basis for interoperability, because using the common dictionary ensures that software capability units and the software catalogues have the definite semantics. Template for manufacturing software capability description can be constructed using identification guide. This template conforms to ISO 22745 [5] where the item is the class and the property-

value pairs represents the characteristic data. This template constrains the use of concepts in the dictionary in the manufacturing software catalogues.

The manufacturing software catalogue contains the profiles of manufacturing software in terms of its capabilities. A manufacturing application may refer to multiple manufacturing software catalogues. This software catalogue is constructed by querying the existing manufacturing software and find the profiles which meets the requirements specified by identification guide.

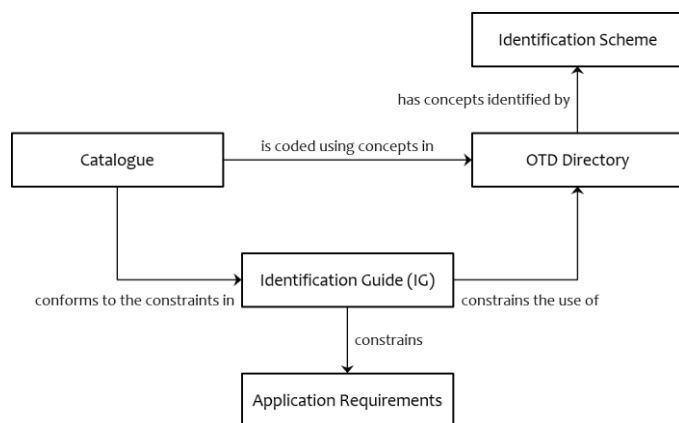


Figure 1. Relationship among catalogue, dictionary and application requirements

B. Overview of ISO 16100

A manufacturing application is modelled as a set of manufacturing processes that are enabled, controlled and automated by a set of manufacturing resources (Figure 2).

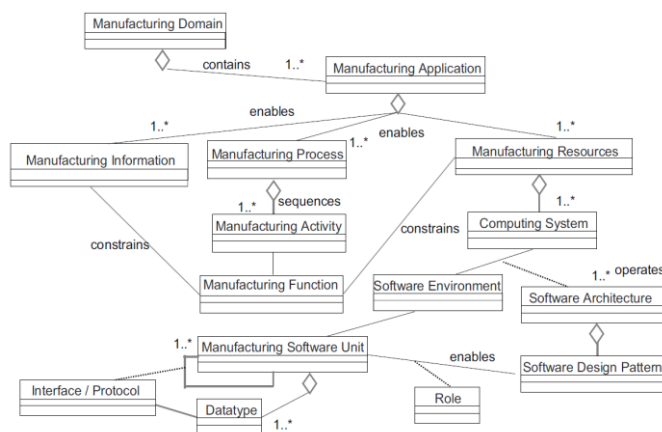


Figure 2. Class diagram of a software unit and associations within manufacturing application (ISO 16100-1 [2])

In order to fulfil the requirements of a manufacturing application, a set of manufacturing software units (MSU) shall be sequenced and scheduled to accomplish the combined set(s) of required manufacturing functions for all the manufacturing activities associated with the set of manufacturing processes that constitute a manufacturing application.

Each manufacturing function can be accomplished by a set of manufacturing software units.

Manufacturing software unit shall be modelled as a type of manufacturing resource that can satisfy a set of interoperability criteria. These criteria shall be determined by the required sequence and timing of the specific set of manufacturing functions that have to be accomplished by manufacturing software unit and the information exchanges that it has to support.

As shown in Figure 3, manufacturing software unit provides several interfaces to its capabilities, including its capability profile to be accessed by Capability Profile Interface (CPI). Information about the other interfaces is included in the capability profile and therefore the information is accessible via the CPI.

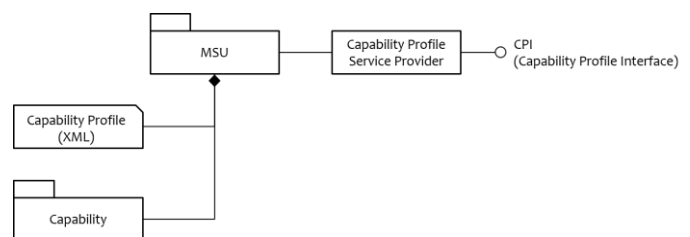


Figure 3. Manufacturing software unit with its capability and corresponding capability interfaces

Figure 4 shows the relation between capability class, capability template and capability profile.

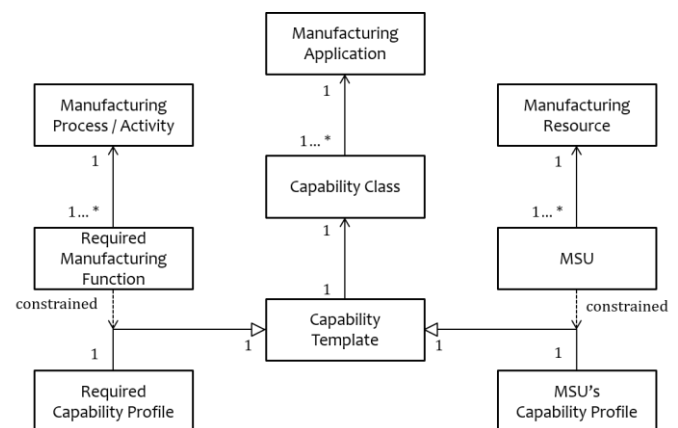


Figure 4. Class diagram of capability

The structure of manufacturing software unit capability profile template shall be derived from the manufacturing capability class structure [1]. Capability profiles pertaining to manufacturing software unit or a capability requirement for a manufacturing activity shall be matched according to ISO 16100-2 [3]. A capability profile for manufacturing software unit is an instance of capability template filled with appropriate information.

IV. MODELING OF MANUFACTURING SOFTWARE CATALOGUE

As described earlier, manufacturing software catalogue is description of software capability in terms of its functions

provided in the way of representing the software capability, how and what it can do.

A. Software unit cataloguing using software capability unit

This paper defines two terms, software unit catalogue and software capability unit for modelling of manufacturing software catalogue as follows:

1) **Software unit catalogue:** a set of manufacturing software unit (MSU) capability profiles constructed by querying the existing MSU capability profiles which meet the requirements specified by identification guide

2) **Software capability unit:** each entry of software unit catalogue.

Based on these definitions, software unit catalogue and software capability unit can be shown as Figure 5 using class diagram of capability shown in Figure 4.

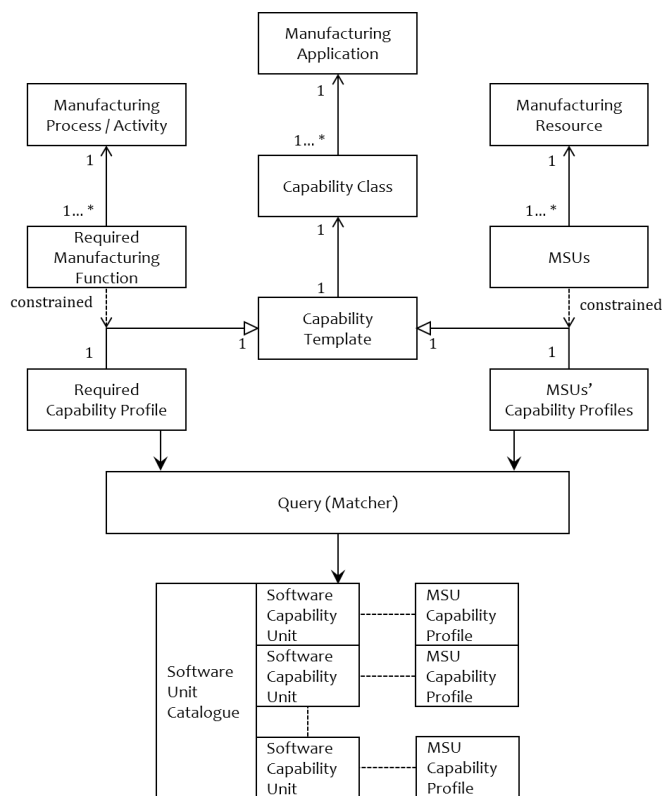


Figure 5. Class diagram of software unit catalogue and software capability unit

According to Figure 5, the procedures of software unit cataloguing are as follows:

- 1) Required manufacturing functions should be represented as required capability profile using capability template. The required capability profile for manufacturing software unit is an instance of capability template filled with appropriate information.
- 2) Query (Matcher) compares required capability profile with manufacturing software unit's capability profiles which are registered in capability profile service provider (see Figure 3).

- 3) When matches are found, software unit catalogue is created with a set of manufacturing software unit capability profiles satisfying the required capability profiles.
- 4) Software capability units are manufacturing software units identified by each entry of software unit catalogue.

B. Capability template

Capability template is a schema to describe software capability unit. According to class diagram in Figure 5, capability template shall conform to manufacturing capability class and shall be defined using XML conventions for creating schema based on ISO 16100-3 [4] which defines a template consisting of a common part for all capability class and application specific part for specific capability class.

Following is an example of a schema of capability template given in ISO 16100-3 [4].

Common Part

```

Template ID
Capability Class Name
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
NumberOfShipments
IntendedSafetyIntegrityLevel
CertificationBody
Support Policy
Price Data
Reference Dictionary Name
NumberOfMethods
.
.
.

```

Part Specific to Capability Class

```

.
.
.

```

ISO 16100-2 [3] defines two software capability units can be regarded as compatible and have a match when the attributes and the operations of one template form a subset of the attributes and the operations of another.

V. SOFTWARE UNIT CATALOGUING INTERFACES

Figure 6 shows procedures and interface for creating software unit catalogue based on clause III.

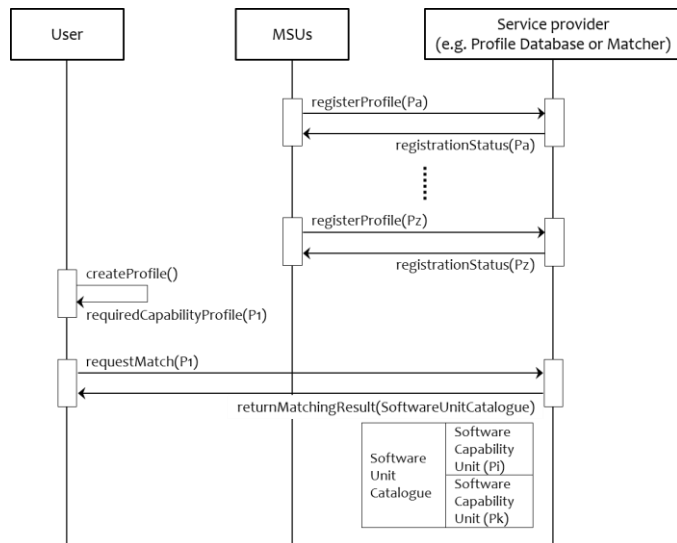


Figure 6. Software unit cataloguing interface

Software unit catalogue consists of the following steps:

- 1) Each manufacturing software unit (MSU) registers its capability profile to service provider which acts profile database or matcher. Capability profile should be represented by capability template.
- 2) User (Profile user) identifies required manufacturing functions and creates capability profile (required capability profile) by required manufacturing functions. Capability profile should be represented by capability template.
- 3) Upon creation of required capability profile, user requests service provider to match required capability profile and registered capability profile by manufacturing software units (MSUs).
- 4) Service provider retrieves registered capability profiles which match to the required capability profile.
- 5) Service provider creates software unit catalogue which is as set of capability profiles constructed by the existing manufacturing software unit (MSU) capability profiles which meet the requirements specified by user.
- 6) Each entry of software unit catalogue is software capability unit as a form of manufacturing software unit (MSU) capability profile.

VI. CONCLUSIONS

This paper proposes how to describe the capability of manufacturing software that can be mapped to the requirements of target manufacturing application by user and suppliers of manufacturing software for interoperability. In this paper, software capability profiling is the basis for providing software interoperability services.

This software unit cataloguing ensures:

- integrating an automation application system by combining capabilities of a set of software components provided by various sources,
- substituting another software component in a software unit to provide an equivalent capability required by the automation application system,
- validating and verifying the capability of a software unit to meet the automation application system requirements.
- cataloguing a software unit in terms of its capability for interoperability in an automation application system to support wide distribution.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. R0127-16-1046, Development of Smart Factory interworking and interoperability standards).

REFERENCES

- [1] ISO 16100 Series, Industrial automation systems and integration – Manufacturing software capability profiling for interoperability
- [2] ISO 16100-1, Industrial automation systems and integration – Manufacturing software capability profiling for interoperability – Part 1: Framework
- [3] ISO 16100-2, Industrial automation systems and integration – Manufacturing software capability profiling for interoperability – Part 2: Profiling methodology
- [4] ISO 16100-3, Industrial automation systems and integration – Manufacturing software capability profiling for interoperability – Part 3: Interface services, protocols and capability templates
- [5] ISO 22745 Series, Industrial automation systems and integration – Open technical dictionaries and their application to master data – Part 1, Part 2, Part 10, Part 11, Part 13, Part 14, Part 20, Part 30, Part 35, Part 40 and Part 45



Sangkeun Yoo received the M.S degrees from Chungnam National University, Korea in 1999. He developed micro payment systems in start-up from 1999 to 2000. Since 2001, he has worked for ETRI in areas of information security, RFID, sensor networks and IoT, recently smart factory. He has been a Convenor of ISO/IEC JTC 1/WG 10 (Working Group on Internet of Things) since 2014. His current research interests include IoT, smart factory and digital manufacturing.

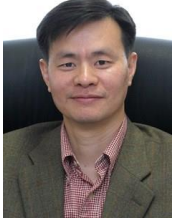


Hyunjeong Lee received her BS, MS, and PhD degrees in computer science from Chungbuk National University, Cheongju, Rep. of Korea, in 1997, 1999, and 2015, respectively. Since 1999, she has been working at ETRI. She has been engaged in the research and development of communication protocols, home network services, context-aware frameworks, and content transformation technology. She is currently working as a senior engineer of the Convergence Standards Research Section, ETRI. Her current research interests include smart factory, smart city, and energy efficiency.



Yong-Woon Kim studied electronics engineering at Dong-A Univ. and majored in computer networks and communication for the MS degree at POSTECH. He is currently a principal research engineer at the Protocol

Engineering Center, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include IoT applications and services in the fields of smart city, smart factory, smart grid and smart water grid, and also IT sustainability works in terms of standardization. His research results have been contributed to ITU-T and ISO/IEC JTC 1 for development of relevant standards where he is a vice chairman of ITU-T SG 5 WP 3 (ICT and Climate Change) and the convenor of ISO/IEC JTC 1/SC 39(Sustainability for and by IT)/WG 2(Green ICT).



Hoon Choi is a faculty member of the Department of Computer Science and Engineering, the Chungnam National University (CNU).

Before joining the CNU in 1996, was he a senior member of technical staff at the Electronics and Telecommunications Research Institute (ETRI) in Korea since 1983. His research area is the mobile, distributed computing. He teaches various courses on computer operating systems and distributed systems of the undergraduate, graduate programs in CNU, and he received the Outstanding Teaching Award in 2004. He received a BSEE from the Department of Computer Engineering of Seoul National University, Korea in 1983, and an MS and a Ph.D. in computer science from Duke University in 1990 and 1993, respectively. He was awarded an IBM Graduate Fellowship in 1991 and 1992.