9

AN INTEROPERABILITY FRAMEWORK AND CAPABILITY PROFILING FOR MANUFACTURING SOFTWARE

M. Matsuda[1], E. Arai[2], N. Nakano[3], H. Wakai[4],
H. Takeda[5], M. Takata[6] and H. Sasaki [7 ¶]
[1]*Kanagawa Institute of Technology, Japan*
[2]*Osaka University, Japan*
[3]*Mitsubishi Electric Mechatronics Software Co., Japan*
[4]*Komatsu Ltd., Japan*
[5]*National Institute of Informatics, Japan*
[6]*The University of Electro-Communications, Japan*
[7]*Yokogawa Electric Co., Japan*
matsuda@ic.kanagawa-it.ac.jp

Abstract: ISO/TC184/SC5/WG4 is working on ISO16100: Manufacturing software capability profiling for interoperability. This paper reports on a manufacturing software interoperability framework and a capability profiling methodology which were proposed and developed through this international standardization activity. Within the context of manufacturing application, a manufacturing software unit is considered to be capable of performing a specific set of function defined by a manufacturing software system architecture. A manufacturing software interoperability framework consists of a set of elements and rules for describing the capability of software units to support the requirements of a manufacturing application. The capability profiling methodology makes use of the domain-specific attributes and methods associated with each specific software unit to describe capability profiles in terms of unit name, manufacturing functions, and other needed class properties. In this methodology, manufacturing software requirements are expressed in terns of software unit capability profiles.

Key words: Manufacturing software, Interoperability, Capability profiling, International standardization

¶ All authors are members of Japanese mirror committee of ISO/TC 184/SC 5/WG 4.

1. INTRODUCTION

Few years ago, an ISO/TC184/SC5 Study Group (Title: Manufacturing software capability profiling, Convenor: Dr. U. Graefe) identified the problem that a unified characterisation might be expected to address in terms of the two view points of user requirements and contribution to interoperability. User requirements on the manufacturing application were identified assembling a new functionality, selecting appropriate software, substituting one software component with another, migrating to another platform, managing software inventory, certifying software to a capability profile, distributing software to the mass market, and registering new software. Interoperability issues in manufacturing software were identified as the ability to describe software in unambiguous terms to enable a common understanding, the characterising of the business benefits delivered by software components, the ability to find enabling candidate software components automatically using search engines, expressing the dependencies of one software component on other application or operating system components, and the management of the implications of change. As a conclusion, the study group suggested launching a standardization work item on these view points [1].

At present, ISO/TC184/SC5/WG4 (Title: Manufacturing software and its environment, Convenor: Dr. M. Matsuda) is developing a 16100 series International Standard which is titled "Manufacturing software capability profiling for interoperability". This project addresses concerns of users and suppliers of manufacturing software with regard to user requirements and interoperability of software in the area of industrial automation. This paper discusses a manufacturing software interoperability framework and a capability profiling methodology which were proposed and developed in this international standardization activity.

2. MANUFACTURING SOFTWARE INTEROPERABILITY FRAMEWORK

2.1 Manufacturing software unit interoperability

The interoperability framework for manufacturing software is based upon a more general interoperability framework for manufacturing applications. An integrated manufacturing application is modeled as a combination of a set of manufacturing resources and a set of information units whose data structure, semantics, and behaviour can be shared and

exchanged among the manufacturing resources. The set of integrated manufacturing resources forms a manufacturing system architecture that fulfils a set of manufacturing application requirements. These manufacturing resources, including the manufacturing software units, provide the functions associated with the manufacturing processes, as shown in Figure 1. The combined capabilities of the various software units, in an appropriate operating environment, provides the required functionality to control and monitor the manufacturing processes according to the production plan and the allocated resources [2].

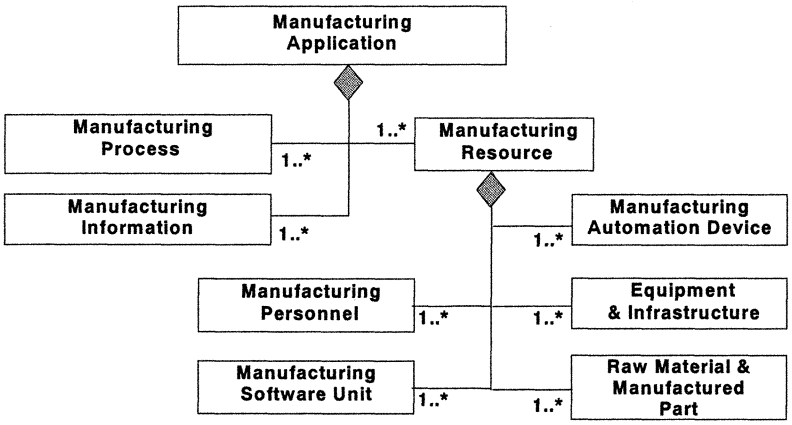

Figure 1. Class diagram of a manufacturing application [2].

A manufacturing software unit is a class of software resource, consisting of one or more manufacturing software components, performing a definite function or role within a manufacturing activity while supporting a common information exchange mechanism with other units. The software interoperability of a set of manufacturing activities is described in terms of the interoperability of the set of software units associated with each manufacturing activity [2].

2.2 Conceptual framework for manufacturing software unit interoperability

A manufacturing software interoperability framework consists of a set of elements and rules for describing the capability of software units to support the requirements of a manufacturing application. The capability to support the requirements cover the ability of the software unit to execute and to

exchange data with other software units operating in the same manufacturing system or in different manufacturing systems used in the manufacturing application. A manufacturing software interoperability framework is based on syntax and semantics shared between manufacturing software units, functional relationships between the manufacturing software units, services, interfaces, and protocols offered by the manufacturing software units, and ability to provide manufacturing software unit capability profiling [2].

Figure 2 shows the conceptual interoperability framework. In Figure 2, the interoperability of software units can be described in terms of their capabilities that are associated with the aspects of functionality, interface and structure. 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. The software units capability profile definition is registered in an appropriate capability profile database after passing the conformance test. The profile database has a set of taxonomies for use in describing the capability profiles. When developing new manufacturing software or reusing a software unit, the profile database is referred to and searched [3].

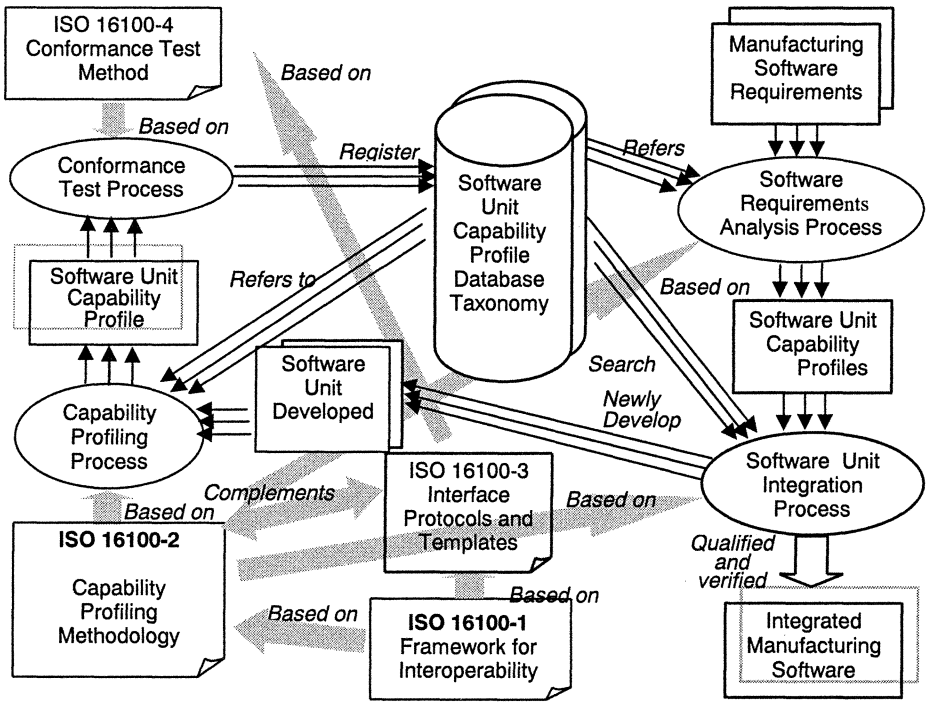

Figure 2. Conceptual framework for manufacturing software unit interoperability.

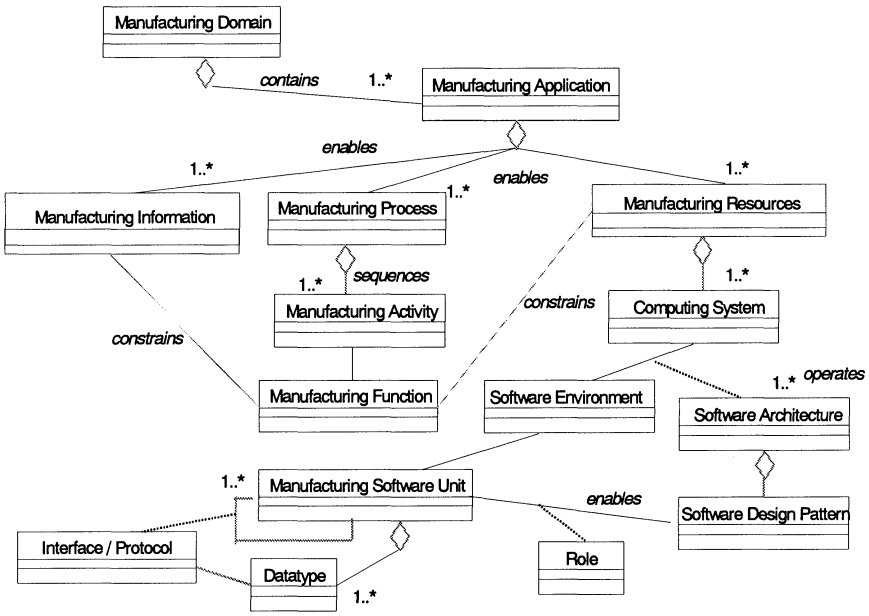

Figure 3. Class diagram of a software unit and its surroundings and associations within a manufacturing application [2].

3. ELEMENTS IN A MANUFACTURING SOFTWARE INTEROPERABILITY FRAMEWORK

3.1 Capability classes

The capability of a manufacturing software unit is expressed in terms of capability classes. These classes is derived from the manufacturing activities noted in Figure 3. These classes also denote the manufacturing function, resource, and information handled by the manufacturing software unit according to the requirements of the manufacturing process [3].

A manufacturing process has a structure that is both nested and hierarchical. At each level, the manufacturing software requirements can be modelled as a set of capability classes organized in a similar structure as shown in Figure 4.

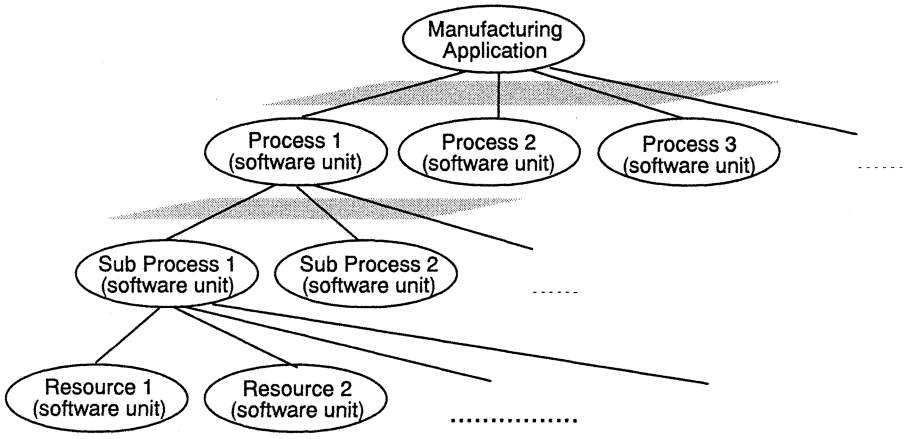

Figure 4. Hierarchical structure in manufacturing application.

3.2 Capability templates

A software unit that enables or supports an activity with an associated capability class is concisely described in a capability template. The structure of a software capability template follows the structure of a manufacturing capability class as shown in Figure 4. In a hierarchical structure, a capability template is associated with each capability defined at each level of the structure. In a nested structure, a similar association exists between each capability class and a template at each level of the structure [3].

3.3 Capability profiles and software unit profile database

Capability profiles are capability templates with, at a minimum, a profiled software unit name instantiated. Other items are fulfilled according to the specification level [3].

A set of taxonomies, a set of capability classes, a set of capability templates, and a set of capability profiles are stored in software unit profile databases, and are distinguished by their differing dictionary names. The databases may be structured as a free combination of the above four elements to provide the necessary services. A taxonomy, a capability class, a capability template, and a capability profile are unique when entered in an adequate corresponding dictionary.

Matching capability profiles are used in the analysis of software unit in the capability profiling process, the decomposition of requirements in the manufacturing software requirements analysis process, and the database search for each profile in the software unit selection and verification process. Matching is attempted between software unit descriptions, manufacturing

software requirements, or required software unit capability profiles in these processes and that of capability profiles in the database [3].

4. DETAILED PROCESSES IN THE MANUFACTURING SOFTWARE INTEROPERABILITY FRAMEWORK

4.1 Capability profiling process

The capability profiling process shown in Figure 2 is detailed in Figure 5. A software unit to be profiled is analyzed in terms of the supported paths within the capability class structure. The supported paths then are used in the search for a matching template from the database. When a matching template is found, the fields of the template is filled to make a profile. When no matching template is found, a new template is formed using the set of capability classes.

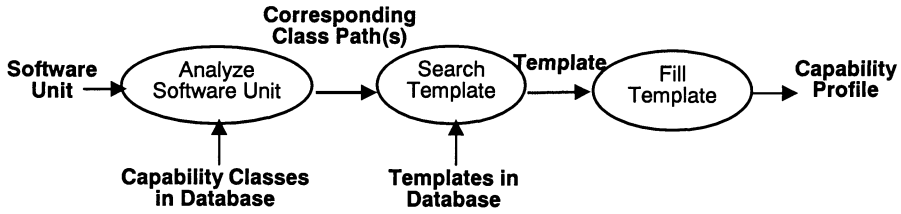

Figure 5. Capability profiling process [3].

4.2 Software requirements analysis process

The software requirements analysis process shown in Figure 2 is detailed in Figure 6. Capability profiles for each manufacturing software unit are derived from manufacturing software requirements in the software requirements analysis process. As a first step, manufacturing software requirements are decomposed into several primitive requirements which are fulfilled by capability classes that are selected from the database. When a template that corresponds to the class exists, the template is filled with specific requirements in order to generate a required capability profile. When such a template does not exist, a new template is created based on rules for template creation.

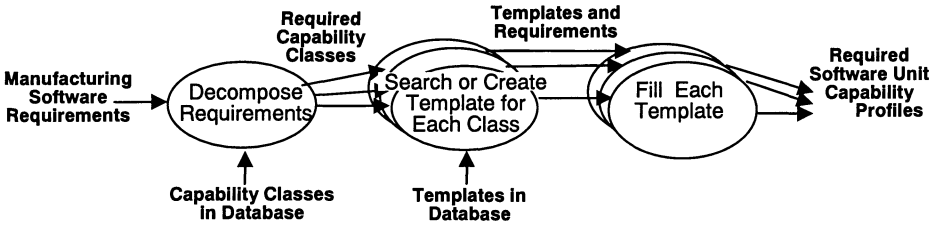

Figure 6. Manufacturing software requirements analysis process [3]

4.3 Software unit selection and verification process

The software unit selection and verification process shown in Figure 2 is detailed in Figure 7. For each required capability profile, a search of matching capability profiles that represent available software units are performed. Matching is performed according to the following rules. When a match exists, the software unit is added to a list of candidates. When a match does not exist, one of the following occurs: a) a new software unit is developed to meet the required profile, b) the required profile is decomposed into a combination of several profiles, or c) requirements are reconsidered against existing profiles. The profile for the new software unit is registered to the database according to the profiling process in 4.1. The selected software units is verified against the manufacturing software requirements according to interoperability criteria.

5. CONCEPTUAL STRUCTURE OF CAPABILITY CLASSES AND TEMPLATES

5.1 Capability class structure

Software capability classes are defined in the structure shown in Figure 8. The contents of a software unit capability class include, but may not be limited to type of manufacturing domain, type of manufacturing activity as differentiated by the process it is part of, the resources involved in conducting the activity, and the information types exchanged during the activity, type of computing system as differentiated by the operating environment, the software architecture, and the design pattern used, type of services, protocol, and data types used in running the software unit, supplier

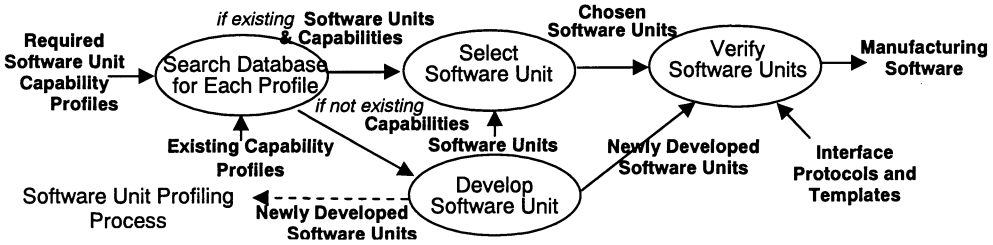

Figure 7. Software unit selection and verification process [3].

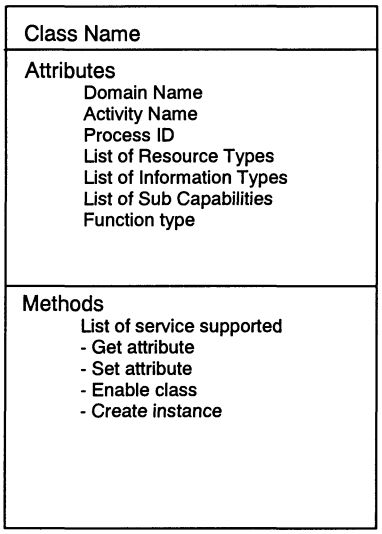

Figure 8. Capability class structure [3].

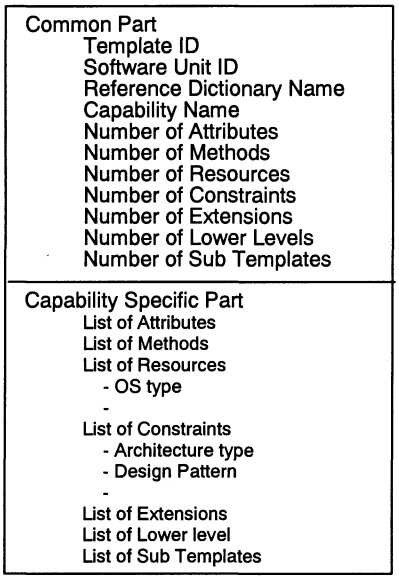

Figure 9. Capability template structure [3].

name, software version, and change history, performance benchmarks, reliability indices, service and support policy, and pricing terms and conditions of use.

5.2 Capability templates structure

A conceptual structure of a capability template is shown in Figure 9. The structure consists of a part that is common to all templates and another part that is specific to capability class. When a capability class is specified in a template and such a class has been instantiated, then the instantiated class represents an object. Two capability templates are identical if their respective attributes and operations are identical. When the attributes of one template form a subset of the attributes of another and the operations of one template form a subset of the operations of another, then the two capability templates are considered to be compatible and have a match.

6. CONCLUSIONS

The ISO 16100 series enable manufacturing software integration by providing the following: a) standard interface specifications that allow information exchange among software units in industrial automation systems developed by different vendors, b) software capability profiling, using a standardized method to enable users to select software units that meet their functional requirements, and c) conformance tests that ensure the integrity of the software integration. The ISO 16100 series consist of four parts. Part 1 specifies a framework for interoperability of a set of manufacturing software products. Part 2 specifies a methodology for constructing profiles of manufacturing software capabilities. Part 3 will specify the interface protocol and capability templates. Part 4 will specify the concepts and rules for the conformity assessment of the other parts of ISO 16100 [2, 3].

ISO 16100 Part 1 has been published. Part 2 is under preparation for publishing. This paper described the concepts and methodology which were newly proposed in the development procedure for Parts 1 and 2. Now, Part 3 is under development.

ACKNOWLEDGEMENTS

The authors thank ISO/TC 184/SC 5/ WG 4 members and its Japanese mirror committee members for fruitful discussions. Especially, the authors are grateful to Mr. E. delaHostoria, Mr. G. Winchester, Mr. B. Sunder and Dr. K. Kosanke of ISO/TC 184/SC 5/ WG 4 member for their useful effort to complete international standards. The authors are also grateful to Dr. U. Graefe, the previous convenor of ISO/TC 184/SC 5/ WG 4 for his helpful assistance with the writing of this paper in English.

REFERENCES

1. ISO/TC 184/SC 5/Study Group (1999). Study Group Report to SC5 Manufacturing Software Capability Profiling, *SC5 study Group N-18*
2. ISO/TC 184/SC 5/WG 4 (2002). Industrial automation systems and integration - manufacturing capability profiling for interoperability - Part 1: Framework, *ISO/IEC 16100-1*
3. ISO/TC 184/SC 5/WG 4 (2002). Industrial automation systems and integration - manufacturing capability profiling for interoperability - Part 2: Profiling methodology, *ISO/TC 184 DIS 16100-2*