INTERNATIONAL STANDARD

ISO 16100-5

First edition
2009-03-01

Industrial automation systems and integration — Manufacturing software capability profiling for interoperability —

Part 5: Methodology for profile matching using multiple capability class structures

Systèmes d'automatisation industrielle et intégration — Profil d'aptitude du logiciel de fabrication pour interopérabilité —

Partie 5: Méthodologie pour la correspondance de profil utilisant des structures de classe de capacité multiple

Reference number
ISO 16100-5:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 16100-5 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 5, *Architecture, communications and integration frameworks*.

ISO 16100 consists of the following parts, under the general title *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability*:

— *Part 1: Framework*

— *Part 2: Profiling methodology*

— *Part 3: Interface services, protocols and capability templates*

— *Part 4: Conformance test methods, criteria and reports*

— *Part 5: Methodology for profile matching using multiple capability class structures*

The following part is planned:

— *Part 6: Interface services and protocols for matching profiles using multiple capability class structures*

Introduction

The motivation for this part of ISO 16100 stems from the industrial and economic environment noted by TC 184/SC 5 in its strategic planning discussions, in particular:

— a growing base of vendor-specific solutions;

— user difficulties in applying standards;

— the need to move to modular sets of system integration tools;

— the recognition that application software and the expertise to apply that software are assets of the enterprise.

ISO 16100 is an International Standard for the computer-interpretable and human-readable representation of a software capability profile. Its goal is to provide a method to represent the capability of manufacturing software relative to its role throughout the life cycle of a manufacturing application, independent of a particular system architecture or implementation platform. This can lead to reduced production and information management costs to users and vendors/suppliers of manufacturing applications.

Certain diagrams in this part of ISO 16100 are constructed following unified modeling language (UML) conventions. Because not all concepts embodied in these diagrams are explained in the text, some familiarity with UML on the part of the reader is assumed.

Annex A describes the process for generating a manufacturing domain model (MDM) and manufacturing domain data (MDD).

Annex B gives an example of profile matching using multiple capability classes.

Industrial automation systems and integration — Manufacturing software capability profiling for interoperability —

Part 5: Methodology for profile matching using multiple capability class structures

1 Scope

This part of ISO 16100 specifies the methods and rules for matching existing manufacturing software unit (MSU) capability profiles and required capability profiles derived from multiple capability class structures. The methods and rules allow MSUs in manufacturing applications to be evaluated for interoperability and, in some cases, for interchangeability.

The following are outside the scope of this part of ISO 16100:

— services for creating, registering and accessing the various templates for the reference manufacturing domain models, the manufacturing domain data and the capability class structures;

— the conformance tables that reference Type 1 services defined and specified in ISO 16100-3;

— additional services needed for managing these templates in a database or equivalent object repository.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 16100-1, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 1: Framework*

ISO 16100-2:2003, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 2: Profiling methodology*

ISO 16100-3:2005, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 3: Interface services, protocols and capability templates*

ISO 16100-4:2006, *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 4: Conformance test methods, criteria and reports*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 16100-1, ISO 16100-2, ISO 16100-3, ISO 16100-4 and the following apply.

3.1
capability class
element within the capability profiling method that represents software unit functionality and behaviour with regard to the software unit's role in a manufacturing activity, as denoted in a capability inheritance structure and as deployed in a capability aggregation structure

NOTE 1 The role of an MSU changes when used in different manufacturing activities; however, the MSU's corresponding capability class is positioned uniquely in an inheritance structure, but can assume different positions in an aggregation structure.

NOTE 2 In this part of ISO 16100, a capability class template is identical to a capability template (see ISO 16100-2:2003, 6.3, for requirements for capability templates).

NOTE 3 Adapted from ISO 16100-2:2003, 3.3.

3.2
capability class structure template
extensible markup language (XML) schema representing a hierarchy of capability classes

3.3
manufacturing domain data
unified modeling language (UML) class representing information about manufacturing resources, manufacturing activities, or items exchanged among manufacturing resources within a particular manufacturing domain

3.4
manufacturing domain data template
eXtensible markup language (XML) schema representing a manufacturing domain data

3.5
manufacturing domain model
particular view of a manufacturing domain, consisting of manufacturing domain data and relationships among them, corresponding to the domain's applications

3.6
manufacturing domain model template
eXtensible markup language (XML) schema representing a manufacturing domain model

4 Abbreviated terms

CCS Capability Class Structure

CSI Conformance Statement for the Implementation

MDD Manufacturing Domain Data

MDM Manufacturing Domain Model

MES Manufacturing Operations Management

MSU Manufacturing Software Unit

UML Unified Modeling Language

XML eXtensible Markup Language

5 Multiple CCSs referenced in manufacturing applications and in MSUs

5.1 Profile matching concept

Figure 1 shows the concept of the profile matching using multiple capability class structures.

NOTE 1 The capability class structures of the provided MSUs (left side of figure) are assumed to be based on the existence of a common capability class inheritance tree.

NOTE 2 The actual process for matching profiles uses the same algorithms that exist for matching XML schemas.

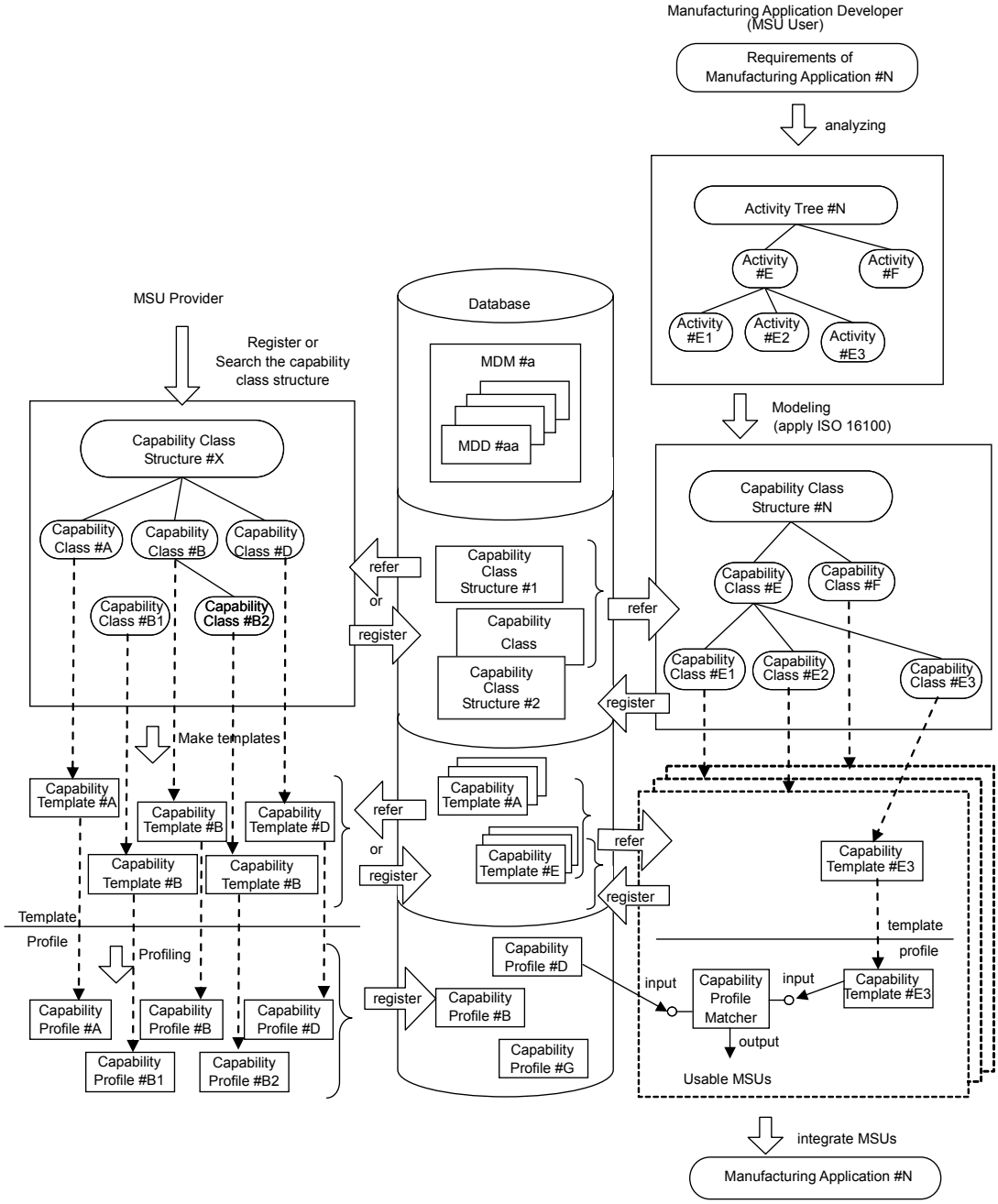

Figure 1 — Concept of profile matching using multiple capability class structures

5.2 Reuse of MSUs

To increase the efficiency of manufacturing application development, MSUs previously deployed in a similar manufacturing application should be reused. The MSU user shall only consider as candidates for reuse those MSUs whose capability profiles (see left side of Figure 1) meet his defined matching criteria for its capability profile (see right side of Figure 1). This part of ISO 16100 specifies a profile matching process wherein the capability class structures of the profiles being matched can be different, thus resulting in a larger number of candidate MSUs for reuse.

5.3 Registration of MSUs

A MSU provider registers an MSU so that it is widely available to many potential users of the MSU. The MSU provider shall perform the following procedures to prepare and register an MSU.

a) Analyze the set of activities that the MSU enables. The MSU can enable one or more activities.

b) Identify the capability class corresponding to each activity and search for the associated CCS to which the capability class belongs. If an MSU provides capabilities for two or more activities, those activities can belong to the same CCS or the different CCS.

c) Select the capability template for each capability class identified.

d) If there is no suitable CCS, construct the appropriate CCS and register it using appropriate database management methods. Then, generate the corresponding template and register that similarly.

e) Create the MSU capability profile by filling in the template(s) selected in procedure (c) or the new template(s) created in procedure (d), and register the template(s) using appropriate database management methods.

5.4 Applying reusable MSUs to meet requirements of a new manufacturing application

When a new manufacturing application is developed, the MSU user shall perform the following procedures.

a) Analyze the functional capability requirements of the manufacturing application and create an activity tree.

b) Create a CCS using existing or new capability classes to match the activity tree created in procedure (a), or select an existing CCS by using the methodology of ISO 16100-2.

c) For each capability class in the created or selected CCS, fill in the corresponding capability class template to create the set of required capability profiles.

d) Using a Type 2 capability profile matcher as described in ISO 16100-3, compare the set of required capability profiles to the available set of MSU capability profiles to find a set of existing MSUs that matches the set of required capability profiles.

e) Select the set of existing MSUs that meets the requirements of the new manufacturing application.

f) If the set of MSUs that meets the requirements is not found, develop a set of missing MSUs.

g) Combine the set of reused MSUs from procedure (e) and any set of developed MSUs from procedure (f) to meet the requirements of the new manufacturing application.

Figure 2 shows the implementation of the concept in Figure 1 to develop a new manufacturing application.

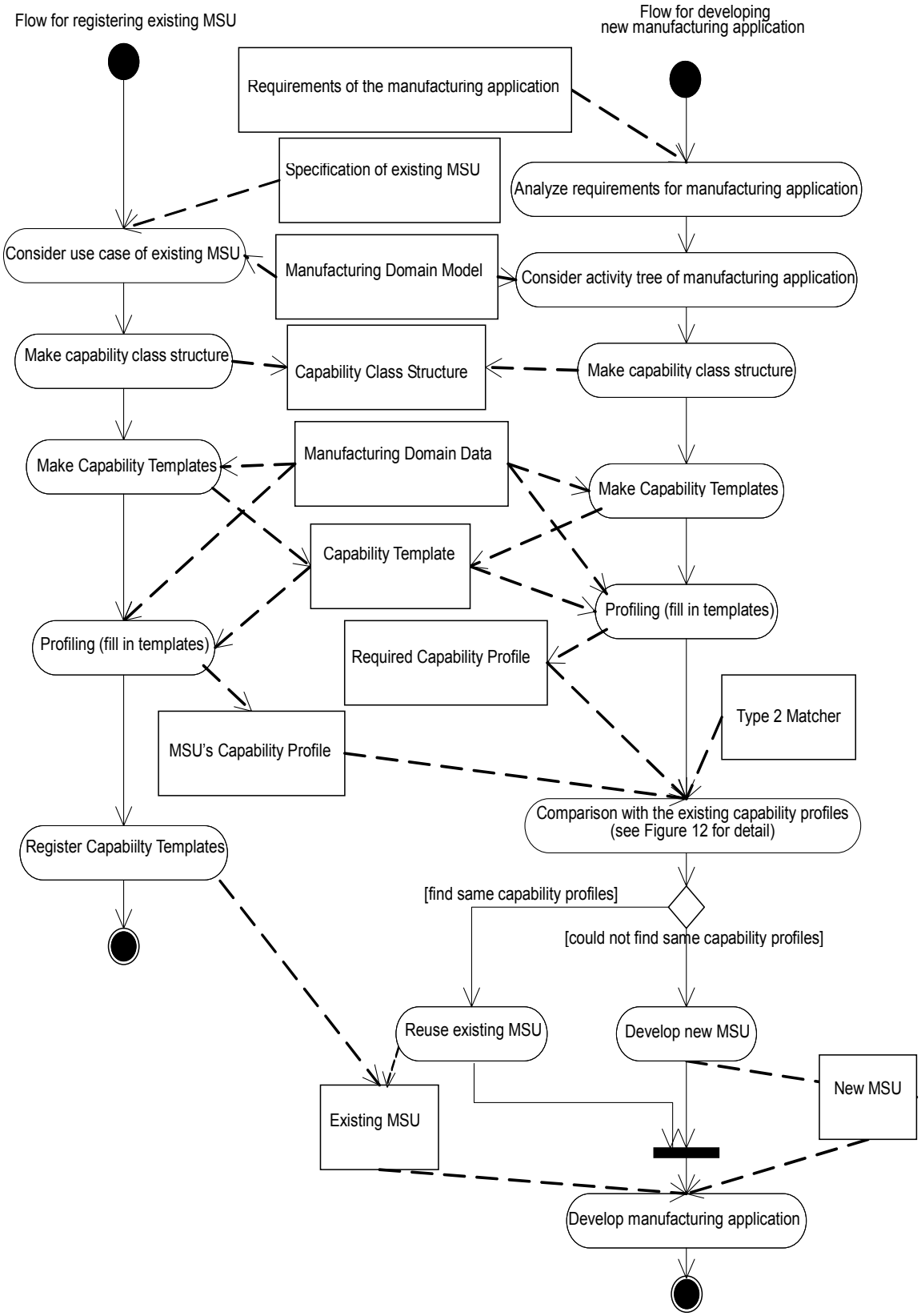

Figure 2 — Application development process with capability template, capability profile, and CCS

5.5 Manufacturing Domain Data

Figure 3 shows the MDD, MDM, CCS, Capability Class and their associations with the other manufacturing domain elements.

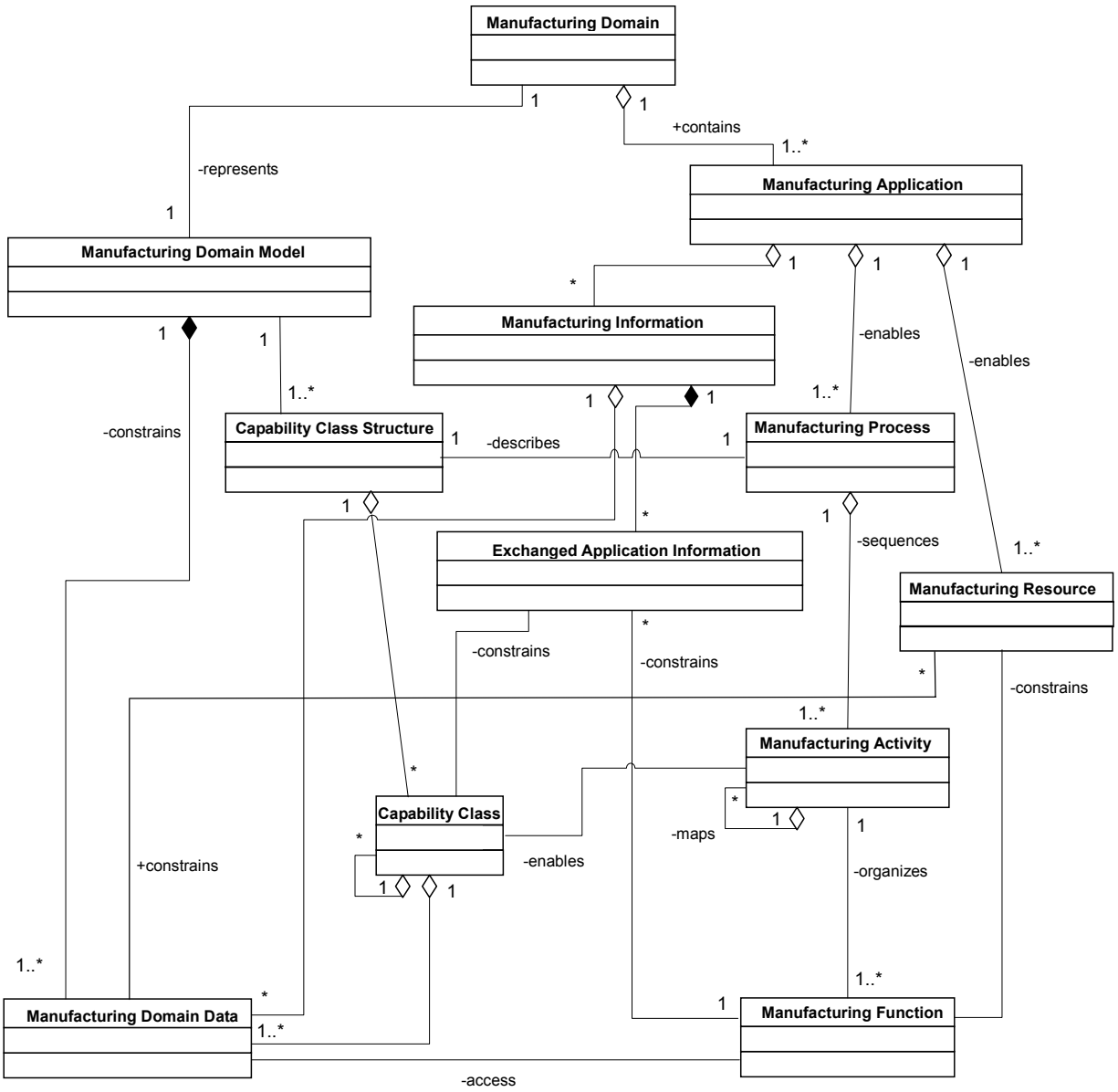

Figure 3 — Relationship between manufacturing domain and manufacturing application

MDDs represent different types of manufacturing information, including those that are exchanged between the resources within an application and between applications.

Figure 4 shows an example of a structure of a MDM with multiple MDDs. The process an MDM creator follows to create an MDM and MDDs is described in Annex A.

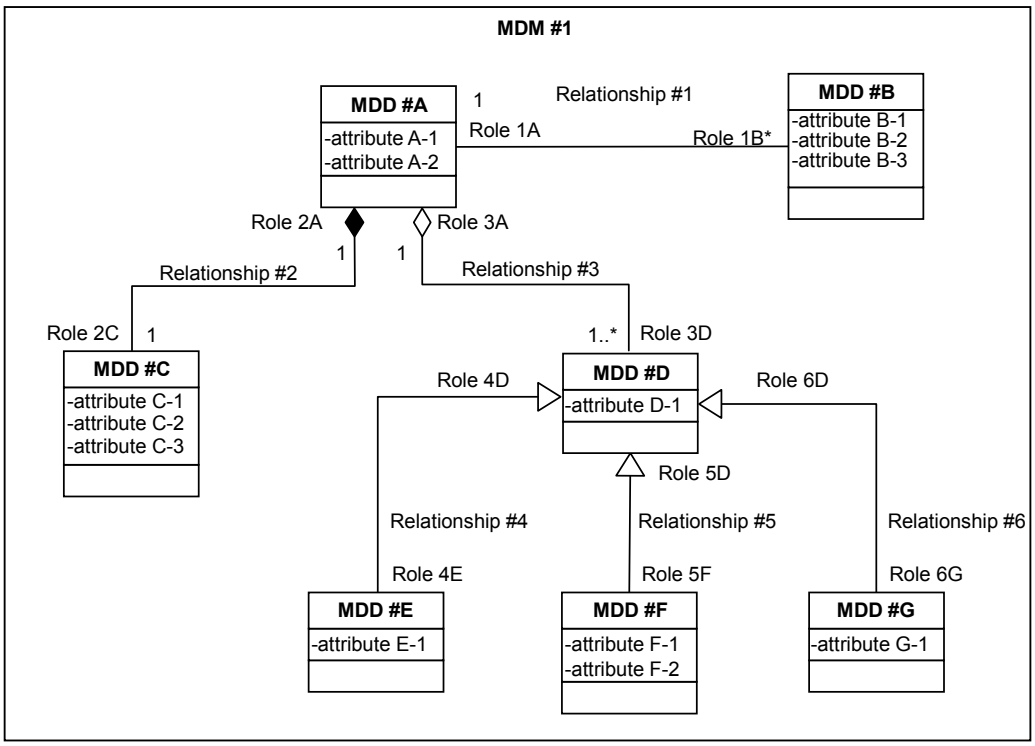

Figure 4 — Example structure of a MDM

Within a specific manufacturing domain, the MDM creator can represent a manufacturing application as a set of MDDs. An MDD provides information about various aspects of a manufacturing application such as:

a) manufacturing resources (e.g. MSU, equipment, automation devices, personnel, material, work-in-process inventory);

b) manufacturing processes (e.g. operations, activities);

c) manufacturing information exchanged (e.g. product data, recipe, manufacturing data, quality data);

d) relationships among the resources, processes and information exchanged.(e.g. data flow, network configuration, work flow).

In Figure 5, each MDD within a specific manufacturing domain consists of attributes and a set of relationships with other MDDs in the same domain using a relationship class. The relationship constraint and relationship type attributes in the relationship class delineate the allowed relationships among the MDDs within the specific MDM. The MDM creator shall descriptively name the MDD exchanged among manufacturing functions or among manufacturing activities such that each MDD is unique in the target manufacturing domain.

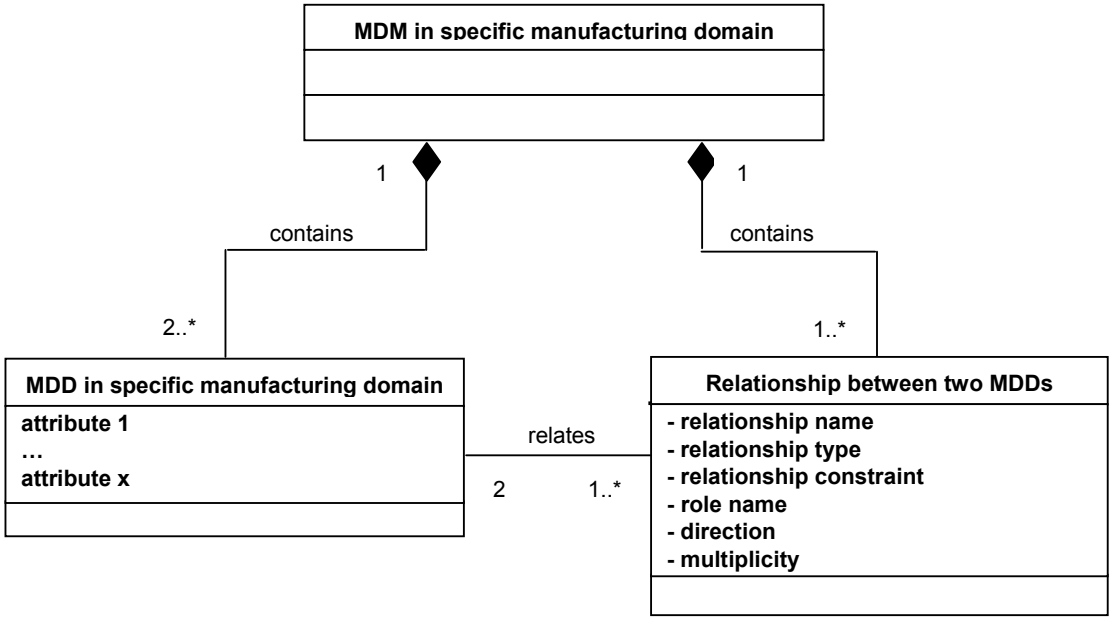

Figure 5 — Relationship between MDD and MDM

Figure 5 shows the relationship between a MDM and a minimum of two MDDs. These MDDs can typically form tree structures and have their relationships constrained by the relationship class noted in Figure 5. There is also the possibility of implementing specific class definitions specified as external classes from some related standards applicable to a particular manufacturing domain.

5.6 Mapping capability classes to MDDs

The MSU provider or MSU user models his activity tree based on the MDM from the requirements of the manufacturing application. The MSU provider or MSU user shall, in order to distinguish a particular activity in an activity tree, label an activity with an unambiguous and unique name, along with semantic information expressed in terms of a sequence of MDDs. The activities in the activity tree form the CCS. MSU providers and manufacturing application developers specify capability classes using a common set of MDDs.

Figure 6 shows two different CCSs mapped from their respective manufacturing activity trees. CCS #1 and CCS #2 are distinct structures, with some of the capability classes in the two CCSs being the same. These same capability classes can be recognized when a capability profile corresponding to a capability class is described using MDDs from the same MDM.

Each capability class in a structure is formed based on the services of a MDD or combined MDDs in the MDM. The canonical expression of a capability class includes specific lists of attributes, methods, and resources.

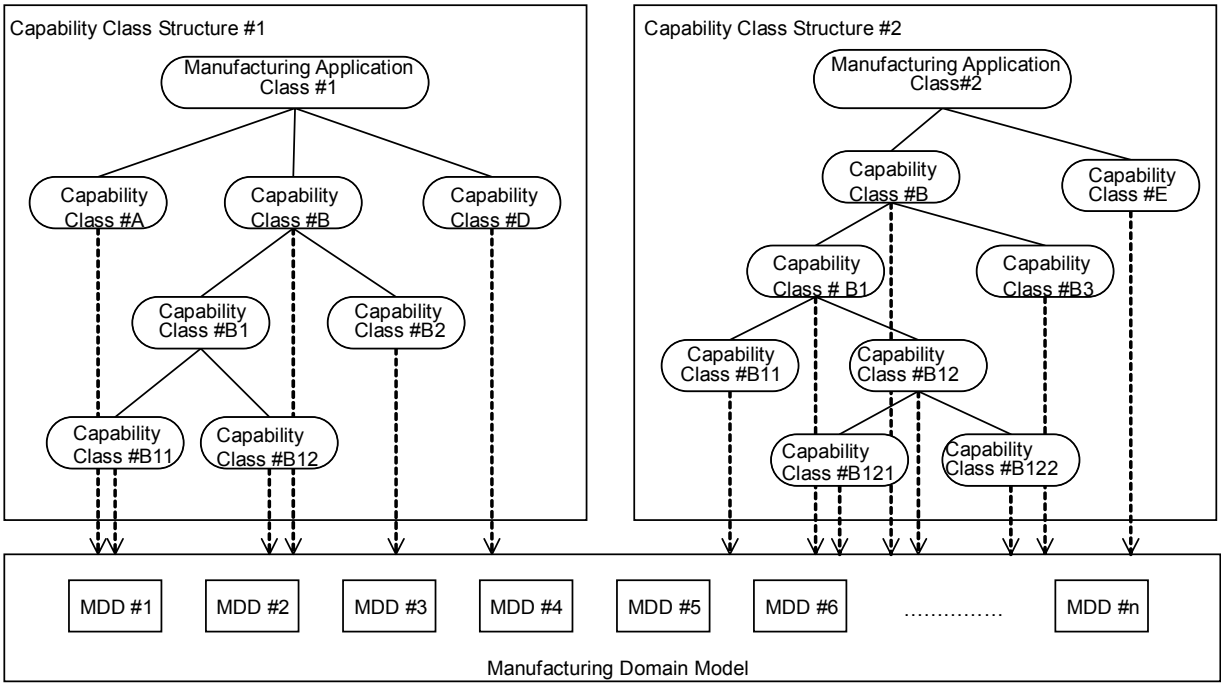

Figure 6 — Multiple capability classes described using MDDs from the same MDM

6 Methods and rules for capability profiling

6.1 Templates for MSU capability profiling

The following four templates are used in MSU capability profiling when there are multiple capability class structures:

a) CCS template;

b) Capability template;

c) MDM template;

d) MDD template.

6.2 CCS template

6.2.1 Conceptual structure

The CCS template shall contain, at a minimum, the following elements:

a) CCS Creator Name;

b) CCS ID;

c) ID for each capability class;

d) Parent Node ID for each capability class (the root node has no value for its Parent Node ID);

e) Child Node ID(s) for each capability class.

Figure 7 shows a conceptual structure of a CCS template.

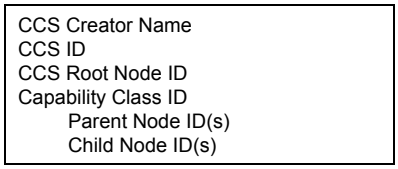

Figure 7 — Conceptual structure of a CCS template

6.2.2 Formal structure

The MSU provider or MSU user shall describe CCS templates using XML schemas. The formal structure of the CCS template shall be as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="CapabilityClassStructure">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="CCS_Creator_Name">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="CCS_ID">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="CCS_Root_Node_ID">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:ID" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Capability_Class" type="RecursionType"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="RecursionType">
        <xs:sequence>
            <xs:element name="Capability_Class_ID">
                <xs:complexType>
                    <xs:attribute name="id" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:sequence maxOccurs="unbounded">
                <xs:element name="Parent_Node_ID">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
                <xs:element name="Child_Node_ID">
                    <xs:complexType>
                        <xs:sequence>
                            <!--the follwing sentence shows the recurdion definition of
capability_class-->
                            <xs:element name="Capability_Class" type="RecursionType"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
```

```
            </xs:sequence>
          </xs:sequence>
      </xs:complexType>
</xs:schema>
```

6.3 Capability profile template

6.3.1 Conceptual structure

A capability profile template shall contain a Common Part in accordance with ISO 16100-2:2003, 6.3, and ISO 16100-3:2005, 7.2.2, and a Specific Part in accordance with ISO 16100-2:2003, 6.3. The Specific Part shall, at a minimum, contain the elements identified in ISO 16100-2 along with the following additional elements:

a) Reference MDM Name;

b) MDD Description Format (e.g. list of MDD objects);

c) MDD Description (e.g. time ordered access to MDD objects).

Figure 8 shows a conceptual structure of a capability profile template.

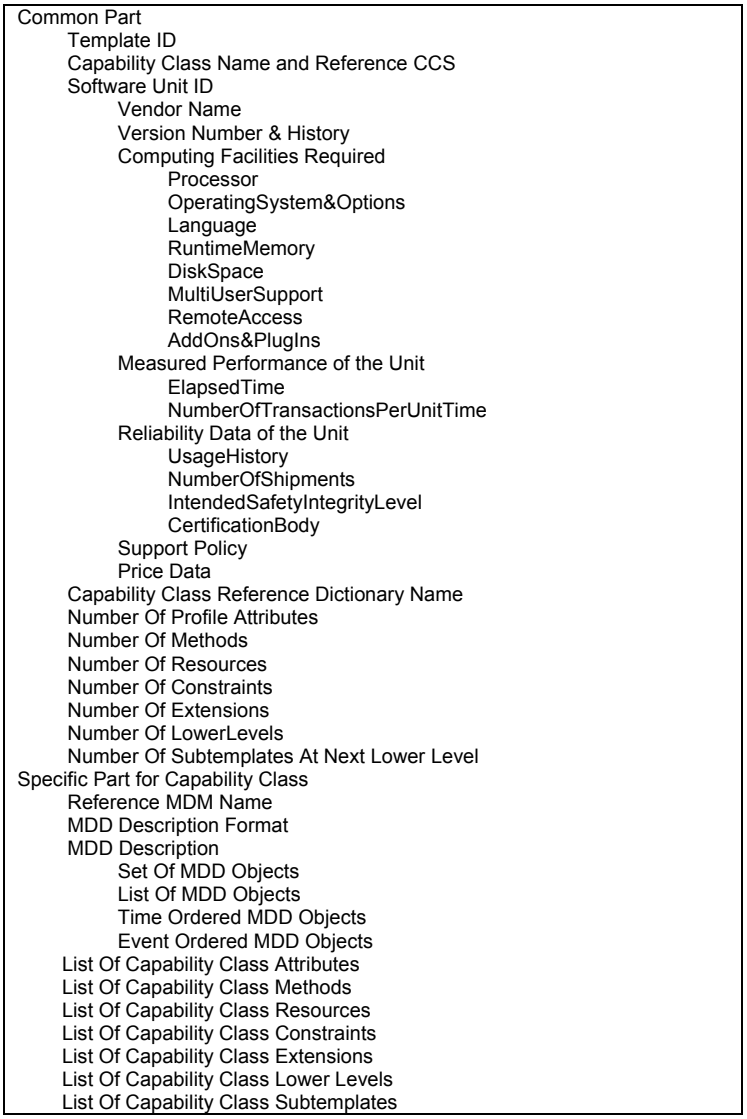

Figure 8 — Conceptual structure of a capability profile template

6.3.2 Formal structure

The MSU provider or MSU user shall describe capability profile templates using XML schemas. The formal structure of a capability profile template shall be as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="CapabilityProfiling">
        <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
                <xs:element name="Type">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:string" use="required"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="CapabilityProfile">
                    <xs:complexType>
                        <xs:sequence>
```

```
                            <xs:element name="Pkgtype">
                                <xs:complexType>
                                    <xs:attribute name="version" type="xs:string"
form="unqualified"/>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Common" type="CommonPartType"/>
                            <xs:element name="Specific" type="SpecificPartType"/>
                        </xs:sequence>
                        <xs:attribute name="date" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="CommonPartType">
        <xs:sequence>
            <xs:choice>
                <xs:element name="Requirement">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="ID" type="xs:string"/>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="MSU_Capability">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="ID" type="xs:string"/>
                        </xs:sequence>
                        <xs:attribute name="id" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
            <xs:sequence maxOccurs="unbounded">
                <xs:element name="ReferenceCapabilityClassStructure">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:string" form="unqualified"/>
                        <xs:attribute name="name" type="xs:string" form="unqualified"/>
                        <xs:attribute name="version" type="xs:string" form="unqualified"/>
                        <xs:attribute name="url" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="TemplateID">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:element name="Capability_Class_Name">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Reference_Capability_Class_Structure_Name">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Version">
                <xs:complexType>
                    <xs:attribute name="major" type="xs:string" form="unqualified"/>
                    <xs:attribute name="minor" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Owner">
                <xs:complexType>
                    <xs:sequence>
```

```
                        <xs:element name="Name" type="xs:string" minOccurs="0"/>
                        <xs:element name="Street" type="xs:string" minOccurs="0"/>
                        <xs:element name="City" type="xs:string" minOccurs="0"/>
                        <xs:element name="Zip" type="xs:string" minOccurs="0"/>
                        <xs:element name="State" type="xs:string" minOccurs="0"/>
                        <xs:element name="Country" type="xs:string" minOccurs="0"/>
                        <xs:element name="Comment" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="ComputingFacilities" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Processor" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="type" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="OperatingSystem" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="type" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Language" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="name" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Memory" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="size" type="xs:string" form="unqualified"/>
                                <xs:attribute name="unit" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="DiskSpace" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="size" type="xs:string" form="unqualified"/>
                                <xs:attribute name="unit" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="type" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Performance" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="elapsedTime" type="xs:string" form="unqualified"/>
                    <xs:attribute name="transactionsPerUnitTime" type="xs:string"
form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="ReliabilityData" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="UsageHistory" type="xs:string" minOccurs="0"/>
                        <xs:element name="Shipments" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="number" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="IntendedSafetyIntegrity" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="level" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Certification" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:attribute name="number" type="xs:string" form="unqualified"/>
```

```
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="SupportPolicy" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="index" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="PriceData" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="invest" type="xs:string" form="unqualified"/>
                    <xs:attribute name="annualSupport" type="xs:string" form="unqualified"/>
                    <xs:attribute name="unit" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="ReferenceDictionaryName">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfProfileAttributes" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfMethods" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfResources" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfConstraints" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfExtensions" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfLowerLevels" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="NumberOfSubtemplatesAtNextLowerLevel" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="number" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="SpecificPartType">
        <xs:sequence>
            <xs:element name="Reference_MDM_Name">
                <xs:complexType>
                    <xs:attribute name="domain_name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="MDD_Description_Format">
                <xs:complexType>
```

```
                    <xs:attribute name="format_name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="MDD_Description">
                <xs:complexType>
                    <xs:sequence>
                        <xs:choice>
                            <xs:element name="Set_Of_MDD_Objects">
                                <xs:complexType>
                                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                        <xs:element name="MDD_Name">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string"
form="unqualified"/>
                                                <xs:attribute name="action" type="xs:string"
form="unqualified"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="List_Of_MDD_Objects">
                                <xs:complexType>
                                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                        <xs:element name="MDD_Name">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string"
form="unqualified"/>
                                                <xs:attribute name="action" type="xs:string"
form="unqualified"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Time_Ordered_MDD_Objects">
                                <xs:complexType>
                                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                        <xs:element name="Time_Occurrence_Of_MDD_Object">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="MDD_Name">
                                                        <xs:complexType>
                                                            <xs:attribute name="name"
type="xs:string" form="unqualified"/>
                                                            <xs:attribute name="action"
type="xs:string" form="unqualified"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                    <xs:element name="MDD_Qualifiers">
                                                        <xs:complexType>
                                                            <xs:sequence minOccurs="0"
maxOccurs="unbounded">
                                                                <xs:element name="Qualifier_Name">
                                                                    <xs:complexType>
                                                                        <xs:attribute name="name"
type="xs:string" form="unqualified"/>
                                                                    </xs:complexType>
                                                                </xs:element>
                                                            </xs:sequence>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Event_Ordered_MDD_Objects">
```

```
                                <xs:complexType>
                                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                        <xs:element name="Event_Occurrence_Of_MDD_Object">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="MDD_Name">
                                                        <xs:complexType>
                                                            <xs:attribute name="name"
type="xs:string" form="unqualified"/>
                                                            <xs:attribute name="action"
type="xs:string" form="unqualified"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                    <xs:element name="MDD_Qualifiers">
                                                        <xs:complexType>
                                                            <xs:sequence minOccurs="0"
maxOccurs="unbounded">
                                                                <xs:element name="Qualifier_Name">
                                                                    <xs:complexType>
                                                                        <xs:attribute name="name"
type="xs:string" form="unqualified"/>
                                                                    </xs:complexType>
                                                                </xs:element>
                                                            </xs:sequence>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:choice>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="List_Of_CC_Attributes" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="List_Of_CC_Methods" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="List_Of_CC_Resources" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="List_Of_CC_Constraints" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="List_Of_CC_Extensions" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="List_Of_CC_Lower_Levels" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="List_Of_CC_Subtemplates" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

6.4 MDM Template

6.4.1 Conceptual structure

A MDM template shall contain a base part and an extension part. The base part shall include the following elements:

a) MDM Name;

b) Domain Reference Dictionary Name;

NOTE The Domain Reference Dictionary is composed of definitions specific to a manufacturing domain. The manufacturing domain is delimited by either a functional level in an enterprise or a group of activities within a functional level of an enterprise.

c) MDD Name — for each MDD within a MDM;

d) Relationship Type — for each MDD, an ordered list describing the connection(s) with the destination MDD(s)

e) Destination MDD Name — for each MDD, an ordered list of the name(s) of the other MDD(s) in the relationship;

f) Connection Point Direction — for each MDD, an ordered list of the connection point directions of the MDD(s) in the relationship;

g) Role Name — for each MDD, an ordered list of the name(s) of MDD role(s) in the relationship;

h) Multiplicity — for each MDD, an ordered list of multiplicity number(s) for the instances in the relationship.

The extension part contains elements that are specific either to the industry domain, the industry organization, or the enterprise functional domains.

Figure 9 shows a conceptual structure of a MDM template.

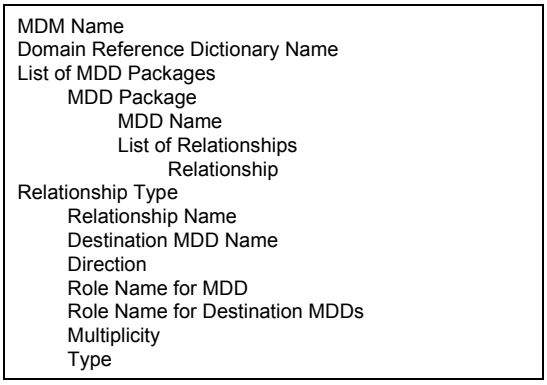

Figure 9 — Conceptual structure of a MDM template

6.4.2 Formal structure

The MDM creator shall describe MDM templates using XML schemas. The formal structure of a MDM template shall be as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="MDM">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="MDM_Name">
                    <xs:complexType>
                        <xs:attribute name="domain_name" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Domain_Reference_Dictionary_Name">
                    <xs:complexType>
                        <xs:attribute name="dictionary_name" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="List_Of_MDD_Packages">
                    <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                            <xs:element name="MDD_Package">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="MDD_Name">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string"
form="unqualified"/>
```

```
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="List_Of_Relationships">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="Relationship"
type="Relationship_Type"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                                <xs:attribute name="id" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="Relationship_Type">
        <xs:sequence>
            <xs:element name="Relationship_Name">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Destination_MDD_Name">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Direction">
                <xs:complexType>
                    <xs:attribute name="direction" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Role_Name_For_MDD">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Role_Name_For_Destination_MDD">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Multiplicity">
                <xs:complexType>
                    <xs:attribute name="multiplicity" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Type">
                <xs:complexType>
                    <xs:attribute name="type" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" form="unqualified"/>
    </xs:complexType>
</xs:schema>
```

6.5 MDD templates

6.5.1 Conceptual structure

A MDD template shall contain a base part and an extension part. The base part shall include the following elements:

a) MDD Name;

b) Reference MDM Name;

c) MDD Type

NOTE MDD Type can be used to distinguish a manufacturing resource, a manufacturing function, or a manufacturing information item represented by the MDD.

d) Attribute Name -- for each attribute in a MDD;

e) Attribute Type -- for each attribute in a MDD.

The extension part of a MDD template consists of other attributes to support MDD types that are specific to either an industry domain, an industry organization, or an industry application.

Figure 10 shows a conceptual structure of a MDD template.

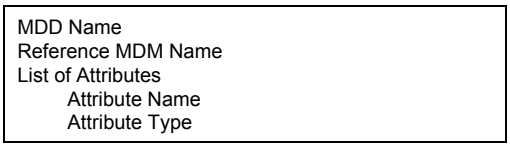

Figure 10 — Conceptual structure of a MDD template

6.5.2 Formal structure

The MDM creator shall describe MDD templates using XML schemas. The formal template structure of the MDD is as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="MDD">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="MDD_Name">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Reference_MDM_Name">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="List_Of_Attributes">
                    <xs:complexType>
                        <xs:sequence minOccurs="0" maxOccurs="unbounded">
                            <xs:element name="Attribute">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Attribute_Name">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string"
form="unqualified"/>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="Attribute_Type">
                                            <xs:complexType>
                                                <xs:attribute name="type" type="xs:string"
form="unqualified"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                    <xs:attribute name="id" type="xs:string" form="unqualified"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
```

```
    </xs:element>
</xs:schema>
```

7 Profile matching based on multiple capability class structures

7.1 Capability profile matching procedure

As shown in Figure 11, a Type 2 Matcher is used to determine if a functional correspondence exists between two capability profiles. The Type 2 Matcher compares the features of manufacturing functions described in a required capability profile and a MSU capability profile. A MSU provider or manufacturing application developer can evaluate the existence of a functional correspondence of these profiles even when capability templates are based on different capability class structures within the same manufacturing domain.

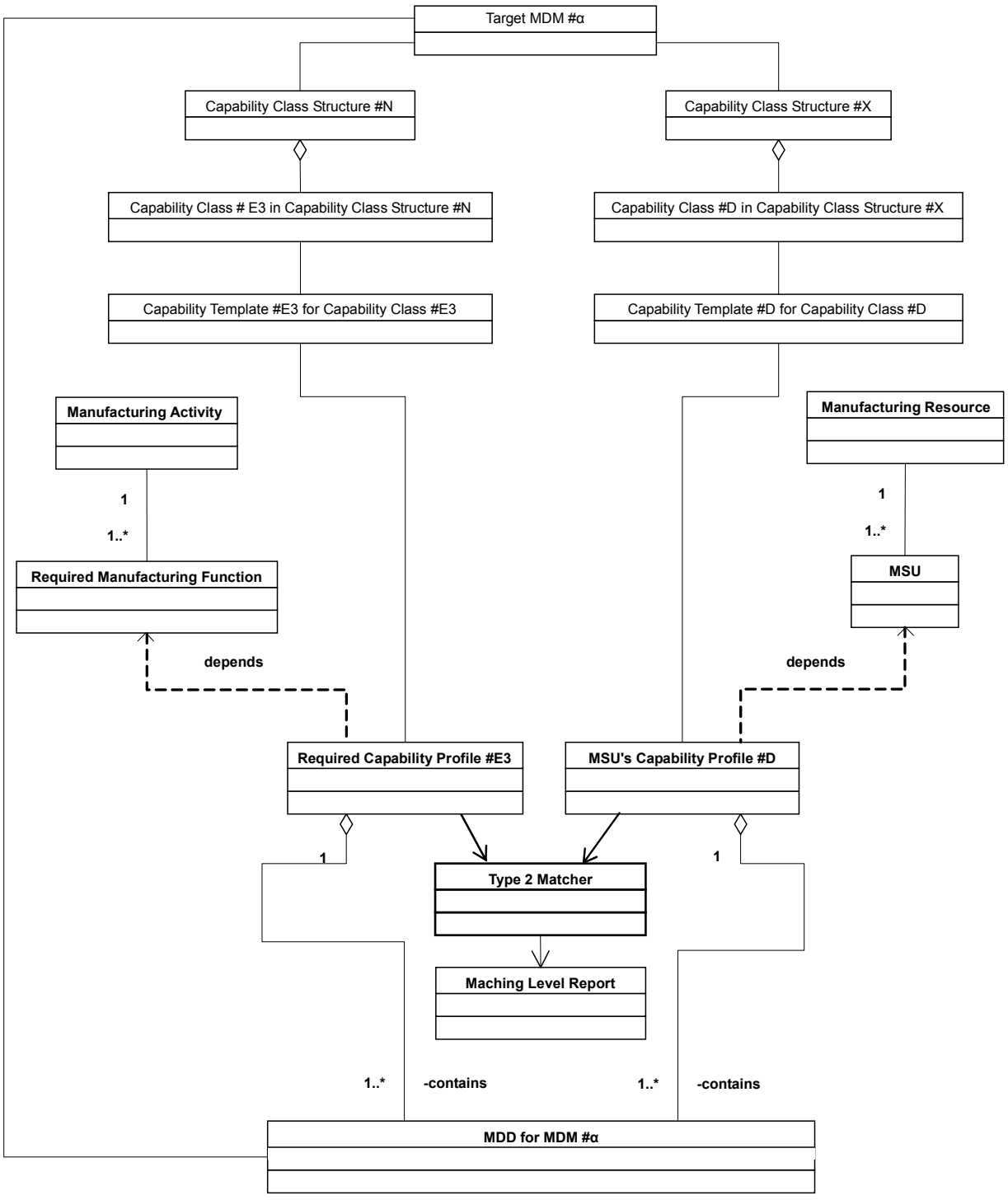

Figure 11 — Matching capability profiles with Type 2 Matcher

A Type 2 Matcher shall match a required capability profile and a MSU capability profile according to the matching procedure depicted in Figure 12. A Type 2 Matcher shall make use of the reference CCS names and related information from the two inputted capability profiles in order to determine if these profiles are based on a common MDM and common set of MDDs. When these profiles are based on a common MDM and common set of MDDs, a Type 2 Matcher shall be able to evaluate the existence of a functional correspondence between these profiles.

The capability profile matching process using a Type 2 Matcher begins with extracting the capability class reference dictionary IDs from the inputted capability profiles and compares these IDs. If these IDs are the same, then the Type 2 Matcher follows the procedure as for a Type 1 Matcher defined in ISO 16100-2. If these IDs are not the same, then the Matcher extracts the reference MDM IDs from the inputted capability profiles and compares these MDM IDs. If these MDM IDs are not the same, then the Type 2 Matcher reports that a comparison of the inputted profiles cannot be made. If these MDM IDs are the same, then the Type 2 Matcher extracts the capability definition formats from the inputted profiles and compares these formats. If these formats are the not the same, the MDDs in the capability definitions are converted to a single capability definition format by a means external to the Type 2 Matcher. If these formats are the same, no conversion is made. Then the Type 2 Matcher extracts the sets of MDDs contained in the capability definitions for these and compares them to determine the existence of functional correspondence between the profiles. The Type 2 Matcher then reports the matching level of the MSU profile relative to the required profile.

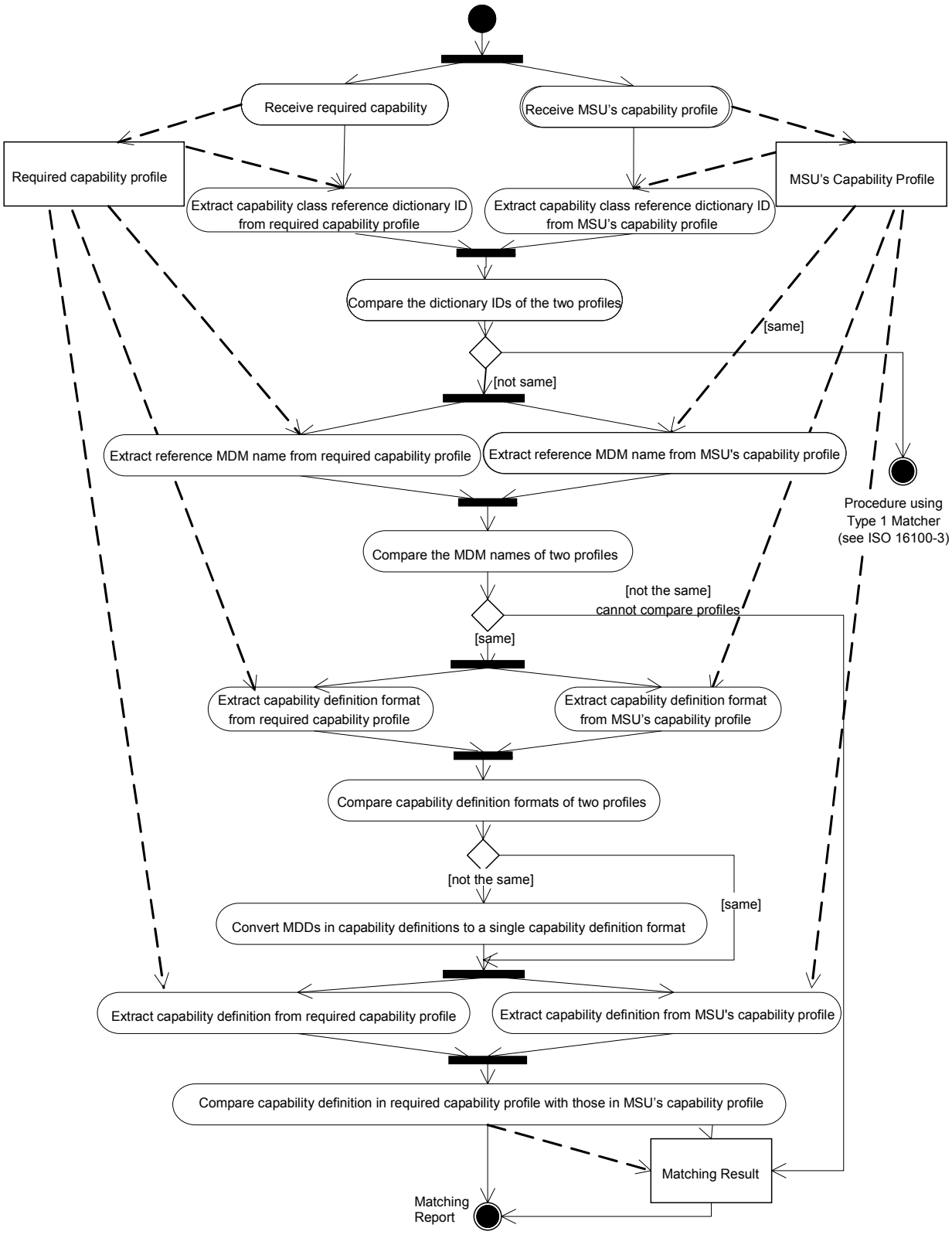

Figure 12 — Capability profile matching procedure

7.2 Matching report

The matching level generated by a Type 2 Matcher, when comparing the contents of two target capability profiles, shall assume one of following values.

a) Complete Match — all the manufacturing functions referenced in the required capability profile match completely with all the corresponding functions referenced in the MSU capability profile. This means that both sets of manufacturing functions are fully equivalent in terms of both the MDD objects being equivalent and the time ordering of these objects being equivalent.

b) All Mandatory Match — all the mandatory functions in the required capability profile are completely matched with a corresponding set of manufacturing functions referenced in the MSU capability profile. The matching level report includes information about the details of the MSU functions in the corresponding set.

c) Some Mandatory Match — the required capability profile is matched partially by the MSU capability profile. The matching level report includes information about the details of the MSU functions that matched the functions referenced in the required capability profile.

d) No Mandatory Match — none of the mandatory functions referenced in the required capability profile match the functions referenced in the MSU capability profile.

8 Conformance

The conformance methodology specified in ISO 16100-4 applies to this part of ISO 16100. This clause adds CSIs, as defined in ISO 16100-4:2006, 6.1.3, for a CCS template (see Table 1), for a MDM template (see Table 3) and for a MDD template (see Table 4). In addition, this clause adds CSIs for elements of a capability profile template (see Table 2) not addressed in ISO 16100-4. The conformance point types shown in the tables in this clause are defined in ISO 16100-4:2006, Table 5.

Table 1 — CSIs of a CCS template

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index_1	XML format	ISO 16100-5: 2009, 6.2.2	A	CCS in XML schema format.
index_2	CCS_Creator_Name	ISO 16100-5: 2009, 6.2.2	A	Presence and uniqueness of attribute "name"
index _3	CCS_ID	ISO 16100-5: 2009, 6.2.2	A	Presence and uniqueness of attribute "id"
index _4	CCS_Root_Node_ID	ISO 16100-5: 2009, 6.2.2	A	Presence and uniqueness of attribute "id"
Index_5	Capability_Class	ISO 16100-5: 2009, 6.2.2	A	Presence and uniqueness of attribute "id" and all Parent_ Node(s) and Child_Node(s)
index_5.1	Parent_Node_ID	ISO 16100-5: 2009, 6.2.2	A	Presence and location of all direct Parent_ Node(s) with attribute "id"(s)
index_5.2	Child_Node_ID	ISO 16100-5: 2009, 6.2.2	A	Presence and location of all Child_Node(s) with capability_ class type

Table 2 — CSIs of a capability profile template

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index_1	XML format	ISO 16100-5: 2009, 6.3.2	A	Capability template in XML schema format
Index_2	CapabilityProfiling	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema component
index_2.1	Type	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema component
Index_2.2	CapabilityProfile	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema component
Index_2.2.1	Pkgtype and version	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema components
Index_2.2.2	CommonPartType and SpecificPartType	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema components
Index _3	CommonPartType	ISO 16100-5: 2009, 6.3.2		
Index_3.1	Choice of capability profile type	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema component and ID of either "Requirement" or "MSU_Capability"
Index_3.2	ReferenceCapability ClassStructure	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attributes "id", "name", "version", and "url"
Index_3.3	Capability_Class _ Name	ISO 16100-5: 2009, 6.3.2	A	Presence of Capability_Class _ Name
Index_3.4	TemplateID	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema component
Index_3.5	Version	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema component
Index_3.6	Owner	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components “Street”, “city”, “zip”, “state”, “county”, “comments”

Table 2 (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index_3.7	ComputingFacilities	ISO 16100-5: 2009, 6.3.2	C	Presence of unbounded list components "Processor0", "OperatingSystem0", "Language", "Memory", "DiskSpace"
Index_3.8	Additional ComputingFacilities elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components satisfying criteria in index_4.7
Index_3.9	Performance	ISO 16100-5: 2009, 6.3.2	C	Presence and location of attributes "ElapsedTime" and "TransactionsPerUnitTime"
Index_3.10	Additional Performance elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components satisfying criteria in index_4.9
Index_3.11	ReliabilityData	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components "UsageHistory", "Shipments", "IntendedSafetyIntegrity", and "Certification"
Index_3.12	Additional ReliabilityData elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components satisfying criteria in index_4.9
Index_3.13	SupportPolicy	ISO 16100-5: 2009, 6.3.2	D	Presence and location of attribute "index"
Index_3.14	Additional SupportPolicy elements	ISO 16100-5: 2009, 6.3.2	D	Presence and location of schema components satisfying criteria in index_4.13
Index_3.15	PriceData	ISO 16100-5: 2009, 6.3.2	D	Presence and location of attributes "invest", "annualSupport", and "unit"

Table 2 (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index_3.16	Additional PriceData elements	ISO 16100-5: 2009, 6.3.2	D	Presence and location of schema components satisfying criteria in index_4.13
Index_3.17	CapabilityClass ReferenceDictionary Name	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute "name"
Index_3.18	NumberOfProfile Attributes	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attributes "number"
Index_3.19	NumberOfMethods	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute "number"
Index_3.20	NumberOfResources	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attributes "number"
Index_3.21	NumberOfConstraints	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute"number"
Index_3.22	NumberOfExtentions	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute "number"
Index_3.23	NumberOfLowerLevels	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute "number"
Index_3.24	NumberOf SubtemplatesAt NextLowerLevel	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute "number"
Index _4	SpecificPartType	ISO 16100-5: 2009, 6.3.2		
Index _4.1	Reference_MDM_ Name	ISO 16100-5: 2009, 6.3.2	A	Presence and location of attribute "Domain_Name"

Table 2 (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index _4.2	Capability_Definition _Format	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema components with the attribute "Format_Name" having a value of either: "Set_of_MDD_objects" or "List_of_MDD_Objects" or "Time_Ordered_ MDD_Objects" or "Event_Ordered _MDD_Objects"
Index _4.3	Capability_Definition	ISO 16100-5: 2009, 6.3.2	A	Presence and location of schema components
Index _4.3.1	Set_of_MDD_ objects	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.1.1	MDD_Name	ISO 16100-5: 2009, 6.3.2	C	Presence and location of the schema component
Index _4.3.1.2	Additional MDD_Name elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.2	List_of_MDD_Objects	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.2.1	MDD_Name	ISO 16100-5: 2009, 6.3.2	C	Presence and location of attributes “name” and "action”
Index _4.3.2.2	Additional MDD_Name elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.3	Time_Ordered_ MDD_Objects	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index_4.3.3.1	Time_Occurrence_ of_MDD_Objects	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.3.1.1	MDD_Name	ISO 16100-5: 2009, 6.3.2	C	Presence and location of attributes “name” and "action"

Table 2 (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index _4.3.3.1.2	MDD_Qualifiers	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.3.1.2.1	Qualifier_Name	ISO 16100-5: 2009, 6.3.2	C	Presence and location of attribute "name"
Index_4.3.3.1.2 .2	Additional Qualifier_Name elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index_4.3.4	Event_Ordered_ MDD_Objects	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index_4.3.4.1	Event_Occurrence_ of_MDD_Object	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.4.1.1	MDD_Name	ISO 16100-5: 2009, 6.3.2	C	Presence and location of attributes "name" and "action"
Index _4.3.4.1.2	MDD_Qualifiers	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.3.4.1.2.1	Qualifier_Name	ISO 16100-5: 2009, 6.3.2	C	Presence and location of attribute "name"
Index_4.3.4.1.2 .2	Additional Qualifier_Name elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.4	List_Of_CC_ Attributes	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.4.1	Additional List_Of_CC_ Attributes elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.5	List_Of_CC_ Methods	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.5.1	Additional List_Of_CC_ Methods elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.6	List_Of_CC_ Resources	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components

Table 2 (*continued*)

Conformance point or set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
Index _4.6.1	Additional List_Of_CC_ Resources elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.7	List_Of_Constraints	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.7.1	Additional List_Of_CC_ Constraints elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.8	List_Of_CC_ Extensions	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.8.1	Additional List_Of_CC_ Extensions elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.9	List_Of_CC_Lower_ Levels	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.9.1	Additional List_Of_CC_Lower_ Levels elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.10	List_Of_CC_ Subtemplates	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components
Index _4.10.1	Additional List_Of_CC_ Subtemplates elements	ISO 16100-5: 2009, 6.3.2	C	Presence and location of schema components

In order to reference an unambiguous and unique MDM ID in a capability profile, a MDM creator shall use the template defined in 6.4.2 and register the MDM in the database using Table 3.

Table 3 — CSIs of a MDM Template

Conformance point and set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
index_1	XML Format	ISO 16100-5: 2009, 6.4.2	A	MDM Template in XML format.
Index_2	MDM_Name	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attribute “Domain_Name”
Index_3	Domain_Reference_ Dictionary_Name	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attribute “Dictionary_Name”
Index_4	List_of_MDD_ Packages	ISO 16100-5: 2009, 6.4.2	A	Presence and location of schema components
Index_4.1	MDD_Package	ISO 16100-5: 2009, 6.4.2	A	Presence and location of unbounded attribute "id"
Index_4.1.1	MDD_Name	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attribute “name”
Index_4.1.2	List_of _Relationships	ISO 16100-5: 2009, 6.4.2	A	Presence and location of schema components
Index_4.1.2.1	Relationship	ISO 16100-5: 2009, 6.4.2	A	Presence and location of unbounded attribute "Relationship_Name" with type "Relationship_Type"
Index_4.1.2.2	Additional Relationship elements	ISO 16100-5: 2009, 6.4.2	A	Presence and location of unbounded attribute "Relationship_Name" with type "Relationship_Type"
Index _5	Relationship_Type	ISO 16100-5: 2009, 6.4.2	A	Presence and location of schema components
Index _5.1	Relationship_Name	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attribute “name”
Index_5.2	Destination_MDD_ Name	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attribute "name"
Index_5.3	Direction	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attributes "direction"
Index_5.4	Role_Name_for_ Destination_MDD	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attributes "name"
Index_5.5	Multiplicity	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attributes "multiplicity"
Index_5.6	Type	ISO 16100-5: 2009, 6.4.2	A	Presence and location of attributes "type"

Table 4 — CSIs for a MDD template

Conformance point and set number	Conformance point description	Specification reference	Conformance point type	Abstract test criteria
index_1	XML Format	ISO 16100-5: 2009, 6.5.2	A	MDM Template in XML format.
Index_2	MDD_Name	ISO 16100-5: 2009, 6.5.2	A	Presence and location of attribute “name”
Index_3	Reference_MDM_ Name	ISO 16100-5: 2009, 6.5.2	A	Presence and location of attribute “name”
Index_4	List_of_Attributes	ISO 16100-5: 2009, 6.5.2	A	Presence and location of schema components
Index_4.1	Attribute	ISO 16100-5: 2009, 6.5.2	A	Presence and location of attribute "id"
Index_4.1.1	Additional Attribute elements	ISO 16100-5: 2009, 6.5.2	A	Presence and location of attribute "id"
Index_4.1.2	Attribute_Name	ISO 16100-5: 2009, 6.5.2	A	Presence and location of attribute "name"
Index_4.1.3	Attribute_Type	ISO 16100-5: 2009, 6.5.2	A	Presence and location of attribute "type"

Annex A
(informative)

Process for Generating a MDM and MDDs

Figure A.1 shows a typical procedure a modeler would follow to generate a MDM and MDDs. The modeler begins by determining the target manufacturing domain upon which the MDM and MDDs will be based. Using his past experience and refering to suitable existing models contained in international standards or other publications, the modeler collects a number of typical manufacturing applications relevant to the target manufacturing domain. The modeler then extracts the manufacturing functions from these manufacturing applications and analyzes the functions to identify the manufacturing information and manufacturing resources shared and exchanged between the functions. Using the manufacturing information and manufacturing resources identified, the modeler finally generates a MDM and accompanying MDDs.

In practice, the modeler also will validate the generated MDM and MDDs so that users, such as CCS designers, will have confidence in applying them to create capability profile templates and capability profiles.

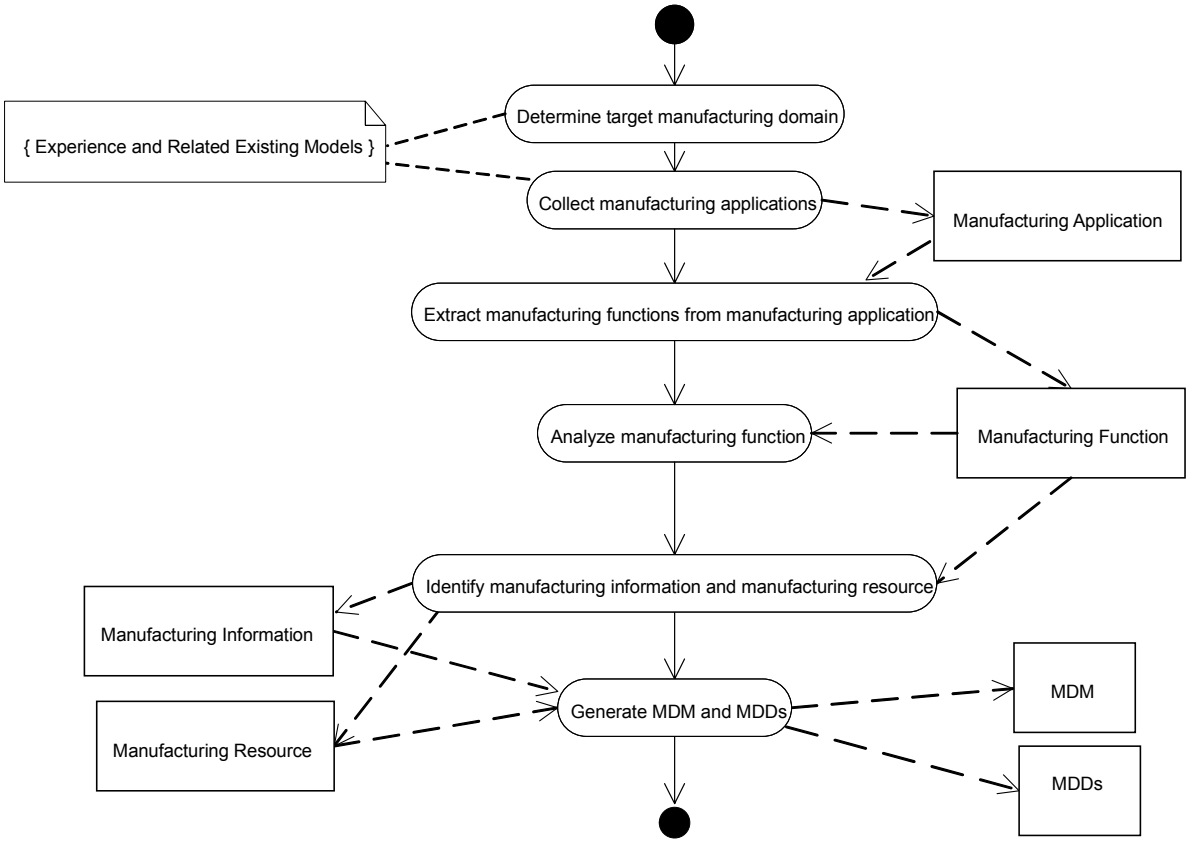

Figure A.1 — Generation process of MDD and MDM

Annex B
(informative)

Example of profile matching using multiple capability classes

B.1 Example of a Manufacturing Domain Model

Figure B.1 shows an example of a MDM in the MES[1] domain.

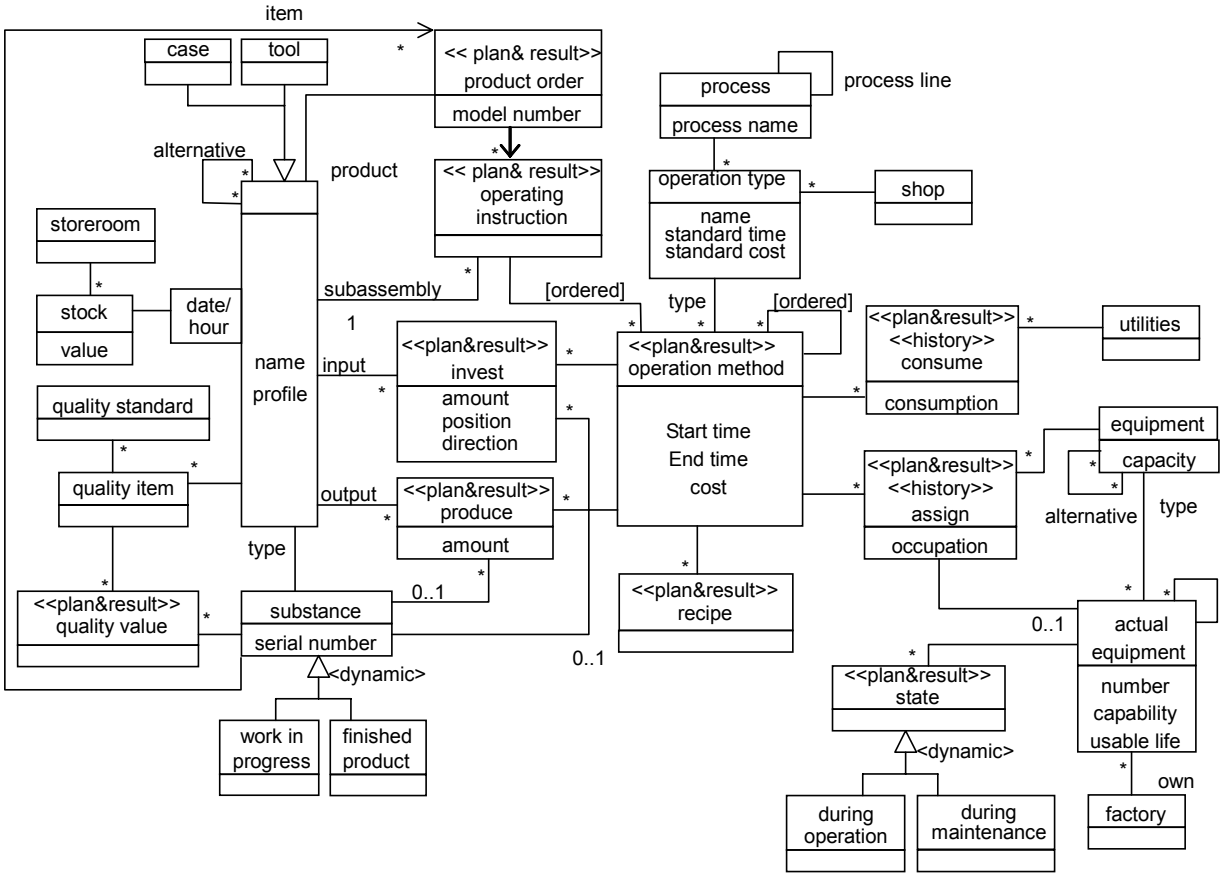

Figure B.1 — Example of MDM in MES Domain

1 MES, Manufacturing Execution System, is part of Manufacturing Operations Management as defined in IEC 62264-3.

The meaning of the MDD elements appearing in Figure 1 are as follows:

a) item — general term encompassing raw materials, parts, work-in-progress (see (f) below), substance (see (e) below) and finished products (see (g) below);

b) quality item — name of an attribute measured for product acceptance;

c) quality standard — specified nominal value of an attribute measured for product acceptance;

d) quality value — actual nominal value of an attribute measured for product acceptance;

e) substance — items (as defined in (a) above) flowing through a production activity;

f) work-in-progress — items (as defined in (a) above) that have begun, but have not completed, their production activities;

g) finished product — items (as defined in (a) above) that have completed all of their production activities;

h) recipe — set of operation parameters of the equipment assigned to the operation method (see (i) below);

i) operation method — activity that produces a finished product (see (g) above) or work-in-progress (see (f) above);

j) operation type — a general description of related operations; e.g. the operation type for turning, drilling, and milling operations is "machining";

k) process — ordered list of operation methods (see (i) above) employed in producing an item;

l) stock — residual quantity of an item at a given time;

m) storeroom — location used for the storage of items;

n) shop — location in which production occurs;

o) utility — e.g. water, air, electricity, fuel;

p) equipment — general term encompassing machines, tool, and the manual-labor-capability of a worker;

q) actual equipment — equipment (as defined in (p) above) used in a production activity;

r) tool — attached or detached device used with equipment, e.g. metallic mold;

s) invest — name of a MDD listing items (see (a) above) and their quantity that are an input to an operation method (see (i) above);

t) produce — name of a MDD listing items (see (a) above) and their quantity that is the output of an operation method (see (i) above);

u) consume — name of a MDD listing the utilities and their quantity consumed in an operation method (see (i) above);

v) assign — name of a MDD listing the actual equipment (i.e. the equipment assigned to an operation method; see (p) and (q) above), and the quantity and capacity of this equipment, for an operation method.

Figure B.1 contains two stereotype classes, <<plan&result>> and <<history>>. The structure of the <<plan&result>> stereotype class is shown in Figure B.2. The structure of the <<history>> stereotype class is shown in Figure B.3.

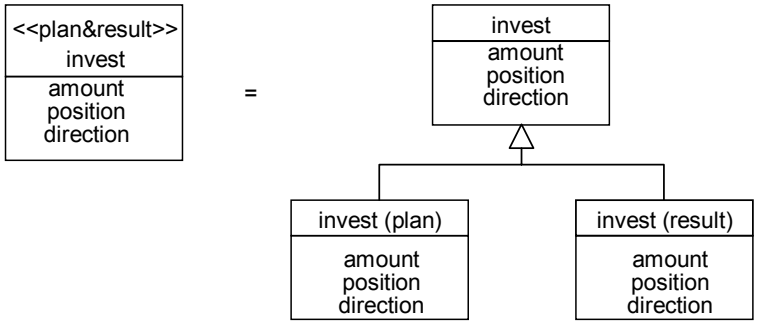

Figure B.2 — Meaning of stereotype <<plan&result>>

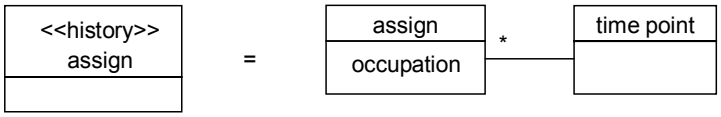

Figure B.3 — Meaning of stereotype <<history >>

B.2 Example of an MSU capability profile

B.2.1 Example of an activity tree for an MES package

Figure B.4 shows an example of a manufacturing activity tree for a MES package developed by a software vendor. This MES package is decomposed into seven activities of a production operation management activity model by applying the MDM shown in Figure B.1. The seven activities have been further decomposed into sub-activities through an analysis of the function of each activity. Table B.1 lists these activities and sub-activities, along with the MDDs and actions of each sub-activity.

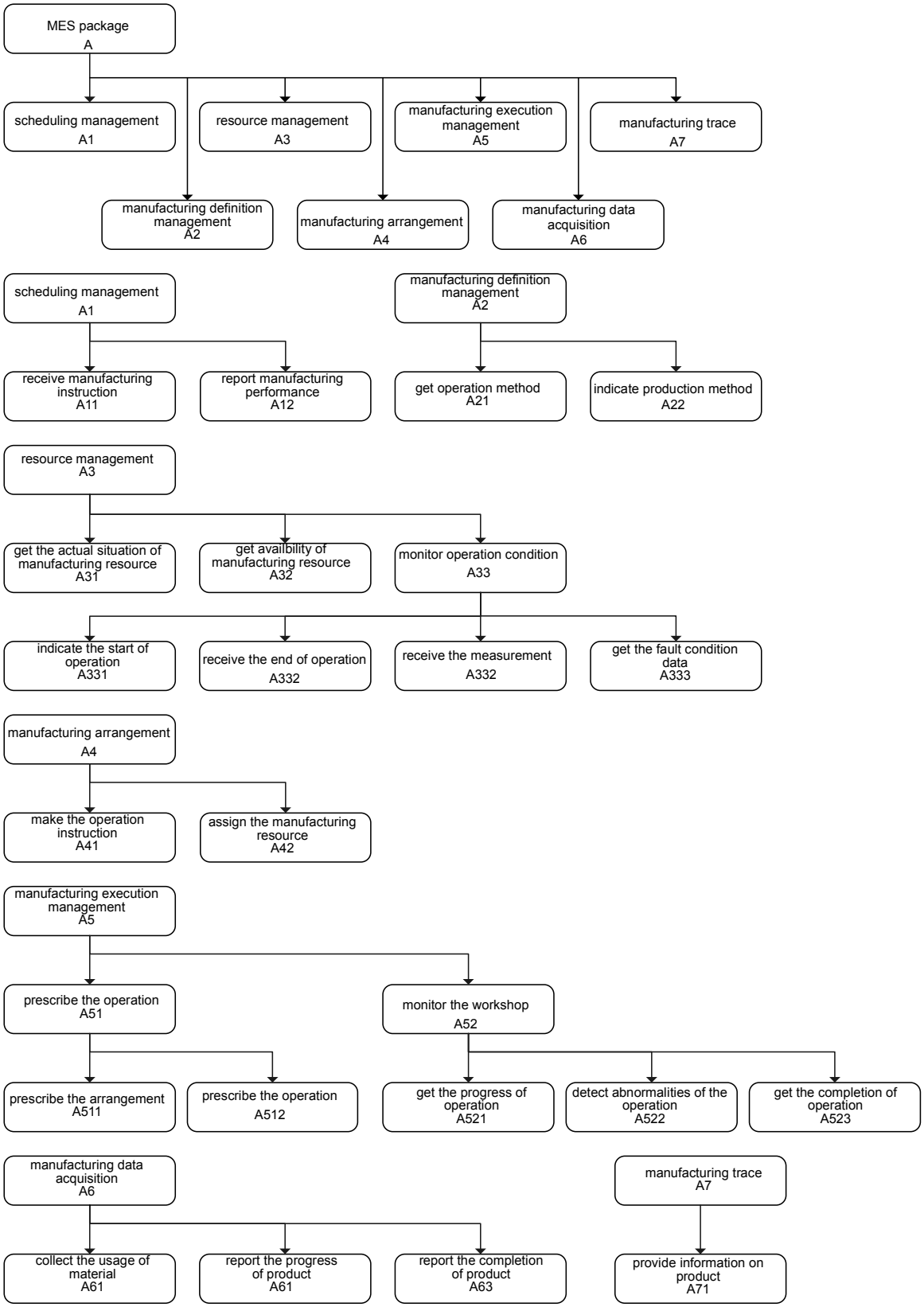

Figure B.4 — Activity tree of MES package A

Table B.1 — Decomposition of MES package A (see Figure B.4)

ID	Activity Name	Associated MDD	Action
A1	scheduling management		
A11	receive manufacturing instruction	product order (plan)	Get
		operating instruction (plan)	Get
		item	Get
A12	report manufacturing performance	product order (result)	Set
		operating instruction (result)	Set
		invest (result)	Set
		produce (result)	Set
		substance	Set
A2	manufacturing definition management		
A21	get operation method	operation type	Set
		operation method (plan)	Get
		recipe (plan)	Get
A22	indicate production method	operation type	Set
		operation method (plan)	Set
		recipe (plan)	Set
A3	resource management		
A31	get the actual situation of manufacturing resource	utilities	Get
		actual equipment	Get
		factory	Get
A32	get availability of manufacturing resource	actual equipment	Set
		state (plan)	Get
A33	monitor operation condition	actual equipment	Set
		state (result)	Get
A331	indicate the start of operation	operation method (result)	Set
		assign (result)	Set
		actual equipment	Set
A332	receive the end of operation	operation method (result)	Get
		assign (result)	Get
		actual equipment	Get
		state (result)	Get
A333	receive the measurement data	operation method (result)	Get
		produce (result)	Get
		substance	Get
		quality value (result)	Get
A334	get the fault condition	operation method (result)	Get
		assign (result)	Get
		actual equipment	Get

Table B.1 (*continued*)

ID	Activity Name	Associated MDD	Action
A4	manufacturing arrangement		
A41	make the operating instruction	operating instruction (plan)	Set
		operation method (plan)	Set
		item	Set
		invest (plan)	Set
		produce (plan)	Set
A42	assign the manufacturing resource	operating instruction (plan)	Set
		operation method (plan)	Set
		assign (plan)	Set
		equipment	Set
		actual equipment	Set
A5	manufacturing execution management		
A51	prescribe the operation	operating instruction (result)	Set
		operation method (result)	Set
		invest (result)	Set
		produce (result)	Set
		substance	Set
A511	prescribe the arrangement	operating instruction (result)	Set
		operation method (result)	Set
		recipe (result)	Set
A512	prescribe the begin of operation	operating instruction (result)	Set
		operation method (result)	Set
A52	monitor the workshop	shop	Get
A521	get the progress of operation	operation method (plan)	Get
		operation method (result)	Get
		invest (plan)	Get
		produce (plan)	Get
		item	Get
		invest (result)	Get
		produce (result)	Get
		substance	Get
A522	detect abnormalities of the operation	operation method (result)	Get
A523	get the completion of operation	operation method (plan)	Get
		operation method (result)	Get
		produce (plan)	Get
		item	Get
		produce (result)	Get
		finish product	Get
A6	manufacturing data acquisition		

Table B.1 (*continued*)

ID	Activity Name	Associated MDD	Action
A61	collect the usage of material	operating instruction (result)	Set
		invest (result)	Get
		substance	Get
		stock	Get
A62	collect the progress of product	operating instruction (result)	Set
		produce (result)	Get
		substance	Get
		work-in-progress	Get
A63	report the completion of product	product order (result)	Set
		operating instruction (result)	Set
		invest (result)	Get
		produce (result)	Get
		substance	Get
		quality value	Get
		finish product	Get
A7	manufacturing trace		
A71	provide information on product	substance	Get
		item	Get
		quality value	Get

B.2.2 XML description of example MSU capability profile

The following XML description shows the capability profile of activity A11 in Table B.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<CapabilityProfiling xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNaMESpaceSchemaLocation="C:\ISO16100\Capability_Template.xsd">
    <type id="MES SW-A011"/>
    <CapabilityProfile date="2006-02-01">
        <pkgtype version="1.0.0"/>
        <Common>
            <MSU_Capability ID="MSU81-0001">
                <id="pilot_only"/>
            </MSU_Capability>
            <ReferenceCapabilityClassStructure/>
            <TemplateID id="A11"/>
            <Capability_Class_Name name="A11_ReceiveOrder_Activity"/>
            <Reference_Capability_Class_Structure_Name name="MESSW_Structure"/>
            <Version major="1" minor="1"/>
            <Owner>
                <name>MES Product Inc.</name>
                <city>Tokyo</city>
                <country>Japan</country>
            </Owner>
            <ReferenceDictionaryName/>
            <NumberOfProfileAttributes/>
            <NumberOfMethods/>
            <NumberOfResources/>
            <NumberOfConstraints/>
            <NumberOfExtensions/>
            <NumberOfLowerLevels/>
            <NumberOfSubtemplatesAtNextLowerLevel/>
        </Common>
        <Specific>
```

```
            <Reference_MDM_Name domain_name="MESX Domain Conceputual Model"/>
            <MDD_Description_Format format_name="List_Of_MDD_Objects"/>
            <MDD Description>
                <List_Of_MDD_Objects>
                    <MDD_Name name="product order (plan)" action="Get"/>
                    <MDD_Name name="operating instruction (plan)" action="Get"/>
                    <MDD_Name name="item" action="Get"/>
                </List_of_MDD_Objects>
            </MDD Description>
        </Specific>
    </CapabilityProfile>
</CapabilityProfiling>
```

B.3 Example of a required capability profile

B.3.1 Example of an activity tree for a system requested specification

Figure B.5 shows an example of a manufacturing activity tree for a MES package required by a system integrator. The required MES package is decomposed into six activities of a production operation management activity model by applying the MDM model in Figure B.1. The six activities have been further decomposed into sub-activities through an analysis of the function of each activity. Table B.2 lists these activities and sub-activities, along with the MDDs and actions of each sub-activity.

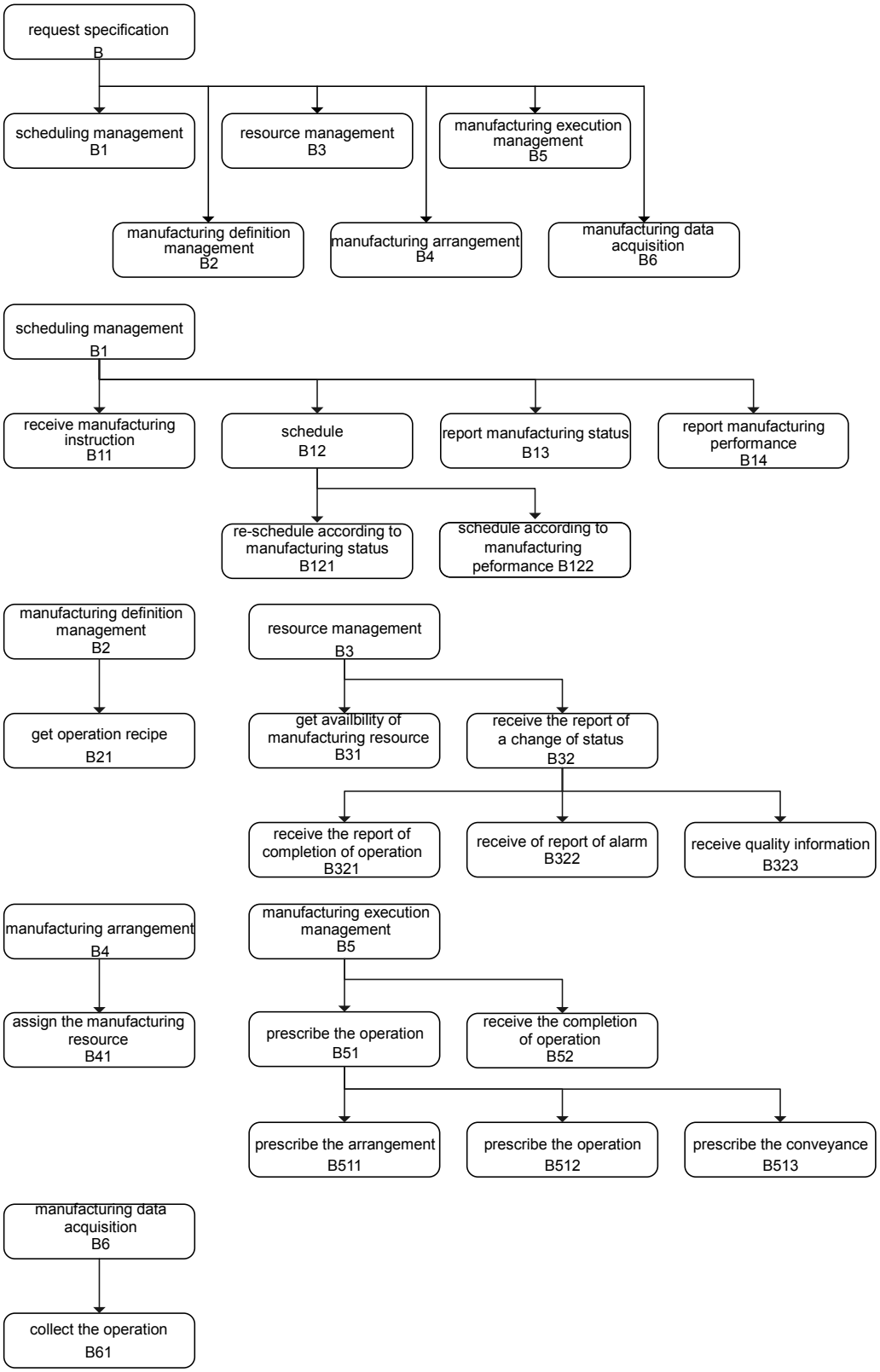

Figure B.5 — Activity tree of MES package B

Table B.2 — Decomposition of MES package B (see Figure B.5)

ID	Activity Name	Referred MDD	Action
B1	scheduling management		
B11	receive manufacturing instruction	product order (plan)	Get
		operating instruction (plan)	Get
		item	Get
B12	schedule	operating instruction (plan)	Set
		operating method (plan)	Set
		assign (plan)	Set
		equipment	Set
B121	re-scheduling according to manufacturing status	operating instruction (plan)	Set
		operating method (result)	Get
		operating method (plan)	Set
		assign (result)	Set
		equipment	Set
B122	re-scheduling according to manufacturing performance	operating instruction (plan)	Set
		operating method (plan)	Set
		assign (result)	Set
		equipment	Set
		product order (result)	Get
		produce (result)	Get
		substance	Get
B13	report manufacturing status	operating instruction (result)	Set
		operation method (result)	Set
B14	report manufacturing performance	product order (result)	Set
		operating instruction (result)	Set
		invest (result)	Set
		produce (result)	Set
		substance	Set
B2	manufacturing definition management		
B21	get operation recipe	operation method (plan)	Set
		recipe (plan)	Get
B3	resource management		
B31	get availability of manufacturing resource	actual equipment	Set
		state (plan)	Get
B32	receive the report of a change of status	actual equipment	Get
		state (result)	Get
B321	receive the report of completion of operation	operation method (result)	Get
		assign (result)	Get
		actual equipment	Get
		state (result)	Get

Table B.2 (*continued*)

ID	Activity Name	Referred MDD	Action
B322	receive the report of alarm	actual equipment	Get
		state (result)	Get
B323	receive quality information	product order (result)	Get
		operating instruction (result)	Get
		operation method (result)	Get
		produce (result)	Get
		substance	Get
		quality value (result)	Get
B4	manufacturing arrangement		
B41	assign the manufacturing resource	operation method (plan)	Get
		assign (plan)	Set
		actual equipment	Set
		recipe (plan)	Set
B5	manufacturing execution management		
B51	prescribe the operation	operating instruction (plan)	Get
		operation method (plan)	Set
		invest (plan)	Set
		produce (plan)	Set
		item	Set
B511	prescribe the arrangement	operation method (plan)	Get
		invest (plan)	Set
		item	Set
B512	prescribe the operation	operation method (plan)	Get
		invest (result)	Set
		produce (result)	Set
		substance	Set
B513	prescribe the conveyance	operation method (plan)	Get
		assign (result)	Set
		actual equipment	Set
		substance	Set
B52	receive the completion of operation	product order (result)	Get
		operating instruction (result)	Get
		operation method (result)	Get
B6	manufacturing data acquisition		
B61	collect the operation performance	product order (result)	Set
		operating instruction (result)	Set
		operation method (result)	Get
		produce (result)	Get
		finish product	Get

B.3.2 Example of a requested capability profile

The following XML description shows the capability profile of activity B11 in Table B.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<CapabilityProfiling xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNaMESpaceSchemaLocation="C: \ISO16100\Capability_Template.xsd">
    <type id="Requirement Profile"/>
    <CapabilityProfile date="2006-02-10">
        <pkgtype version="V01.01.03"/>
        <Common>
            <Requirement ID="SYS-Req2006-0001">
                <id="production_ready"/>
            </Requirement>
            <ReferenceCapabilityClassStructure/>
            <Template_ID id="B11"/>
            <Capability_Class_Name name="B11_ReceiveOrder_Activity"/>
            <Reference_Capability_Class_Structure_Name name="REQ_Structure"/>
            <Version major="1" minor="1"/>
            <Owner>
                <name>MES User Inc.</name>
                <city>SoftCity</city>
                <state>Alabama</state>
                <country>USA</country>
            </Owner>
            <ReferenceDictionaryName/>
            <NumberOfProfileAttributes/>
            <NumberOfMethods/>
            <NumberOfResources/>
            <NumberOfConstraints/>
            <NumberOfExtensions/>
            <NumberOfLowerLevels/>
            <NumberOfSubtemplatesAtNextLowerLevel/>
        </Common>
        <Specific>
            <Reference_MDM_Name domain_name="MESX Domain Conceptual Model"/>
            <MDD_Description_Format format_name="List_of_MDD_Objects"/>
            <MDD Description>
                <List_of_MDD_Objects>
                    <MDD_Name name="product order (plan)" action="Get"/>
                    <MDD_Name name="operating instruction (plan)" action="Get"/>
                    <MDD_Name name="item" action="Get"/>
                </List_of_MDD_Objects>
            </MDD Description>
        </Specific>
    </CapabilityProfile>
</CapabilityProfiling>
```

B.4 Matching example using a Type 2 Matcher

A system integrator compares his required capability profiles (see B.3 for an example) to existing MSU capability profiles (see B.2 for an example) using the procedure detailed in clause 7. The Type 2 Matcher compares the MDD names and actions associated with each sub-activity. For each match of a sub-activity, the Type 2 Matcher then evaluates the hit rate, i.e. the percent of MDD names and actions in the required profile matched to MDD names and actions in the MSU's profile.

Figure B.6 and Table B.3 show an example of the Type 2 Matcher's evaluation. In Figure B.6, the Type 2 Matcher has matched activity B61 and activity A63, both having a MDD name of “product order (result)” and an action name of “Set”. The Type 2 Matcher calculates the hit rate as 4 of 5 MDD names and action names for activity B61 in the required capability profile -- i.e. a 80% rate. The Matcher calculates the hit rates of all the activities in the required capability profile that were matched to an activity in the MSU's profile and generates a report as shown in Table B.3. The Matcher also calculates the matching level (see 7.2) for the activities in the required capability profile and indicates by a shading the report cell if a "All Mandatory Match" was achieved. In the example of activity B61, even though there was only a 80% hit rate, the 4 MDD names and action names relate to mandatory functions. Thus, the Matcher shades the B61-A63 cell in Table B.3.

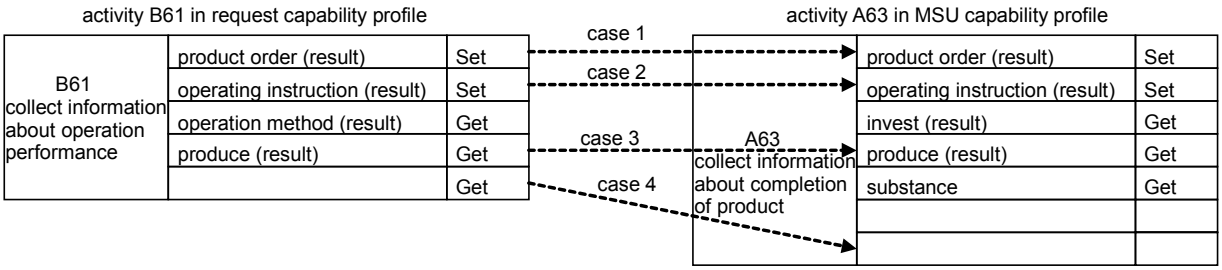

Figure B.6 — Example of profile matching using a Type 2 Matcher

Table B.3 — Example of profile matching results using a Type 2 Matcher

Required Activities that Matched MSU Activities	MSU Activities (MES Package A)														
	A11	A12	A21	A22	A31	A32	A33	A41	A42	A51	A511	A512	A61	A62	A63
B11	100	0	--	--	--	--	--	--	--	--	--	--	--	--	--
B12	0	0	--	--	0	0	0	50	100	--	--	--	--	--	--
B121	--	--	--	--	--	--	--	40	40	--	--	--	--	--	--
B122	--	--	--	--	--	--	--	28	42	--	--	--	--	--	--
B13	0	50	--	--	--	--	--	--	--	--	--	--	--	--	--
B14	0	100	--	--	--	--	--	--	--	--	--	--	--	--	--
B21	--	--	100	50	--	--	--	--	--	--	--	--	--	--	--
B31	--	--	--	--	0	100	50	--	--	--	--	--	--	--	--
B32	--	--	--	--	50	50	50	--	--	--	--	--	--	--	--
B321	--	--	--	--	25	0	50	--	--	--	--	--	--	--	--
B322	--	--	--	--	50	50	50	--	--	--	--	--	--	--	--
B323	--	--	--	--	0	0	0	--	--	--	--	--	--	--	--
B41	--	--	--	--	--	--	--	25	50	--	--	--	--	--	--
B51	--	--	--	--	--	--	--	--	--	0	0	0	--	--	--
B511	--	--	--	--	--	--	--	--	--	0	0	0	--	--	--
B512	--	--	--	--	--	--	--	--	--	75	0	0	--	--	--
B513	--	--	--	--	--	--	--	--	--	50	0	0	--	--	--
B52	--	--	--	--	--	--	--	--	--	0	0	0	--	--	--
B61	--	--	--	--	--	--	--	--	--	--	--	--	20	40	80

When a system integrator analyzes Table B.3, he will, by the shaded report cells, see that seven MSUs have achieved an "All Mandatory Match" level. Thus, these seven MSUs are candidates for use (and possible reuse) by the system integrator to fulfill his requirements, such as for the MES Package B in B.3.

Figure B.7 shows a manufacturing activity tree for the MES Package B that applies the usable MSUs shown in Table B.3.

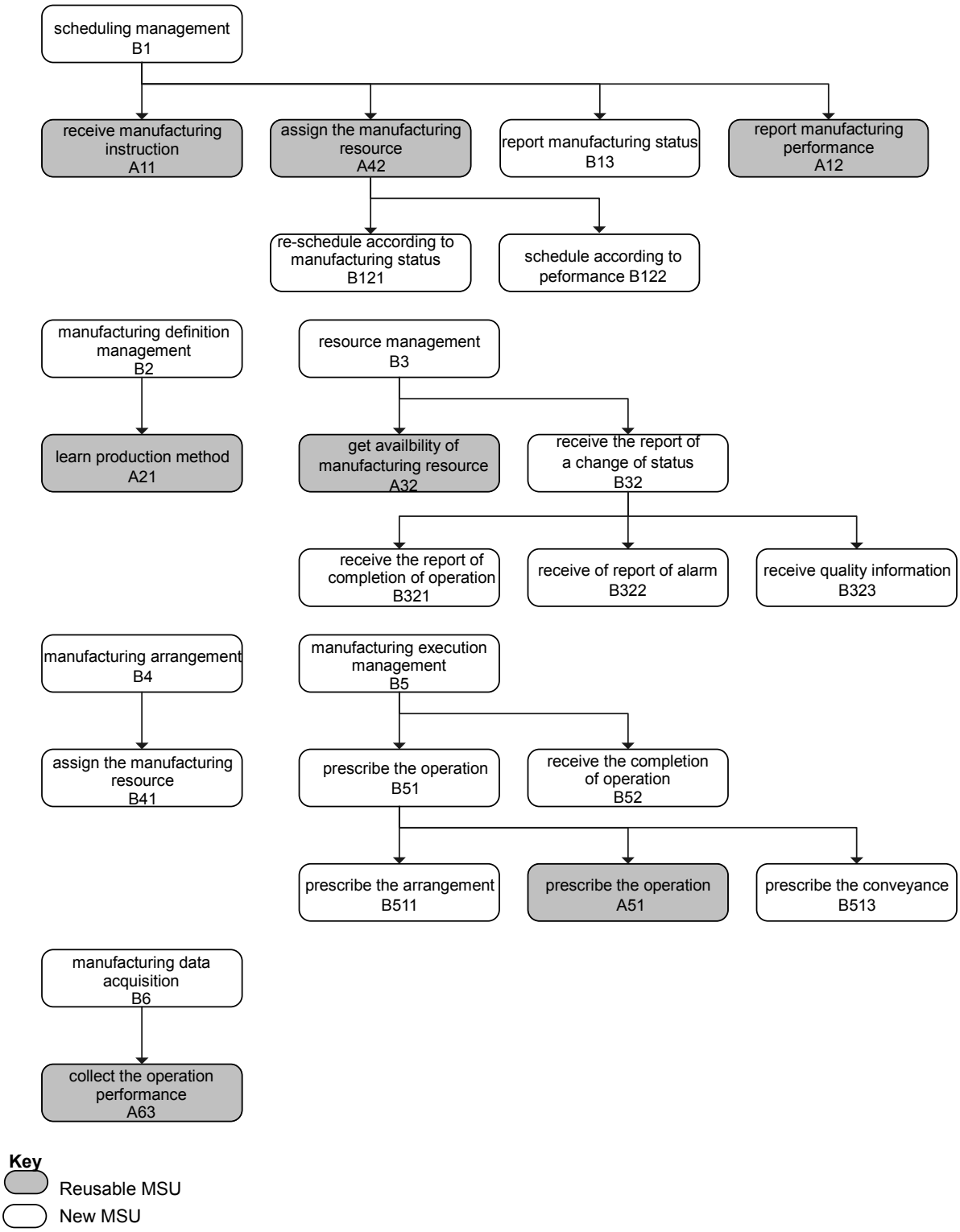

Figure B.7 — Example of an activity tree for a system requested specification showing reusable and new MSUs

Bibliography

[1] ISO 15745-1, *Industrial automation systems and integration — Open systems application integration framework — Part 1: Generic reference description*

[2] ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

[3] IEC 62264-3, *Enterprise-control system integration — Part 3: Activity models of manufacturing operations management*

ICS 25.040.01

Price based on 50 pages