INTERNATIONAL STANDARD

ISO 16100-1

First edition
2002-12-15

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

Part 1: Framework

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

Partie 1: Cadre

Reference number
ISO 16100-1:2002(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

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 electro-technical 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 was prepared by Technical Committee ISO/TC 184, *Industrial 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 protocols and templates*

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

Introduction

The motivation for ISO 16100 stems from the industrial and economic environment noted by ISO/TC 184/SC5. In particular, there is broad recognition by industry that application software and the expertise to apply that software are assets of the enterprise. Industry feedback has noted the need for improvement and continued development of current design and manufacturing standards to enable software interoperability.

ISO 16100 specifies a manufacturing information model that characterizes software-interfacing requirements. With interfacing requirements clearly expressed, standard interfaces can be more easily and quickly developed using the Interface Definition Language (IDL) or an appropriate programming language, such as Java and C++. These standard interfaces are expected to enable the interoperability among manufacturing software tools (modules or systems).

The Unified Modelling Language (UML) is used in this International Standard for modelling these interfaces. Also, the manufacturing information model can be used to develop commonly sharable database schema using languages such as the eXtensible Markup Language (XML).

Sectors of the manufacturing industry — such as automotive, aerospace, machine tool manufacturing, computer peripheral manufacturing, and mold and die manufacturing — that intensively use computer-aided design (CAD), computer-aided manufacturing (CAM), numerical control (NC) programming, computer-aided engineering (CAE), product data management (PDM), and manufacturing execution systems (MES) will directly benefit from ISO 16100. The software interface requirements in ISO 16100 will facilitate the development of:

a) interoperable design and manufacturing software tools leading to shortened product development time;

b) new software tools that can be easily integrated with current technologies leading to more choices in the market;

c) new application software leading to reduced capital expenditures to replace legacy systems;

d) programming interfaces and database schema leading to cost savings by not having to develop proprietary interfaces for point-to-point software integration.

The end result will be a reduction in product and manufacturing information management cost and lower product costs.

ISO 16100 enables 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;

c) conformance tests that ensure the integrity of the software integration.

ISO 16100 consists of four parts. Part 1 specifies a framework for interoperability of a set of manufacturing software products used in the manufacturing domain and its integration into a manufacturing application. Part 2 specifies a methodology for constructing profiles of manufacturing software capabilities, and includes a methodology for creating manufacturing software capability profiles as well as for using these profiles at the developing stage of manufacturing applications. Part 3 specifies the interface protocol and templates for various manufacturing application areas. Part 4 specifies the concepts and rules for the conformity assessment of the other parts of ISO 16100.

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

Part 1: **Framework**

1 Scope

Part 1 of ISO 16100 specifies a framework for the interoperability of a set of software products used in the manufacturing domain and to facilitate its integration into a manufacturing application (see Annex A for a discussion of a manufacturing application). This framework addresses information exchange models, software object models, interfaces, services, protocols, capability profiles, and conformance test methods.

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 15704, *Industrial automation systems — Requirements for enterprise-reference architectures and methodologies*

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

ISO/IEC 19501-1, *Information technology — Unified Modelling Language (UML) — Part 1: Specification*

IEC 62264-1, *Enterprise-Control System Integration — Part 1: Models and Terminology*

IEEE 1320.1-1998, *Standard for Functional Modelling Language — Syntax and Semantics for IDEF0*

W3C Recommendation Feb 1998, *Extensible Markup Language (XML) 1.0*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. Other relevant terms are defined in Annex D.

3.1
advanced planning
production planning over time horizons of months or years using constraint models that treat both materials and capacity

NOTE In some cases, the planning system includes master production scheduling, material requirements planning, or capacity planning.

3.2
CAD/PDM
computer systems that are used for product design and modelling, engineering, product data management, and process data management

3.3
capability
<software> set of functions and services with a set of criteria for evaluating the performance of a capability provider

NOTE This definition differs from that given in ISO 15531-1 and ISO/DIS 19439, where capability is defined as the quality of being able to perform a given activity. See IEC 62264-1 for a general definition of capability.

3.4
capability profiling
selection of a set of offered services defined by a particular interface within a software interoperability framework

3.5
CAPP/CAM
computer systems that are used for process planning and programming of numerically controlled machines

3.6
controller
<digital systems> hybrid hardware/software systems that are used for controlling machines

EXAMPLES Distributed control systems (DCS), programmable logic controllers (PLC), numerical controller (NC), and supervisory control and data acquisition (SCADA) systems.

3.7
data collection
gathering of information on workpieces, timing, personnel, lots, and other critical entities for production management in a timely manner

3.8
design knowledge
rules and logic that a human designer brings to bear on design problems, including design and implementation techniques

NOTE Many different types of design knowledge are used in different design activities, such as decomposition knowledge, assignment knowledge, consolidation knowledge, and optimization knowledge.

3.9
design pattern
medium-scale patterns, smaller in scale than architectural patterns, but are at a higher level than the programming language-specific idioms[1)]

NOTE The application of a design pattern has no effect on the fundamental structure of a software system, but may have a strong influence on the architecture of a subsystem.

3.10
manufacturing software
type of software resource within an automation system that provides value to a manufacturing application (e.g. CAD/PDM) by enabling the flow of control and information among the automation system components involved in the manufacturing processes, between these components and other enterprise resources, and between enterprises in a supply chain or demand chain

3.11
manufacturing software component
class of manufacturing software resource intended to support the execution of a particular manufacturing task

---

1) Taken from *Pattern-oriented Software Architecture*, John Wiley & Sons, June 2000.

3.12
manufacturing software unit
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

NOTE A software unit can be modeled using UML as a software object.

3.13
manufacturing system
system coordinated by a particular information model to support the execution and control of manufacturing processes involving the flow of information, material, and energy in a manufacturing plant

3.14
manufacturing software capability
set of manufacturing software functions and services against a set of criteria for evaluating performance under a given set of manufacturing conditions

NOTE See Annex C for use cases and related scenarios involving manufacturing software capability.

3.15
manufacturing software capability profile
concise representation of a manufacturing software capability to meet a requirement of a manufacturing application

3.16
software architecture
fundamental organization of a software system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution

[IEEE 1471-2000]

3.17
software environment
other manufacturing resources within the computing system that affect the operational aspects of the manufacturing software unit

NOTE The software environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. The environment determines the boundaries that define the scope of the system of interest relative to other systems.

3.18
supply chain planning
usage of information technology to address planning and logistics problems at different levels and granularities of detail using models for a product line, a production plant, or a full chain of multiple demand sources, suppliers, production plants, and distribution means

NOTE Supply chain planning can be used to synchronize production, balancing constraints based on goals including on-time delivery, minimal inventory, and maximum profit.

4 Abbreviations

AGV Automatic Guided Vehicle

APT Automated Programmed Tool

BOM Bill of Materials

CAD Computer Aided Design

CAM Computer Aided Manufacturing

CAPP Computer Aided Process Planning

ERP Enterprise Resource Planning

MES Manufacturing Execution System

NC Numerical Control

PDM Product Data Management

SCM Supply Chain Management

SCADA Supervisory Control and Data Acquisition

SQC Statistical Quality control

XML eXtensible Markup Language

UML Unified Modelling Language

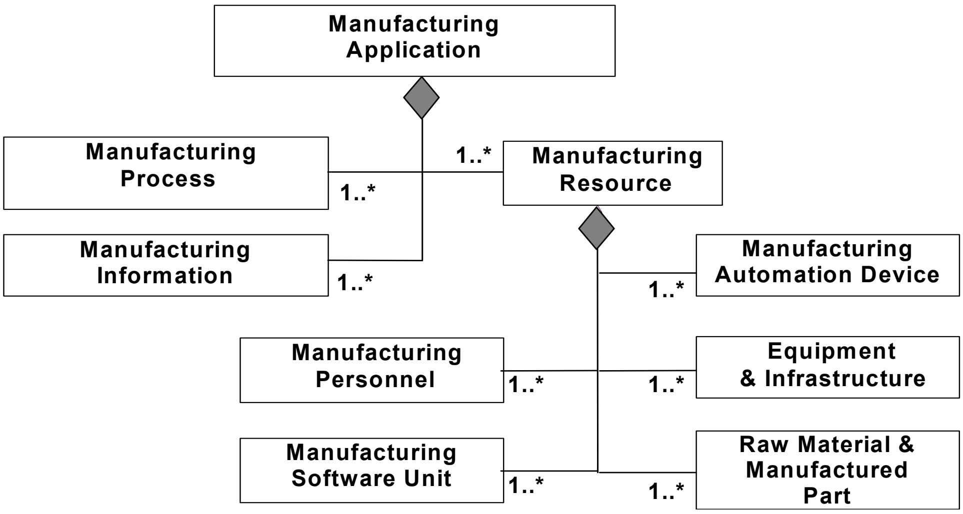
5 Manufacturing application

5.1 Reference application framework

The interoperability framework for manufacturing software is based upon a more general interoperability framework for manufacturing applications. Such an application interoperability framework, which is explained in further detail in ISO 15745-1, provides a basis for integrating an automation and control system architecture within a manufacturing application architecture.

An integrated manufacturing application shall be 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, as shown in Figure 1. Manufacturing resources are communication networks, devices, software, equipment, material, and personnel necessary to support the processes and information exchanges required by the application.

In this application integration model, the various elements of the model have shared interfaces and exchange material, energy, and information in a cooperative and coordinated manner. The manufacturing processes can cooperate with each other if the functions performed by the various elements of the model can interoperate with each other. When software units perform some of these functions, it is necessary for the software units to be interoperable with the other elements, as well as with each other.



NOTE Boxes represent classes of objects (things). Lines connecting boxes represent associations between objects (things). An association has two roles (one in each direction). A role may optionally be named by a label. A role from A to B is closest to B, and vice versa. Roles are one-to-one unless otherwise noted. A role can have a multiplicity, e.g. a role marked with "1..*" is used to denote *many* as in a one-to-many or many-to-many association. A diamond at the end of an association line denotes a *part-of* relationship. A black diamond at the end of an association line denotes a *composition aggregation* relationship. For example, Manufacturing Application owns (is comprised of) Manufacturing Process, Manufacturing Information, and Manufacturing Resources. This notation is taken from ISO/IEC 19501-1.

Figure 1 — Class diagram of a partial model of a manufacturing application

5.2 Manufacturing domain

The manufacturing domain that includes discrete, batch, and continuous control encompasses many types of industries. The automotive industry is an example of an industry employing discrete control; the pharmaceutical industry is an example of an industry employing batch control; the petrochemical industry is an example of an industry employing continuous control. For manufacturing software, the interface between plant management systems and floor control systems is described by the same method regardless of whether control systems are discrete, batch, or continuous. Similarly, the control flow inside a control system is also described by the same method regardless of whether the system is discrete, batch, or continuous.

Even as the manufacturing domain applies to many industries, the relationship between firms in these industries is changing rapidly due to recent developments in IT infrastructure, as is the case in supply chain management systems. Therefore, ISO 16100 sets a target manufacturing domain to include the manufacturing operation and control activity, the discrete control activity, the batch control activity, the continuous control activity, and the manufacturing process design activity, as shown in Figure 2.

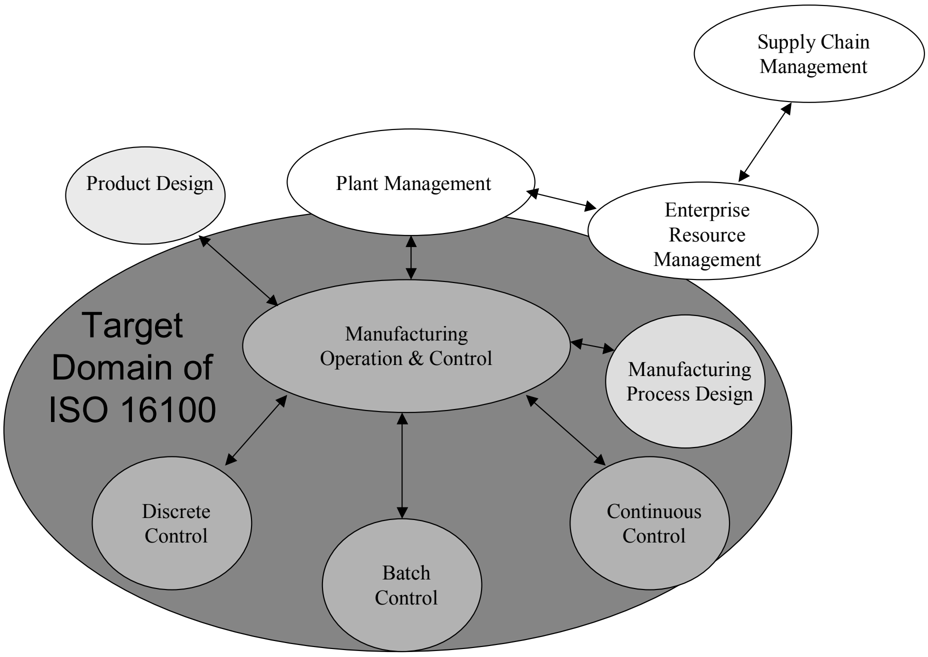


Figure 2 — Target domain of ISO 16100

5.3 Manufacturing processes

A manufacturing process shall be modeled as a set of activities that follow a specific sequence. Each activity shall be associated with a set of functions performed according to a time schedule or triggered by a set of events.

The functions associated with a manufacturing process shall be viewed as being implemented through a set of manufacturing resources. The manufacturing resources shall be considered to be selected and configured to support the material, information, and energy flows required by the specified sequence of manufacturing activities associated with a process.

When a manufacturing process must cooperate and coordinate with another process, the respective functions of these interacting processes are considered to be able to cooperate and coordinate with each other. Such a situation requires that the cooperating and coordinating functions meet a common set of criteria and a set of conditions for interoperability. The software units that implement these functions shall meet a related set of criteria and conditions for interoperability.

5.4 Manufacturing resources

The manufacturing resources required by a manufacturing application shall be organized in terms of the type of flow being managed and supported among the manufacturing processes — material, control, information, or energy flow. The set of integrated flows can be used to represent an integrated manufacturing application or manufacturing system architecture.

The set of integrated manufacturing resources shall form a manufacturing system architecture that fulfills a set of manufacturing application requirements. These manufacturing resources, including the manufacturing software units, shall provide the functions associated with the manufacturing processes.

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.

An operating environment shall be distinguished by the manufacturing resources needed by the associated set of software units. These manufacturing resources include the processing, storage, user interface, communications, and peripheral devices, as well as other system software required for executing the software units.

5.5 Manufacturing information

A set of information structures shall provide the knowledge infrastructure to manage the various types of flows within a manufacturing application. These information sets shall include data pertaining to the product, the process, and the equipment.

The manufacturing software units shall be the primary means for handling, transforming, and maintaining these information structures.

6 Manufacturing software interoperability framework

6.1 Manufacturing software unit interoperability

Within a context of a manufacturing application, a manufacturing software unit is considered to be capable of performing a specific set of functions defined by a manufacturing system architecture. In performing these sets of functions, the manufacturing software unit is cooperating and conducting transactions with other manufacturing software units.

The functions performed by each software unit shall be those as described by the manufacturing application architecture. The information exchanged between these software units shall enable the coordinated execution of these manufacturing functions.

The software interoperability of a set of manufacturing activities shall be described in terms of the interoperability of the set of software units associated with each manufacturing activity.

A 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 shall 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 application.

A software interoperability framework shall be based on the following aspects:

a) syntax and semantics shared between manufacturing software units;

b) functional relationships between the manufacturing software units;

c) services, interfaces, and protocols offered by the manufacturing software units;

d) ability to provide manufacturing software unit capability profiling.

The framework elements shall consist of the roles, the activities, and the artifacts associated with the software entities when dealing with the manufacturing process, information, and resources. The framework rules shall address the relationships, templates, and conformance statements needed to construct a capability class (see ISO 16100-2), a profile class (see ISO 16100-2), and a component class (see ISO 16100-3).

The organization, relationships, and tasks pertaining to the software unit and its manufacturing software components shall be expressed in terms of the framework elements and rules in ISO 16100-3.

Figure 3 shows the relationships between the aspects of the software interoperability framework and the derivation of this framework from a generic application interoperability framework.

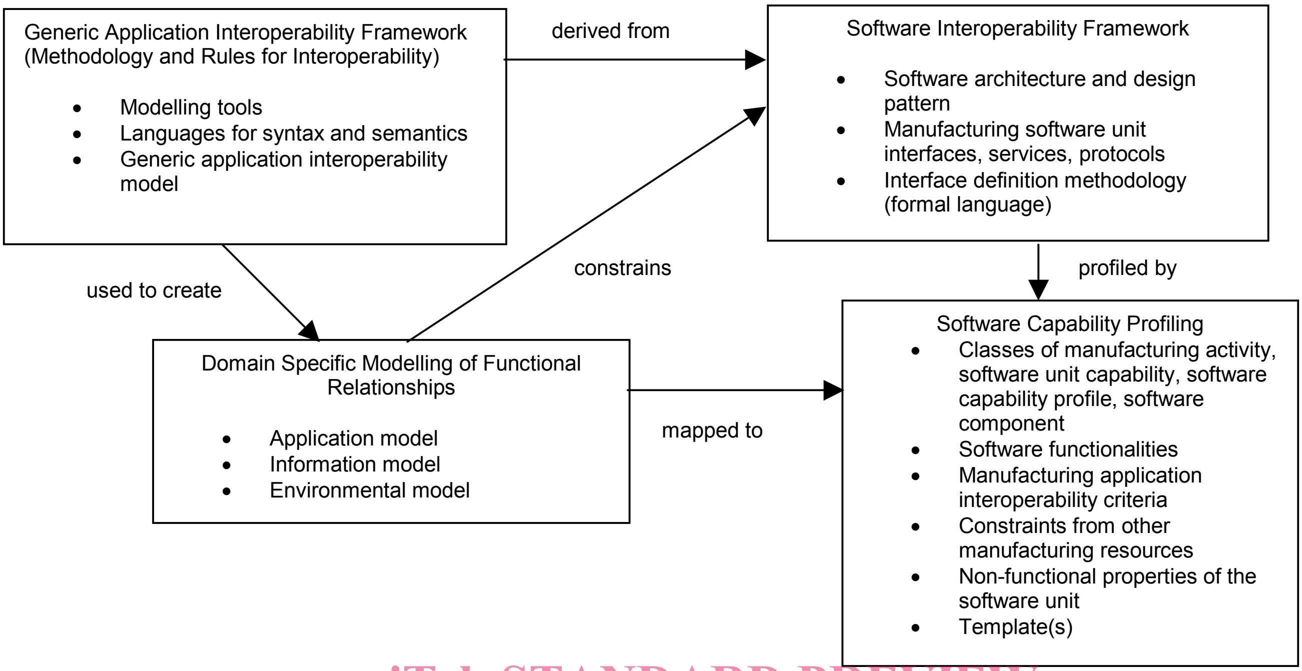


Figure 3 — Relationships of software interoperability aspects

6.2 Functional relationships between the manufacturing software units

Within the manufacturing domain shown in Figure 2, there can be one or more operational software units that cooperate through a specific interface/protocol to perform a single manufacturing function required in that domain. This is realized in the software environment of a specific computing system as one of the components of the manufacturing resources, enabled by a specific software design pattern performing a specific role. Conversely, a single software unit can perform one or more manufacturing functions. One or more manufacturing functions can interoperate with each other to execute, control, monitor, or manage a particular manufacturing activity. A series of activities can be conducted in a particular sequence to complete a manufacturing process. Figure 4 depicts the classes of a software unit and its surroundings and associations.

In this framework, the sequence and schedule of functions performed is determined by the sequence and schedule of the activities that comprise a particular process. The manufacturing software units deployed to perform the functions are considered to execute according to the required sequence and schedule of their associated functions.

The interoperability of the manufacturing processes shall be viewed in terms of the interoperability of the functions, which in turn, shall be viewed in terms of the interoperability of the manufacturing resources, including the manufacturing software units. Examples of information flow among design, manufacturing planning, and execution activities are provided in Annex B.

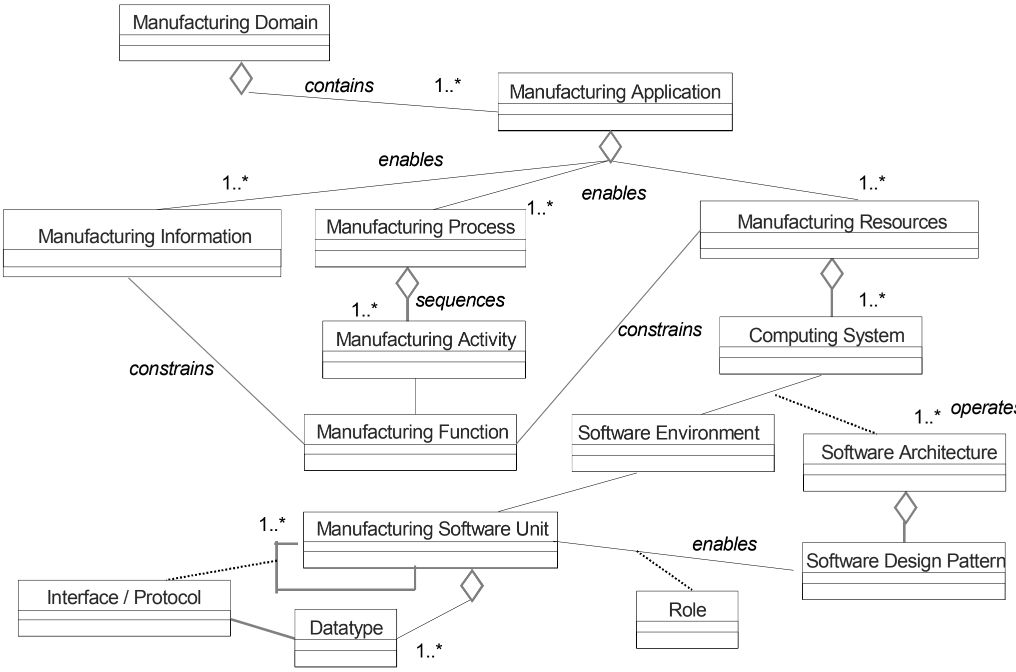


Figure 4 — Class diagram of a software unit and its surroundings and associations within a manufacturing application

A software unit shall be modeled as a set of software components that have been linked to perform a definite manufacturing function. Each software unit shall be represented as a UML object.

A manufacturing software unit shall provide a service interface for use in its configuration, execution, and maintenance.

The capability of a software unit to perform a manufacturing function shall include a description of the set of services available at its service interface. The capability of a manufacturing software unit shall be concisely stated in a capability profile described in XML.

The sequence and timing of the manufacturing activities determines the specified criteria for the interoperability of the associated set of manufacturing software units.

Information structures included or referenced in a capability profile are defined in ISO 16100-2.

6.3 Services, interfaces, and protocols

A manufacturing software unit shall be modeled as a set of manufacturing software components that have been linked to perform a definite manufacturing function.

Manufacturing software units shall interoperate with one another, in support of a manufacturing activity, when the services requested by the former can be provided by the latter, using the same operating environment.

The services, interfaces, and protocols are defined in ISO 16100-3.