



# 中华人民共和国国家标准

GB/T 19902.2—2005/ISO 16100-2:2003

---

## 工业自动化系统与集成 制造软件互操作 性能力建规 第2部分：建规方法论

Industrial automation systems and integration—Manufacturing software  
capability profiling for interoperability—Part 2: Profiling methodology

(ISO 16100-2:2003, IDT)

2005-09-12 发布

2006-04-01 实施

---

中华人民共和国国家质量监督检验检疫总局 发布  
中国国家标准化管理委员会

## 目 次

前言 .....	I
引言 .....	II
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
4 缩略语 .....	2
5 能力建规的方法 .....	3
6 能力建规的元素和规则 .....	5
7 一致性 .....	11
附录 A (资料性附录) 参考方法 .....	12
参考文献 .....	14

## 前 言

GB/T 19902《工业自动化系统与集成 制造软件互操作性能力建规》由 4 部分组成:

- 第 1 部分:框架;
- 第 2 部分:建规方法论;
- 第 3 部分:接口协议及模板;
- 第 4 部分:一致性测试方法、标准及报告。

本部分为 GB/T 19902 的第 2 部分。

本部分等同采用 ISO 16100-2:2003《工业自动化系统与集成 制造软件互操作性能力专规 第 2 部分:建规方法论》,其技术内容和结构与 ISO 16100-2:2003 一致。

国际标准 ISO 16100-2:2003 的原文中以下部分有明显编辑性错误:

- a) 6.2.3.1.1 节的内容;
- b) A.2 中“PROCESSOR:”和“LANGUAGE:”的说明部分。

在编写本部分过程中对上述错误进行了纠正。

本部分的附录 A 为资料性附录。

本部分由中国机械工业联合会提出。

本部分由全国工业自动化系统标准化技术委员会归口。

本部分主要起草单位:北京机械工业自动化研究所。

本部分参加起草单位:东南大学。

本部分主要起草人:黎晓东、王茜。

## 引 言

编制 GB/T 19902(ISO 16100, IDT)的动机来自于 ISO/TC 184/SC5 战略计划中提出的工业和经济环境,具体如下:

- a) 逐步增强的卖方解决方案基础;
- b) 用户应用标准过程中的困难;
- c) 模块化对系统集成工具的需求;
- d) 认识到应用软件以及使用这些软件的专业技术是企业的资产。

GB/T 19902(所有部分)是针对软件能力专规描述的国际标准转化来的,这种描述是计算机可读的,而且也是人可读的。它的目标是提供描述制造软件能力的方法,这些软件与其整个制造应用生命周期中的任务相关,不依赖于具体系统体系结构和执行平台。

# 工业自动化系统与集成 制造软件互操作性能力建规 第2部分:建规方法论

## 1 范围

GB/T 19902 的本部分规定了构造制造软件能力专规的方法论,可应用于制造领域中使用的软件产品。

## 2 规范性引用文件

下列文件中的条款通过 GB/T 19902 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 19902.1—2005 工业自动化系统与集成 制造软件互操作性能力建规 第1部分:框架 (ISO 16100-1:2002, IDT)

ISO 16100-3:2005 工业自动化系统与集成 制造软件互操作性能力建规 第3部分:接口协议及模板

ISO 16100-4 工业自动化系统与集成 制造软件互操作性能力建规 第4部分:一致性测试方法、标准及报告

REC-xml-20010502 XML 模式,第1部分:结构,万维网联盟(W3C)建议 2001年5月2日

REC-xml-20010502 XML 模式,第2部分:数据类型,W3C 建议 2001年5月2日

## 3 术语和定义

下列术语和定义适用于 GB/T 19902 的本部分。

### 3.1

#### 关联 association

两个或两个以上类之间的语义关系,说明了其实例间的关系。

[ISO/IEC 19501-1]

### 3.2

#### 基础规范 base specification

基础标准或被广泛接受和可用的标准。

### 3.3

#### 能力类 capability class

能力专规方法中的元素,描述了制造活动中与软件单元的任务相关的功能和行为。

### 3.4

#### 能力专规集成 capability profile integration

两个或两个以上软件单元互操作的过程,过程中使用相当的接口,这些接口用一致的方式进行配置,这一点在软件能力专规中有说明。

### 3.5

#### 分类 classifier

描述行为和结构特征的机制。

注：分类包括接口、类、数据类型和组件。

[ISO/IEC 19501-1]

### 3.6

#### 元素 element

模型的原子要素。

[ISO/IEC 19501-1]

### 3.7

#### 实体 entity

所关注的具体的或抽象的事物。

[ISO/IEC 10746-2]

### 3.8

#### 接口 interface

对象的行为抽象,由该对象交互作用的子集以及可能出现的约束组成。

[ISO/IEC 10746-2]

### 3.9

#### 对象 object

实体的模型。

注：对象用其行为和状态来描述其特征。一个对象是区别于另一个对象的。对象是封装的(隐蔽的),即：任何状态变化可能是一个内部活动的结果或是与其环境交互作用的结果。对象在其交互作用点上与其环境进行交互作用。取决于观察视点,重点可能在行为上也可能在状态上。当重点在行为上时,对象执行功能和提供服务(某个对象使某个功能有效,称为提供服务),这种说法是非正式的。建模时,这些功能和服务是按照对象及其接口的行为进行详细说明。一个对象可以执行一个以上的功能,一个功能可以由几个对象合作执行。

[ISO/IEC 10746-2]

### 3.10

#### 专规 profile

一个或多个基础规范和/或子专规的集合,经常用的是,所选择的类的标识、一致的子集、那些基础规范的选项和参数,或为完成具体的功能、活动或关系所必需的子专规。

注：这个定义来自于 ISO/IEC TR 10000-1。

### 3.11

#### 任务 role

参与到具体环境中的实体所具有的命名的指定的行为。

注：任务可以是静态的(如,关联的末端)或动态的(如,协作任务)。

[ISO/IEC 19501-1]

### 3.12

#### 分类法 taxonomy

为了明确的指定专规或专规组的分类方案。

[ISO/IEC TR 10000-1]

## 4 缩略语

CORBA	公共对象请求代理体系结构
IDL	接口定义语言
OMG	对象管理组
PSL	过程规范(描述)语言
UML	统一建模语言

## 5 能力建规的方法

### 5.1 能力建规的概念

GB/T 19902 重点是考虑制造软件互操作。图 1 描述了使用能力专规的概念来集成可互操作的软件。

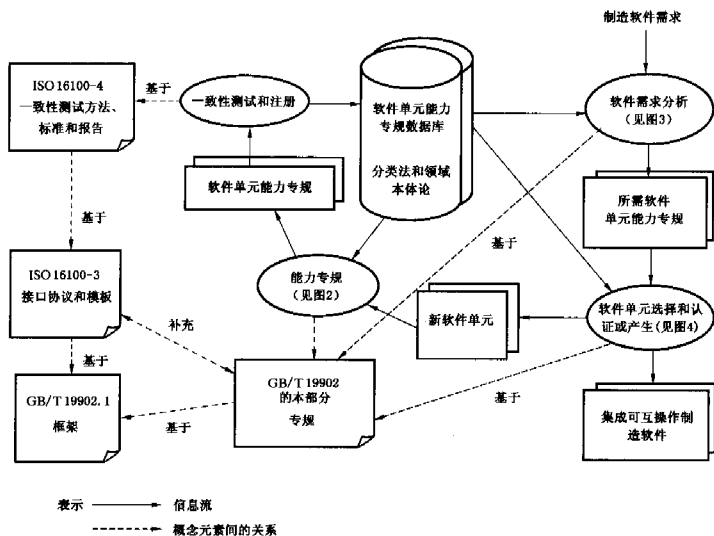


图 1 软件互操作能力专规的概念

软件单元的互操作可以按照与功能、接口和结构方面相关的能力来进行描述。基于 GB/T 19902.1—2005 中定义的框架和指定领域的应用系统模型, 这些方面在第 5 章和第 6 章中进行了定义, 在 ISO 16100-3 的第 3 部分中还要对它们进行详细的描述。制造过程的结构既可以是嵌套的又可以是分层的。在每个层次上, 制造软件的需求可以建模为一组能力类, 这些类具有相似的结构。制造软件需求是通过几个制造软件单元的集成来满足的。

在本方法论中, 制造软件需求应当按照软件能力专规来进行表示。建立软件单元专规包括产生软件单元的制造能力的精确说明, 这些能力是指软件单元所能执行的功能、提供的接口和支持的协议, 这些功能、接口和协议与所需求的目标制造能力一样。

能力建规方法论应当按照第 6 章提供的规则和元素进行定义。该方法论应当使用特定领域属性与每个特定软件单元相关的方法来描述能力专规, 并按照单元名称、制造功能和其他需要的属性类进行描述。

所需的专规与数据库中已有的专规进行比较。当匹配时, 便认为已经建立了专规的软件单元已经为集成做好了准备。当没有匹配时, 应当开发具有所需能力的软件单元, 并为其建立专规, 在能力专规数据库中注册。

在通过一致性测试后, 软件单元能力专规应当在适当的数据库中注册, 用于一致性测试的方法论及其抽象测试套件在 ISO 16100-4 中定义。

专规数据库应当具有一套分类法,用于描述能力专规。

### 5.2 能力建规的过程

图1所表示的软件互操作性能力专规的概念中与能力建规过程相关的部分在图2中进行了详细说明。

要建立专规的软件单元应当按照6.2.1中所描述的能力类结构中所支持的路径进行分析,这种能力类结构本身在ISO 16100-3中进行了定义。

所支持的路径应当用于在数据库中查找匹配的模板。当找到了匹配的模板,应当填写模板的字段,用于建立专规。当找不到匹配的模板时,应当用组能力类形成新的模板。

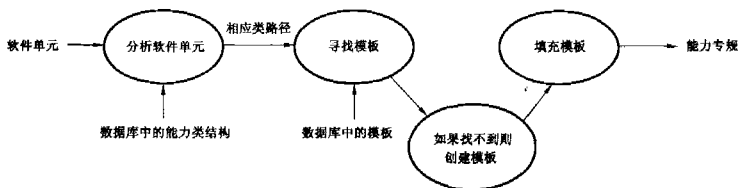


图2 能力建规过程

### 5.3 软件需求分析过程

图1所表示的软件互操作能力专规的概念中与软件需求分析过程相关的部分在图3中进行了详细说明。制造软件单元能力专规应当起源于软件需求分析过程中的制造软件需求。作为第一步,制造软件需求应当分解为几个简单的需求,这几个简单的需求是由从数据库中选择的能力类实现的。当存在与类相应的模板时,应当用特定的需求填写模板,以产生所需的能力专规。当没有这样的模板时,应当根据6.3中描述的模板规则产生新模板。

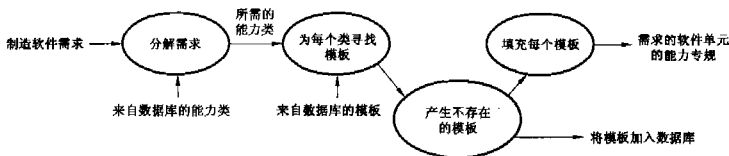
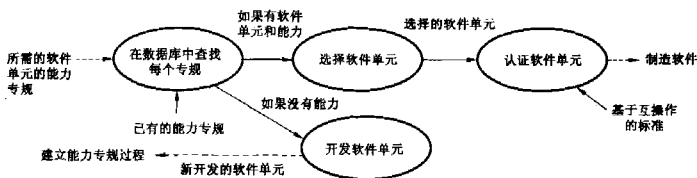


图3 软件需求分析过程

### 5.4 软件单元选择和认证或产生的过程

图1所表示的能力专规集成中与软件单元选择和认证或产生过程相关的部分在图4中进行了详细说明。



表示 —— 过程中的流  
 ----- 来自另一个过程或流向另一个过程的流, 另一个过程存在于图1的能力建规的概念中

图4 软件单元选择和认证或产生过程

对于每个所需的能力专规,应当搜索可用的软件单元的匹配的能力专规。应当按照 6.6 中给出的规则进行匹配。当存在匹配时,应当将软件单元加到候选列表中。当没有匹配时,应当进行下列某一项:

- a) 开发新的软件单元以满足所需专规;
- b) 将所需专规分解成为几个专规的合成;
- c) 针对已有的专规考虑需求。

新的软件单元的专规应当按照 5.2 中的建规过程在数据库中注册。应当按照互操作标准对选择的软件单元按照制造软件需求进行验证。

## 6 能力建规的元素和规则

### 6.1 分类法

能力专规分类法的关键是它能够识别能力类定义的内容。所构造的分类法应当提供一种方法来交换能力信息。

分类法应当描述制造企业生命周期中所承担的部分活动。如果需要向分类法中加入一项新的活动,则应按照 6.2 中的规则构造与新活动相配合的能力类。

### 6.2 能力类及其内容

#### 6.2.1 软件单元能力类内容

制造软件单元能力应当按照能力类进行说明。这些类应当来源于制造活动,这一点在 GB/T 19902.1—2005 的图 4 中作了说明。这些类也应当表示制造功能、资源和制造软件单元按照制造过程需求处理的信息。

以下是软件单元能力类应当包括的内容,但不限于这些:

- a) 制造领域类型;
- b) 制造活动类型(它是由它所参与的过程区分的),所参与的活动中的资源和活动中交换信息的类型;
- c) 由操作环境区分的计算机系统类型、软件体系和使用的设计模式;
- d) 服务类型、协议和运行的软件单元中使用的数据类型;
- e) 供应商名称、软件版本和变化历史;
- f) 性能基准;
- g) 可靠性指标;
- h) 服务和支持策略;
- i) 定价条款和使用条件。

更多的类内容规则及其详细情况在 ISO 16100-3 中进行了描述。

图 5 表明了能力类的概念性结构。

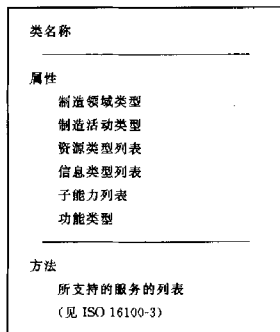


图 5 能力类的概念性结构

## 6.2.2 制造应用领域

### 6.2.2.1 制造应用活动模型

GB/T 19902.1—2005 图 4 中的特定领域的制造应用活动模型及其相关的信息、过程和资源三个模型使用了公共的需求和软件互操作性框架来描述为解决一个制造应用的需求和框架所提供的整套软件单元。

由于提供的软件单元可能只是整个应用的一部分,应当用注册的分类法和活动模型中的序列和层次编号对整个模型中的目标部分进行标注。

样例(见 GB/T 19902.1—2005 的图 2)

#### A) 制造活动

##### AA) 开发产品

##### AA1) 设计产品

##### AA11) 开展概念设计

##### AA111) 定义产品功能和约束

##### AA112) 产生产品行为

##### AA113) 分解功能约束和行为

##### AA114) 规定产品配置

##### AA12) 开展详细设计

##### AA121) 设计系统/组件

##### AA122) 分析系统/组件

##### AA123) 评价系统/组件设计

##### AA124) 优化设计

##### AA125) 确定系统/组件设计

##### AA126) 产生装配图

##### AA2) 设计工艺过程

##### AA21) 开展概念工艺过程设计

##### AA211) 选择制造工艺过程

##### AA212) 选择制造资源

##### AA2121) 选择机床

##### AA2122) 选择工具/夹具

##### AA2123) 选择劳动技能

- AA213) 估算制造成本/时间
- AA22) 开展详细工艺过程设计
- AA221) 产生工艺过程序列
- AA222) 产生工序
- AA2221) 确定中间加工特性
- AA2222) 规定加工件的设置和加工资源
- AA2223) 计算中间的加工公差
- AA2224) 开发加工指令
- AA223) 定义制造参数
- AA224) 产生控制程序
- AA2241) 产生工具路径
- AA2242) 规定工艺过程控制参数
- AA2243) 产生机床控制程序
- AA225) 产生车间路径
- AA2251) 确定车间配置
- AA2252) 确定运送方式
- AA2253) 规定时间
- AA3) 计划企业资源
- AA4) 获取资源
- AA5) 执行制造订单
- AA51) 开发工序序列和详细调度
- AA52) 分派生产单元
- AA53) 跟踪生产单元和资源
- AA54) 管理工厂-车间数据/文档
- AA6) 控制设备和工艺过程

#### 6.2.2.2 制造过程模型及其专规

制造过程模型类是一种(自动的功能)模型,这种模型来源于所需求的特定应用活动,适当的分类法和在系统中索引号(见6.2.2.1中的例)特定的资源及其信息流。

使用像 IDEF0 这样的建模语言,活动模型描述应用系统的需求和数据流。

这种过程模型描述了自动的功能,所选择的资源以及其中的信息流。过程模型应当合适地命名和标注,使用注册了的分类法和相关的(目标)活动模型的序列和层次编号。

每个过程模型应当表达成为一个恰当的专规。

#### 6.2.2.3 制造资源模型及其专规

制造资源模型描述了过程模型中所选择的资源,如仪器、设备、通信网络、人员和材料,这些资源用于满足信息模型的需求,这种信息模型规定了资源间的信息流。

资源模型应当使用注册的分类法和相关的(目标)活动模型的序列和层次编号适当地命名和标注。

每个资源模型应当表达成为一个恰当的专规。

#### 6.2.2.4 制造信息模型及其专规

制造信息模型描述了过程模型中的事件的数据类型和资源间交换的数据,过程模型描述了应用活动模型中活动的特定范围。

信息模型应当使用注册的分类法和相关的(目标)活动模型的序列和层次编号适当地命名和标注。

每个资源模型应当表达成为一个恰当的专规。

### 6.2.3 计算模型及其相关类

计算模型描述了 6.2.2 中描述的过程模型、资源模型和信息模型的映射。

#### 6.2.3.1 软件单元的类表达

##### 6.2.3.1.1 类名称

应当列出名称。

##### 6.2.3.1.2 类属性

应当列出类属性,以及它们所具有的数据类型和接受外部访问能力。

##### 6.2.3.1.3 类操作

应当列出类操作,以及它们所具有的特征和对外部服务的能力。

可以将软件单元打包,其中包含多种子类。这种情况中,应当列出包含的全部类。

#### 6.2.3.2 相关的软件体系结构、软件设计模式

应当列出软件体系结构典型的特点和软件单元框架使用的软件设计模式以及软件单元担任的角色。

以下列出了体系结构设计模式及其结构和角色的样例<sup>1)</sup>。

##### a) 层次体系结构

示例 结构:应用可以分解为多个子任务组,应用中的子任务组在抽象的特定层次上。角色:第 N 层实体担任为第 N+1 层实体服务的角色。

##### b) 代理体系结构

示例 结构:具有分离组件的分布软件系统,这些组件通过远程服务调用进行相互作用。角色:客户、服务器、代理、桥路、客户代理、服务器代理。

##### c) 模型-视图-控制器体系结构

示例 结构:模型包含了核心功能和数据、视图向用户显示信息、控制器处理用户的输入。角色:模型、观察者、视图、控制器。

##### d) 主机-终端

示例 结构:主机组件将工作分配给同样的终端组件并根据这些终端返回的结果计算出最终结果。角色:客户、主机、终端 1、终端 2、...、终端 N。

##### e) 代理

示例 结构:使组件的客户与其代表进行通信,而不是与组件本身通信。角色:客户、代理、原物。

##### f) 出版者-订户

示例 结构:出版者向任何数量的订户通报其状态的转变。角色:出版者、订户。

#### 6.2.3.3 服务或协议类

软件单元的接口应当描述成为服务(如,在层次体系结构中,第 N 层实体应当为第 N+1 层实体服务)或协议(如,在客户服务器体系结构中,客户对于服务器具有特定的协议的接口)和相关的数据类型。

### 6.2.4 软件单元的非功能属性

6.2.2 和 6.2.3 描述了功能方面的分类,与之相反,以下的软件单元属性是软件单元的非功能方面的属性。

#### 6.2.4.1 单元的经销商、版本和历史

软件单元的能力专规应当包括经销商(供应商)名称和联系地址、软件的最新版本和它的修正历史。

#### 6.2.4.2 使用的计算工具

以下信息应当包含于软件单元的能力专规:

a) 处理器——处理器的类型是关键而且重要的,但是也应考虑它的性能,因为低性能可能严重妨碍软件的效率和及时地执行软件。

<sup>1)</sup> 本例来自 F. Buschmann et al, “面向模板的软件体系结构”, John Wiley & Sons, 2000 年 6 月。

- b) 操作系统和所需选项——运行软件组件所需的适当的操作系统和发布的版本。这些信息的可以包括任何支持向上兼容的特性。
- c) 语言——软件组件的源语言,包括生成组件过程中使用的编辑器、编译器、连接器和调试器的版本。这些信息的可以包括任何支持向上兼容的特性。
- d) 运行时的存储器——软件组件运行时的存储器的类型和容量以及其他的支持运行的信息。
- e) 磁盘空间——存储运行中的信息和软件组件的源文件所需的媒体类型和容量。这类信息应当包括任意的数据,例如,所需要的操作变量数据,处理结果的数据和错误恢复机制所需数据。
- f) 多用户支持——软件组件处理多用户、多客户或订户的能力。
- g) 远程访问——软件组件,以及其他的在执行前进行下载和上载的软件组件,支持远程访问、控制和管理的的能力。
- h) 附件和插件——为了支持软件能力运行时的行为需要的软件扩展,如解释导入的图形或筛选输入的非本国应用格式的数据。

#### 6.2.4.3 软件单元标准的性能

软件单元的能力专规应当包括软件单元在特定的计算设备上运行所花费的实际时间(时间的临界值)的性能数据。这些性能数据应当包括:

- a) 输入特定的数据和约束所消耗(执行)的时间(基本的性能数据);
- b) 单位时间执行特定事务处理的数量(集成的性能数据)。

#### 6.2.4.4 软件单元的可靠性数据

软件单元的能力专规应当包括使用的历史、软件单元的装载数量、它的计划安全等级以及它的安全等级是由其自身决定的还是由第三方决定的。

#### 6.2.4.5 资格

软件单元的能力专规应当包括关于每个经销商资格的信息,如:授权、承诺和应用软件单元的策略;许可要求;车间策略;操作员培训要求。

#### 6.2.4.6 价格数据

软件单元的能力专规应当包括软件单元的初始价格和运行的花费。

### 6.3 能力模板和规则

在能力模板中对使能或支持活动的软件单元进行了精确描述,这些活动有与之相关的能力类。软件能力模板的结构应当遵循制造能力类的结构。

注:在层次结构中,每个能力模板与结构中定义在每个层次的每个能力有关。在嵌套结构中,每个能力类和结构的每个层次上的模板间存在相似的关联。

图6表示了一个模板的概念结构的示例。这个结构应当包括一个全部模板共有的部分和一个该能力类特有的部分。模板的形式化结构在ISO 16100-3中进行定义。

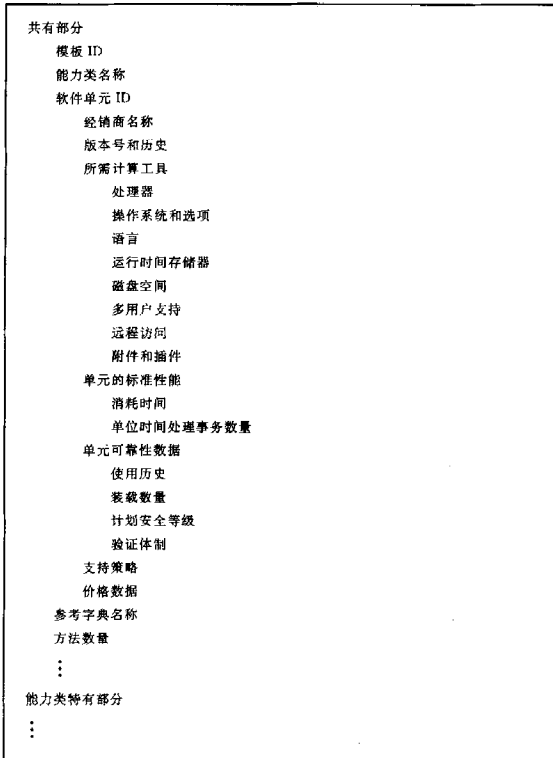


图 6 模板结构示例

共有部分中应当包含以下系列元素：

- a) 模板 ID——对象模板的标识；
- b) 软件单元 ID——使能制造功能的制造软件单元的标识；
- c) 参考字典名称——包含能力类定义的字典的名称；
- d) 能力类名称——参考能力类名称；
- e) 专规属性数量——从相应的能力类继承的属性的数量；
- f) 方法数量——软件单元提供的方法的数量；
- g) 资源数量——软件环境中需求资源的数量；
- h) 约束数量——运行软件单元所需条件的数量；
- i) 扩展数量——单元供应商提供的其他软件单元的数量；
- j) 低层数量——参考能力类结构的层次中的嵌套的程度或最深层；
- k) 下层子模板的数量——在层次结构或嵌套结构中，在当前层的下一层与子能力相关的模板的数量，这些子能力组成了当前层、与该模板相关的目标能力。

能力类的特有部分中应当包括下列元素：

- a) 属性列表；
- b) 方法列表；
- c) 资源列表，如操作系统类型；
- d) 约束列表，如体系结构类型、设计模式；
- e) 扩展列表；
- f) 低层列表；
- g) 子模板列表。

应当使用 XML 惯例创建 XML 模式来定义能力模板（参见 REC-xmlschema-1-20010502 和 REC-xmlschema-2-20010502）。应当用 XML 惯例进行 XML 模式和 XML 文件转换的方式进行能力模板间关系的说明。当模板中规定了一个能力类，而且对这个类做了实例化，那么这个实例就表示一个对象。如果两个能力模板各自的属性和操作相同，那么这两个能力模板是相同的。当一个模板的属性构成了另一个模板的属性子集，并且这个模板的操作构造了另一个模板的操作子集，则认为这两个能力模板是兼容的，而且具有匹配。

#### 6.4 能力专规和规则

能力专规是至少具有专规软件单元名称实例的能力模板。其他项要根据规范等级填写。

应当使用 XML 惯例产生 XML 文件的方式定义能力专规。应当用 XML 惯例转换 XML 文件的方式说明能力专规间的关系。当能力专规引用了一个能力模板并且已经填写了这个模板时，这个已填写了的模板描述了一个专规对象。

#### 6.5 软件单元专规数据库

下列元素都保存在数据库中，每个元素都以字典名称进行区分，并在从 6.1~6.4 中进行了描述：

- a) 一组分类法；
- b) 一组能力类；
- c) 一组能力模板；
- d) 一组能力专规。

可以将上述 4 项元素的自由组合构成数据库提供给必要的服务。

当把一个分类法输入分类法字典时，这个分类法应当是唯一的。当把一个能力类输入人类字典时，这个能力类应当是唯一的。当把一个能力模板输入模板字典时，这个能力模板应当是唯一的。当把一个能力专规输入专规字典时，这个能力专规应当是唯一的。

#### 6.6 匹配能力专规的规则

在以下过程中要用匹配能力专规：

- a) 在能力建规过程中的软件单元分析过程（见图 2）；
- b) 在软件需求分析过程中的需求分解过程（见图 3）；
- c) 在软件单元选择和认证或产生过程中对每个专规进行数据库搜索过程（见图 4）。

匹配是在这些过程中的软件单元描述、制造软件需求或需求的软件单元能力专规与数据库中的软件单元能力专规间进行的。应当参照为数据库准备的分类法、能力类的内容和现有的模板，用 XML 的惯例来描述软件单元和制造软件需求。将产生的软件单元、制造软件需求或需求的软件单元能力专规的描述与 6.2 中描述的数据库中的能力类内容进行对比。如果至少有一部分匹配，则数据库中的能力类描述就是匹配的结果。用能力类的输出内容在数据库中搜索模板。

#### 6.7 互操作性的标准

用互操作性的标准来执行 5.4 中的认证过程。

### 7 一致性

本部分的一致性包含于 ISO 16100-4 中。

**附录 A**  
**(资料性附录)**  
**参考方法**

### A.1 可扩展置标语言 (XML)

可扩展置标语言(XML) (Schuldt, 1998)具有一些特性,可以直接或间接地用于建立软件能力专规的工作中。XML是一种语言,它可以将“文档”中的词汇元素表示为有向图,尤其是在网络中。词汇元素可以由用户定义。XML实际上是 SGML 的子集,并提供像 HTML 的标记。任何 XML 文档也能检查 XML 的有效性。然而,应当说明,在建立软件能力专规的过程中使用 XML 是因为 XML 具有 XML 名称空间用于名称空间的协调和注册。

### A.2 软件包的词汇、定义和交换格式:开放软件描述 (OSD)和频道定义格式 (CDF)

开放软件描述<sup>2)</sup>(OSD)是基于 XML 的词汇,用于描述软件包和它们的内部依赖关系。OSD 能够用于软件分布环境,也能用于用户启动(“拉”)或自动(“推”)状态。OSD 可能是在 Web 上分布软件使用的一种潜在标准,采用这两种模式中的一种。

基于拉式的软件分布包括用户发现、下载和升级软件的活动。虽然 OSD 简化了自动化下载和安装所需软件组件的过程,仍需要 Web 用户浏览启动软件安装过程的 HTML 页。HTML 4.0 规范中的“OBJECT”标记用于在 Web 上为新软件“做广告”。在 OSD 资源里检测 OBJECT 的过程中,掌握 OSD 的用户代理能够自动下载并更新必要的软件组件。

频道定义格式(CDF)也是基于 XML 的,为 HTML 页和其他 Web 资源间的内部关系描述提供了元数据词汇。掌握 CDF 的客户可以使用“敏捷拉式”技术自动下载 Web 内容。掌握 CDF 的服务器可以执行“真实推式”机制自动将内容从客户分布给服务器。这样 CDF 就为“推”内容提供了一种语言,为使软件能够“推”或为软件的自动分布提供了理想的杠杆支点。为使 CDF 实现软件“推”,CDF 文件需要包含基于 OSD 软件包的参考。

OSD 词汇包括了扩展的词汇,它们描述了软件元素。这些词汇包括以下内容:

SOFTPKG:	定义一个通用软件包
IMPLEMENTATION:	用于描述一个软件包的执行
DEPENDENCY:	用于指明软件分布或组件间的依赖关系
TITLE:	提供软件包的标题或“友好名称”
ABSTRACT:	提供一种简短的描述,概括软件分布的种类和目的
LICENSE:	指明获取许可证明和版权声明的地点
CODEBASE:	指明软件分布存在的地点(通常在网上)
OS:	指明所需操作系统
OSVERSION:	指明所需操作系统的版本
PROCESSOR:	指明所需的处理器
LANGUAGE:	指明软件用户接口所需的自然语言。
VM:	指明所需虚拟机
MEMSIZE:	指明所需运行时间存储器的数量

<sup>2)</sup> Van Hoff et al., 1997.

DISKSIZE;	指明所需磁盘空间的数量
IMPLTYPE;	指明执行的类型

### A.3 分布软件服务:开放分布式处理(ODP)和公共对象请求代理体系结构(CORBA)

ODP 参考模型描述了信息系统的重要属性,为使系统开放或分布,可以对其定义约束。这种参考模型是详细定义这些需求的系列标准的基础。

对象管理组(OMG)是非赢利性的经销商、软件开发人员和最终用户的团体,成立于1989年5月。它试图为分布的面向对象的应用提供一个公用的体系结构框架,这些应用基于广泛有效的接口规范,如同ODP提供的那样。对象管理体系结构(OMA)是OMG承担的所有活动的中心,它由一种参考模型(1992发布的)构成,这种模型识别了组件、接口和协议,并描述了它们的特征,它们组成了OMA,但OMA并没有详细定义它们。

CORBA是由对象管理组开发的,是动态绑定分布对象的体系结构和协议。绑定只在语法和词汇方面进行,局限于IDL的能力。

这种参考模型有5个组件:

- 对象请求代理——提供一种允许对象交互信息的基础结构,它独立于执行对象使用的特定平台和技术。
- 对象服务——将对象的生命周期管理标准化。提供的接口用于生成对象,控制对对象的访问并持续跟踪重新部署的对象。对象服务是为应用的持续性提供的。
- 公用工具——提供一套通用应用功能,这些功能能够按照特定配置的特定需求进行配置。这些工具更容易被最终用户识别,如打印和电子邮件。
- 领域接口——描述特定应用领域中向最终用户提供关注功能的纵向范围,这些领域如制造、金融和保健。
- 应用对象——不是OMG标准化活动的一部分,但OMG认为这些对于应用的成功开发至关重要。

OMG方法用接口定义语言(IDL)为分布对象定义接口。OMG已经发现了对更富语义的描述语言的需求。其结果是,即将发布大量的RFP,这些RFP覆盖了商务对象领域,而且特别注意了对更富语义的描述语言的需求。最近,OMG已经开始了大量商务对象的标准化;如任务通用,将来它将使得经销商在构造软件时使用一些基础的标准化对象,进而有利于实现互操作性。

参 考 文 献

- [1] ISO/IEC TR 10000-1:1998 信息技术 国际标准化专规框架和分类 第1部分：一般规则和文档框架
  - [2] ISO/IEC 10746-2:1996 信息技术 开放分布处理 第2部分：参考模型：基础
  - [3] ISO/DIS 18629-1 工业自动化系统与集成 过程描述语言 第1部分：综述与基本原理
  - [4] ISO/IEC 19501-1 信息技术 统一建模语言（UML） 第1部分：规范
  - [5] IEEE 1320.1-1998 功能建模语言标准 IDEF0 的语法和语义
  - [6] REC-xml-20001006 可扩展标记语言（XML）1.0，第2版，W3C 建议 2000.10.6
-