

STEP 模型映射技术及其在开放环境中的应用*

杨子江, 李善平, 董金祥

(浙江大学 计算机科学与工程系, 浙江 杭州 310027)

E-mail: yzj@cad.zju.edu.cn

http://www.zju.edu.cn

摘要: 针对互操作所导致的异构系统间数据交换构造 STEP 模型映射算法,为基于不同 STEP 应用协议的 CAx 系统集成提出了一种方法论.首先对 STEP/EXPRESS 数据模型作出形式化定义,在此基础上提出了两种 STEP 模型映射技术——静态映射与动态映射,并阐述了基于 EXPRESS-X 的实现及其在开放式环境中的一种典型应用.提出了模型映射技术独立于具体系统,为异构系统提供透明的集成方式,从而降低了信息集成在系统开发中的开销.

关键词: STEP;EXPRESS;CIMS;模型映射;开放环境

中图法分类号: TP391 **文献标识码:** A

20 世纪 90 年代以来,工程设计的复杂性对 CIMS 环境提出了更高的开放性需求.制造系统从单一的集中式的体系结构向多元分布式体系结构转化,不同企业的异构系统之间通过 Internet/Intranet 实现互操作^[1].由于异构系统往往采用不同的数据模型,它们之间的互操作意味着大量基于异构模型的数据交换,这一特点导致了模型映射技术的需求.

STEP^[2]技术的出现为产品模型的标准化提供了新的方法论,只要系统都采用符合同一 STEP 协议的产品模型,就可以通过 STEP 工具集相互交换数据.然而,STEP 标准本身包括很多不同的应用协议(AP),在采用不同应用协议的系统间交换数据超出了传统 STEP 工具的应用范围.

J.Liang 提出通过建立集成数据共享环境(IPDE)来实现跨模型的信息集成^[3].IPDE 在多个 STEP 应用协议中分别确定不同系统所需的信息实体的子集,将这些子集的并作为一个集成产品数据库(IPDB)模式.IPDB 由共享数据管理器(SDM)直接操作,不同的应用通过各自的数据访问接口与 SDM 交互,并得到所需的数据视图.这种方法实现了分布式异构系统的信息共享,其局限在于难以控制信息实体的粒度、难以解决数据模式的完备性/一致性与信息模型冗余性的矛盾.

实现异构系统间信息共享的另一种选择是模型映射技术,该技术将一种模型的结构和语义转换成另一种模型的等价表达. William C. Burkett 在文献[4]中给出了将 EXPRESS 数据模型转换为 XML 模型的范例.这种转换包括了结构转换和语义转换,可以在基于 STEP 和基于 XML 的异构系统间实现信息集成. Youchon Oh 提出了基于通用建模语言 UML 的模型映射方法,并以此实现 PDM 与 CAD 系统的集成^[5].该方法采用 UML 作为映射中介,适用于传统数据模型与 STEP 数据模型之间的映射.由于 UML 语言的计算机可理解性较差,因此映射过程需要大量的人工参与. Yanping (Paul) Zhang 为基于不同 STEP 应用协议的异构系统间实现互操作提出了一种系统方法论^[6].其思想是在理解应用上下文环境的基础上,从应用解释构造(application interpreted constructs)、功

* 收稿日期: 2000-12-09; 修改日期: 2001-07-12

基金项目: 国家自然科学基金资助项目(69684006); 国家 863 高科技发展计划资助项目(863-511-820-019)

作者简介: 杨子江(1975 -),男,安徽安庆人,博士生,主要研究领域为信息集成,产品数据标准;李善平(1963 -),男,浙江宁波人,博士,教授,博士生导师,主要研究领域为信息集成,产品数据标准,嵌入式系统;董金祥(1945 -),男,浙江温州人,教授,博士生导师,主要研究领域为 CAD,CIMS,工程数据库.

能单元(unit of functionality)与概念(concept)这 3 个层次实现产品模型间的数据转换,强调对模型语义信息的理解以及针对模型相似度的不同而采取不同的映射策略.从实现的角度来看,缺乏对 STEP 数据模型的构造分析,映射器的生成缺乏自动化支持,需要较多的手工编程.

针对现有模型转换技术的不足^[3-6],本文首先分析了 STEP/EXPRESS 数据模型的逻辑结构及其面向对象特性,在此基础上给出 STEP 模型映射的形式化定义及算法,最后以浙江大学自主开发的集成化 STEP 工具集 GS-STEP^[7]为背景,阐述了该算法的实现及其在开放环境中的应用方法论.本文提出的算法与文献[3~6]中提出的方法相比具有以下特点:

(1) 形式化.给出 EXPRESS 数据模型及其映射的形式化定义,在此基础上针对模型的逻辑结构构造映射算法,保证了映射描述及实现的一致性与完备性.

(2) 映射自动化.充分利用 EXPRESS 模型的面向对象特性,通过对 EXPRESS 继承及约束机制的计算机处理实现模型映射自动化.与上述其他方法相比,在较大程度上减少了模型映射中的人工参与.

(3) 协议无关性与系统独立性.算法仅从 EXPRESS 数据模型的层次处理产品信息,而不局限于任何特定应用协议;软件的设计独立于具体应用系统,以信息集成平台的形式实现.

(4) 标准化.以 ISO 标准模型映射语言 EXPRESS-X^[8]描述映射关系,适应开放式环境对信息交换标准化的需求.

任何一种模型映射技术只适用于特定的领域.文献[4,5]讨论的方法用于 STEP 模型与非 STEP 模型的转换,文献[6]与本文提出的方法针对不同 STEP 应用协议间的映射.

1 EXPRESS 数据模型的形式化定义

EXPRESS 是一种面向对象的形式化信息建模语言^[9].为了描述上的规范性,首先给出 EXPRESS 数据模型的形式化定义.

定义 0. EXPRESS 数据模型是类的集合.

定义 1. 类是一个二元组 (S, C) ,其中 S 表示类的结构, C 表示类的约束;类的合法实例称为对象,对于指定类 A 的所有对象的集合称为类 A 的域,记作 $dom(A)$.

定义 2. 给定下列简单数据类型:整型 δ 、实数型 ψ 、字符串型 ζ 、逻辑型 ξ 、布尔型 λ 、枚举型 θ 、二进制型 β ,若类 $\Sigma \in \{\delta, \psi, \zeta, \xi, \lambda, \theta, \beta\}$,其中 Σ 的结构 S 为简单数据,约束 C 为对应数据类型的值域,则称 Σ 为原子类.

定义 3. 设 $n \geq 0, I$ 是一个类, $D_1, D_2, \dots, D_n = dom(I)$,若类 Π 的域 $dom(\Pi) = D_1 \times D_2 \times \dots \times D_n$,则称 Π 是聚集(aggregation)类.

定义 4. 设 $n \geq 0, \Gamma_1, \Gamma_2, \dots, \Gamma_n$ 是类, $I = (a_1: \Gamma_1, a_2: \Gamma_2, \dots, a_n: \Gamma_n)$,则 I 是类,其中 a_1, a_2, \dots, a_n 称作 I 的属性.

EXPRESS 数据模型中的类同时支持单一继承和多重继承,分别定义如下:

定义 5. Γ_0, Γ_1 是类,若 $(\Gamma_1 \subseteq \Gamma_0) \wedge \forall \Gamma [(I \neq \emptyset \wedge I \subset \Gamma_0) \rightarrow (I \not\subset \Gamma_0 - \Gamma_1)]$,则称 Γ_0 单一继承 Γ_1 .

定义 6. 设 $n \geq 2, \Gamma_0, \Gamma_1, \dots, \Gamma_n$ 是类,若 $\forall i \forall j [(1 \leq i, j \leq n \wedge i \neq j) \rightarrow (\Gamma_i \subseteq \Gamma_0 \wedge \Gamma_i \neq \Gamma_j)]$,则称 Γ_0 多重继承 $\Gamma_1, \dots, \Gamma_n$.

单一继承和多重继承描述的是父类与导出类之间的关系,EXPRESS 对继承关系附加了 3 个算子 ONEOF, AND, ANDOR^[9],并通过这 3 个算子描述了导出类与导出类之间的关系,分别定义如下:

定义 7. 设 $n \geq 1, \Gamma_0, \Gamma_1, \dots, \Gamma_n$ 是类, $O_m \in dom(\Gamma_m), m = 1, 2, \dots, n$,若 $\forall i [(1 \leq i \leq n) \rightarrow \Gamma_0 \subseteq \Gamma_i] \wedge \forall j \forall k [(1 \leq j, k \leq n \wedge j \neq k) \rightarrow (O_j \cap O_k = \emptyset)]$,则称 Γ_0 与 $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ 之间存在 ONEOF 继承关系,记作 ONEOF($\Gamma_0 | \Gamma_1, \Gamma_2, \dots, \Gamma_n$).

定义 8. 设 $n \geq 2, \Gamma_0, \Gamma_1, \dots, \Gamma_n$ 是类,若 $\forall i [(1 \leq i \leq n) \rightarrow (\Gamma_0 \subseteq \Gamma_i \wedge dom(\Gamma_i) = \emptyset)] \wedge dom(\Gamma_1 \cup \Gamma_2 \cup \dots \cup \Gamma_n) \neq \emptyset$,则称 Γ_0 与 $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ 之间存在 AND 继承关系,记作 AND($\Gamma_0 | \Gamma_1, \Gamma_2, \dots, \Gamma_n$).

定义 9. 设 $n \geq 2, \Gamma_0, \Gamma_1, \dots, \Gamma_n$ 是类, $O_m \in dom(\Gamma_m), m = 1, 2, \dots, n$,若 $\forall i [(1 \leq i \leq n) \rightarrow \Gamma_0 \subseteq \Gamma_i] \wedge \forall j \forall k [(1 \leq j, k \leq n \wedge j \neq k) \rightarrow (O_j \cap O_k = \emptyset)] \wedge dom(\Gamma_1 \cup \Gamma_2 \cup \dots \cup \Gamma_n) \neq \emptyset$,则称 Γ_0 与 $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ 之间存在 ANDOR 继承关系,记作 ANDOR($\Gamma_0 | \Gamma_1, \Gamma_2, \dots, \Gamma_n$).

定义 10. 设 $n \geq 1, \Gamma_1, \Gamma_2, \dots, \Gamma_n$ 是类,则它们之间的继承关系构成一个有向着色超图(hypergraph) $G(\Gamma, E)$.其中 $\Gamma = \{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$ 是顶点的集合, E 是边的集合,定义如下:

1) $\forall i\{(1 \leq i \leq n) \wedge (\psi_0, \psi_1, \dots, \psi_i \in \Gamma) \wedge \text{ONEOF}(\psi_0 | \psi_1, \psi_2, \dots, \psi_i) \rightarrow \exists e_1 \exists e_2 \dots \exists e_i [e_1, e_2, \dots, e_i \in E \wedge e_1 = (\psi_0, \psi_1) \wedge e_2 = (\psi_0, \psi_2) \dots \wedge e_i = (\psi_0, \psi_i)]\}$;

2) $\forall i\{(1 \leq i \leq n) \wedge (\psi_0, \psi_1, \dots, \psi_i \in \Gamma) \wedge \text{AND}(\psi_0 | \psi_1, \psi_2, \dots, \psi_i) \rightarrow \exists e [e \in E \wedge e = (\psi_0, \psi_1, \dots, \psi_i)]\}$;

3) $\forall i\{(1 \leq i \leq n) \wedge (\psi_0, \psi_1, \dots, \psi_i \in \Gamma) \wedge \text{ANDOR}(\psi_0 | \psi_1, \psi_2, \dots, \psi_i) \rightarrow \exists e_0 \exists e_1 \dots \exists e_i [e_0, e_1, \dots, e_i \in E \wedge e_0 = (\psi_0, \psi_1, \dots, \psi_i) \wedge e_1 = (\psi_0, \psi_1) \wedge e_2 = (\psi_0, \psi_2) \dots \wedge e_i = (\psi_0, \psi_i)]\}$;

4) 谓词 1)、3) 中的 e_1, e_2, \dots, e_i 为 ONEOF 边, 谓词 2) 中的 e , 谓词 3) 中的 e_0 为 AND 边, ONEOF 和 AND 是继承关系图的颜色。

定义 11. 设 M_1, M_2 是数据模型, $\Gamma_1 = (S_1, C_1), \Gamma_2 = (S_2, C_2)$ 是类, 且 $\Gamma_1 \in M_1, \Gamma_2 \in M_2, \Gamma_1, \Gamma_2$ 同构当且仅当满足下列条件:

1) $S_1 = S_2$;

2) G_1, G_2 分别是 Γ_1, Γ_2 的继承关系图, 若删除 Γ_1, Γ_2 所有导出类的顶点及其相关边的子图为 G'_1, G'_2 , 则 G'_1, G'_2 同构;

3) 设 $n \geq 0, \Gamma_1$ 的父类为 $\Gamma_{11}, \Gamma_{12}, \dots, \Gamma_{1m}, \Gamma_2$ 的父类为 $\Gamma_{21}, \Gamma_{22}, \dots, \Gamma_{2n}$, 则 Γ_{1i} 与 Γ_{2i} 同构, 其中 $i = 1, 2, \dots, n$.

定义 12. Γ 是一个类, γ 是 Γ 的任意一个实例, Σ 是 γ 所在的 STEP 数据集, 若存在谓词 C , 使得 $\gamma \in \text{dom}(\Gamma) \rightarrow C(\gamma, \Sigma)$, 则称 C 是 Γ 的约束。

定义中引入 是因为 EXPRESS 数据模型的约束可能涉及其他类的对象。

2 EXPRESS 数据模型映射原理

STEP 模型映射的目标是在两个不同的 EXPRESS 数据模型或其子集之间建立映射关系. 根据对模型上下文环境处理方法的不同, 分为静态映射与动态映射两种。

2.1 静态模型映射

定义 13. 设 M_1, M_2 是数据模型, ψ_1, ψ_2 分别是 M_1, M_2 的幂集, 若存在映射 $f: \psi_1 \rightarrow \psi_2$, 则称 f 是从 M_1 到 M_2 的一个静态模型映射, 并称 M_1 可部分映射至 M_2 。

将模型映射定义在幂集上是为了表达多对多的对应关系, 即 M_1 中的一个类可能对应 M_2 中的多个类, 反之亦然. 从类与类之间对应的角度, 静态模型映射分为 1:1 映射、1:n 映射、n:1 和 m:n 映射。

定义 14. 设 M_1, M_2 是数据模型, ψ_1, ψ_2 分别是 M_1, M_2 的幂集, $e_1 \in \psi_1, e_2 \in \psi_2$, 且 $f(e_1) = e_2$, 则:

1) 若 $|e_1| = |e_2| = 1$, 称 f 将 e_1 1:1 映射至 e_2 ;

2) 若 $|e_1| = 1, |e_2| > 1$, 称 f 将 e_1 1:n 映射至 e_2 ;

3) 若 $|e_1| > 1, |e_2| = 1$, 称 f 将 e_1 n:1 映射至 e_2 ;

4) 若 $|e_1| > 1, |e_2| > 1$, 称 f 将 e_1 m:n 映射至 e_2 。

1:1 映射用于源模型和目标模型中存在对等类的情况. 1:n 映射用目标模型中的多个类对应源模型中的一个类, 即目标模型中以多个类的集合来表达该类信息. 与此相反的是 n:1 映射, 需要以源模型中多个类的聚合对应目标模型中的一个类. m:n 映射是指源模型的多个类的任意一个都不能进行单独映射, 必须将其信息作为整体处理, 并对应到目标模型中的若干个类。

建立静态模型映射首先需要分别确定源数据模型和目标数据模型的封闭子集作为映射的定义域和值域, 一般可以取 STEP 应用协议的一个一致性分类 (conformance class). 若该子集引用了其他一致性分类或集成资源中的类, 必须将其加入该子集中, 以确保封闭性. 从继承关系的角度来看, 定义域包含若干个互不连通的继承图, 每个连通子图对应一个 STEP 应用协议所规定的功能单元 (unit of functionality), 模型映射以连通子图为单位进行. 下面给出建立映射关系的算法。

算法 1.

- (1) 从继承图取出一个无导出类的节点 (叶节点) ω , 若 $\omega = \emptyset$, 则结束;
- (2) if (ω 无相关边或只与 ONEOF 边相关) and (ω 与目标模型中的对应类同构)
建立 1:1 映射, 从继承关系图中删除 ω , 转 (1);

```

else
  转(3);
(3) if ( $\omega$ 无相关边或只与 ONEOF 边相关) and (目标模型中存在类的集合与  $\omega$ 对应)
  建立 1:n 映射,从继承关系图中删除  $\omega$ ,转(1);
else
  转(4);
(4) if ( $\omega$ 与 AND 边  $e=(\omega_0, \omega_1, \dots, \omega_n)$ 相关)
  begin
  (4.1) 建立中间类  $\Gamma$ 与  $\{\omega_0, \omega_1, \dots, \omega_n\}$ 对应;
  (4.2) 删除边  $e$ ;
  if (目标模型中有与  $\Gamma$ 对应的类)
  (4.3) 建立  $n:1$  映射;
  else if (目标模型中有与  $\Gamma$ 对应的类的集合)
  (4.4) 建立  $m:n$  映射;
  else
  (4.5) 转(1);
  (4.6) for ( $i=1, 2, \dots, n$ )
    if ( $\omega_i$ 在继承关系图中无相关边)
      从继承关系图中删除  $\omega_i$ ;
    转(1);
  end
else
  转(5);
(5) 从继承关系图中删除  $\omega$ ,并把该节点加入不可映射集中,转(1).

```

设 A 是一个给定的映射关系构造算法, M_1, M_2 是任意两个部分可映射模型, ψ_1, ψ_2 分别是 M_1, M_2 的幂集, 且 $A(M_1, M_2) = f: \psi_1 \rightarrow \psi_2$. 对 $\forall e_1 \in \psi_1, \forall e_2 \in \psi_2$, 若 e_1, e_2 之间存在对应关系, 必有 $f(e_1) = e_2$, 则称 A 是完备的; 反之, 必有 $f(e_1) \neq e_2$, 则称 A 是一致的.

定理 1. 算法 1 对 EXPRESS 数据模型是完备且一致的.

证明: 设源模型 M_1 , 目标模型 M_2 是任意两个部分可映射的 EXPRESS 数据模型, 由前述封闭性假设知 M_1, M_2 是连通子图的集合, 每个连通子图对应应用协议的一个功能单元. 由于功能单元在语义上是完备的, 因而对应关系只存在于 M_1 的功能单元与 M_2 的功能单元之间, 即存在于单个的连通子图与连通子图之间. 由定义 10 可知, 连通继承关系图的构造可由以下 4 个操作完成:

- 1) 在 $G = \emptyset$ 中添加根节点;
- 2) 对 G 增加 1 条 ONEOF 边 $e = (\omega_0, \omega)$, 其中 ω_0 为 G 中原有节点, ω 为新增叶节点.
- 3) 对 G 增加 1 条 AND 边 $e = (\omega_0, \omega_1, \dots, \omega_i)$, 其中 ω_0 为 G 中原有节点, $\omega_1, \dots, \omega_i$ 为新增叶节点.
- 4) 对 G 增加 1 条 AND 边 $e = (\omega_0, \omega_1, \dots, \omega_i)$, 其中 $\omega_0, \omega_1, \dots, \omega_i$ 为 G 中原有节点, 且 $\omega_1, \dots, \omega_i$ 为叶节点.

其中 1) 为初始操作, 只执行 1 次; 2)~4) 称作对 G 的单边扩张, 可反复执行.

令连通图 $G, G_1 \in M_1, G_2 \in M_2$, 下面对 EXPRESS 继承关系图的结构使用归纳法.

(1) 当 G 只包含一个节点 ω 时, 其边集为空, 算法 1 的执行路径只能是 $p_1: 1 \rightarrow 2 \rightarrow 1$ 或 $p_2: 1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ 或 $p_3: 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$.

若 ω 可映射, 分两种情况:

- (I) G_2 中存在与 ω 对应的单一类且两者同构, 则执行路径为 p_1 , 建立 1:1 映射并删除 ω ;
- (II) G_2 中存在与 ω 对应的类的集合, 则执行路径为 p_2 , 建立 1:n 映射并删除 ω .

因此, 算法 1 的输出包含且只包含 $\{\omega\}$ 的对应关系, 对单节点图是完备的.

若 ω 不可映射, 则执行路径为 p_3 , 不建立任何映射关系, 注意到 ω 可映射时算法 1 的输出只包含 $\{\omega\}$ 的对应关系, 所以对单节点图是一致的.

(2) 假设算法 1 对连通图 G 是完备且一致的, G_1 是对 G 的单边扩张. 根据归纳假设, 只须考虑与新增边和节点相关的对应关系. 分 3 种情况:

(I) G_1 是对 G 增加 1 条 ONEOF 边 $e = (\omega_0, \omega)$ 所形成的图, 其中 $\omega_0 \in G, \omega$ 为新增叶节点. 算法 1 对 ω_0 的处理是在完成处理并删除 ω 之后, 因而不受影响, 对其他边和节点的处理也不受影响, 所以对 G 的处理不变. 由于 ω 只与 1

条 ONEOF 边相关,算法 1 对 ω 的处理与 ω 无相关边时相同,即执行路径为 1) 中的 p_1, p_2, p_3 之一. 因此同理可证算法 1 对 e 及 ω 也是完备且一致的, 因而对 G_1 完备且一致.

(II) G_1 是对 G 增加 1 条 AND 边 $e=(\omega_0, \omega_1, \dots, \omega_i)$ 所形成的图, 其中 $\omega_0 \in G, \omega_1, \dots, \omega_i$ 为新增叶节点. 算法 1 对 ω_0 的单独处理是在完成处理并删除 e 和 $\omega_1, \dots, \omega_i$ 之后, 故不受影响, 对其他边和节点的处理也不受影响, 所以对 G 的处理不变. 由于 $\omega_1, \dots, \omega_i$ 只与 1 条 AND 边相关, 算法 1 对它们的执行路径为 $p_4: 1 \ 2 \ 3 \ 4 \ 1$.

若 $\{\omega_0, \omega_1, \dots, \omega_i\}$ 可映射, 分两种情况: a) G_2 中存在与之对应的单一类, 则第 4 步执行路径为 4.1 4.2 4.3 4.6, 建立 $n:1$ 映射并删除 e 和 $\omega_1, \dots, \omega_i$; b) G_2 中存在与之对应的类的集合, 则第 4 步执行路径为 4.1 4.2 4.4 4.6, 建立 $m:n$ 映射并删除 e 和 $\omega_1, \dots, \omega_i$.

因此, 算法 1 在图 G 扩张后的新增输出包含且只包含 $\{\omega_0, \omega_1, \dots, \omega_i\}$ 的对应关系, 对 e 及 $\omega_0, \omega_1, \dots, \omega_i$ 是完备的, 所以对 G_1 也完备.

若 $\{\omega_0, \omega_1, \dots, \omega_i\}$ 不可映射, 则第 4 步执行路径为 4.1 4.2 4.5, 不建立任何映射关系, 注意到可映射时算法 1 的新增输出映射只包含 $\{\omega_0, \omega_1, \dots, \omega_i\}$ 的对应关系, 所以对 G_1 是一致的.

(III) G_1 是对 G 增加 1 条 AND 边 $e=(\omega_0, \omega_1, \dots, \omega_i)$ 所形成的图, 其中 $\omega_0, \omega_1, \dots, \omega_i \in G$ 且 $\omega_1, \dots, \omega_i$ 为叶节点. 算法 1 对 $\omega_0, \omega_1, \dots, \omega_i$ 的单独处理是在完成处理并删除 e 之后, 因此不受影响, 对其他边和节点的处理也不受影响, 所以对 G 的处理不变. 此时算法 1 的执行路径与情况(II)的惟一区别在于, 步骤 4.6 不删除节点 $\omega_1, \dots, \omega_i$, 所以与(II)同理可证算法 1 对 e 及 $\omega_0, \omega_1, \dots, \omega_i$ 是完备且一致的, 因而对 G_1 也是完备且一致的.

综上所述, 算法 1 对 G 进行单边扩张后所形成的图是完备且一致的.

由归纳基础 1) 及 EXPRESS 继承关系图的构成可知, 算法 1 对任何 EXPRESS 数据模型是完备且一致的. \square

定理 2. 算法 1 的时间与空间复杂度为 $O(V+E)$, 其中 V 为连通图 G 的节点数, E 为 G 的边数.

证明: 由定理 1 的证明过程可知, 对于只与 ONEOF 边相关的节点, 算法 1 的执行路径不包含内部循环, 所耗费时间用于建立节点的映射关系与删除 ONEOF 边, 因此是被删除的节点数与边数的线性函数. 对于只与 AND 边相关的节点, 算法 1 的执行路径包括了两个循环, 即建立中间类(步骤 4.1)和删除节点(步骤 4.6). 这两个循环的复杂度均为被删除的节点数的线性函数, 因而所耗费时间也是被删除节点数与边数的线性函数. 对于既与 AND 边又与 ONEOF 边相关的节点, 注意到每个节点作为子类至多与 1 条 AND 边相关(否则与 EXPRESS 语义矛盾), 因而其处理时间为前两种情况之和(先处理 AND 继承, 再处理 ONEOF 继承), 仍然是节点数与边数的线性函数. 由于算法 1 在 $G=\emptyset$ 时结束, 综合上述分析可知, 总体执行时间为 G 的节点数与边数的线性函数, 即时间复杂度为 $O(V+E)$.

算法 1 所需的内存分为两部分, 一部分用于存储继承关系图, 另一部分用于构造中间类, 前者为 G 的节点数与边数的线性函数, 后者小于前者, 因而算法的空间复杂度为 $O(V+E)$. \square

静态模型映射适用于数学关系明确的数据模型, 如从流形曲面模型(AP203 CC4)到面片 B-rep 模型(AP203 CC5)的映射. 这类映射关系有严格的数学定义, 因而只与类本身的语义相关, 而不依赖于模型的上下文环境.

2.2 动态模型映射

在很多情况下, 映射关系的建立不仅与类本身的语义相关, 还与对象所在的模型上下文环境有关. 映射过程必须建立在根据上下文环境对对象进行“解释”的基础上. 例如, 一个基于 CSG 表达的圆柱实体在特征模型中既可以是孔特征, 也可以是凸台特征. 此时, 对圆柱对象的映射就必须考虑其所在的上下文, 即圆柱所在零件的形状特征定义或圆柱相对基准面的参考方向. 根据特征定义或方向的不同, 可以将圆柱映射成不同特征.

在 STEP 应用协议中, 产品模型的应用上下文由应用参考模型(ARM)描述, 而模型的信息结构则取自集成资源(IR)的子集. 应用解释模型(AIM)是这两者之间的桥梁^[6], 它不仅从 IR 中选择合适的信息实体对模型结构进行形式化定义, 还以约束的形式对 ARM 中的上下文进行描述. AIM 为 STEP 模型的物理实现提供模式支持, 是应用协议中可由计算机处理的部分. 因此, 对自动化的 STEP 模型映射而言, 映射语义上下文是 AIM 中 EXPRESS 类的相关约束的集合.

EXPRESS 语言的约束有效地表达了产品的应用环境. 下面我们针对 EXPRESS 约束机制, 给出映射语义上

下文与动态模型映射的定义.

定义 15. 设 M_1, M_2 是数据模型, 类 $\Gamma \in M_1, C = C_1 \vee C_2 \vee \dots \vee C_m$ 是 Γ 的约束, f_1, f_2, \dots, f_n 是从 M_1 到 M_2 的静态模型映射, 若存在 $f(C_1, C_2, \dots, C_m) \in \{f_1, f_2, \dots, f_n\}$, 即 f 是依赖于 C_1, C_2, \dots, C_m 的真值的映射, 则称 C 是类 Γ 的映射语义上下文, f 是从 M_1 到 M_2 的一个动态模型映射.

定义 15 说明, 在任何一个确定的映射语义上下文中, 映射机制是惟一确定(或静态)的, 其动态性体现在对映射语义上下文的解释以及由此对映射机制的选择上. 动态模型映射的实质是在静态映射的基础上, 利用映射语义上下文动态确定目标类的映射方法, 因而依赖于对 EXPRESS 约束机制的解释^[10]. 建立动态模型映射的方法是:

- (1) 对无映射语义上下文的类建立静态映射关系;
- (2) 对有映射语义上下文的类: a) 建立该类的对象在所有语义上下文环境中的静态映射; b) 确定依赖于语义上下文的映射选择机制, 并表达为 EXPRESS 约束的形式; c) 将该类标记为动态映射类;
- (3) 在模型映射器中对动态映射类调用 EXPRESS 语言解释器, 并根据对约束解释的结果选择确定的映射模块.

动态模型映射是一种机制, 而非算法. 其正确性依赖于对映射上下文的处理, 即对 EXPRESS 约束机制的解释. 参考文献[10]对此作了详细的论述. 动态模型映射的时间与空间复杂度取决于类的约束函数的复杂度, 是由模型本身的特性确定的.

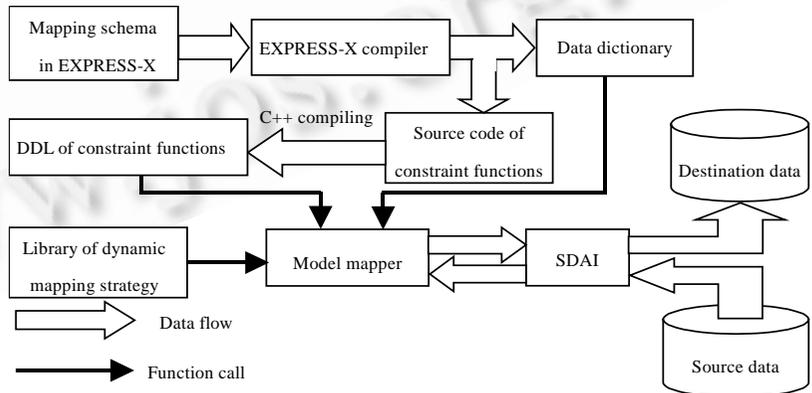
3 STEP 模型映射的实现与应用

3.1 实现方法

模型映射自动化的前提是采用计算机可处理的形式化语言描述映射关系. 有多种语言可以定义 STEP 模型映射关系, 如 EXPRESS-M, EXPRESS-V, EXPRESS-X 等. M.Hardwick 建议在虚拟企业的构建中采用 EXPRESS-X 语言实现异构模型的映射与信息共享^[11]. EXPRESS-X 是对 STEP 建模语言 EXPRESS 的扩充, 可用于定义 EXPRESS 数据模型之间的静态映射, 是 STEP 模型映射的标准化工具. 本文讨论的映射采用 EXPRESS-X 语言描述.

EXPRESS-X 模式经编译器编译后, 生成两种数据, 一种是描述其 MAP 结构和操作的数据字典, 另一种是约束函数的 C++ 代码. 数据字典记录了每个 MAP 的属性、执行映射的操作以及与 MAP 对应的约束函数名. 约束函数代码由 C++ 编译器编译成动态库, 可与模型映射器实现动态联编. 利用约束函数可以对映射数据的内存映像进行约束满足检查.

由于 EXPRESS-X 只能描述静态映射, 对于动态映射类需要提供映射选择策略, 同时, 映射模式中必须包含它在所有语义上下文中的静态映射. 在处理动态映射时, 模型映射器根据源数据的上下文环境和映射策略动态选择映射模式中的 MAP, 并执行映射. 为保证系统描述的一致性和模型映射器的通用性, 动态映射策略库用 EXPRESS 语言描述, 并编译为 C++ 动态库的形式由模型



EXPRESS-X 映射模式, EXPRESS-X 编译器, 数据字典, 约束函数动态库, C++ 编译, 约束函数源代码, 目标数据, 动态映射策略库, 模型映射器, 标准数据存取界面, 源数据, 数据流, 调用关系.

Fig.1 Model mapping system for STEP

图 1 STEP 模型映射系统

述的一致性和模型映射器的通用性, 动态映射策略库用 EXPRESS 语言描述, 并编译为 C++ 动态库的形式由模型

映射器调用.

若以标准数据存取界面(SDAI)访问 STEP 数据,则系统的体系结构如图 1 所示.

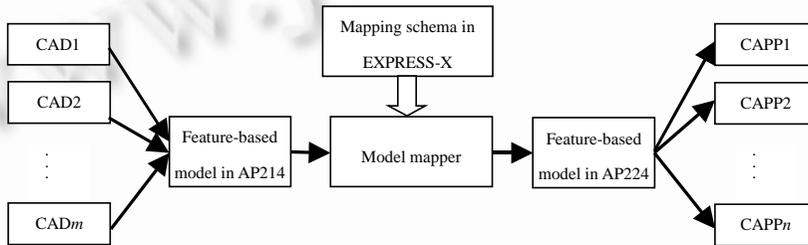
3.2 在开放环境中的典型应用

随着 STEP 标准在制造业的应用日益广泛,异构系统间的信息集成往往归结为不同 STEP 协议间的映射.因此,STEP 模型映射技术在 CIMS 环境中有着广泛的应用前景.本节阐述其中的一种典型应用.

由于设计特征与制造特征之间存在较大差异,进行 CAD/CAPP 信息集成时必须把 CAD 特征模型转换为 CAPP 特征模型.假设要在 m 个 CAD 系统与 n 个 CAPP 系统之间进行数据交换,那么软件设计的复杂度将是 $O(m \times n)$,原因在于每两个 CAD, CAPP 系统之间都要设计特征转换接口.

由于越来越多的 CAD/CAPP 系统将提供 AP214/224 接口,因此,只要能完成从 AP214 特征模型到 AP224 特征模型的映射,就可以实现大多数 CAD 与 CAPP 之间的特征信息集成^[12].采用 STEP 模型映射工具可以方便地实现这一目标.

如图 2 所示,利用 EXPRESS-X 构造特征映射模式,它记录了两种特征模型之间的映射关系.模型映射器通过 SDAI 读取 AP214 特征模型,在特征映射模式的支持下输出符合 AP224 标准的特征模型.CAD, CAPP 软件通过各自的 STEP 接口分别输出和输入特征数据,而不必考虑与其他系统的差异,从而实现了两者之间的透明集成.系统无关性使模型映射器可应用于多种 CAD/CAPP 系统的集成.



特征映射模式(EXPRESS-X), AP214 特征模型, 模型映射器, AP224 特征模型.

Fig.2 Integration of CAD/CAPP based on STEP model mapping

图 2 基于 STEP 模型映射的 CAD/CAPP 信息集成

4 结 论

STEP 模型映射技术为采用不同应用协议的系统间集成提供了标准接口,是对传统 STEP 技术的扩充.本文在分析 EXPRESS 数据模型的基础上论述了两种 STEP 模型映射技术——静态映射与动态映射.静态映射只考虑 EXPRESS 类的结构和类本身的语义;动态映射建立在静态映射的基础上,并考虑了映射语义上下文,涉及到对 EXPRESS 约束机制的处理.在 CIMS 集成环境中,这两种技术有着各自不同的应用背景,应根据实际需要选择合适的模型映射方式.

References:

- [1] Szykman, S., Fenves, S.J., Keirouz, W., *et al.* A foundation for interoperability in next-generation product development systems. *Computer-Aided Design*, 2001,33(7):545~559.
- [2] ISO TC184/SC4. Industrial automation systems and integration——product data representation and exchange——Part 1: overview and fundamental principles. ISO 10303-1, 1994.
- [3] Liang, J., Shah, J.J., D'Souza, R., *et al.* Synthesis of consolidated data schema for engineering analysis from multiple STEP application protocols. *Computer-Aided Design*, 1999,31(7):429~447.
- [4] Burkett, W.C. Product data markup language: a new paradigm for product data exchange and integration. *Computer-Aided Design*, 2001,33(7):489~500.
- [5] Oh, Youchon, Han, Soon-hung, Suh, Hyowon. Mapping product structures between CAD and PDM systems using UML. *Computer-Aided Design*, 2001,33(7):521~529.

- [6] Zhang, Y., Zhang, C., Wang, H-P. Interoperation of STEP application protocols for product data management. *Concurrent Engineering: Research and Applications*, 1998,6(2):161~169.
- [7] Yang, Zi-jiang, Yang, Xiao-hu, Li, Shan-ping, *et al.* Design and implementation of GS-STEP: an open environment-oriented integrated STEP toolkit. *Journal of Computer Aided Design and Computer Graphics*, 2001,13(11):1002~1007 (in Chinese).
- [8] ISO TC184/SC4. Industrial automation systems and integration——product data representation and exchange——Part 14: EXPRESS-X mapping language. ISO 10303-14, 1999.
- [9] ISO TC184/SC4. Industrial automation systems and integration——product data representation and exchange——Part 11: description methods: the EXPRESS language reference manual. ISO 10303-11, 1994.
- [10] Yang, Zi-jiang, Li, Shan-ping. Application of model conversion techniques in STEP. In: *Proceedings of the 6th International Conference on CAD&CG*. 1999. 191~195.
- [11] Hardwick, M., Morris, K.C., Spooner, D.L., *et al.* Lessons learned developing protocols for the industrial virtual enterprise. *Computer-Aided Design*, 2000,32(2):159~166.
- [12] Yang, Zi-jiang, Yang, Xiao-hu, Li, Shan-ping, *et al.* Research on STEP feature model and its application in the integrated environment of CIM. *Journal of Computer Aided Design and Computer Graphics*, 2001,13(5):438~443 (in Chinese).

附中文参考文献:

- [7] 杨子江,杨小虎,李善平,等.面向开放环境的集成化 STEP 工具集 GS-STEP 的设计与实现.计算机辅助设计与图形学学报,2001,13(11):1002~1007.
- [12] 杨子江,杨小虎,李善平,等.STEP 特征模型及其在集成环境中的应用研究.计算机辅助设计与图形学学报,2001,13(5):438~443.

Model Mapping Techniques for STEP and Its Application in Open Environment*

YANG Zi-jiang, LI Shan-ping, DONG Jin-xiang

(Department of Computer Science and Engineering, Zhejiang University, Hangzhou 310027, China)

E-mail: yzj@cad.zju.edu.cn

<http://www.zju.edu.cn>

Abstract: To achieve the data exchange between heterogeneous systems introduced by interoperability, the algorithms for model mapping in STEP are presented, which provide a methodology to the integration of CAX systems based on various application protocols of STEP. A formal definition of STEP/EXPRESS data model is introduced firstly. Based on this definition, two kinds of model mapping techniques for STEP, i.e. the static mapping and the dynamic mapping, along with the implementation by using EXPRESS-X mapping language are elaborated. Finally, a typical application of this technique in the open environment is addressed. Being independent of any specified system, the model mapping techniques presented in this paper achieve transparent integration between heterogeneous systems, which reduce the cost of information integration in system development.

Key words: STEP; EXPRESS; CIMS; model mapping; open environment

* Received December 9, 2000; accepted July 12, 2001

Supported by the National Natural Science Foundation of China under Grant No.69684006; the National High Technology Development 863 Program of China under Grant No.863-511-820-019