

装配知识库模型及其核心设计*

余卫东 陆玉昌 张 钺

(清华大学计算机系,北京 100084)

摘要 本文分析了研究和建立基于智能机器人装配系统的知识库的必要性,并探讨了实现途径.接着,我们提出了支持装配知识库的知识模型及其核心的体系结构,并讨论了支持该知识模型的建模工具 EXPRESS 语言.最后,给出结论.

关键词 知识模型,装配知识库,数据与知识,装配系统,EXPRESS 语言.

为了实现产品的自动装配,智能机器人装配系统除了需要关于产品的信息之外,还需要关于机械手、装配工具、周边装置等几何和物理特性,工作空间的布局等大量信息,以及装配工艺的经验知识^[1-3].这些数据 and 知识不仅种类繁多、数据浩大、冗余度高,而且难以量化,这些特点使其难以为应用程序所有效地存储、存取和充分利用.因此,如何管理装配数据与知识,为装配规划和实时执行提供必要的信息,是我们面临的主要问题.

目前,尽管工程图纸广泛应用于工业界,同时越来越多的 CAD 系统走向成熟、实用,但它们均不能满足装配规划与实时执行的需要.因为工程图纸所包含的大量的数据和知识,很难为计算机所利用,CAD 系统提供的大多是较低层次上的几何信息,无法满足机器人装配系统对较高层次的装配数据和知识的需求.解决这个问题的常用办法是:搜集一些零件产品的特征,通过用户接口(UI)输入计算机内,存储在一个或多个文件中;然后结合零件的 CAD 数据模型,在装配规划和自动执行时应用这些特征信息^[1-3].不过,该方法的缺点是:由于大量的数据和知识缺乏有效的管理,可能会引起开销、效率和复杂性等技术问题^[4].

因此,对装配数据和知识建模,建立装配系统的知识库(简称装配知识库,或 AKBs)是必要的.这些装配知识库具有下列特点:(1)并发控制相对简单;(2)没有版本控制的要求;(3)支持 CAD 数据类型,如零件的 B-Rep 和 CSG 表示;(4)支持复杂对象的结构与语义,自然包括表示与推理;(5)知识库必须尽可能小,与装配系统紧耦合,支持装配规划与实时执行.

很明显,传统的关系数据库管理系统(RDBMS)不能满足这些要求,主要是因为它们不能支持复杂对象的存储与管理,同时通常因为系统较大,不能有效地支持实时应用.目前,对

* 本文 1993-12-09 收到,1994-03-30 定稿

本研究项目得到 863 高技术支持.作者余卫东,1966 年生,助教,博士生,主要研究领域为数据库技术,知识库.陆玉昌,1939 年生,副教授,主要研究领域为知识工程,机器学习.张钺,1935 年生,教授,博士生导师,主要研究领域为计算机应用,人工智能.

本文通讯联系人:余卫东,北京 100084,清华大学计算机系

于我们而言,购买现有的 OODBMS 产品(如 ONTOS、ObjectStore 等),作为装配知识库的对象存储管理子系统也是困难的.这就是我们为什么要设计并实现一个支持装配知识库的核心系统的原因.

在本文中,我们把重点放在知识模型、知识库系统核心的体系结构和设计讨论上.文章第 1 节回顾设计背景,并讨论了系统体系结构;第 2 节描述对象模型的语义,将其作为知识模型;第 3 节讨论知识定义语言接口 EXPRESS;最后给出结论.

1 背景与系统体系结构

过去 10 年,我们针对智能机器人装配系统开展了大量的工作,并建立了一个实验原型系统 SAROT^[2],它包括装配序列规划子系统(SarotASP)、装配操作调度子系统(SarotOS)、装配过程图形仿真子系统(SarotGS)以及基于 VME 总线多处理机实时控制子系统(SarotRTC)四部分.目前,我们是利用清华大学 CAD 中心开发的 GEMS3.0 实体建模软件包作为装配件的 CAD 数据模型.不过,GEMS3.0 仅仅只能产生低层几何信息,生成图形模拟显示中必要的几何图形.建立 AKBs 的目的就是为了行之有效地管理和维护同装配件相关的数据和知识,如零件的装配特征等.SAROT 原型系统、装配知识库系统(AKBS)和 GEMS3.0 的关系如图 1 所示.

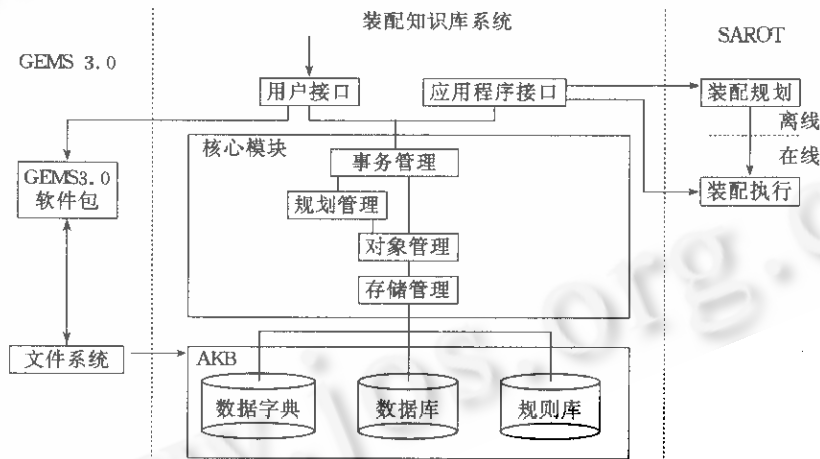


图1 SAROT、KBS和GEMS3.0的关系

其中,装配知识库系统包含接口、核心模块和装配知识库三部分.

接口分为用户接口(UI)和应用程序接口(API).用户接口包括三部分:知识定义语言接口、知识操作语言接口和几何建模辅助工具.我们选择了 EXPRESS 语言作为知识定义语言接口(将在第 3 节详细讨论),扩展的 SQL 语言作为知识操作语言接口,通过它们,用户和知识库管理者(KBA)可以输入、查询和更新存储在装配知识库中的信息.同时,用户和 KBA 还可以利用几何建模辅助工具操作 GEMS3.0,创建、删除和更新装配件的几何信息,存储在文件系统中,并可检入到 AKBs 中.通过应用程序接口,AKBs 可以装配规划和实时执行提供必要的数据和知识.

核心模块包括事务管理、规则管理、对象管理和存储管理四部分.从全局层来看,装配知

识库的一致性,也就是通过 UI 和 API 输入、更新和存取装配数据和知识的正确性,是由事务管理模块予以保证的(相关技术参见文献[5]).规则管理子系统建立了复杂对象推理的统一框架(详细讨论参见文献[6]).对象管理子系统主要负责复杂对象的存取与操作,提供对象的高层语义,如继承等特性.存储管理是物理数据库设计一个重要环节,其目的是为了快速存取和管理存储在 AKBs 上的数据和知识,该模块包含索引管理、数据缓冲区的管理以及多存取路径的支持等功能.

知识库包括三部分:数据词典、数据库和规则库.数据词典内存放的信息包括数据模式描述(包括类型定义、实体定义、通用计算过程)、授权以及数据库和规则库之间的联系等.数据库主要用来存储合法的复杂对象、索引、计算过程参数等信息.规则库主要用来存储语义约束、推理规则以及存取控制策略等信息.

2 知识模型

设计一个知识模型的目的,就在于允许用户和知识库管理者(KBA)直接、无歧义地描述一个知识库的语义特性,并允许建立在该模型之上的 KBMS 有序、一致地处理和操作知识库,以达到保护知识语义的目的.语义在这里指的是装配数据与知识的结构特性、一致性约束和与之关联的操作规则.因此,为智能机器人装配系统设计一个知识模型,或者说评估这样一个模型,我们必须覆盖必要的语义特性,以便直接定义能支持装配规划和自动装配执行的装配数据和知识.

通常,语义数据模型能较好地满足知识模型的需求.如日本的富士通有限实验室基于改进的函数数据模型(FDM),设计并实现了一个面向对象的多媒体 KBMS^[7].迄今为止,已有许多语义数据模型得到了发展,如 E-R^[8]、广义数据模型^[9]、RM/T^[10]、SAM*^[11]、TAXIS、SDM、Functional Data Model、Event Data Model、SHM+等^[12,13].我们针对这些模型作了系统的分析和比较,认为 SAM* 能更好地适应智能机器人装配系统,并结合装配知识库的特性,作了较大的改进.

2.1 知识模型的特征

2.1.1 复杂数据类型

将现有的商用 DBMS 用于装配系统,最明显的不足也许就是它们有限的数据类型.通常,仅仅只有一些基本数据类型,如整型、浮点型、字符型和布尔型可以识别,更复杂的数据类型,如矢量、矩阵、集合、日期、结构通常不能被这些数据模型或 DBMS 所处理,而是在应用程序中通过宿主程序编程实现的.在装配知识库中,具有复杂数据类型的数据和知识是很普遍的,应该能够直接处理.基于这个理由,该知识库核心应该能直接应用隶属这些复杂数据型的操作去操纵数据和知识.在该模型中,我们把数据类型分为以下四类:

(1)基本数据类型

基本数据类型包括数型 NUMBER(整型 INTEGER,短整型 SHORT,长整型 LONG,浮点型 FLOAT,双精度浮点型 DOUBLE),逻辑型 LOGIC,字符型 CHAR(定长字符串),字符串型 STRING(变长字符串),文件型 FILE 等.

(2)构造数据类型

支持构造数据类型,为用户定义新的数据类型提供了一种有效的手段.定义构造数据类

型的原则是将语义上不可分割的最小数据单位作为一个基型的定义,其大小通常是可以静态确定的,其递归定义是极其有限的.例如,一个结构型的成员型不能是集合型,或大小不能确定的数组型和结构型.当然这不等于说,装配知识库的核心不支持复杂数据对象的递归定义,我们将在 2.2 小节中给出详细讨论.

$$\bullet \text{ 枚举型} \quad T = \text{Enum}\{\langle \text{string} \rangle, \dots, \langle \text{string} \rangle\} \quad (1)$$

其中, T 是被定义的型(以下同). 其域由多个字符串组成,每个字符串为不超过 20 位 ASCII 码,应用中取其一作为值.

$$\bullet \text{ 区间型} \quad T = \text{Between}(T_1, T_2) \text{ of } T_3 \quad (2)$$

其中, T_1 和 T_2 是相同的数型, T_3 是数型.

$$\bullet \text{ 偶对型} \quad T = \text{Cons}(T_1, T_2) \quad (3)$$

其中, T_1 和 T_2 相同的数型.事实上,偶对型是一个特殊的结构型,主要用来表示容差.我们将其抽取出来,作为一个构造数据类型定义和使用,有助于提高系统的效率.

$$\bullet \text{ 向量型和矩阵型} \quad T = \text{Vector}(\text{INTEGER}) \text{ of } T_1 \quad (4)$$

$$T = \text{Matrix}[\text{INTEGER}, \text{INTEGER}] \text{ of } T_1 \quad (5)$$

其中, T_1 是数型.

$$\bullet \text{ 集合型} \quad T = \text{set of } T' \quad (6)$$

$$T = \text{Dset of } T' \quad (7)$$

$$T = \text{OSet}(\text{INTEGER}) \text{ of } T' \quad (8)$$

其中, T' 是基型,它或者是基本数据类型,或者是日期型,或者为大小固定的预定义的结构数据类型((9)式的 T' , (10)式的 T_1, T_2, \dots, T_n 含义相同).上述三个定义的不同之处就在于:针对型(6),知识库不允许重复元素的存在;针对型(7),重复元素是许可的;针对型(8),元素有序且数目有限.

$$\bullet \text{ 数组型} \quad T = \text{Array}[\text{INTEGER}] \dots [\text{INTEGER}] \text{ of } T' \quad (9)$$

从语法上,数组的维数不受限制;同有序集相比,数组型元素的位置是固定的.

$$\bullet \text{ 结构型} \quad T = \text{Struct} \{A_1; T_1, A_2; T_2, \dots, A_n; T_n\} \quad (10)$$

其中, A_1, A_2, \dots, A_n 是属性名, T_1, T_2, \dots, T_n 是基型.

(3) 日期数据类型

事实上,日期型(Data)是一种特殊的结构类型,它包含年、月、日、时、分、秒等信息.用户可以根据需要,可以定义相应的格式输入、更新和显示日期和时间.在存储子系统中,仅仅只需存储一个无符号整数,可以利用 UNIX 操作系统提供的系统函数 `localtime()` 和 `timelocal()` 实现从系统时间到用户时间和用户时间到系统时间的自动转换.

(4) 特殊数据类型

特殊数据类型包括计算型(Compute)、引用型(Refer)和规则型(Rule)等.在我们设计的核心系统中,当作知识处理,在此我们不作详细讨论,有关规则定义和知识管理等详细讨论,参见文献[6].

从上述讨论中不难发现,数据类型的定义是根据装配应用的需求所确定的.该知识模型可以较好地支持数据结构的分层递阶定义.例如,一个机械手的六个自由度可以用一个矢量表示;若该矢量的基型为偶对型,则该矢量不仅表示了六个自由度的大小,而且还表示它们

的误差范围,同时,装配应用程序可以通过应用程序接口(API)访问这些数据.

2.1.2 复杂对象

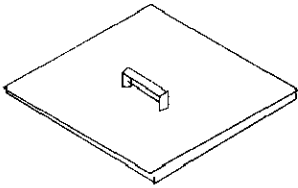


图2 装配件Plate

在知识库中,一个对象是由其成员确定的.例如,如图2所示,一个装配件 Plate 是由零件 Handle 和 Lidpart 通过一定的装配关系组成,而一个 Handle 又是两个 Peg 特征和一个 Bridge 特征组成.用规范的关系是难以表达这种复杂的装配关系的,这是因为规范关系的一个属性值必须是一个原子元素,不可能是原子元素的集合或者复杂对象.因此,要求装配知识库的核心支持复杂对象的存储和处理是必要的.

2.2 四种语义关联

本文提出的四种语义关联是从 Su 提出的 SAM * 模型^[11]中的七种语义关联发展而来的.结合智能机器人装配系统的特点,无论是在定义上,还是表示方法上,我们都作了较大的相应改进.

• 成员语义关联

成员语义关联是指一组同构原子概念组成的集合.该集体成员有相同的数据类型,并形成一个论域.定义在该关联语义的成员约束指定了属于该域的合法成员.该语义关联建模了自然语言的“成员”(is a member of)或“示例”(is an instance of)语义.

在核心系统中,支持复杂数据类型定义的目的,一方面是为获取必要的语义特性,并直接用于定义装配数据;另一方面是为了定义更明确的成员语义约束.

有时,仅仅支持构造数据类型的定义作为一个论域的成员约束是不够的,这就是该模型为什么要支持特殊数据类型(Compute 型, Rule 型等)的原因之一. Compute 型是和论域相关联的,其值可以由一个计算公式导出,也可以由一些预定义的计算过程对其参数赋值求得. Rule 型引入装配知识库,一方面是为了更好地定义约束;另一方面是为了支持复杂对象的推理,其详细讨论参见文献[6].规则可以当作数据类型处理(查询与更新),在运行时当一定的数据条件被满足,规则被触发执行.一条规则的执行可能又会激活其它规则的执行.该知识模型支持前向链(forward-chaining)推理.

成员语义关联在网络表示中,用结点 M 表示.

• 聚合语义关联

一个概念(一个物理对象、抽象事物或者事件)可以根据其属性或特征进行定义.我们将一组属性或特征定义另一概念,称之为聚合语义关联.我们认为聚合语义关联提供了一种将低层次型之间的关系抽象为高层次型的手段,如部件与零件、零件与特征之间的语义关系,就属于聚合语义关联.如图3所示,

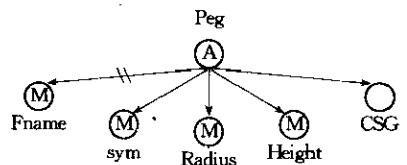


图3 特征Peg的聚合定义

一个 Peg 特征可以定义为五个属性:Fname、Sym(对称组)、Radius、Height 和 CSG 表示的聚合语义关联,每个属性用一个自聚合结点的有向弧指示的结点表示,其属性值由其指向结点的论域所确定,每个属性有一不同于域的名和数据型(即定义于其上的结构),属性名和数据型可用来共同标识对应的弧.

在网络表示中,聚合语义关联用结点 A 表示,它表示一元组集(类似关系数据库意义下的元组),是其成员属性的论域集的笛卡尔乘积的真子集.键属性用标识弧(\rightarrow)表示.

同时,聚合语义关联反映一个对象和其成员的关系.通常不是任意几个成员均可以构成一个对象,一个对象是由其成员通过一定的语义关系所组成.语义的表达是定义聚合语义关联的难点.我们将给出一个更有代表性、更完整的例子(参见例 1).

• 引用语义关联

假定存在两个类 T_1 和 T_2 , $I(T_1)$ 和 $I(T_2)$ 分别是其对应的示例集,对每个子集 $I_{11} \subset I(T_1)$,一定存在一个相应的子集 $I_{22} \subset I(T_2)$,使得 I_{11} 和 I_{22} 之间存在引用和被引用关系,我们则称类 T_1 和 T_2 之间存在引用语义关联.类 T_1 的示例称为主对象, T_2 的示例称为属对象.

在装配知识库中,被引用的对象(实体)通常当作主对象(或主实体)的成员定义,这与装配知识库支持 Refer 型的定义是一致的.装配知识库支持对象标识符(OID),OID 能将一个对象和其它对象加以区分,满足唯一命名假设(UNA),它具有将对象看成自己本身,而不是依靠它的值而存在.由于装配知识库支持 OID、Refer 型和引用语义关联,尽管数据型的递归定义是有限的,而复杂数据对象的递归定义却是无限的.在网络表示中,引用结点 R 表示一个被引用的对象,而它通常是由另一个聚合语义关联单独表示的.

• 概括/从属语义关联

设 T 为一类对象, $\{Y\}$ 是不含 T 且元素个数大于等于 2 的对象类集合.若 T 表示的语义是 $\{Y\}$ 所有元素的语义的共同语义抽象,则称 T 和集合 $\{Y\}$ 之间存在着概括语义关联, T 称为超类, $\{Y\}$ 中各元素称为子类.这种语义关联反映了自底自上的模式设计方法.如果采用自顶向下的方法设计模式,保持继承性,我们则称子类与超类之间的语义关联为从属语义关联.

事实上,概括语义关联和从属语义关联是相对的,它们均反映了共性与个性语义关系,它们的示例是从不同抽象层次描述的另一事物.在网络表示中,我们用结点 G 表示,出自一个超类结点的两条有向弧所指向的两个结点,一定存在下列四种关系之一:集合相等(SE—SE)、集合互斥(SX—SX)、集合与子集(ST—SS)或集合相交(SI—SI).我们用 IS—A 有向弧表示从属语义关联.

从概括语义关联,我们可得到自身内部属性信息、子类信息以及属于该类的任意两个子类之间的关系.同时,从从属语义关联,我们不仅可以得到一个子类自身定义的成员属性信息,而且还可以获悉从其超类可继承哪些静态属性和动态特性(如规则、过程等).在装配知识库中,对象之间的语义关系是复杂的,支持多继承是必要的.事实上,从属语义关联和多继承是不相冲突的,一个超类可以有多个子类;相反,一个子类也可以有多个超类.自然,多继承特性可能会引起命名冲突,我们在实现对象管理子系统时,采用属性重定位的方法已经解决了这个问题,因篇幅有限,在此不作讨论.

例 1:我们列举装配零件的几个特征类,如表 1 所示:

表 1 几个常见的特征表

特征类	特征属性说明
feature	CSG, B-Rep, mmc, additive, sym
gear	radius, num-teeth
peg	radius, height, bevel radius bevel height
threaded-peg	radius, height, chamfer radius, chamfer height
hole	height

采用四种语义关联,我们可以给出表示特征类的分层递阶结构,如图 4 所示.

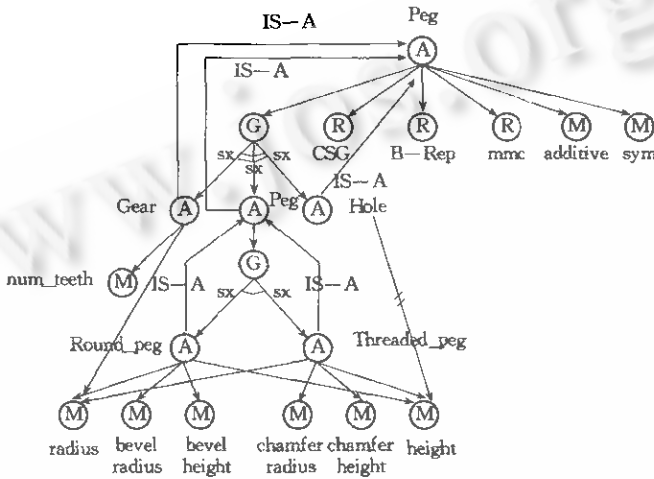


图4 几个特征类的分层递阶结构

3 知识定义语言接口——EXPRESS 语言

EXPRESS 语言 IPO (IGES/PDES Organization) 开发的一种功能很强的信息建模语言,为描述产品的数据结构和行为特性提供了一个很好的工具,并在构造产品信息模型上起了关键作用.我们认为 EXPRESS 语言非常适合描述复杂对象的成员和分层递阶结构.事实上,知识模型的设计、知识库核心的实现,与 EXPRESS 语言的语义和功能是一致的. EXPRESS 语言具有下列基本功能:对感兴趣的事物建模,对规则建模,定义施加于事物上的约束;对操作建模,定义事物参与的操作.

EXPRESS 语言的描述元素和层次结构为:

```

SCHEMA 模式名
  TYPE
    型体;
  END_TYPE;
  ...
  ENTITY 实体名
    实体体;
  END_ENTITY;
  ...

```

```

FUNCTION 函数名
    函数体;
END_FUNCTION;
...
RULE 规则名
    规则体;
END_RULE;
...
END_SCHEMA;

```

其中,TYPE 块允许用户定义和使用知识模型所许可的数据类型;ENTITY 块允许用户使用知识模型所许可的语义关系对复杂对象建模,包括其超类与子类的定义、自身成员属性的定义、一个或多个属性约束的定义、以及语义关联的定义;FUNCTION 块能实现特殊数据类型 Compute 的功能;RULE 块允许用户定义施加于实体上的约束。

例2:我们用 EXPRESS 语言描述例1所列举的特征如下:

```

SCHEMA feature_class
    TYPE
        RADIUS: cons (INTEGER, INTEGER)
    END_TYPE;
    ENTITY gear
        superclass: feature;
        radius: RADIUS;
        num_teeth: INTEGER;
    END_ENTITY;
    ENTITY peg
        superclass: feature;
        subclass: round_peg XOR threaded_peg;
    END_ENTITY;
    ...
END_SCHEMA

```

在知识库核心中,我们提供一个编辑界面,允许用户书写和更新 EXPRESS 语言程序,经过词法分析、语法检查和语义分析,将 EXPRESS 语言转换生成调度执行计划(SEP),提交给事务管理模块处理,最后将数据模式以及相应的通用计算过程存放在数据词典内,规则存放在规则库内。EXPRESS 语言处理部分,我们是用 UNIX 操作系统提供的 LEX 和 YACC 工具完成的。

4 结 论

考虑智能机器人装配系统的特点,我们提出了一个与 EXPRESS 语言语义一致的知识模型,并基于该模型,建立了一个支持装配知识库的核心。在该核心中,存储管理子系统、对象管理子系统和规则管理子系统是紧密集成的。建立在该核心之上的装配知识库具有下列特点:并发控制简单,支持 CAD 数据模型,支持复杂对象的推理,能和装配系统紧耦合,支持装配规划与实时执行。

我们相信,通过与装配知识库、GEMS3.0集成,SAROT 系统将变得更加成熟与实用。初步实验表明,用装配知识库管理的数据和知识有下列优点:层次清晰、扩展、维护容易等。

参考文献

- 1 Gottschlich S N, Kak A C. AMP—CAD: automatic assembly motion planning using CAD models of parts. Techni-

- cal Report. TR-EE-91-12, School of Electrical Engineering, Purdue University, 1991.
- 2 鲁涛. 智能装配机器人系统——体系结构、任务规划和原型系统[博士论文]. 清华大学, 1993.
 - 3 滕少华, 余卫东, 陆玉昌. 面向对象的装配数据与知识表示及处理. 刘大有编著, 人工智能学术会议'93论文集, 人工智能学术会议'93, 南京, 1993, 长春: 吉林大学出版社, 1993. 281-289.
 - 4 Yagi T. Modelling design objects and processes. Berlin Heidelberg: Springer-Verlag, 1991. 4-5.
 - 5 余卫东. 超微机分布式数据库系统 WDDBS-32 的分布事务管理[硕士论文]. 武汉大学, 1991.
 - 6 余卫东, 陆玉昌, 张钹. 工程数据库系统中的知识管理. 计算机研究与发展, 1994, 31(6): 35-40.
 - 7 Ishikawa H, Suxuki F *et al.* The model, language, and implementation of an object-oriented multimedia knowledge base management system. ACM Transactions on Database Systems, 1993, 18(1): 1-50.
 - 8 Chen P P-S. The entity-relationship model: toward a unified view of data. ACM Transactions on Database Systems, 1976, 1(1): 9-36.
 - 9 陈其明. 广义数据模型[博士论文]. 清华大学, 1988.
 - 10 Codd E F. Extending the Database Relational Model of Capture More Meanings. ACM Transactions on Database Systems, 1979, 4(4): 397-434.
 - 11 Su S Y W. Modelling integrated manufacturing data with SAM*. IEEE Computer Magazine, 1986, 19(1): 34-49.
 - 12 Hull R, King R. Semantic database modelling: surveys, applications and research issues. ACM Computing Surveys, 1987, 19(3): 201-260.
 - 13 Peckham J, Maryanski F. Semantic data models. ACM Computing Surveys, 1988, 20(3): 153-189.

THE MODEL AND DESIGN OF THE KERNEL OF ASSEMBLY KNOWLEDGE BASE SYSTEMS

Yu Weidong Lu Yuchang Zhang Bo

(Department of Computer Science, Tsinghua University, Beijing 100084)

Abstract In order to efficiently manage data and knowledge related with assemblies, and supply necessary information for assembly planning and real-time execution, it is necessary to build the knowledge bases for intelligent robotic assembly systems (also called assembly knowledge bases). In this paper, after the research background is introduced, the system architecture and the knowledge model are proposed; then, the knowledge definition language interface EXPRESS language is described, which is consistent with the knowledge model. Finally, this paper gives the conclusion.

Key words Knowledge model, assembly knowledge bases, data and knowledge, assembly systems, EXPRESS language.