

Abstract

With the development of Computer Aided Design (ACD), it is increasingly urgent to study the computer management of engineer data. New data models have been the focus of many researchers to overcome the traditional data models' deficiency of not meeting the describing and management of engineer data. Several kinds of data model have been proposed and some EDBMS experimental systems have been established. Among those models, the object-oriented data model has some outstanding advantages, which will be the mainstream for establishing an engineer database in the future. In this thesis, the object-oriented data model and the realization technology of engineer database management system are deeply studied.

The engineer data are analyzed according to the manage requirement of mechanical CAD engineer data. The engineer data are classified four: management data, graph data, design data and soft package data. Definition of each class is given as well as the features and the differences from commercial data.

The concept of part feature is discussed. The part information model is built based on the part features. Different kinds of part have different information models. Similar methods are used to built the information model of assemble body from its character. These information models are applied to describe detailly the different instances.

In order to fully describe engineer data, the object-oriented technology are combined with the database technology. The definition of OODM and relative concepts are illuminated such as class, object, heritage, attribution, operation and identification. Three basic methods to construct object are discussed: sets, relation and sequence. And the integrated application of above three methods is also introduced to

construct various complicated objects. One object has both internal operations and external operations, which can realize the management and maintenance of corresponding object. To tell these basic concepts clearly and sharply, the axis, gear and other instance objects are exemplified. In order to efficiently manage engineer data, the system construction of engineer database management system is designed based on OODM, and concerning critical technology are deeply studied. Guided by preceding researches, the user interface, the method of establishing and maintenance of system base, the method of management of sub system base, the model description of active base data and the management of relative engineer data are all realized by VC 6.0, among which emphases are the ODBC transfer system and the system establishing of axis parts in active base.

An experimental EDBMS is set up, which is used to describe and manage the engineer data of axis parts during design. From that, the effectiveness of OODM, the feasibility of system construction of EDBMS and the feasibility of its realization technology are proved.

Key Words engineer data; information model; object-oriented data model;
engineer database management system

第 1 章 绪论

本章首先介绍数据库技术的基本概念以及传统数据模型和它的局限性，然后介绍面向对象数据模型以及工程数据库技术研究的现状，并分析确定论文的题目和研究内容。

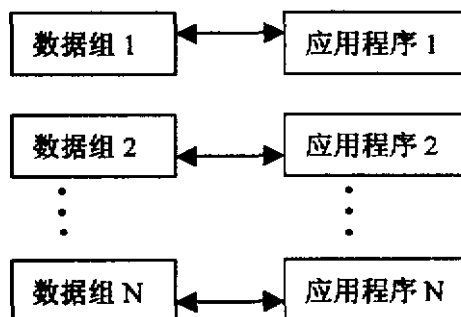
1.1 数据库技术基础

1.1.1 数据管理技术的发展

在人们的日常生活和工作中，常常接触到大量的数据。为了充分地利用这些数据，必须进行有效地组织、编目、定位、存储、检索和维护，这是数据管理的核心任务。数据管理技术经历了一个由简单到复杂的不断完善的发展过程，大致可分为三个阶段。

1. 人工管理

在 50 年代中期以前，用户进行数据处理时，不仅要编制自己的课题程序，而且还必须考虑数据的逻辑定义和组织以及数据在计算机内的物理存贮方式。数据的引用完全按物理地址由编码者设定。数据管理基本是手工的、分散的，严重地影响了计算机的使用效率。这个阶段的数据与程序的关系如图 1-1 所示：数据和程序是一一对应的，程序之间不能共享数



据，数据存在大量的冗余，不利于对大量数据的管理。

2. 文件管理

从 50 年代后期到 60 年代中期，人们把数据组织成文件的形式进行管理。文件管理是将数据按照一定的规则组织起来，成为一个有序的数据组合体，且赋予

它一个名字，称为文件名或文件标识。这种情况下用户可不必考虑文件组织的物理细节，而由文件系统充当应用程序与数据文件之间的接口，实现数据与应用程序的分离，具有一定的独立性，较人工管理先进得多。其相互关系如图 1-2 所示。

虽然文件管理比人工管理具有较大的先进性，但是仍存在较多的

缺点：

(1) 文件基本上还是对应于一个或多个应用程序，数据还是面向应用，缺少通用性。

(2) 从逻辑上讲，与人工管理没有多大变化。

(3) 数据冗余度大，存储空间浪费严重。

(4) 文件不易扩充。

3. 数据库管理

从 60 年代后期开始，磁盘技术日趋成熟，并广泛地作为主要外存应用，为数据管理技术的发展奠定了物质基础。由于计算机大量应用于管理和计算，从而处理的数据量急剧增加，规模庞大，对数据进行集中管理并充分提供数据共享的要求日益迫切。在

这样的背景下，产生了新的数据管理技术——数据库技术。数据库技术是用数据库管理软件作为用户程序与数据的接口。数据库管理软件能完成

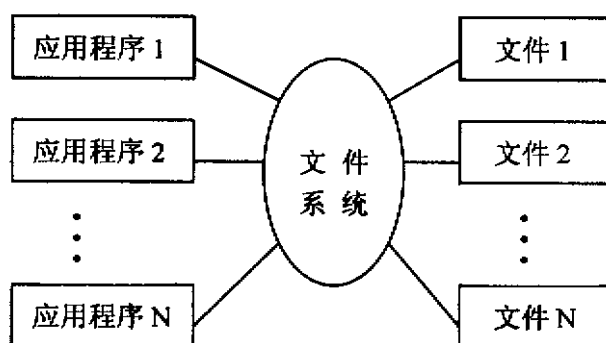


图 1-2 文件管理数据与应用程序的关系

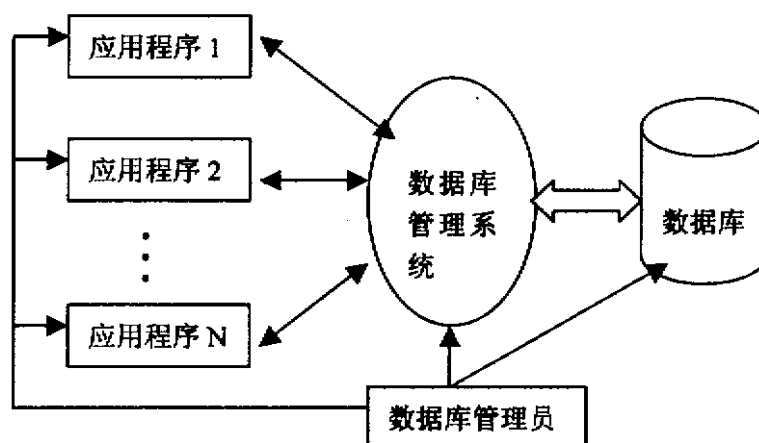


图 1-3 数据库系统示意图

绝大部分数据的管理、维护和操作,保证数据的有效性、完整性和一致性。程序与数据的关系如图 1-3 所示,其优点:

- (1)解决了多用户共享,能实现数据的集中管理。
- (2)较高的数据独立性,可以各自独立设计和维护。
- (3)便于为各种数据提供安全保护措施,避免数据被破坏。
- (4)减少了数据冗余度,充分利用了物理存储空间。

1.1.2 数据库技术的基本概念

要掌握数据库技术,必须理解数据库技术的一些基本概念。

1. 数据、数据处理和数据管理

数据是描述现实世界中各种具体事物或抽象概念的可存储并具有明确意义的代码组合。数据处理是对数据施加各种有效操作以改变其状态的行为。数据管理是通过对数据施加操作使数据保持有序或方便使用的状态。从这些概念可看出数据是基础,是数据处理和数据管理的对象,数据管理包含数据处理。

2. 数据库、数据库管理系统和数据库系统

数据库(Data Base——DB)是为满足某一组织中多个用户的多种应用需要,在计算机系统中由数据库管理系统按照一定的数据模型建立、运用、管理和维护的相互联系的数据集合。数据集合指关于企事业运行的所有业务数据的集成。数据库管理系统(Data Base Management System——DBMS)是一组专门处理访问数据库的程序,它能够对数据库进行有效的管理,是数据库系统的核心。它的功能包括存储管理、安全性管理以及完整性管理。数据库系统(Data Base System——DBS)是组织、存取和维护数据库中数据的人机管理系统,是由计算机硬件、软件、数据库和有关人员组成的有机整体。数据库系统中的人员包括数据库管理员、系统分析员、应用程序员以及用户,软件包括操作系统、数据库管理系统应用开发工具及应用程序。

3. 数据模型

数据模型是描述数据的一组概念和定义。一般来讲,数据描述包括两方面,即对数据的静态特性和动态特性的描述。数据静态特性指数据的基本结构、数据间的联系以及数据间的约束,而数据的动态特性指定义在数据上的一组操作,完成数据的使用和维护。数据模型是数据库系统的基础。在数据库系统中,针对不

同的使用对象和应用层次，可采用三级数据模型。

(1) 概念数据模型

概念数据模型是针对用户、方便用户与数据库之间进行通信的最外层数据模型。一般非计算机专业用户很难理解面向计算机的逻辑和物理数据模型，采用概念数据模型使数据库设计人员在设计的开始阶段就能集中精力描述用户所要求的现实世界，以更好地满足用户需要。

(2) 逻辑数据模型

逻辑数据模型是用户从数据库中所看到数据的数据模型，如层次、网状和关系数据模型。它与 DBMS 有关，DBMS 常以所采用的逻辑数据模型进行分类。在概念数据模型中描述的数据必须转化为逻辑数据模型描述的数据，才能在 DBMS 中实现，逻辑数据模型既要面向用户又要面向实现。

(3) 物理数据模型

物理数据模型是数据存贮结构的数学模型，数据库中的数据最终必须存储在物理存贮介质上。每种逻辑数据模型实现时，必须有对应的物理数据模型。物理数据模型不但与 DBMS 有关，而且还与操作系统和硬件有关。

4. 数据模式

数据模式是按照给定的数据模型对具体的数据进行描述的结果，是数据模型在某一特定条件下的实例。在 DBMS 中，常常采用三级数据模式。

(1) 概念模式

概念模式是用逻辑数据模型对一个单位所用到的数据进行描述。概念模式设计是数据库设计的基本任务。概念模式又称为逻辑模式，它是数据库的全局模式，负责外模式与内模式之间的映射转换。

(2) 外模式

外模式是用逻辑数据模型对用户所用到的那部分数据的描述。由于每一个用户感兴趣的数据不一样，因此每一个用户的外模式不一定相同。外模式是由概念模式推导而来，主要针对特定用户设计。

(3) 内模式

内模式是用物理数据模型对数据的描述。概念模式与内模式之间可以互相映射，这个映射由 DBMS 完成。内模式对用户而言是透明的，它主要是用于数据库的

设计，其好坏直接影响数据库的性能。

1.1.3 数据库语言

每个数据库系统都为用户提供了一种数据库语言，用户可以使用这个语言定义和操纵数据库。数据库语言一般由两部分组成：数据定义语言和数据操纵语言。一般情况下，这两种语言以合并在一起的形式出现。数据库语言与数据模型紧密相关，具有不同数据模型的数据库系统具有不同的数据库语言。

1. 数据定义语言

数据定义语言（Data Define Language——DDL）是用来定义数据库模式、数据库存储结构以及存取方法的数据库语言。数据库定义语言对应的处理程序分为两部分。一部分称为数据库模式定义处理程序，它负责接收用 DDL 表示的数据库模式定义并把它转换为内部表示形式存储到称为数据字典的特殊文件中，同时负责数据模式的删除和修改。另一部分称为存储结构及存取方法定义处理程序，它接收用 DDL 表示的数据库存储结构和存取方法的定义，在存储设备上创建相关的数据库文件，建立物理数据库。

2. 数据操纵语言

数据操纵语言（Data Manipulate Language——DML）是用来表示用户对数据库的操纵请求的数据库语言。一般要求 DML 能够完成如下几种操作：

- （1）查询数据库中的信息。
- （2）向数据库中插入新的信息。
- （3）从数据库中删除信息。
- （4）修改数据库中的信息。

数据操纵语言根据它的表现形式可分为过程性语言和非过程性语言。过程性语言既要求用户说明取什么数据，又要求说明怎样搜索这些数据。非过程性语言则只要求用户说明取什么数据即可，同时比过程性语言易于学习和掌握，但是非过程性语言产生的处理程序的代码比过程性语言的代码多。

1.2 传统数据模型及其局限性

1.2.1 传统数据模型的类型

从数据库技术诞生之日起,对数据模型的研究就从未间断过。根据不同的应用目的提出了多种数据模型,一般把层次、网状以及关系数据模型称为传统数据模型。

1. 层次数据模型

1969 年 IBM 公司提出层次数据模型,并研制了基于层次数据模型的数据库管理系统 IMS (Information Management System)。层次数据模型是基于记录和字段描述现实世界中的层次关系的一种模型。

2. 网状数据模型

在 60 年代末 70 年代初,美国数据库系统语言协会下属的数据库任务组 (Data Base Task Group——DBTG) 对网状数据库方法进行了系统研究,提出了若干 DBTG 报告。这些报告确定并建立了网状数据库系统的许多概念、方法和技术。网状数据模型是以记录、系、数据项等基本概念描述数据。

3. 关系数据模型

1970 年 IBM 公司 San Jose 研究实验室的研究员 E. F. Codd 发表题为“大型共享数据库数据的关系模型”的论文,提出关系数据模型,开创了关系数据库方法和关系数据库理论,为关系数据库技术奠定了理论基础。在 70 年代开发了 System R、INGRES 等关系数据库系统,直到现在关系数据库系统仍被大量使用。

1.2.2 关系数据模型

1. 关系数据模型的基本概念

关系数据模型是以集合论中的关系概念为基础发展起来的数据模型,具有如下一些基本概念。

(1) 关系与元组

关系是指描述某一实体集合的属性集与各个个体值的二维表。其笛卡尔定义为: 设 $D_1, D_2, \dots, D_n$ 为任意一组集合, 定义 $D_1, D_2, \dots, D_n$ 的笛卡儿积为:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_1, \dots, d_n) / d_i \in D_i, i=1, 2, \dots, n\}$$

其中每一个元素 $(d_1, d_2, \dots, d_n)$ 叫一个元组, 元组是关系所表示实体集中的个体。元组中的每一个 d_i 叫元组的一个分量, D_i 叫 d_i 的值域。

(2) 属性和域

属性指描述某一事物的性质, 域指属性的取值范围。在关系的定义中指出关系的属性, 而模式定义中定义相应的域。属性名和域名可以相同, 也可以不同。

(3) 关键字

关键字是用以标识实体的属性。一个实体至少有一个关键字, 其中能唯一标识该实体的属性称为主关键字, 其余的关键字称为候选关键字。作为主关键字的属性不能是空值。

(4) 关系约束

关系约束指关系数据库中为了保证数据的完整性和有效性而给数据附加的各种约束条件。常用的有语义约束、域的完整性约束、实体完整性约束、引用完整性约束、一般完整性约束。

2. 关系运算

关系运算是实现关系模型中数据之间联系的运算, 是实现数据操作语言的基础。关系运算可分为两类: 一类是传统的集合运算, 如并、差、交运算及笛卡尔积, 另一类是针对关系专门设计的运算, 如投影、选择、联接及除法运算。现分述如下:

(1) 并运算。设关系 R 、 S 是同类关系, 则二者的并运算为:

$$R \cup S = \{t \mid t \in R \vee t \in S\}$$

其中, $\cup$ 为并运算符, t 为元组变量。并运算得到的关系是 R 与 S 元组的并集。

(2) 差运算。设关系 R 、 S 是同类关系, 定义二者的差运算为:

$$R - S = \{t \mid t \in R \wedge t \notin S\}$$

式中, $-$ 为差运算符。即是从 R 中减去与 S 相同的那些元组, 其结果仍是集合。

(3) 交运算。设 R 和 S 是同类关系, 其交运算为:

$$R \cap S = \{t \mid t \in R \wedge t \in S\}$$

式中, $\cap$ 是交运算符, 其结果仍为关系集合。

(4) 笛卡尔积。设 R 、 S 为关系集, 则二者的笛卡尔积:

$$R \times S = \{\langle t, g \rangle \mid t \in R, g \in S\}$$

式中, $\times$ 为笛卡尔积运算符, 其笛卡尔积仍为元组的集合。

(5) 投影运算。设 R 是 K 元关系, 其元组变量 $t^k = \langle t_1, t_2, \dots, t_k \rangle$, 那么关系 R 在其属性 $A_{j_1}, A_{j_2}, \dots, A_{j_m}$ ($j_1, j_2, \dots, j_m$ 为 1 到 K 之间互不相同的整数) 上的投影定义为:

$$\Pi_{A_{j_1}, A_{j_2}, \dots, A_{j_m}}(R) = \{t \mid t = \langle t_{j_1}, t_{j_2}, \dots, t_{j_m} \rangle \wedge \langle t_1, t_2, \dots, t_k \rangle \in R\}$$

式中, Π 是投影运算符, 其含义是从关系 R 中按 $A_{j_1}, A_{j_2}, \dots, A_{j_m}$ 的次序取出 m 列, 删除结果中的重复元组, 构成一个以 $A_{j_1}, A_{j_2}, \dots, A_{j_m}$ 为列次序的 m 元关系。

(6) 选择运算。设 F 是一个命题公式, 其运算对象是常量或元组分量, 运算符是算术比较运算符、逻辑运算符。则是关系 R (K 元关系) 上的 F 选择运算定义为:

$$\sigma_F(R) = \{t \mid t = \langle t_1, t_2, \dots, t_k \rangle \wedge t \in R \wedge F = \text{True}\}$$

式中, σ 为选择运算符, $\sigma_F(R)$ 表示从关系 R 中挑选满足 F 为真的元组, 新关系是原关系的子集且与原关系同类。

(7) 联接运算。设 R 是 K_1 元关系, S 是 K_2 元关系, θ 为算术比较符。则关系 R 、 S 关于 R 的第 i 列和 S 的第 j 列的 θ 联接运算定义为:

$$R \bowtie S = \{t \mid t = \langle t^{k_1}, t^{k_2} \rangle \wedge t^{k_1} \in R \wedge t^{k_2} \in S \wedge t^{k_1}_i \theta t^{k_2}_j\}$$

其中, $\bowtie$ 是联接运算符, 结果仍为关系集合。联接运算实际是有选择的笛卡尔积, 选择条件是 θ 。另外, 还有自然联接和自然半联接运算。

(8) 除法运算。设 R 是 K_1 元关系, S 是 K_2 元关系, 且 $K_1 > K_2$, $S \neq \phi$, 以及 S 的 K_2 个属性名与 R 的最后 K_2 个属性名相同, 则 R 除以 S 定义为:

$$R \div S = \{t \mid t = \langle t^{k_1}, t^{k_1}, \dots, t^{k_1-k_2} \rangle\}$$

式中, $\div$ 为除法运算符, 除法是笛卡尔积的逆运算, 其结果仍是关系集合。

1.2.3 传统数据模型的局限性

虽然传统数据模型在某些领域中得到广泛应用, 但是它具有较大的局限性, 不能应用到一些新的领域中, 主要表现在如下几个方面。

1. 以记录为基础的描述形式不能很好的面向应用程序。现在的程序设计大都采用面向对象的程序设计方法, 以对象为单位组织应用程序, 因此以记录为基础的数据描述形式必须转换成对象才能被应用程序使用。

2. 不能以自然的形式表示实体间的联系, 并且层次和网状描述靠物理手段实现, 破坏了物理独立性。现实中的实体都被抽象为记录的形式进行描述, 人们要

把记录经过转换才能恢复实体的自然形式。对层次和网状数据是通过关键字与物理存储地址建立映射联系进行描述,使记录的存储地址改变非常困难。

3. 语义贫乏,各个属性间的关系与作用不明确,实际语义靠用户掌握,给用户增加了负担。在关系数据模型中记录各个字段的意义由用户定义,甚至记录以及表之间的关系也需如此,给用户的使用带来了困难。

4. 数据类型太少,难以满足需要。在传统数据模型中,数据类型不能进行扩充,因此不管定义多少类型,它总是有限的。对于各种新的需求,它总不能完全满足需要。

5. 传统数据模型强调数据的结构化,具有单调性,无法表示现实世界中复杂对象。

1.3 面向对象数据模型的产生及研究现状

正如上一节所述因为传统数据模型具有一定的局限性,在七十年代后期,人们开始应用新技术探索新的数据模型,陆续出现了多种新的数据模型(通称为后关系数据模型),其中公认应用前景广泛的是基于面向对象技术的面向对象数据模型。本节综述其相关概念及该数据模型的发展现状。

1.3.1 面向对象技术基本概念

始于 60 年代中期的面向对象技术既是一种新的程序设计方法学,也是一种认知方法学,在 90 年代得到突飞猛进的发展。面向对象不仅是软件开发技术和策略,而且是关于如何看待软件系统与现实世界的关系,以怎样的观点来研究问题并进行求解,以及如何如何进行系统构造的方法学。

1. 对象(Object)

对象也称客体,可泛指任何事物,包括实体和抽象概念。对象是现实世界的基本单元。每个对象都有自己的属性,包括静态和动态两个部分。

2. 类(Class)

类指的是具有某些共性的对象的抽象模型。它描述了同类对象的特征和行为,是对象的数据类型。

3. 消息(Message)

消息是一个对象要求另一个对象执行某个操作的规格说明。在面向对象技术中,消息是用于对象与对象之间进行某种操作或响应的信息,并且对象与对象之间只能通过消息传递来触发相应的操作。

4. 方法(Function)

方法是实现某个功能或操作的计算步骤及代码集合。方法描述的是对象的各种操作,也即是对象的行为,属于对象的动态特性,方法包括界面和实现两部分。

5. 数据封装

数据封装是指将数据结构和作用于数据结构上的操作组成一个实体,使数据的表示方式和对数据的操作细节隐藏起来,用户通过操作接口对数据进行操作。数据封装将程序的设计和使用分开,提供了代码的重用性,保护了数据的完整性与安全性。

6. 继承性

对类进行考察可以看出,不仅实体对象能够聚成类,而且抽象对象也能聚成类,抽象的类还能再进一步聚集成更高层的类,即父类;同时,一个类也可以派生出一些子类。子类具有父类的所有性质,即继承。类之间的继承关系,使类之间具有层次结构,并且可共享程序代码及数据,还可使派生类具有不同的数据结构和操作,有利于程序的扩展。

7. 多态性

多态性指不同对象接收同一消息可产生不同行为或一个名字多种语义以及相同界面多种实现方法。引入多态主要考虑到同一对象在不同的情况、条件下应具有不同的行为和方法。

1.3.2 面向对象数据模型的产生

面向对象数据模型提出于 70 年代末、80 年代初^[41],它吸取了语义数据模型的一些概念,同时融合面向对象技术的一些思想。在 80 年代,面向对象程序设计(OOP)方法开始使用,它在本质上不同于传统的程序设计思想^[19]。它将程序看作包含数据元素和程序行为的数据结构,而不是看作处理的指令序列。这种数据结构很难用关系数据模型进行描述,同时随着技术的进步,出现了许多新的应用领域,如 CAD,这些新的领域中的数据同样很难用传统的数据模型进行描述。在

这种情况下基于先进的面向对象技术,提出了面向对象数据模型的概念。面向对象数据模型(Object Oriented Data Model——OODM)是建立在对象集合的基础之上,应用对象、类等概念描述现实世界的信息,它的基本概念是对象和类等。

1.3.3 面向对象数据模型的研究现状

面向对象数据模型产生以后,大量的学者致力于这个模型的研究,取得了一定的进展。但是至今对这个模型还没有统一的定义,更不用说完整有效的体系结构,还需要进一步的研究。

1. 作为前期工作,人们研究了 CAD 系统中工程数据描述的要求,为后继者设计 OODM 提供了目标。

2. 明确了 OODM 既要面向用户,又要面向实现。

3. 提出了面向对象数据模型的一些初步概念。如认为任何现实世界中的实体都被模型化成一个对象,并赋予它一个唯一的标识符^[17]。具有相同属性和方法的所有对象组织在一起构成一个类^[19]。

4. OODM 还不能大量的推广使用,还需要进一步完善

1.3.4 面向对象数据模型的主要待研问题

面向对象数据模型虽然有许多学者进行研究,取得了一定的进展,但还存在较多的问题有待解决,主要有如下几个方面。

1. 对类和对象的定义还需要进行研究。不同学者根据自己对这些概念的理解,结合具体研究领域给出了不同的定义。

2. 面向对象数据模型的定义还未统一,需要进行研究。

3. OODM 的概念体系需要进一步的规范。对 OODM 应包含哪些概念,还没有达成共识。虽然所有 OODM 都含有对象、类、属性和方法等基本概念,但其余概念则基本不相同。

4. OODM 的一些概念仅仅是 O O P 概念的翻版,不完全符合 OODM 的要求,还需要结合具体领域作深入的研究。

1.4 工程数据库系统综述

1.4.1 工程数据库的产生

早在计算机时代之前,工程师们已经使用组表、图表和工程图来描述整个产品设计的流程和结构,进行相互交流。直到现在这种传统而古老的方式仍在大量使用。组表和图表记录数据很不严格,用工程图交换信息因人而异,很粗糙,不适于计算机处理。到了 50 年代中期,有限元法和计算机取得了巨大的发展,使分析和设计大型系统成为可能。在微型计算机上处理大型矩阵、大量有限元数据的要求,迫使工程师们寻找一种正确、有效的数据管理方法,从而开始了对工程数据库的研究。

20 世纪 70 年代,随着计算机对商务数据的成功管理,数据管理技术积累了大量知识和经验,提出了非常有效的数据库理论。在计算机辅助设计(CAD)出现后,迫切需要对大量复杂的工程数据进行管理,因而科技工作者和计算机专家开始致力于机械设计中的工程数据库的研究。1975 年,美国洛克希德公司的 Eastman 首先描述了一个可用于 CAD 的数据库,对 CAD 领域产生了重大影响。1985 年,第四届国际工程软件会议(在伦敦开)详细讨论了工程数据库在集成工程设计中的显著作用、工程数据库管理系统 EDBMS(Engineering Data Base Management System)的特点、类型以及它们的术语。这些讨论加速了工程领域对现行三种数据模型的吸收、开发并应用到未来的 CAD 系统中。对图形数据关系的组织和 CAD 系统中交互数据处理的一些方面提出了类型(Type)、对象(Object)与关系(Relation)和函数(Function)概念的一个 CAD 数据模型。这个时期的主要成果是以文件系统和数据库相结合的方式出现,这标志着工程数据库理论的确立。

1.4.2 工程数据管理的现状

随着 CAD 技术的出现和发展,特别是计算机集成制造系统 CIMS(Computer Integrated Manufacturing System)的提出,在 CAD 应用领域出现了大量复杂的工程数据需要管理。但是到目前为止,对这个问题尚未彻底解决。

1. 大部分工程数据仍是手工管理

在机械行业,工程数据管理的主要任务是图纸管理。虽然现在大部分企业实

现了计算机绘图,但仅仅是计算机代替了手工绘图,而并不能实现图纸的管理。人们还是得将绘好的图纸,利用绘图机绘出后进行管理。

2. 仅仅实现了部分静态工程数据的数据库管理

如北京理工大学研制的工程数据库实现了齿轮精度数据的管理。还有其它一些工程数据库,它们都只是实现了部分静态工程数据的管理。

3. 图形数据还不能进行很好的管理

几乎已建成的工程数据库都不能实现图形数据的管理或者不能很好实现。

4. 设计中产生的动态数据的管理还很困难

1.4.3 工程数据库的定义和特点

1. 工程数据库的定义

工程数据库是各种工程应用领域中,为了有效管理与具体工程相关的所有工程数据,存储在一起的相关数据集合。根据工程应用需要,它能完成对工程数据的修改、插入以及删除等操作。工程数据库是对图形等所有工程数据进行有效管理的工具,而不是仅仅能对管理性数据进行管理。根据工程数据的特点,工程数据库有其自身的一套理论,不完全等同于传统的数据库,是新一代数据库。由于CAD的兴起和广泛应用,现阶段提到工程数据库一般指机械行业用到的工程数据库。

2. 工程数据库的特点

(1)支持长事务和并行工程

长事务指较长时期地独占某些工程数据,而不能被其它人使用的事务。工程设计是一项长时间的设计过程,因而不能简单的利用共享/互斥锁实现数据共享。同时为了缩短产品开发周期、提高产品质量、降低产品成本,在产品开发早期阶段应及早考虑后续阶段的各种因素,要求支持并行工程。

(2)支持有效的各种版本管理

工程设计是一个不断修改的过程,并且有时还会返回到以前已经废弃的数据,因而必须将所有工程数据都进行保存,以备后期工作查询。

(3)支持数据模式的动态定义和修改

工程设计的对象经常都会改变,因而要求工程数据库系统能够动态地定义数据模式,并且能够在设计过程中根据需要修改数据模式。

(4)支持变长结构数据实体的处理

工程数据中有一类非常重要的数据——动态数据。它的数据类型不固定,数据量也不确定,要求工程数据库系统能够较好地处理这类数据。

(5)支持复杂对象的表示和处理

工程数据中图形数据的表示和处理都非常复杂,要求工程数据库系统具有良好的数据模型,具备较强的数据表示能力。

(6)支持复杂多样的工程数据的存储和管理

为了加速数据的查询和处理,必须设计良好的物理存储模式,有利于提高工程数据库系统的效率。

(7)支持工程数据的一致性和完整性

数据的完整性和一致性是数据库系统的基本要求,只有这样才能保证数据的有效性。

(8)支持多用户工作环境

现在的工程设计为了缩短周期,一般需要多人协作共同完成,因此工程数据库系统要能支持多用户。

(9)提供友好交互的用户界面

友好的用户界面能方便用户使用,才能有效地提高设计者的工作效率。现在一般都是使用图形用户界面,参照人机工程学的观点,考虑人机交互的方便性。

1.4.4 工程数据库的研究现状

1. 开展的研究工作

为了建立适合 CAD 应用的工程数据库管理系统 EDBMS,已经作了如下几方面的研究工作:

(1)系统地阐述了 CAD 自动化对工程数据管理的需求,研究了 CAD 中某些数据类型,提出了多种数据模型。

(2)研究 CAD 对 EDBMS 的具体要求。这方面的工作主要是对 CAD 的需求进行规范化,做了建立工程数据库的前期工作。

(3)建立了一些局部工程数据库管理系统,能管理一些工程数据,辅助完成一些局部的设计工作。

纵观上述工作,大部分都是解决建立工程数据库的外围问题或解决建立工程

数据库某一方面的问题,都未全面切入工程数据库的核心,都未完全解决 CAD 自动化的需求。对工程数据库的研究,还需做大量的工作。

2. 提出了一些新的数据模型

(1) 面向对象数据模型

前面已经叙述,这里不再重复。

(2) 扩充关系数据模型

这种数据模型利用类型、客体、关系和功能四个基本概念描述 CAD 系统中的工程数据。

(3) 扩充网状数据模型

这种数据模型是在网状数据模型的基础,允许实体和系可以用任何的组合方式进行组合,包括同构组合和递归组合等。

(4) 语义网络数据模型

这种数据模型采用对象型、对象实例、语义距离、分割等概念描述工程数据,以弧和结点模拟客观世界的静态和动态特性。

3. 典型的工程数据库管理系统

自 80 年代以来,国内外开发的比较典型的工程数据库管理系统有:

(1) 北京理工大学根据国家标准研制开发的齿轮精度标准数据库系统。仅仅是处理了机械设计中很小一部分静态的工程数据。

(2) 美国波音公司开发的 IPAD(Integrated Programs for Aerospace Vehicle Design)工程中使用的 EDBMS IPIP。支持网状、关系、层次三种数据模型,分布式管理,支持用户接口,编程语言,交互数据操纵和描述性语言,多库同时存取。

(3) 日本 Nippon 公司开发的面向 CAD 的多层数据库系统 MLDB。它是基于多重组织的数据库和自描述的动态数据模型,具有多级库分布式管理(当前的发展趋势)。

(4) 挪威工业中心研究所研制的面向 CAD 的扩展网状模型的工程数据库管理系统 TORNADO。能处理复杂的数据结构,可变长记录和动态长度记录,支持七种不同的数据类型,用户接口有数据定义语言和操纵语言,使用了缓冲区技术。

(5) 西德 PHILIPS 研究实验室为 CAD 一体化研制的专用于小型机的 EDBMS PHIDAS。支持网状数据模型,具有存储结构描述语言 SSDL,可写内模式,具有模

式和子模式定义语言 DDL, 数据操纵语言 DML。

所有这些工程数据库系统都解决了 CAD 应用中的一些工程数据管理问题, 都具有一定的实用性, 但它们都是针对某一类具体应用而开发的, 不具有通用性, 不能完全支持 CAD 自动化, 不能完全管理 CAD 中的工程数据, 还不能进行大范围的推广使用。

4. 工程数据库研究中存在的问题

- (1) 对复杂的工程数据还不能很好地管理和支持。
- (2) 数据库模式的动态定义和修改还很困难。
- (3) 对工程长事务、图形数据还没有很好地支持和处理。
- (4) 没有很好支持多用户工作环境。
- (5) 不论国内还是国外都还没有建立一个统一的通用的工程数据库系统。

1.4.5 工程数据库的发展趋势

随着技术的进步和新的应用需求的出现, 工程数据库的发展具有如下几个趋势。

1. 工程数据库与面向对象技术相结合

面向对象技术是当今描述现实世界比较理想的一种方法, 它符合人们的思维习惯。为了建立一个功能强大的工程数据库, 必须研究一个语义丰富、功能完善的数据模型, 即面向对象数据模型, 并开发基于 OODM 的 EDBMS, 使其具有通用性。

2. 工程数据库与人工智能相结合

由于设计过程中有许多经验性的知识要使用, 这部分工程数据利用一般的工程数据库不能进行管理, 必须求助于知识库, 所以应研究开发可以处理知识的工程数据库。

3. 发展分布式工程数据库

随计算机网络的发展和设计制造环境的改变, 要求企业具有更大的适应性。为了对分布在各地的工程数据进行管理, 必须建立分布式工程数据库。

1.5 论文选题

随着科学技术的发展, 企业在具有一定的计算机辅助工具的基础上, 需要有

一个统一的工具将各种应用软件集成起来,并对其产生的大量数据进行统一管理与控制,发挥整体效益。这就需要工程数据库的支持,因此很有必要对工程数据库及其应用开发技术进行深入研究。

1.5.1 研究工程数据库的意义

1. 推动基础理论研究

我国 CAD 发展纲要将工程数据库系统作为关键的攻关技术。工程数据库是开发其它应用软件的基础支撑软件。只有工程数据库提供有效的工程数据管理和 CAD 应用程序的接口,才能开发出高效的应用程序,才能实现 CAD 的智能化和自动化。因此我国 863 计划把工程数据库的研究作为其核心之一。现阶段发展 CAD 的主要任务是改善实施 CAD 的基础条件,包括建造工程数据库、推广 CAD 技术等。

2. 推动数据库技术的发展

数据库技术从诞生之日起,就不断地发展、更新,以适应新的应用需求。在新技术高速发展的今天,对数据管理的要求越来越高,数据库技术必须不断地更新才能满足新的要求。

3. 增强企业竞争力的需要

随着科学技术的进步,使生产得到了巨大发展,市场竞争变得愈来愈激烈,给企业造成了严酷的生存环境。市场需求不断变化,对产品结构、性能的需求越来越苛刻,变化越来越频繁,从而使新产品的研制变得越来越复杂。企业必需在最短的时间内设计出满足市场需求的产品,促使企业采用各种计算机辅助手段缩短产品开发周期。要达到这个目的,必需实现 CAD 自动化,必需研究工程数据库理论和技术。只有这样,企业才能在激烈的市场竞争中,增强自己竞争力。

4. 通过工程数据库的研究可以较全面地掌握数据库技术,增强自己的科学研究能力和学术水平。

1.5.2 论文命题及主要研究内容

1. 论文命题

通过前面的分析可以看出,对工程数据库的研究不论是从理论还是从实际应用看,都具有很大的必要性。因此选择的研究题目是《面向对象的数据模型及其在工程数据库开发中的应用研究》。

2. 论文研究的主要内容

(1) 研究工程数据的类型及其特点

通过研究工程数据类型, 分析工程数据之间的联系和区别, 明确关系数据库系统处理工程数据的不足, 把握研究的主要问题。

(2) 研究信息模型

研究零件特征的定义、构成及相互间的关系。根据面向对象数据模型研究的需要, 面向用户建立基于特征的零件信息模型, 并将建立零件信息模型的方法推广到装配体信息模型的建立。

(3) 探讨面向对象数据模型的基本概念

既面向用户又面向实现, 研究面向对象数据模型的定义及相关概念: 类、对象等。针对机械 CAD 研究对象的构造方式以及操作, 并应用面向对象数据模型对实例对象进行描述。这是本文研究的核心。

(4) 探讨工程数据库管理系统的体系结构

探讨面向对象数据模型在工程数据库开发中的应用, 构造工程数据库管理系统的体系结构。

(5) 研究工程数据库管理系统的实现技术

根据工程数据库管理系统的体系结构, 用 Visual C++ 6.0 编程验证 EDBMS 的基本实现技术, 并建立一个试验系统。

1.5.3 开发工具和运行环境

科学技术在不断发展, 先进的技术总要借助先进的工具。当前面向对象程序方法是最先进的软件开发方法。它应用类、对象等概念, 将数据和对数据的操作进行封装, 增加软件的重用度。同时本论文应用的是面向对象数据模型开发工程数据库管理系统, 当然最好是利用面向对象的开发工具, 而 C++ 是最成功的面向对象的程序设计语言之一。因此论文的软件采用 Visual C++6.0 开发、调试。运行的软件环境是 Microsoft 公司的 Windows98, 硬件环境是 Pentium MMX400/64M 以上内存的微机。

第 2 章 工程数据相关概念的研究

本章通过对工程数据的分析,研究工程数据的类型以及它们的特点,较全面地掌握工程数据的本质特征,在此基础上研究工程数据与商用数据的区别。最后研究工程数据库系统与商用数据库系统的区别和联系。

2.1 工程数据的定义和分类

工程设计是一个非常复杂、艰辛的过程,要经过多个阶段才能完成,涉及的工程数据类型多样、关系复杂、处理困难。为了能有效地管理这些复杂的工程数据,必须对工程数据作深入细致的分析。

2.1.1 工程数据的定义

在机械行业中,工程数据指产品生命周期中所涉及到的与产品有关的所有数据,特别是机械产品的设计与制造过程中使用和产生的数据。产品生命周期指从产品的市场调研开始,产生设计任务书,经设计、制造到使用等多个阶段,直到最后产品报废为止的整个过程。在这过程中涉及到的与产品有关的数据包括各种报表、材料属性、标准零部件及其参数、设计的零部件的几何结构和性能参数、产品的使用说明等等。工程数据与传统的管理型数据有明显的区别,有其自身的特点,必须采用不同的方法进行管理。由于工程数据数量庞大、关系复杂,很难一下全部分清它们的特性,结合本文针对的领域——机械 CAD,在以后的叙述中如无特别申明,工程数据特指产品设计过程中的工程数据。

2.1.2 工程数据的分类法

工程数据不仅类型多样、形态复杂,而且处理过程也很复杂,为了对它进行有效的管理,必须建立工程数据库系统。为了建立工程数据库系统,不仅要掌握工程数据的自然属性,而且还要透过工程数据的自然属性,抓住工程数据的本质特征进行全面的分析。

1. 按数据的应用目的分类

(1) 产品定义数据

产品定义数据指通过设计产生的对产品进行描述和分析的所有数据。它由两部分构成：产品描述数据和产品分析数据。所有产品定义数据一般都是以图纸的形式表现出来，一般又称为图形数据，具有动态性，它由两部分构成。

①产品描述数据是指设计人员在设计任务的控制下，经过分析和计算产生的对产品结构等进行描述的数据。它包括对产品几何结构进行描述的各种零件图和装配图、零件的材料以及加工要求的说明等。它是设计过程的主要成果，是下一阶段任务的基础。

②产品分析数据是指对产品性能、力学特性以及结构合理性进行分析、优化的数据。力学特性分析指应用各种分析模型对零件以及机构等主要受力部分进行强度分析，以保证零件在生命期内的安全性。结构分析是对产品整体结构的合理性进行分析并优化的过程。性能分析是对产品或零件的特性进行分析，以了解产品或零件的各种性能，如承载能力、功率等。

(2) 设计与控制数据

设计与控制数据是对产品拓扑结构进行描述和说明的数据，包括用户需求说明、对外形尺寸和结构进行控制的数据、产品计划表以及对各种报表进行管理的数据等。这部分数据的主要作用是对产品的设计过程以及设计的对象进行控制和管理，它一般都是以表格的形式出现，有时又称为非图形数据，大部分具有静态性。它可以分为三个部分：

①普通管理信息：它包括产品的需求说明、产品的性能说明、生产计划、经济核算以及各种报表的管理。

②标准数据：它包括设计规范、公差标准、结构要素、材料特性以及技术规范等。

③对形体和设计过程的语义约束条件。

这种分类法的优点是能将产品定义数据和约束控制的数据区分开，根据它们各自的特点，分别采取不同的措施对它们进行管理。同时它也能大体上将图形数据和非图形数据相区分，有利于对图形数据的认识。其缺点是不能区分软件包数据和突出设计数据，同时数据的动、静特性也不是很分明。

2. 按图形和非图形数据分类

(1) 图形数据

图形数据是指以图纸的形式、以点、线、面以及文字符号为基础构成的表达零部件以及设计过程的工程数据。图形数据包括工程图表、二维结构图、三维几何造型图等。图形数据大部分具有动态性；也有少部分具有静态性，如齿轮的弯曲寿命系数图。图形数据有些是共享的，也有些是独占的。它们的处理方法也不完全相同，对零件图等要求能够进行动态的修改，而对静态的图形数据不要求进行修改，但要能根据给定的参数得到设计者需要的数据。

(2) 非图形数据

非图形数据指工程数据中除去图形数据以外的所有工程数据。它包括各种设计标准、各种报表、零件管理数据等。

这种分类法目标明确，突出图形数据，有利于对图形数据的认识。其缺点是分类较粗糙，很难对非图形数据进行认识，难以根据非图形数据的特征对它们进行管理。

3. 按本质特征分类

(1) 管理型数据

管理型数据主要指产品设计过程中所用到的数据资料，它包括各种技术手册资料、国家标准与设计规范、产品目录文档以及图纸资料文档等。这部分数据一般是对设计过程进行管理和控制，对各种设计资料进行整理的数据。它的特点是：

①数据关系分明。

②数据相对稳定。它的型一般不会改变，而值可以改变。

③语义一致。

这部分数据具有静态特性，对它的操作主要是查询和检索，有很少一部分值的修改。

(2) 设计型数据

设计型数据指 CAD 系统中由已知数据经过工程设计人员的设计处理得到的产品数据，它包括产品结构数据和产品加工数据。产品结构数据是指对设计产品结构、形状进行描述的这部分数据。产品加工数据是指零件的形位公差、表面粗糙度、技术要求、材料以及热处理等进行说明的数据。它的特点：

①具有动态性。这部分数据随着设计的进行，它的型和值都会不断的改变，同时它的存取方式以可能会改变。

②频繁的修改。由于设计是反复试探的过程，随设计的进行，如果设计者对先前取定的值不满意，就会作出修改。

③允许语义的不一致。设计过程是多个设计者协同工作的过程。在设计完成之前，允许各个设计者设计的对象在语义上不一致，但是设计完成之后不允许。

(3) 图形数据

图形数据指以图形形式表现出来的工程图表、二维图、三维图。其中二维图指零件图以及装配图，三维图一般指造型系统为了直观的表示产品而生产的立体图。所有的图形数据都具有动态性，设计者可以根据各方面的需要进行修改。图形数据具有两方面的属性：一方面要反映设计对象的几何属性，另一方面要反映设计对象的拓扑属性。因此图形数据还可以进一步分为几何数据和拓扑数据。

①几何数据是指定义设计对象的点、线、面、体几何尺寸的数据。如传动轴的长度和轴径、齿轮的宽度等。

②拓扑数据指描述几何实体间相互关系的数据。这里的几何实体指构成设计对象的基本单元。这部分数据能反映几何实体间的相互位置及其装配关系。

图形数据由复杂的拓扑关系形成复杂的网状结构，存储、处理非常困难。工程设计的中间结果和最终结果很大一部分是以图形的形式表现出来，因此对这部分数据的管理是工程数据库的一个主要内容，同时也是一个有待解决的技术难点。

(4) 软件包数据

软件包指独立处理局部计算或者图形绘制的软件。这部分数据包括有限元软件、结构优化软件、绘图软件等辅助设计工具。由于这部分数据大多以二进制的形态存在，设计者一般不能对它们进行修改，也不必对它们进行查询，而只能按照它提供的对外接口使用它们，并得到相应的处理结果。

这种分类法反映了工程数据的本质特征，有利于对工程数据的认识；便于设计合适的物理数据结构对工程数据进行有效的存储和检索，有利于工程数据库系统的物理设计；将管理型数据与其它数据相区别，可以有效地利用关系数据库管理管理型数据，简化工程数据库的设计，为工程数据库管理系统的建立提供了一种方法；分析设计型数据、图形数据和软件包数据，进一步明确它们各自的具体

特征，为解决这个难题提供了基础。

2.1.3 工程数据的总体特征

通过对工程数据类型的分析，可以得到工程数据具有如下基本特征：

1. 数据类型多样性

按工程数据的表现形式有图形和非图形数据，按本质区分有管理、设计、图形和软件包数据。正是由于工程数据类型的多样性，使工程数据的管理非常困难。

2. 数据模型的复杂性

由工程数据的多样性导致描述工程数据的数据模型的复杂性。为了描述所有的工程数据，必须设计良好的数据模型。这种数据模型比传统数据模型复杂，传统数据模型都是利用记录表示各种数据，具有很大的局限性（见第一章）。在新一代数据模型中，有的是在传统数据模型的基础上进行扩充增加新的功能，由于受传统数据模型的约束，它们仍不能很好的描述工程数据。

3. 具有动态性

工程数据具有动态性，主要表现在三个方面：

（1）概念模式的动态性

随着设计的进行，设计者可能改变设计对象的整体结构以及形状，从而导致设计对象概念模式的改变。

（2）存储结构的动态性

由于设计对象之间关系的改变，按照设计的物理存储方式相应地改变它们的存储结构。

（3）图形数据的动态性

图形数据随产品结构的改变而改变，并且具有多个版本。

4. 数据量大

工程设计涉及到的各种标准数据量就非常庞大，更不用说由此产生的图形数据等，使工程数据的量猛增。

2.1.4 工程数据与管理（商用）数据的区别

通过对工程数据类型以及总体特征的分析，对工程数据的特性有了一个整体

的认识，它与管理（商用）数据有着明显的区别，总结如表 2-1 所示。

表 2-1 管理（商用）数据与工程数据的区别

	管理（商用）数据	工程数据
记录类型	少	多
记录型数据量	大	不太大
关系	简单	复杂
模式	静态	动态为主
随机存取量	少	大
主要查询类型	整体多	部分多
存取类型	系统存取	过程存取
报表数量	多	少
数据类型	简单	复杂
数据库容量	大	小

2.2 工程数据库系统与商用数据库系统的比较

由工程数据分析知道商用数据库系统与工程数据库系统具有一定的区别和联系。为了建立适合 CAD 系统的工程数据库系统，必须分析商用数据库系统与工程数据库系统的区别和联系，以利于借鉴商用数据库系统的成功经验建立工程数据库系统。

2.2.1 工程数据库系统与商用数据库系统区别

商用数据库系统已经应用了几十年，形成了一套完善的理论和开发方法，而工程数据库系统的研究和开发还不成熟，因而要借助商用数据库系统的一些成功而有效的方法。因此要从各个不同的角度比较商用数据库系统与工程数据库系统的区别，总结如表 2-2 所示。

表 2-2 商用数据库系统与工程数据库系统的区别

	商用数据库系统	工程数据库系统
管理的对象	a) 静态的现实世界; b) 实体之间关系稳定; c) 可事先描述; d) 初始装入。	a) 设计环境、条件数据 (设计规范、方法等); b) 设计对象数据: 随设计的进行而产生——修改——构造, 动态形成的设计结果。
模式的性质	a) 库格式结构可事先描述; b) 库格式结构基本不变。	a) 描述设计对象的库格式是动态构造的; b) 设计结束后, 库格式才生成。
数据的性质	a) 用字符与数字表示; b) 关系类型少; c) 数据量大; d) 数据较稳定。	a) 除字符和数字外, 还有文本和图形数据; b) 实体关系复杂、多样, 有层次、网状和关系类型; c) 开始信息量少, 但增长迅速。
数据库的变化	a) 数值变化; b) 结构变化少; c) 用户修改值, 由数据库管理员修改型。	a) 数据的型经常变化; b) 数据的值经常变化; c) 用户可以存取和改变值以及结构。
数据库的性能	a) 必须具有并发性、一致性和安全性; b) 应规定用户权限来保证数据的安全性; c) 多用户可以同时使用数据库, 为防止冲突, 要对一些数据采取加锁措施。	a) 多用户方式下工作; b) 要求响应速度快; c) 应保证数据 (结构和值) 能动态修改; d) 对数据安全性要求较弱; e) 要求防止错误的存取; f) 允许数据暂时的不一致。
用户的要求	a) 主要是能在一定范围内进行检索; b) 计算需要的数据量小, 复	a) 要求能长时间的计算而且计算中查找信息量大; b) 要求支持反复的、试探性的

	杂性低; c) 用户操作可使用操作语言、交互式图形接口, 但对交互操作要求不强烈。	设计过程; c) 用户操作主要是交互处理, 要求响应速度快, 对用户接口要求高。
处理的数据	a) 查询、增加、删除和修改; b) 每次处理时间短、涉及的记录型少, 而数据量大。	a) 从查询到分析; b) 每次处理周期长、涉及的数据类型多, 处理的数据量少。
用途	a) 管理数据 b) 录入数据 c) 共享数据	a) 工程设计 b) 工程分析

2.2.2 工程数据库系统与商用数据库系统的联系

作为不同用途的数据库系统, 虽然存在上述诸多差别, 但是作为数据集中处理的数据库系统, 它们仍然存在着许多相同的地方。

1. 数据库系统基本特征相同

它们都符合数据库系统的基本特征, 都能实现数据共享, 保证完整性、一致性等, 都需要一定的数据模型, 都要设计数据库系统的数据模式。

2. 数据操作方法有部分相同

都要实现数据的增加、删除以及修改和更新, 都要提供数据的对外接口, 都要实现与应用程序的数据交换。

3. 处理的数据有部分相同

由于工程设计要进行有效的管理, 工程数据中存在管理性数据, 这部分工程数据可以利用商用数据库系统进行处理。

4. 存储介质相同

都是利用磁盘或磁带进行存储, 都要处理物理故障引起的错误, 因而都要进行容错和备份。

5. 功能相近

都是对数据进行有效管理, 都要求能够完成相应数据的各种操作和处理, 实

现数据的计算机管理。

2.3 小结

为了建立工程数据库系统，本章给出工程数据的定义，并详细地分析了机械 CAD 过程中工程数据的分类及其特性。根据工程数据的本质特征将工程数据分类四类：管理数据、图形数据、设计数据和软件包数据。这种分类方法能够体现工程数据间的本质区别，能够针对数据的不同特点进行处理，有利于工程数据的计算机管理。在工程数据类型分析的基础上给出工程数据的总体特征以及与商用数据的区别，分析工程数据库系统与商用数据库系统的区别和联系，为后续章节的信息模型、数据模型的研究以及工程数据库管理系统的开发打下了基础。

第 3 章 基于特征的信息模型研究

本章研究零件特征的基本概念及其分类体系, 根据零件的特征构造零件的特征信息模型, 并应用构造零件特征信息模型的方法构造产品(装配体)的特征信息模型。

3.1 零件特征 (Feature) 概念综述

自从零件特征的概念产生以来, 特征技术的研究一直都非常活跃, 并把它作为一种高层次的零件信息描述方法。但是, 零件特征的定义目前尚未统一, 许多专家和学者对特征技术作进一步研究的同时, 分别从不同的角度对零件特征进行了定义和阐述。

3.1.1 研究零件特征的意义

零件特征的提出和研究, 对计算机辅助设计以及工程数据库的研究具有重要的理论和实践意义。利用零件特征能解决 CAD 系统中零件的描述以及工程数据库信息模型的构造。

1. 计算机辅助设计的需要

随着并行工程的发展, 对 CAD 系统提出了更高的要求。CAD 系统在 CAD/CAM 集成系统中承担着机器或零件设计的任务, 为后续环节, 如: 工艺规划、数控编程、加工检测等提供基础数据; 同时后续的 CAM 系统各个环节对 CAD 系统提出一定的约束条件, 如孔的可加工性等。为了满足这些要求, 零件特征不仅包括组成零件的点、线、面、体等几何、拓扑信息, 而且还包括零件的性能、加工要求等多方面的信息。几何造型技术不能完全满足这些需要, 大多采用特征造型技术, 因此零件特征的概念就显得尤为重要。

2. 建立工程数据库系统的需要

工程数据库是存储各种工程数据及其相互关系的集合, 包括 CAD 系统中的各

种数据。要建立工程数据库必须研究设计对象的信息模型和数据模型。信息模型相当于关系数据库系统中的 E—R 模型,它对数据模型的建立起辅助作用,数据模型根据信息模型对产品(或零件)的描述进行设计。要构造良好的信息模型离不开零件特征,只有利用零件特征建立的信息模型才能全面地描述工程数据。

3. 特征造型理论研究的需要

造型技术是 CAD 的基础,已经有近 40 年的历史,其间经历了多个发展阶段,直到目前的参数化特征造型。特征造型与表面造型和实体造型相比,具有更高层次的抽象,已成为各国研究的重点。但是特征的定义一直都未统一,还很难进行数学描述,制约了特征造型理论的发展。因此零件特征概念的研究可以推动特征造型理论的发展,使特征造型技术得以广泛的使用。

3.1.2 零件特征概念的提出及定义

1. 零件特征概念的提出

零件特征概念是 1978 年由美国麻省理工学院的戈萨德和索尔特斯教授指导研究生论文时提出的,经过二十多年的研究,取得了丰硕的成果。经过大量科研工作者的努力,零件特征概念及其应用已发展成为一门先进的基于特征的造型技术,并得到大量的推广和使用。在大量关于零件特征的研究中,人们从不同的应用角度,对零件特征有不同理解。

2. 零件特征的定义

(1) Pratt 和 Wilson 定义^[93]

特征是产品模型的相关元素集,这些元素必须符合其识别与分类规则,它们被认为是一个独立的实体,并且在一个产品的生命周期中具有一定的功能。特征可分为六类:通道特征、凹陷特征、凸起特征、过渡特征、域特征和变形特征。这种特征的定义和分类方法后来被 STEP 标准所采用。该定义和分类方法主要基于零件形状的描述,无法实现对其它信息的描述,因此这种特征的定义和分类方法无法满足并行工程的需要。其优越性仅在于对形状特征进行了比较详细的描述,为图形数据的计算机处理提供了一种方式。

(2) Shah 的定义^[93]

特征是产品信息传递的载体 (Carrier)，能帮助设计者在设计、制造或其它工程任务之间进行通讯与交流。他把产品信息分为五类特征：形状特征、精度特征、技术特征、材料特征和装配特征。这种特征定义和分类方法描述 CAD/CAM 系统各个阶段设计信息以及传递信息比较有效，因为该定义和分类方法允许包含更多类型的信息，信息较为完整和全面。但是该定义仅强调特征是信息的载体，而忽略了特征本身也是构造零件的基本单元，而且该定义与分类方法过于粗糙，尚不能完全满足机械设计中的建模需要。

3.2 关于特征基本概念的研究

3.2.1 零件特征的定义

为了建立基于零件特征的信息模型，必须对零件特征的定义进行研究。在 CAD 系统中零件特征应包含零件的所有信息，即设计、制造和装配信息。它既要反映零件在设计阶段信息的动态变化，又要强调它对各个阶段信息的描述，以支持并行工程，即强调设计信息的同时也兼顾制造过程的要求。本着这个思想零件特征的定义如下：

零件特征是用于构造零件形状的、具有工程意义及功能的基本结构单元以及依附其上的非几何信息。

零件特征可以描述零件两个方面的信息：作为基本结构单元的几何信息和依附其上的非几何信息。几何信息包括零件的形状特征和精度特征，它们共同描述构造零件形体的信息，是设计阶段产生的主要成果，是制造、检测等过程的依据，同时也是非几何信息描述的基础。非几何信息包括管理特征、功能特征、技术特征、分析特征、装配特征等信息。它是针对零件整体进行描述的信息，是对零件的设计、制造等过程进行管理和控制，它一般不对零件的局部进行描述。需说明的是零件加工所需要的形位公差、表面粗糙度等精度信息不是作为附加信息，而是作为零件特征的一个有机的、不可缺少的组成部分，这样有利于支持并行工程。另外零件特征的分类与零件的类型紧密相关：零件类型不同，具有的特征略有差

异, 相应的分类方法和特征关系也不完全相同。

3.2.2 轴类零件特征的分类

轴类零件是典型的通用零件, 它具有回转体类零件的大部分特性。对轴类零件特征的研究结果, 可以推广应用到所有的回转体类零件。轴类零件特征可分为三类: 总体特征、形状特征和精度特征。

1. 总体特征

零件的总体特征是忽略零件的内部细节而作为作为一个整体进行描述的特征。总体特征主要是描述对零件整体方面的要求以及说明, 它的任何描述对整个零件有效。根据这个定义, 总体特征又可分为管理特征、功能特征、材料特征、技术特征、分析特征以及装配特征。

(1) 管理特征

管理特征指零件的总体管理信息, 包括零件名称、设计者、制图者、审核者、设计日期、零件图号、零件数量、设计单位、所属产品、所属部件等等。与传统的设计图纸相对照, 管理信息主要用于描述标题栏信息。管理特征是对零件设计、加工、检测和装配各个过程进行管理的依据。

(2) 功能特征

功能特征是对零件的功能进行描述的信息。零件的功能特征包括零件的用途、零件的性能等。轴类零件指它的转速、转矩、额定功率、临界转速、安全系数等, 这些参数是设计和使用它的依据。

(3) 材料特征

材料特征是对零件使用的材料以及材料特性进行描述的信息。材料的特性包括材料名称及符号、热处理特性、机械特性、硬度、常见应用等。其中机械特性包括抗拉强度 σ_b 、屈服强度 σ_s 、延伸率 δ_5 、收缩率 δ_5 等。

(4) 技术特征

技术特征是对零件制造、检测以及装配要求进行描述的信息。它包括传统工程图的技术要求和未标注的表面粗糙度。

(5) 分析特征

分析特征是指设计过程中通过计算、推理等对零件的强度等要求进行分析的

结果信息。对轴类零件而言,包括强度信息、刚度信息和临界转速等。其中强度信息又包括许用切应力计算结果、许用弯曲应力计算结果和安全系数校核结果,刚度信息又包括扭角和弯曲变形的大小。

(6) 装配特征

装配特征是指描述该零件与其它零件进行配合安装的信息。装配特征包括参与装配的零件以及装配的几何要素、公称尺寸、配合类型、安装定位、安装顺序等方面的信息。

2. 形状特征

形状特征是零件上一系列几何元素按照一定的拓扑关系形成的具有特定几何形状、特定功能的基本形体单元。形状特征是零件最主要的特征,是精度特征和总体特征的载体,是构造零件几何形状的直接决定因素,是工程数据库系统要处理的主要对象。从构造零件形状、满足零件功能要求所起的作用不同,可将形状特征又分为主要特征、次要特征和辅助特征。

(1) 主要特征

主要特征指用于描述零件的基本几何形体,并相应对零件的工艺路线起主要作用的特征。它是一个独立的几何形状描述,由几何要素按一定的拓扑信息构成,如圆柱体、圆锥体等简单的几何形体。因此主要特征的描述是利用构成它的基本几何要素,这些几何要素同时是精度特征的载体。

(2) 次要特征

次要特征是指那些在位置上从属于某一主要特征,并具有特殊加工要求的形状特征,如键槽等。次要特征是零件完成它特定功能的必不可少的一部分,如减速箱中间轴上的键槽,缺少它,轴和齿轮就不能相互传递动力,就不能完成减速的功能。次要特征的描述是利用它的基本几何要素。

(3) 辅助特征

辅助特征主要是对其所从属的主要特征、次要特征进行局部修饰和说明,起辅助作用的特征。它可以附加于主要特征,也可以附加于次要特征。

3. 精度特征

精度特征是用于描述零件的尺寸公差、形位公差以及表面粗糙度的信息集合。精度特征是工艺信息的主要内容,它的描述要与主要特征、次要特征的相关几何

要素联系起来,不能离开几何要素而独立存在。精度特征可以进一步分为形位公差、尺寸公差和表面粗糙度,如图 3-1 所示。

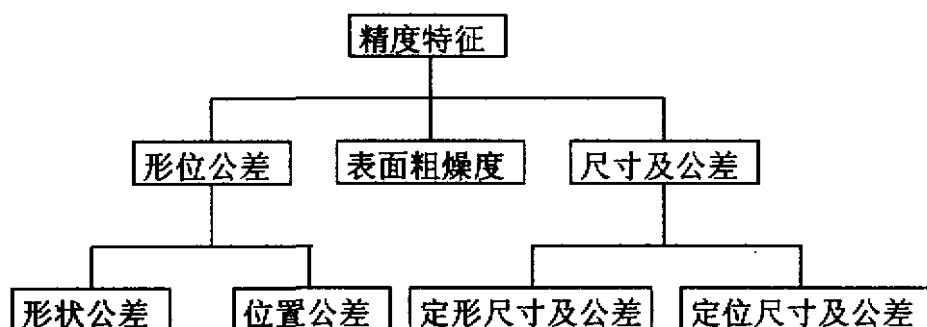


图 3-1 精度特征的分类体系

3.2.3 轴类零件特征之间的关系

零件特征之间的关系简称特征关系,是指零件特征之间或者零件特征与构成零件特征的几何要素之间的相互关系。特征关系与特征相互结合在一起共同描述与零件有关的工程数据,二者缺一不可。通过对 CAD 系统中工程数据的分析,可以得到零件特征之间具有如下几种关系,如图 3-2 所示。

1. 邻接关系

邻接关系反映主要特征之间在空间上的相互近邻的位置关系。具有邻接关系的两个主要特征之间没有主次之分,是相互独立的,但是两者之间始终有固定的相互位置和连接定位点。

2. 从属关系

从属关系描述次要特征或辅助特征对主要特征的附属关系。具有从属关系的两个特征之间有主次之分,次要特征或辅助特征对主要特征具有依赖性,离开主要特征,次要特征或辅助特征就无意义。

3. 包含关系

包含关系指一个特征与该特征内部的各要素之间的关系,即常见到的 IS_PART_OF 关系。具有包含关系的特征与各要素之间是组成与被组成的关系,各个要素是该特征的组成部分,是局部与整体的关系。

由图 3-2 可以看出, 通过上述特征关系使零件特征之间, 以及特征与构成特征的几何要素之间相互关联, 成为一个可访问的复杂网状结构。图中各要素分为总体特征、形状特征和精度特征。零件包括总体特征和形状特征, 总体特征

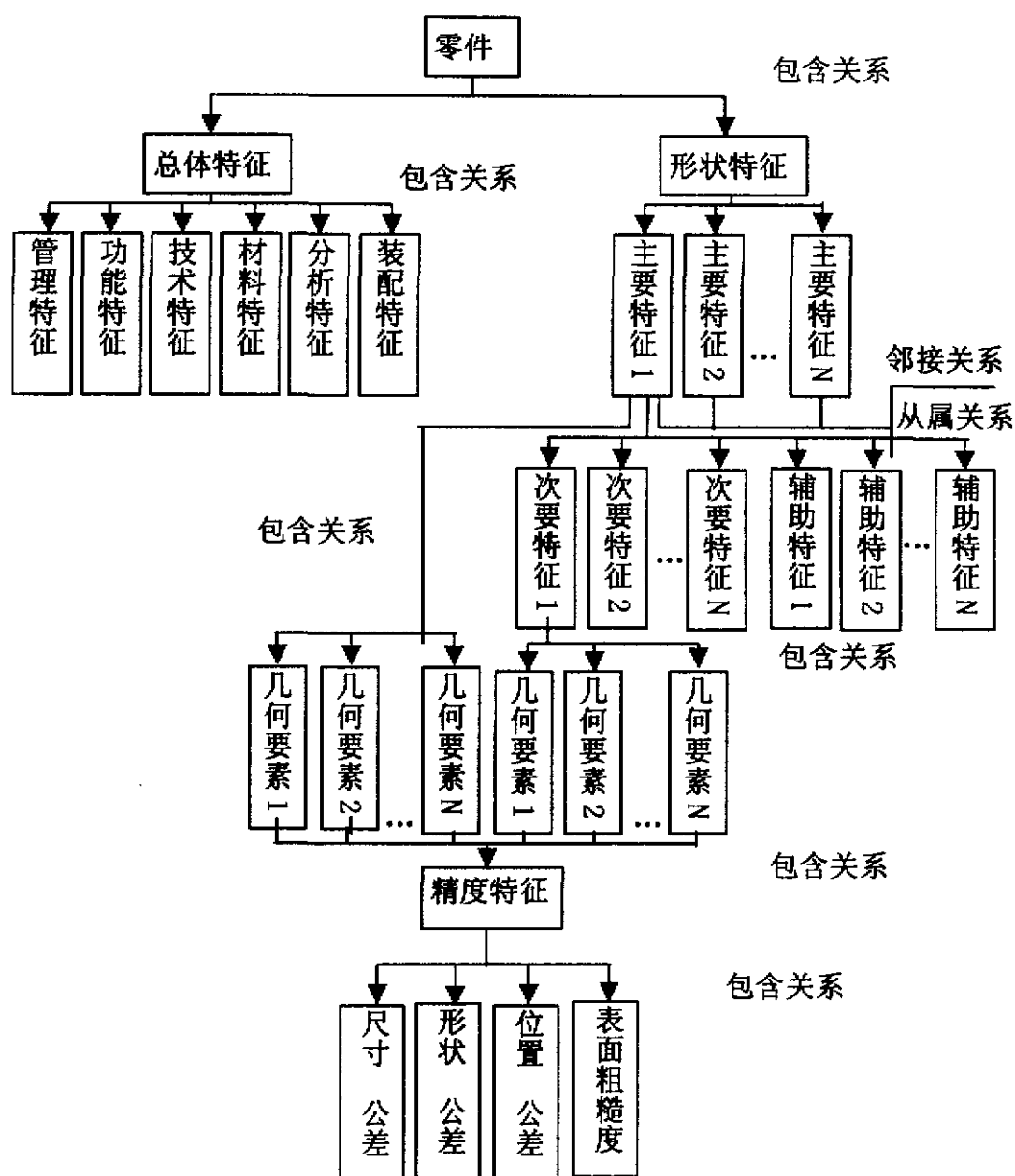


图 3-2 零件特征之间的关系

和形状特征联合起来共同描述零件的信息。此外,精度特征被包含在主要特征或次要特征的几何要素之中,同时借助这些几何要素描述形状特征的精度特征。只要知道零件名就能找到相应的总体特征和形状特征,通过形状特征就能找到各个主要特征,由主要特征能找到相应的几何要素、次要特征和辅助特征以及次要特征的几何要素,通过几何要素可以找到相应的精度特征。由此可看出整个零件的所有信息都能通过这些特征和特征关系进行描述。

3.2.4 齿轮类零件的特征分类及其关系分析

齿轮类零件是又一大类通用零件,一般比较常用的是圆柱齿轮和圆锥齿轮,而以圆柱齿轮用得最多。因此在这里以圆柱齿轮为例讨论齿轮类零件的特征分类以及相应的特征关系,其它的齿轮类零件可以仿照这个方法进行分析。

1. 齿轮类零件的特征分类

通过对轴类零件特征的分析,建立了轴类零件的分类方法以及特征关系,将这种方法引用到齿轮类零件的分析中,同样可将齿轮类零件特征分为三大类:总体特征、形状特征和精度特征。

(1) 总体特征

由于齿轮类零件与轴类零件在结构上具有较大的差异,并且它们的描述方法也存在一定的区别,因此它们的总体特征略微不同。齿轮类零件的总体特征包括管理特征、功能特征、材料特征、技术特征、分析特征、装配特征和齿轮公差。其中管理特征、材料特征、技术特征和装配特征与轴类零件相同,其余特征有一定的差别。

齿轮类零件的功能特征是描述齿轮的主要参数信息,如齿轮模数 m 、齿数 Z 、基本齿廓参数、变位系数 x 和螺旋角 β 等。其中基本齿廓参数包括齿顶高 h_a 、工作齿高 h' 、齿形角 α 、顶隙 c 和齿根圆角半径 ρ_r 。

齿轮类零件的分析特征同样是指设计过程中通过计算、推理等对零件的强度等进行分析的结果信息,但它分析的内容不同。齿轮类零件的分析特征包含齿面接触疲劳强度计算结果、齿根弯曲疲劳强度计算结果和静强度校核计算结果等。

齿轮公差指描述齿轮整体精度的信息。它包括齿轮的精度等级、齿轮所属的公差组、齿轮的检验项目及公差、齿厚极限偏差 E_s 、齿圈径向跳动公差 F_r 、径向

综合公差 F_i'' 、齿形公差 f_r 等。

(2) 形状特征

齿轮类零件的形状特征与轴类的形状特征相似。不同点是齿轮类零件的主要特征可以进一步划分为基特征和宏特征。基特征是指圆柱体、圆锥体等简单三维形状特征，这相当于轴类零件的主要特征。宏特征是指具有相对固定结构形状和相对固定的加工方法的形状特征。宏特征描述的几何形状往往比较复杂，而又不便于进一步细分为其它的形状特征组合，能在整体上反映出零件的基本结构形状。在齿轮类零件中将齿形有关的形状特征看成整体作为宏特征进行描述，其它如齿胚、内孔等作为基特征进行描述。

(3) 精度特征

齿轮类零件的精度特征与轴类的精度特征相似，是对构成齿轮的与齿形无关的几何要素的精度进行描述。

2. 齿轮类零件的特征关系

齿轮类零件的特征关系与轴类零件的特征关系完全相似，同样具有邻接、从属、包含三种关系。利用齿轮特征以及特征关系能描述齿轮的所有信息。

3.2.5 滚动轴承的特征分类及其关系分析

由于滚动轴承是标准件，它与轴类零件以及齿轮类零件的特征具有较大的差别，相应的特征关系也有所不同。在这里以滚动轴承为例进行分析，其它的标准件可以应用同样的方法进行分析。

1. 滚动轴承的特征分类

滚动轴承与相关零件配合时其内孔和外径分别是基准孔和基准轴，对它的精度通常不必进行标注，因此滚动轴承没有精度特征。另外轴承属于标准件，它的形状相对固定，大多用一些参数进行说明，因此它可以不具有形状特征。由此可见轴承类标准零件只具有总体特征。

滚动轴承的总体特征也与轴类和齿轮类的总体特征有些不同，它只具有管理特征、功能特征、分析特征和装配特征，而没有其它特征。

(1) 管理特征

管理特征指滚动轴承的国标、轴承的型号、标注方法等,对轴承总体进行描述的信息。

(2) 功能特征

功能特征指滚动轴承的额定动载荷 C_r 、额定静载荷 C_{or} 、极限转速等,对该滚动轴承的性能进行说明的信息。

(3) 分析特征

分析特征指对滚动轴承进行有关计算和校验的结果,如滚动轴承的基本额定动载荷、基本额定寿命、基本额定静载荷和极限转速等。

(4) 装配特征

装配特征是指滚动轴承与其它部件进行装配时的信息。它包括轴承的公称直径 d 和 D 、轴承宽度和安装尺寸。

2. 滚动轴承的特征关系

利用轴类零件的分析方法,由滚动轴承具有的各个特征可以知道:滚动轴承只具有包含关系。轴承标号通过包含关系与它的总体特征之间建立联系,而总体特征又通过包含关系与管理、功能、分析等特征之间建立联系。由此可见知道轴承标号就能找到轴承的所有相关信息,同时利用总体特征和包含关系也能描述轴承的所有信息。

3.3 基于特征的零件信息模型研究

零件既是构成装配体的基本单元,又是 CAD 设计的对象,同时它还是工程数据库存储的基本数据。另外,根据零件特征的描述,构造零件特征信息模型是构造装配体信息模型的基础。因此本节研究零件特征信息模型的基本概念及零件信息模型的结构。

3.3.1 研究信息模型的目的

为了建立工程数据库管理系统,研究信息模型的目的在于如下两个方面。

1. 面向 CAD, 将设计对象进行抽象化

由于工程数据库解决的主要问题是工程数据的管理,即设计对象及相关数据

的管理。在 CAD 领域中涉及的零件类型多而且复杂, 很难一下弄清它们的构成和各个组成部分之间的相互关系。为了解决这个问题, 可以建立一个辅助工具: 零件信息模型。通过零件信息模型能够很好的分析零件及其各个部分之间的复杂关系, 便于找到较好的方法对零件的构成及内部相互关系的描述。

2. 零件信息模型为数据库的概念设计提供类 E—R 模型的作用

用户和数据库设计者可以通过零件信息模型交换信息。用户利用信息模型描述领域中需要管理的数据, 使数据库设计者能够更好的理解用户需求, 从而更有效地进行数据库的概念设计。同时, 通过利用信息模型对数据的描述, 可以辅助数据库设计者利用数据模型设计数据模式。

3.3.2 零件特征信息模型的定义

在数据库系统中信息模型又称概念模型, 是将现实世界抽象为信息世界, 它不依赖于具体的数据库管理系统, 而仅仅是用一些概念对现实世界进行描述。

零件的特征信息模型 (Feature Based Information Model——FBIM) 是基于零件特征的、表达零件设计和制造过程所需信息的一组人为约定的概念和组织结构。

零件的特征信息模型以零件特征为基础, 通过零件的各个特征去描述零件的有关信息, 它的基本概念是特征以及特征关系。由于不同类型的零件有不同的特征类型以及不同的特征关系, 因此它们信息模型的结构不完全相同, 但是应用的概念非常相似。利用零件的特征信息模型可以实现零件信息的统一描述, 满足各个阶段对零件信息的需求, 保证设计阶段信息的可传递性。

3.3.3 零件信息模型的总体结构

零件信息模型应以零件的各类特征为基础, 不仅是表达零件信息的数据集合, 而且还应反映各类数据的描述方式以及相互之间的关系, 因此零件信息模型应能够描述零件所有的特征及关系。在这里以轴类零件为例重点说明零件信息模型的结构。

轴类零件信息模型的总体结构如图 3-3 所示, 整体而言呈层次状结构。每个零件都包含总体特征、形状特征和精度特征的描述, 每个特征又可进一步细分直到最后对该特征的几何要素或数值、字符说明为止。当然, 其中的某个特征可以是空值, 这样当某个零件不具有该特征时, 通过空值进行表示。注意空值并不表

示零，空值表示无或不具有该特征，而零值表示某个元素的数值是零，在某些时候它还可以是其它数值。通过这个信息模型可以描述 CAD 中的所有的轴类零件。

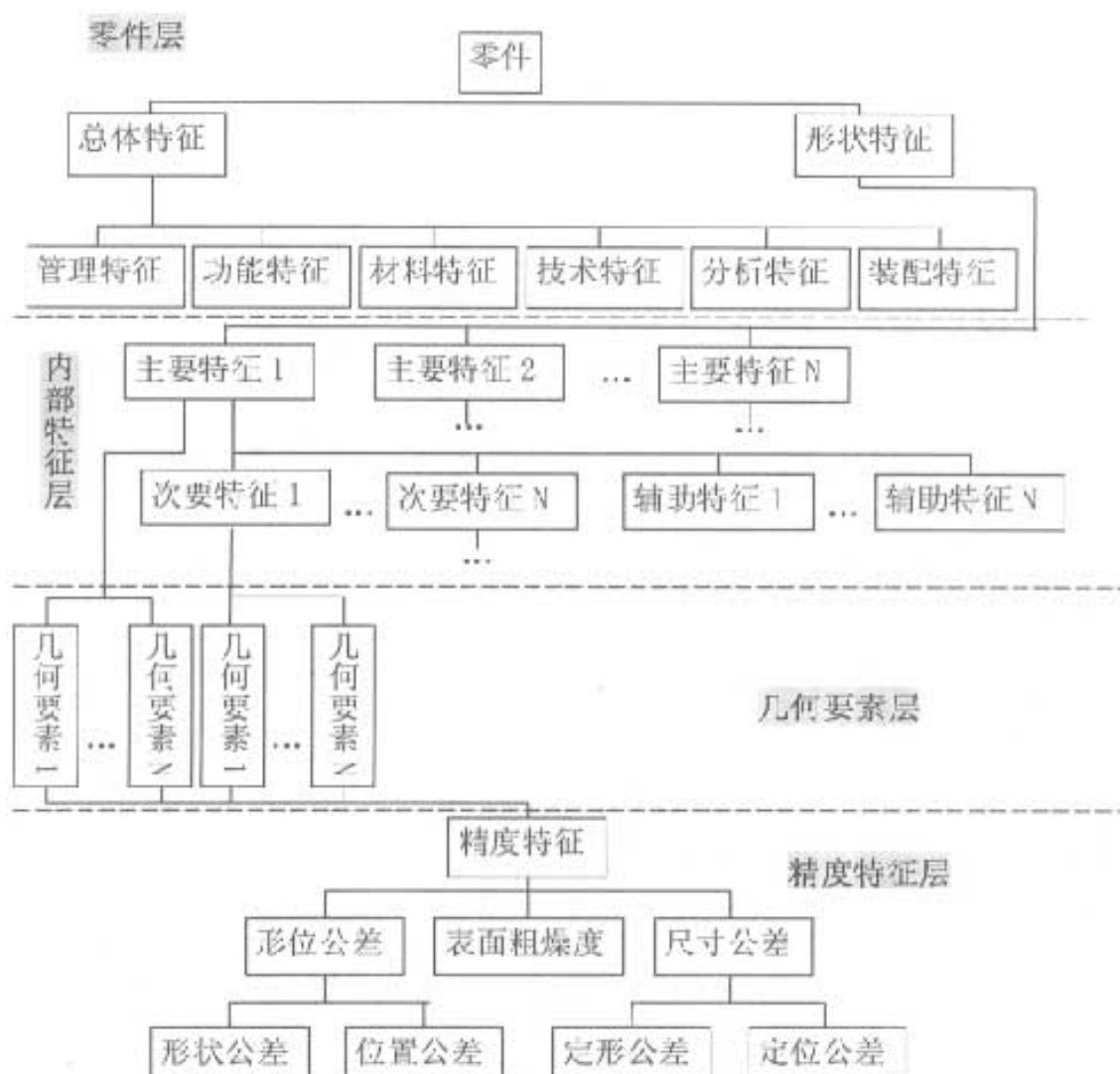


图 3-3 轴类零件信息模型的总体结构

齿轮类零件信息模型的总体结构与轴类信息模型的总体结构相似，标准件滚动轴承信息模型的结构较简单，直接由滚动轴承名称和型号找到总体特征进行描述，下文将以典型零件轴、齿轮和滚动轴承为例描述它们的模式和实例。

3.3.4 零件信息模式及描述实例

利用零件的信息模型分别对轴、齿轮和轴承进行描述,以说明应用零件信息模型对具体零件进行描述的方法。

1. 减速箱中传动轴的信息模式及描述实例

为了说明方法,选用图册上减速箱的传动轴为例,如图 3-4 所示。由图可以得到总体特征中各个特征项的值。根据零件示意图可以把此轴分为 6 段,作为形状特征中的 6 个主要特征,分别标号为①至⑥,如图 3-4 所示;对每个主要特征分析它的几何要素、从属的次要特征和辅助特征以及次要特征的几何要素和辅助特征,各个标号如图 3-4;最后对每个几何要素分析它的精度特征。整个信息模式及描述实例如图 3-5 所示。其中总体特征以管理特征为例分析它的具体描

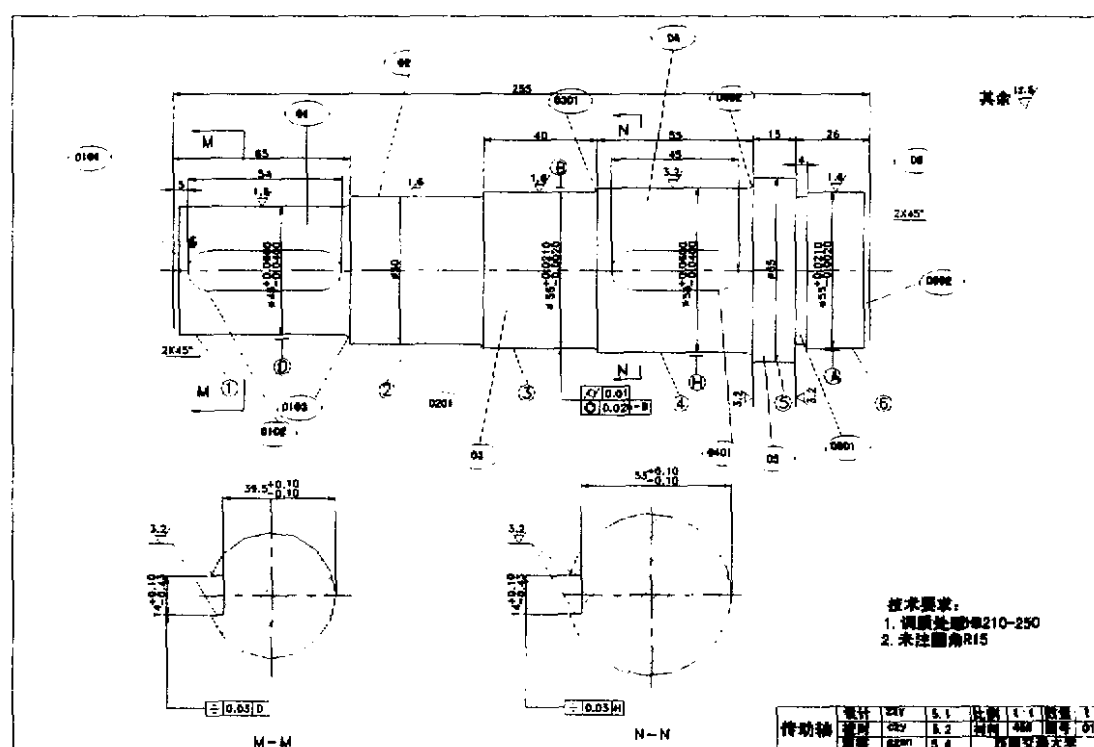


图 3-4 传动轴示意图

述,其它如功能、材料等特征可以用同样的方法进行描述。形状特征中以一个主要特征为例进行描述,其它主要特征的描述方法与此相同。对这个主要特征描述

了它的特征名称、特征标号、几何要素、辅助特征和次要特征，并对几何要素和次要特征之一进行了具体描述，其它可以照此办理。这样就给出了整个传动轴的信息描述框架，根据这个信息描述框架就能描述整个传动轴的信息。

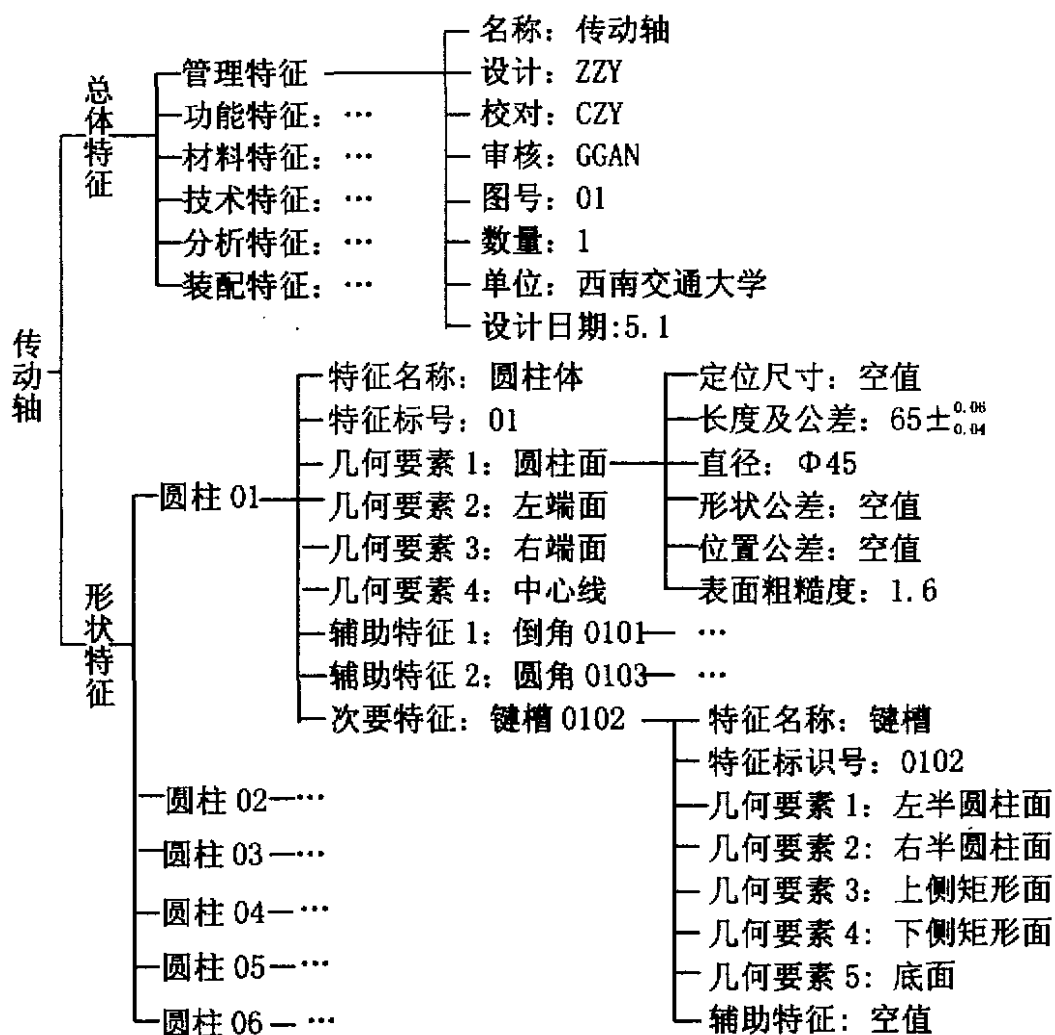
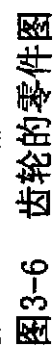


图 3-5 传动轴的信息模式及描述实例

2. 减速箱中齿轮的信息模式及描述实例

利用齿轮类零件信息模型对减速箱中如图 3-6 所示的齿轮进行描述。由图可知它的形状特征包括两个主要特征：齿胚外圆柱和齿胚内圆柱，对它们的分析与轴类零件相似，在这里不进行展开。需注意的是齿胚外圆柱演变为宏特征——齿



形, 在机械领域中对它有规范的表示方法, 在这里不进行具体说明。齿轮的总体特征有它自己的特点, 在这里进行详细的分析, 描述结果如图 3-7 所示。在总体特征中选择功能特征和齿轮公差进行具体描述, 其它特征用同样的方法进行描述。

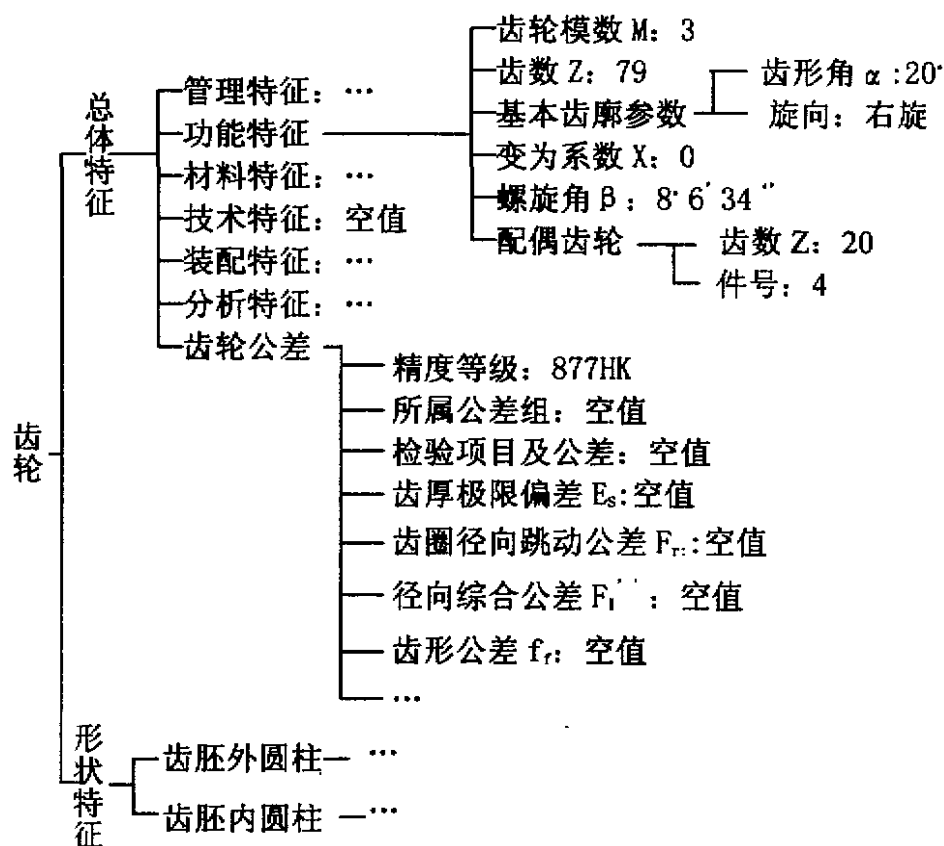


图 3-7 齿轮的信息模式及描述实例

3. 减速箱中滚动轴承的信息模式及描述实例

对如图 3-8 所示的滚动轴承, 采用轴和齿轮类零件信息模式的描述方法, 利用滚动轴承信息模型对它进行描述, 结果如图 3-9 所示。滚动轴承是标准件, 它只具有总体特征, 而总体特征只包含管理、功能等特征。它的管理特征与轴、齿轮的管理特征存在较大的差异, 应给出所采用的国家标准、型号以及标注方法, 具体描述如图 3-9。它的功能特征描述它的性能, 应表明它的额定动载荷、额定静载荷以及极限转速。滚动轴承的描述方法代表了标准件的描述方法, 对其它如键等标准件的描述方法与此相似。这样利用标准件信息模型就能描述相应的标准

件。

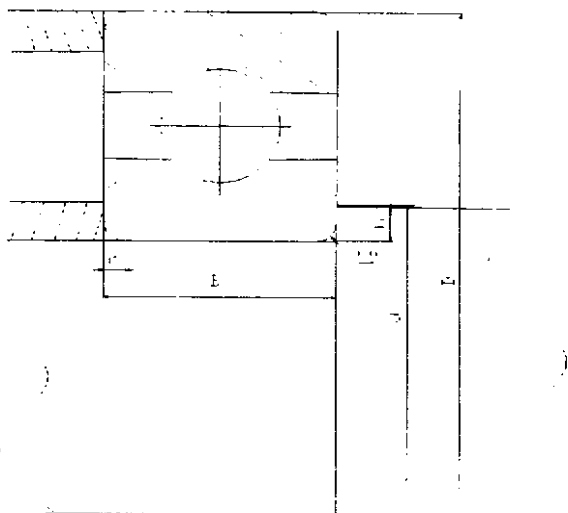


图 3-8 滚动轴承及装配示意图

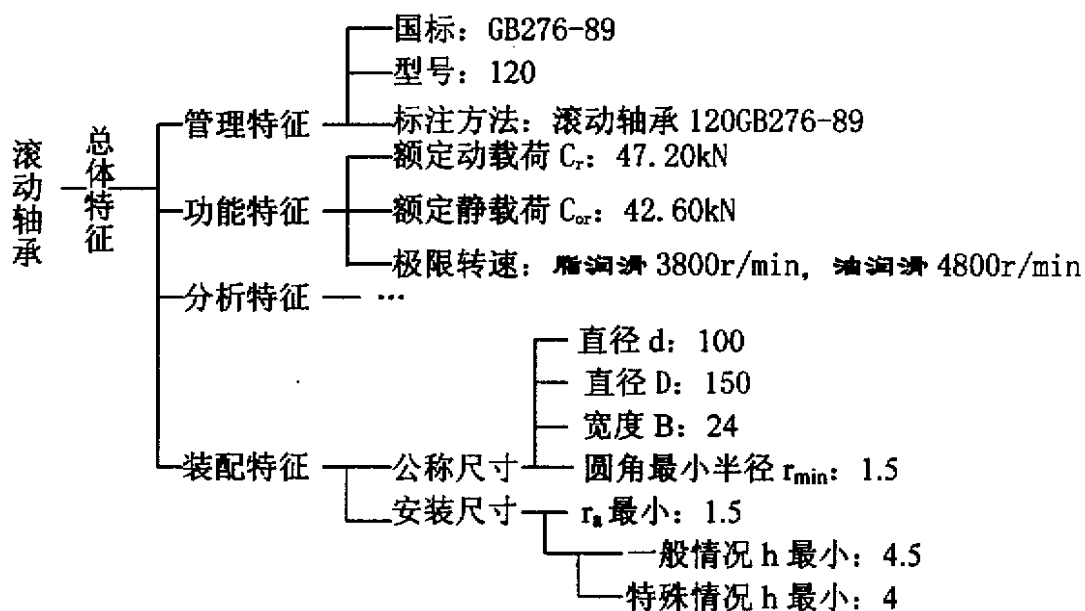


图 3-9 滚动轴承的信息模式及描述实例

3.4 装配体特征信息模型的构造

装配体信息模型是描述装配体设计和制造过程所需信息的一组人为约定的概

念和组织结构。它的描述与零件特征信息模型的描述相似，但是描述的内容有所不同。

3.4.1 构造装配体信息模型考虑的因素

装配体是由多个零件或子装配体按照一定组合方式装配而成的结构。为了能对装配体的有关信息进行描述，建立装配体信息模型时要考虑如下几个方面的因素。

1. 参与装配的对象

一般装配体至少由两个零件或子装配体构成。描述它时必须指出参与装配的零件或子装配体名称，不用对它们进行具体描述。

2. 安装顺序

装配体一般都有固定的安装顺序。这个顺序是正确将各个零件或子装配体安装在一起的保证。

3. 定位方式

定位方式是零件与零件或子装配体、子装配体与子装配体之间进行装配时定位的方法。它一般采用零件形状特征的几何要素进行描述。

4. 联接方式及配合公差

联接方式指零件与零件之间进行联接时的方法，如键联接、销联接等。在联接处，两零件几何要素的配合公差也要考虑。

3.4.2 装配体信息模型的总体结构

根据零件信息模型建立的方法，可以建立装配体的信息模型。装配体信息模型包括总体信息和装配信息。其中总体信息与零件信息模型的总体特征相似，也包括管理信息、功能信息、技术信息和分析信息，但它们描述的具体内容不完全相同。装配信息是装配体的主要信息，它包括各个装配以及装配顺序。每一个装配又包括参与对象、定位方式、联接方式和配合公差。装配体信息模型的总体结构如图 3-10 所示。

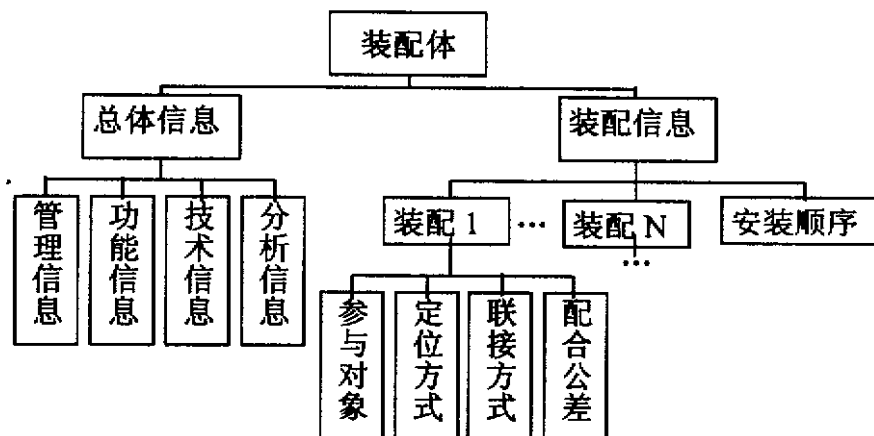


图 3-10 装配体信息模型的总体结构

3.4.3 装配体信息模式及描述实例

利用装配体信息模型及总体结构对由轴、齿轮和键构成的如图 3-11 所示的装配体进行描述，描述的结果如图 3-12 所示。其中对管理信息中的明细栏进行了详细的描述，其它的管理信息以及总体信息可以参照零件信息模式的描述方法进行描述。为了清楚表达安装关系，对装配信息进行了详细描述，有多个装配时可以用同样的方法进行描述，这样就能描述所有的装配信息。

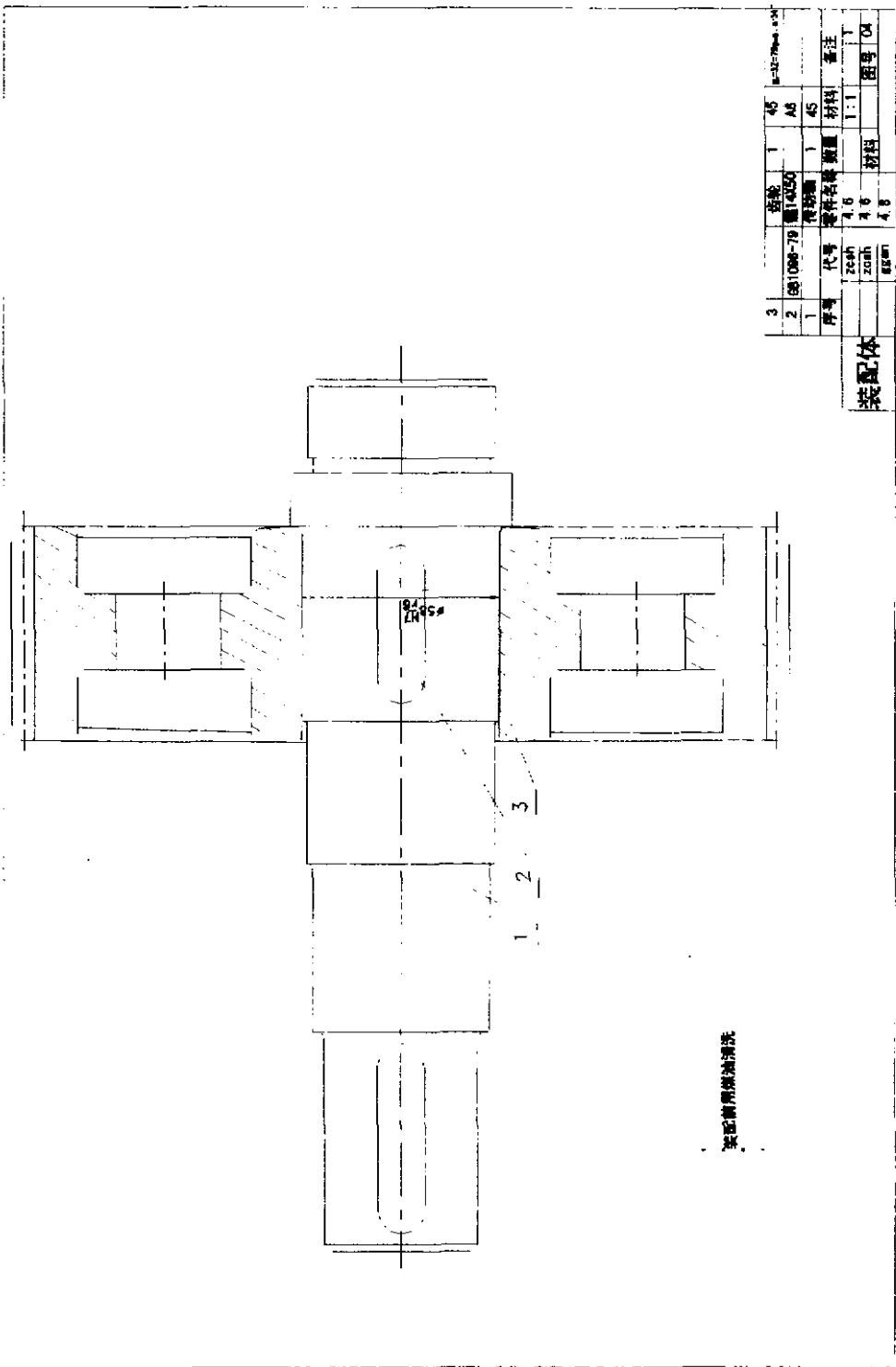


图3-11 装配体示意图

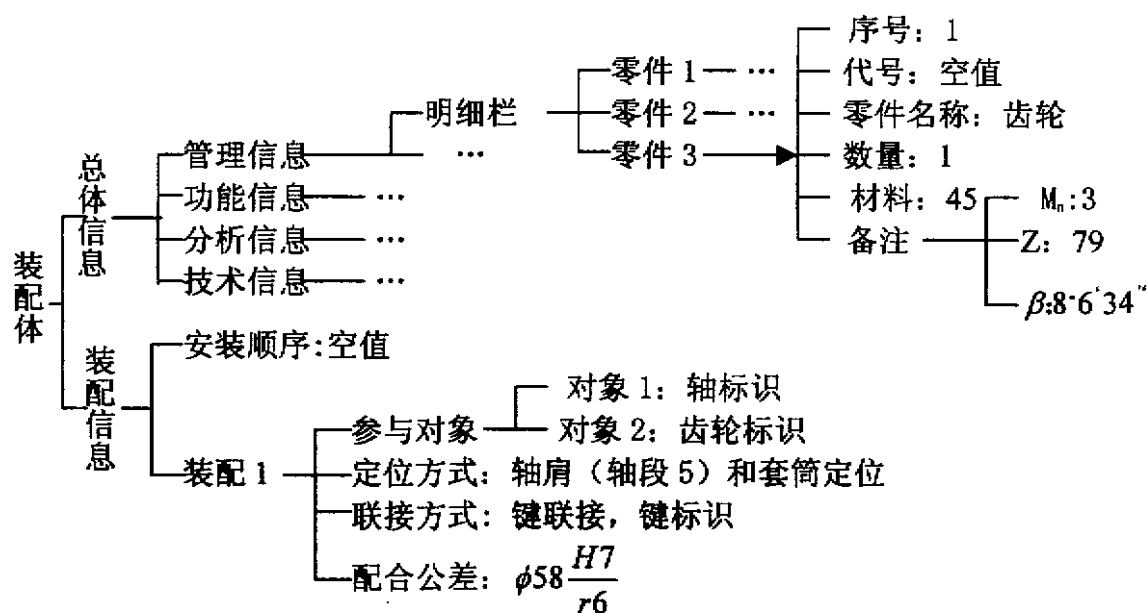


图 3-12 装配体的信息模式及描述实例

3.5 小结

本章研究零件特征的定义以及几种典型零件的特征分类体系，在此基础上给出零件信息模型的定义。由于零件类型不同，它们的特征不完全，结合不同类型零件的特征，建立不同的信息模型。为了验证零件信息模型的有效性，给出了三种具有代表性的典型零件：轴、齿轮和轴承的信息模式并对它们进行具体的描述。最后将建立零件信息模型及总体结构的方法推广到建立装配体信息模型及总体结构，设计装配体信息模型。为了验证装配体信息模型的有效性，给出轴、齿轮、键和轴承组成的装配体的信息模式，并对此装配体进行描述，验证了该信息模型的有效性。

第 4 章 面向对象数据模型研究

数据模型的研究是建立数据库管理系统的关键, 它的类型决定数据库管理系统的类型, 同时在很大程度上决定数据库管理系统的功能。为了建立应用于机械 CAD 的工程数据库管理系统, 本章在前两章工程数据分析以及特征信息模型的基础上, 研究面向对象数据模型以及对象的构造方式和操作, 并应用该数据模型描述减速箱的部分零件以验证其有效性。

4.1 面向对象数据模型的基本概念

鉴于目前对面向对象数据模型的基本概念在不同的应用领域有不同的定义, 还未统一。本节针对 CAD 系统中工程数据描述以及工程数据库管理系统建立的需要, 对面向对象数据模型进行深入研究。在吸取他人研究成果的基础上提出适合于 CAD 系统的面向对象数据模型的定义和一系列相关概念。

4.1.1 面向对象数据模型 (OODM) 的定义

为了解决用户对复杂工程数据的描述和在计算机上的实现, 将面向对象技术的原理和方法应用到工程数据库技术中提出一种适合 CAD 的数据模型——面向对象数据模型 (Object-Oriented Data Model——OODM)。为了有效地设计这种模型, 首先要解决的是什么是面向对象数据模型, 即它的定义。从面向对象数据模型的提出可以看出 OODM 至少包含两方面的特征: 面向对象的特征和数据模型的特征, 因此给出如下定义:

面向对象数据模型是指为了描述一定工程领域中的所有工程数据, 应用面向对象技术的一系列概念、原理和方法提出的一组的相关概念和定义, 这组概念和定义能够说明工程数据的结构、操作以及完整性约束条件。

这个数据模型区别于传统的层次、网状和关系数据模型, 具有面向对象的特征, 同时又集概念数据模型和逻辑数据模型于一体, 既面向用户作为概念数据模

型, 又面向系统实现作为逻辑数据模型。OODM 是现实世界中的对象、对象的约束以及对象间联系的逻辑组织, 同时它能够提供一个对象及其模式的定义、修改和存贮, 数据的查询和更新。

4.1.2 面向对象数据模型的基本概念

在吸取他人研究成果和经验的基础上, 根据工程领域, 特别是机械 CAD 的需要, 提出面向对象数据模型的一组基本概念并给出它们的定义和说明。

1. 类 (Class)

(1) 类的概念

类是对现实世界中一组事物的抽象, 它说明这组事物的共同特征。在日常生活中, 人们认识事物常常是将事物进行分类。“物以类聚, 人以群分”, 就是分类的意思, 可以看出类与分类的概念是密不可分的。通过观察一组相同或相似的事物, 可以归纳出这组事物的共同特征, 对这些共同特征的集中说明就是一个类。由于一组事物的共同特征包括静态和动态两个部分, 因此相应类的说明包括静态特征和动态特征。由类的定义知道类是对一组事物的抽象概括, 是它们共性的说明, 因此类可以看成是这组事物的模板, 通过这个模板可以认识这组事物以及构造某个具体的事物。类的概念是 OODM 中的一个重要概念, 它与其它概念之间存在紧密的联系, 是其它概念存在的基础。需注意的是类与类型是既有区别又有联系。类型是数据结构 (静态特征) 的说明, 不同的数据类型有不同的结构。类可看成是一种特殊的数据类型, 不仅具有静态特征, 而且具有动态特征。一些数据类型不一定是类, 如整型 (int) 等, 因为它不具有动态特征。

(2) 类的描述

由于类是对一组事物的抽象, 因此使用相应类时必须对它进行定义。类的定义包含两个方面: 静态特征的定义和动态特征的定义。静态特征的定义说明这组事物的静态特征, 即类的数据结构。动态特征的定义说明这组事物的动态特征, 即类的行为, 包括协议描述和实现描述。协议描述定义这类事物与其它事物交换信息的接口方式, 对应于函数名和相应的参数类型及参数。实现描述定义这类事物对该信息的具体处理方式, 对应于函数的实现。

2. 继承 (Inherit)

(1) 继承的概念

继承是指一个类能得到另一个类所有特征的两个类之间的关系。其中得到特征的类称为子类，传递特征的类称为父类。子类除具有父类的所有特征外，还具有它自己独有的特征，但子类的特征不能反传给它的父类。子类可以拥有它自己的子类，同样父类也可以拥有它自己的父类。对一个类而言，父类以上的类统称为超类。任意一个类可以拥有多个父类或多个子类，其中多个父类的继承称为多继承，单个父类的继承称为单继承。这样由继承关系可以得到不同的类层次结构（树状或网状结构），称为类树或类网。

(2) 继承的描述

继承是通过在子类的定义中指明其继承的父类的方法进行描述，对父类而言是不可见的。子类在定义时要给出父类的名称，从而子类具有父类的所有特征。在这个过程中可能存在子类和父类某个特征的冲突，解决的办法是在子类中从新定义这个特征。这样当设计者使用该子类的某个特征时，如果在子类中有定义就直接使用，否则根据父类的名称在相应的父类中查找，这样层层向上直到找到为止。如果直到最终的基类也没有找到该特征，则说明此类不具有这个特征。从这个过程可以看出，如果子类和父类同时定义了某个特征，那么使用子类时子类的该特征屏蔽掉父类的这个特征，使用父类时子类的该特征不起作用。在多继承中可能存在多个父类的相同特征冲突，解决办法是通过指定父类的特征重新在子类中定义该特征。如齿轮轴类继承了轴类和齿轮类的特征，其中安全系数特征发生冲突（轴类和齿轮类都具有安全系数），于是在齿轮轴类中重新定义安全系数。

3. 对象 (Object)

(1) 对象的概念

对象也称为客体，泛指现实世界中的任何事物，包括实体和抽象概念，是类的实例。对象是论域中某个具体的事物，具有一定的状态、功能和意义（即对象所属类的特征）。这些状态、功能和意义通过相应类定义的数据结构和行为进行体现，因此对象可以看成是一个软件包。在 CAD 系统中，对象大多指设计的零部件，是工程数据库系统管理的主要数据。对象具有多种不同的构造方式，可以形成各种复杂的结构。

(2) 对象的描述

对象的描述包括两部分：对象型的描述和对象值的描述。对象型的描述给出对象所属的类，即定义了对对象的静态特征和动态特征，对象值的描述给出对象在型的描述下静态特征的具体值。这个值可以是整型、浮点型、字符串等基本类型的值，也可以是另一个类的对象名称（实际是标识）。如果是另一个类的对象名称，那么通过这个对象名称可以找到相应的对象。如一根具体的轴，它所属的类是轴类，它具有强度计算等动态特征，除此之外它还具有静态特征。静态特征中有些是用基本数据类型直接给出值，有些使用其它类的对象名称。

4. 属性 (Attribute)

(1) 属性的概念

属性是指对象具有的某种性质。每个对象可以有一个或多个属性，每个属性描述对象的一个侧面。根据属性的不同组合方式可以构成 CAD 系统中的所有设计对象。属性是对象的静态特性，相当于 OOP 中对象的成员变量、关系数据模型中记录的各个字段，它是对象的有机组成部分。如圆柱具有左端面、右端面、圆柱面和倒角等属性。其中属性可以是空值 (NULL)，空值并不等于零，表示该对象不具有此项属性。注意空值和零是两个不同的概念，零表示属性为零，空值表示属性不存在。如圆柱的属性倒角为空值，则表明此圆柱没有倒角，半径为零，则圆柱退化为中心线。

(2) 属性的描述

对象属性的描述包括两个方面：属性型的描述和属性值的描述。其中属性型的描述是在类的定义中进行，并定义相应的变量名。属性值是在对象中进行描述，可以是数字，也可以是另一个对象的名称。在每个具体对象生成时，给各个属性赋予不同的值，可以得到不同的对象。如齿轮类，当给出不同的模数、齿数以及螺旋角时得到不同的齿轮对象，模数和齿数决定齿轮节圆的大小，螺旋角决定齿形：螺旋角等于零度属于直齿，否则属于斜齿。进行属性描述时要考虑属性的约束条件，要使之具有工程意义，否则就违反了对对象的完整性约束条件。如螺旋角不能任意取值，过大过小都违反工程规范，破坏齿轮对象的完整性。

5. 操作 (Method)

(1) 操作的概念

操作是对象的行为,是对象的动态特性,是对对象以及属性进行操纵和管理的方法。属于同一个类的不同对象具有相同的操作,属于不同类的对象具有不同的操作。对象的操作主要完成对象属性的存储、显示、添加、删除、复制、修改、查询、特定属性的检验和计算等各种处理,是工程数据库组织、管理和应用工程数据的手段和方法。在面向对象技术中,对象具有封装性,OODM 中的对象也不例外,操作和属性是封装在一起,只有通过对象的操作,才能访问对象的属性并对它们进行处理。如设计轴时要求进行强度计算和校核,就必须使用轴这个对象的相应操作才能完成。在关系数据模型中记录和对记录的处理是相互分离的、独立的,无法根据特定记录的要求设计特殊的操作,因此它的数据处理能力受到限制。

(2) 操作的描述

操作的描述在类定义时完成,它可分为两个部分:操作说明和操作实现。操作说明是对操作的对外接口进行定义。它规定操作的结果类型以及得到的值、操作名称、操作的使用方法和需要的参数类型以及值。操作实现是定义操作完成特定功能而对输入的参数进行处理的具体过程。在这个过程中该操作可能调用其它的操作辅助完成该项功能。操作说明和实现一般是相互独立的、分离的。操作说明的设计目标是方便设计者使用,具有一定的稳定性,一般不会改变。操作实现可能由于新方法的出现而改变,但不会改变操作说明。这种改变对设计者而言是不可见的,设计者不需要了解该方法的具体实现过程,只需了解该操作的功能以及操作说明即可。如齿轮的疲劳强度校核操作,设计者只需按要求给出相应的参数,就可得到结果:满足还是不满足,而不用了解怎样得到这个结果。当实现过程改变时,设计者也并不了解,也不需要了解。

6. 对象的标识(Identity)

(1) 标识的概念

标识是对象的名称,是在系统中唯一代表对象的符号,故又称标识符。在关系数据模型中,标识称为关键字,是用记录的状态字段进行表示,又称为状态标识。这种标识机制使系统中不能有两个状态完全相同的记录(对象),同时状态改变可能破坏对象的完整性。在 OOP 中,由于对象具有一定的作用范围,常常采用变量作为标识,称为变量标识。变量一般对应内存中的物理地址,由于存储器地址可重用,易引起灾难性的后果。当一个对象被误删后在同一地址写入另一对

象, 可能导致对象引用出错, 另外地址引用也不易于对象在存储器中的迁移。变量标识还存在一个致命的问题是持久性, 它只在一定的范围内有效, 超出这个范围则不复存在。为了避免这两种标识的不足, 在面向对象数据模型中采用逻辑标识。逻辑标识是用户给具体对象的一个全局性的名称, 即对象名。

(2) 标识的要求

①具有一定的工程意义

在 OODM 中, 标识原则上可以是任意的符号或符号串, 但为了便于设计者理解对象的功能以及数据结构, 标识应具有一定的工程意义。如轴的标识是“轴”, 而不是“齿轮”、“轴承”等符号, 否则就会引起混乱, 不利于工程数据的管理。

②标识具有唯一性

由标识的定义可知, 标识要求具有唯一性。

(3) 逻辑标识的描述

OODM 中的逻辑标识有两种描述形式:

①直接将对象名作为逻辑标识

这种标识在同一个库中能高效地发挥作用, 区分不同的对象, 同时具有较好的工程意义, 但是不能跨库操作, 对长事务的支持较弱。

②对象名前边加用户名作为逻辑标识

一般用户与子库是一一对应的, 因此可以用子库名代替用户名。这种标识可以非常方便地实现跨库操作, 支持工程长事务, 不同的库中可以有相同的对象名。

对一个用户而言, 它的大部分操作是在子库中完成, 因此主要采用第一种标识描述方法, 它具有非常高的效率。当要进行跨子库操作时才使用第二种标识描述方法。这样标识既具有唯一性, 不同的用户又可以有相同的标识。不同的用户可以有自己独立的名字空间, 可以按照自己的爱好和习惯给对象取名。标识一般都是由汉字组成, 也可以是字符串。

4.2 对象的构造方式

CAD 领域中的工程数据对象复杂多样, 要对它们进行有效的管理, 必须对它们的构造方式进行研究, 针对不同对象的属性组合有不同的构造方式。

4.2.1 对象的分类

对象是工程数据库系统处理的基本单元，是属性和操作的封装体。一般每个对象都具有不同的属性，按照属性的不同数据类型可以将对象分为三类：原子对象、简单对象和复合对象。

1. 原子对象

原子对象指由基本数据类型定义的数据，是构成其它对象的基本单元。其中的基本数据类型是由主语言提供的单值数据类型，如整型、浮点型、字符串等。顾名思义原子对象具有原子性，不可再分，它的值是单个的数值或字符串，如用于描述产品长、宽、高等的数据。原子对象是没有类与之对应的特殊对象，因此原子对象没有自己的操作。

2. 简单对象

简单对象指由多个原子对象作为其属性构成的对象。如用横坐标 X 和纵坐标 Y 描述的点，其中的 X 和 Y 都属于原子对象。简单对象一般可以用数值或字符串在对象生成时进行构造，因为它的属性可以由数值或字符串直接给出。简单对象处理非常方便，能直接完成对象属性数据的管理，不会涉及到其它对象的操作。

3. 复合对象

复合对象是一个递归的概念，指至少将一个简单对象作为属性构成的对象。用户可以根据工程的实际需要，将复合对象进行多次复合。复合对象是工程数据库系统处理的主要对象。它的操作一般要涉及到多个对象的操作才能完成，甚至有时会跨几个子库，使处理非常复杂。复合对象根据属性的不同构成情况还可进一步分为简单复合对象和复杂复合对象。

(1) 简单复合对象

简单复合对象指由一个或多个简单对象和原子对象作为属性构成的复合对象。简单复合对象的操作最多只涉及其中简单对象的操作，波及的层次较浅。

(2) 复杂复合对象

复杂复合对象指至少含有一个简单复合对象作为属性的复合对象。复杂复合对象以及简单复合对象在生成时一般不会得到具体的值，而是给各个非原子属性赋予简单对象或复杂复合对象的标识。在对复杂复合对象进行处理时，需要根据

这个标识去层层查找相应的值, 波及的层次较深。

4.2.2 复杂对象的构成方式

原子对象是由单值构成, 因此它的构造方式简单, 在此不作讨论。本文要讨论的对象构成方式主要指简单对象和复杂复合对象以及简单复合对象的聚合构成方式。所谓聚合指由多个对象按照一定的组合构成另一个对象的方式。聚合有多种形式, 可以分为基本聚合方式和组合聚合方式。

1. 基本聚合方式

(1) 集合

集合指同种数据类型的对象没有先后顺序地聚集在一起的方式, 将通过这种方式构造的对象又称为集合对象, 一般用 “ $[O_{s1}, O_{s2}, \dots, O_{sn}]$ ” 表示。其中对象的数据类型包括系统提供的任意数据类型和用户自定义的数据类型, “[]” 是集合构造符 (下文的 “{ }” 是关系构造符, “ $\langle \rangle$ ” 是序列构造符), O_{si} 是参与集合的对象。在工程数据中有的对象就是通过这种方式进行构造。如减速箱中用到的轴承对, 两个轴承完全相同, 并且没有顺序, 可表示为: 轴承对 [轴承 1、轴承 2]。再如关系数据库中的一般表, 它由许多相同结构的记录组成, 分别设为记录 1 到记录 N, 则表的聚合方式可表示为: 表 [记录 1、记录 2... 记录 N]。

(2) 关系

关系指由不同数据类型的对象聚集在一起的方式, 将这种方式构造的对象称为关系对象, 一般用 “ $\{O_{r1}, O_{r2}, \dots, O_{rn}\}$ ” 表示。这种聚合方式强调对象之间数据类型的不同, 是对象构造的主要方式, CAD 系统中的大部分对象都是用这种方式进行构造。由轴的信息模型知道轴是通过关系构造方式进行聚合的, 表示如下: 轴 {总体特征、形状特征、精度特征}, 而总体特征又由管理、功能、材料、分析等不同数据类型的特征构造出来, 表示如下: 总体特征 {管理特征、功能特征、材料特征、分析特征...}。

(3) 序列

序列指多个相同类型的对象按照一定的先后顺序聚集在一起的方式, 将通过这种方式构造的对象称为序列对象, 一般用 “ $\langle O_{s1}, O_{s2}, \dots, O_{sn} \rangle$ ” 表示。这种构造方式强调对象数据类型相同的同时, 并强调各个构成对象之间的顺序。如点的构

造,表示为:点<X、Y、Z>,需指出的是 X、Y、Z 的数据类型要相同,同为整型(int)或同为浮点型(float)。再如轴的主要特征——由多段圆柱构成,可表示为:主要特征<圆柱 1、圆柱 2、...>。

2. 组合聚合方式

组合聚合方式是一种递归的方式,指由多种基本聚合和组合聚合按照一定的排列构造对象的方式。从基本构造方式出发,每种构造方式生成的对象都可以再进行聚合,生成更高一级的对象。

集合对象可以再进行关系构造,也可以进行序列构造,还可以进行集合构造。如几何要素线段它由两个点对象通过序列的方式构成,表示如下:线段<点 1<X、Y>、点 2<X、Y>>。再如关系中的表,假定它的记录的各个字段数据类型相同,则有如下表示:表[记录 1[整型、整型...整型],记录 2[整型、整型...整型]...记录 N[整型、整型...整型]]。

同样关系对象也可以再进行集合、关系和序列构造,形成更为复杂的对象。

序列对象也能仿照同样的方法进行构造。

由这三种构造方式不断地构造下去,可以形成无限多种组合方式。利用所有这些构造方式就能表示 CAD 系统中所有的工程数据,但是描述所有的工程数据并不会用到所有这些构造方式,其中有少数的构造方式最常用。关系对象的集合、关系和序列构造形成更复杂对象在工程数据库最为常用,尤其以关系的关系构造所占的比重最大。如轴类零件它由总体特征和形状特征通过关系构造方式生成,表示为传动轴{总体特征、形状特征}。总体特征又可表示为:总体特征{管理特征、功能特征、材料特征...}。形状特征由多个主要特征构成它可以表示为:形状特征<主要特征 1、主要特征 2...主要特征 N>。主要特征由通过关系构造方式进行构造,表示如下:主要特征 1{几何要素 1、几何要素 2...几何要素 N、次要特征 1...次要特征 N、辅助特征 1...辅助特征 N}。依次下去就能将整个零件构造出来。

4.3 对象的操作

对象的操作能完成对对象的各种处理,辅助设计人员管理工程数据。

4.3.1 操作的分类

对象的操作可以分为两类：内部操作和外部操作。

1. 内部操作

内部操作指某对象具有的，对对象自身的属性进行处理的操作。内部操作具有局部性，它是针对某个特定属性处理需要设计的操作。如设计轴时要对轴进行许用切应力计算、许用弯曲应力、安全系数校核计算以及扭角计算、弯曲变形计算等，这些操作都只是对轴的局部进行处理，都属于内部操作。内部操作可以完成对象属性的赋值、值的修改、值的删除等操作。由工程数据的复杂性可知：每个对象都具有自己的属性，并且不同类型的对象它们的属性大多数不相同，因此不同类型对象的内部操作一般也不相同。只能针对具体对象考虑整个设计过程的需要设计相应的操作，内部操作一般在生成对象的类中进行定义。内部操作的使用也要与具体的类相联系，通过类查找相应的操作。

2. 外部操作

外部操作指将对象作为整体进行处理的操作，如对象的构造、删除、查询等操作。由于这部分操作是以对象为基本单位进行处理，因此它可以不必了解对象的属性。外部操作一般也是在的类中进行定义，但是它们的对外接口有特殊的要求，各个对象的相同操作对外接口要相同。对于某个具体对象，相同的操作，它的实现方式可以不相同，但是这些不同对设计者而言是不可见的。外部操作的实现通常要通过调用对象的有关内部操作完成。通过外部操作，可以提供统一的对外接口，能方便用户的使用。

4.3.2 外部操作

1. 存储

存储操作负责按照一定的物理存储结构，将不同对象作为永久性数据写入物理存储器中，同时要提供它的逻辑标识符作为查找对象的标志。存储操作还要负责修改工程数据库系统中工程数据的版本信息，使存储的对象具有相应的版本。

2. 显示

显示操作是负责将对象按照用户要求的方式在显示器中正确显示，从而使工

程设计人员能见到所设计对象的形状、大小以及具体的参数。显示操作是设计人员感知对象状态的一种途径，其最主要的功能是图形数据的显示。

3. 打印

打印操作负责将对象在打印机上按用户的要求正确进行打印，包括各种打印格式的设置以及纸张大小的选择。通过打印操作可以将设计对象显示在图纸上，设计人员之间利用图纸进行信息的相互交流，与传统的工程数据信息交流方式相似。另外可将对象的图纸进行存档管理，增加工程数据的安全性。

4. 继承

继承操作指在现有类的基础上根据需要派生出新类的操作。派生的新类，即子类具有被继承类（父类）的所有属性和操作。同时子类可以重载父类的某些操作，使两个类的这些操作具有相同的对外接口，而操作的具体实现不同。另外对于多继承要解决继承操作的冲突，即多个父类具有相同操作的处理，一般采用重载的方法，利用父类的操作将该操作在子类中进行重新定义。

5. 构造

构造操作一般是根据需要通过定义相应的类生成新的对象的操作。对象的构造有三种基本方式，并可以在此基础上进行各种不同的组合。因此构造操作要根据设计对象的需要，将各个的属性进行合理的组合。根据构造方式的不同，同一个类的对象不一定具有完全相同的属性，这些都由构造操作完成。

6. 删除

删除操作与构造操作作用相反，是从数据库中删除对象。在设计过程中，可能出现这样的情况：随着设计的进行，以前设计的某个对象不再需要。这就需要调用删除操作，将该设计对象从工程数据库中删除。同时将引用对象的引用记数减一，如果此时被引用对象的引用记数为零，则删除该被引用对象。

7. 复制

复制操作是制造对象的副本，即生成一个基本完全相同的对象，只是它们的标识不相同而矣。复制操作过程分三步完成：第一步是由被复制对象的类生成另一个对象，第二步是将被复制对象的各个属性值赋给新生成的对象，第三步将被原对象引用的对象的引用记数加一。

8. 查询

查询操作是指在设计过程中按照设计者的要求从工程数据库中访问某个对象的操作。查询操作是设计者检索和修改工程数据的基础。对于设计中的各种标准数据,只有通过相关条件的查询才能被找到,从而可被用户使用。对于设计的结果也只有通过查询才能显示或打印。

9. 版本管理操作

版本管理操作是对同一个设计对象的不同版本进行管理时需要的各种操作。它包括添加版本、删除版本、修改版本、设置当前版本等。当对象的新版本生成时,利用添加版本操作在版本数据中增加一项数据结构用于描述该新版本。当对象的一个版本被删除时,则相应的删除版本数据中用于描述该版本的数据结构。有时根据设计需要可以修改对象的版本号或者置某一版本为当前版本。

4.4 对 OODM 的评价

4.4.1 支持信息模型

面向对象数据模型能很好地支持特征信息模型。对一个具体的轴可以把它看成是一个对象。它的标识是“传动轴”,它的属性有总体特征、形状特征和精度特征,而这些属性是按关系聚合方式进行排列。总体特征由管理、材料、功能、分析等特征通过关系聚合的方式进行构造。形状特征由多个主要特征通过一定的构造方式生成。其它的轴也与此相似。因此可把这些相似的零件进行抽象,得到它们的共同特征,利用这些特征定义相应的类,即轴类。分析轴对象可知:为了满足设计过程中对各个属性的处理它的内部操作包括添加属性、强度计算、刚度计算等,它的外部操作包括构造、删除、显示等操作。在设计过程中可由轴类生成具体的轴对象,可以给轴对象赋予具体的标识以及各个属性的值或其它对象的标识。当然还可在这个轴类的基础上,通过继承得到它的许多子类,以满足对它特殊轴的描述。其它的零件以及装配体由于它们的特征信息模型构造方法相似,因此能支持轴类零件的特征信息模型同样能支持它们的特征信息模型。

由上述分析知道面向对象数据模型完全能支持 CAD 系统中零件和装配体的信

息模型，并且描述与产品有关的工程数据更加自然，符合人们的思维习惯，便于理解。

4.4.2. 支持传统数据模型

面向对象数据模型能支持关系数据模型。通过对象的各种聚合方式可将关系表格、记录等都看成是对象，表格对象是由记录对象通过集合方式聚合而成，记录对象是由各个字段通过关系方式聚合而成。关系数据库中对表格以及记录的一些操作成为对象的外部操作进行定义和说明，同时还可根据需要增加表格和记录的内部操作。这样可以增强关系型数据的管理功能，比关系数据模型具有更强的数据管理能力。

同样 OODM 能很好地支持层次数据模型。在 OODM 中，复合对象本身就表示一种层次关系，在由复合对象的构成知道它的属性还可以是复合对象或简单对象，这样层层递推，就可形成层次状结构。这与层次数据模型的描述相一致，因此它能支持层次数据模型。

面向对象数据模型能支持网状数据模型。面向对象数据模型具有继承机制，并且继承分单继承和多继承。通过多继承一个类可以有多个父类，父类还可有多个父类（相当于祖父类）以及更高一级的祖父类，而这些父类或祖父类又可有共同的父类，从而形成复杂的网状结构。利用这种复杂的网状结构可以描述网状数据模型能描述的数据。另外一个复合对象可以是多个其它复合对象的属性，这样也能描述网状数据模型能描述的数据。

4.4.3 支持永久性

在面向对象环境中，可以把对象分为两种：易失对象和永久对象。易失对象被分配在易失存储器中（通常是内存），与程序中普通变量一样，当它们所在的程序结束时，这些对象就消失。永久对象被存储在永久库中（通常是外存），不随程序的运行结束而消亡。在面向对象数据模型中标识能够唯一地代表对象，只要标识存在，对象就存在，对象被删除相应的标识也会被删除。面向对象的工程数据库是永久对象的集合，每个对象用唯一的标识表示，这个标识符被看成永久对象的标志。

4.4.4 有利于工程数据库管理系统的实现

在面向对象技术高速发展的今天,大多数程序都采用了面向对象技术设计和实现,如采用 VC++、VB、DELPHI 等高级语言编程。实际编程时对问题的分析,可以采用面向对象分析方法,弄清面临的问题,然后针对分析的结果,应用面向对象设计方法,设计各种类,并生成相应的对象。面向对象数据模型的分析过程正好与这个过程一致,采用 OODM 可开发工程数据库管理系统 EDBMS,使应用程序语言(VC++)与 EDBMS 的宿主语言一致,可使应用程序与 EDBMS 耦合紧密,而且操作数据更方便。

4.5 实例分析

为了说明面向对象数据模型的有效性,本节应用这个数据模型的一系列概念,将第三章中用信息模型描述的轴和装配体改用面向对象数据模型进行描述。

4.5.1 轴类零件的 OODM 描述模式及实例

对一个轴类零件的描述分为两步:第一步是对轴类零件数据模式的描述,即有关类数据结构的定义,第二步才是由轴类零件的数据模式生成具体对象轴。

1. 轴类零件的数据模式

由面向对象数据模型以及信息模型可以知道轴类零件的数据模式为:轴类零件标识{总体特征标识、形状特征标识},然后可以进一步分析总体特征和形状特征。总体特征的描述模式为:总体特征标识{管理特征标识、功能特征标识...},对管理特征作进一步描述描述为:管理特征标识{名称、设计、校对、审核、图号、数量、单位、设计日期},其它总体特征可以用同样的方法进行描述。形状特征的描述模式为:形状特征标识<主要特征 1 标识、主要特征 2 标识、...主要特征 N 标识>。主要特征 1 的描述模式为:主要特征 1 标识{特征名称、特征标号、几何要素 1 标识、...几何要素 N 标识、辅助特征 1、...辅助特征 N、次要特征 1 标识、...次要特征 N 标识}。几何要素的描述模式为:几何要素标识{定位尺寸及公差标识、定形尺寸及公差标识、形状公差标识、位置公差标识、...表面粗糙度标识},其中定形尺寸及公差的描述模式为:定形尺寸及公差标识{定形尺寸、定形尺寸上偏差、

定形尺寸下偏差}。所有其它的特征及几何要素等可以用同样的方法进行描述,因

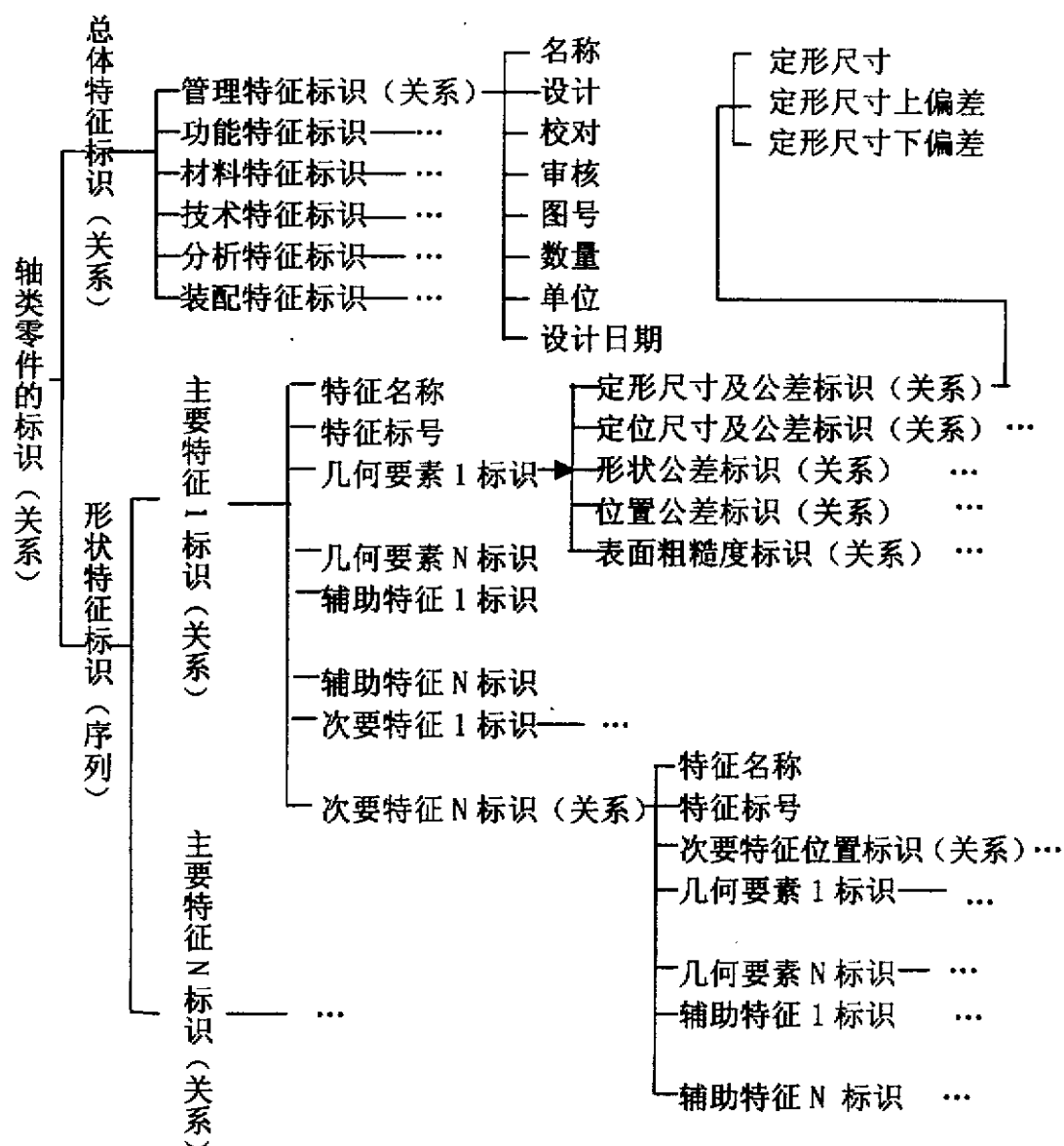


图 4-2 轴类零件的数据模式

此可以得到如图 4-2 所示的轴类零件数据描述模式,其中括号内的文字表示构造方式。

2. 轴类零件的描述实例

轴类零件的描述实例就是将数据模式中的原子对象用具体的数值代替，而将简单对象和复合对象用它们的标识符代替。以图 3-4 所示的传动轴为例描述如图 4-3 所示，其它的总体特征可以参照管理特征进行描述，所有未描述的特征都可以参照相关的特征进行描述。

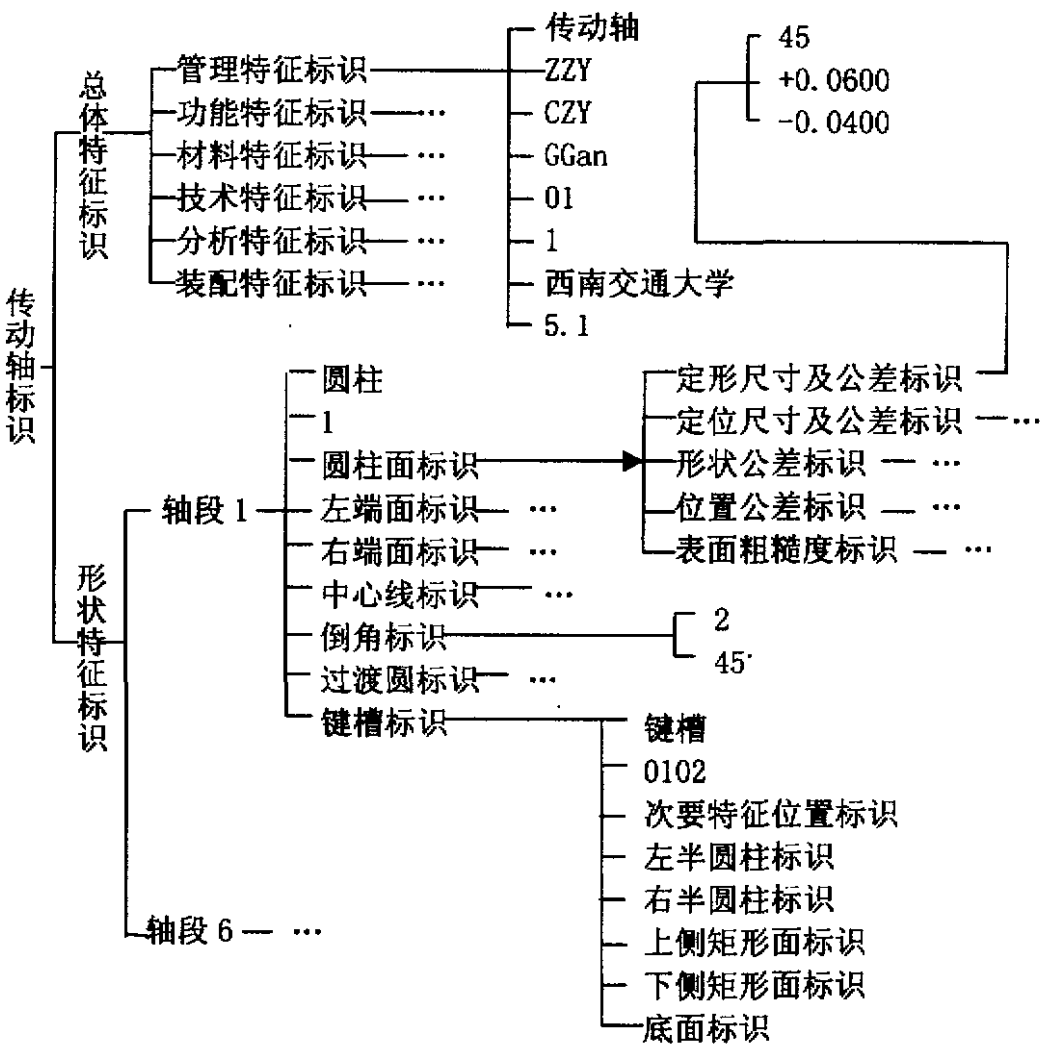


图 4-3 传动轴的简单描述实例

4.5.2 装配体的数据模式及描述实例

借鉴轴类零件数据模式的描述方法,并根据装配体信息模型的描述,利用面向对象数据模型的概念可以设计装配体数据模式如图 4-4 所示。

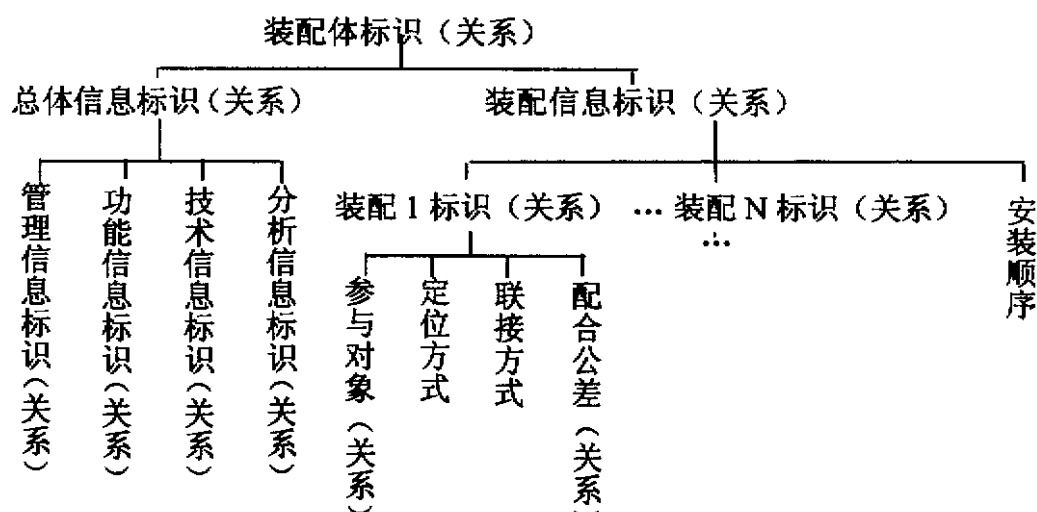


图 4-4 装配体数据模式

利用该数据模式以第三章中的图 3-11 所示的装配体为例进行具体描述,其它装配体可以参照此例进行描述。它的管理信息包括标题栏以及明细栏等所有信息。它的功能信息为空值,因为这是一个简化后的装配体,它基本上没有功能信息。技术信息即是该装配体的技术要求。分析信息对该装配体而言同样为空值。它的装配信息只有一个,因此不存在安装顺序。参与装配的对象有齿轮、轴和键,它的定位方式为轴肩定位,联接方式是键联接,配合公差是基轴制。由于它的总体特征描述与轴类零件相似,在这里不再进行描述,而只描述装配信息,具体描述如图 4-5 所示:

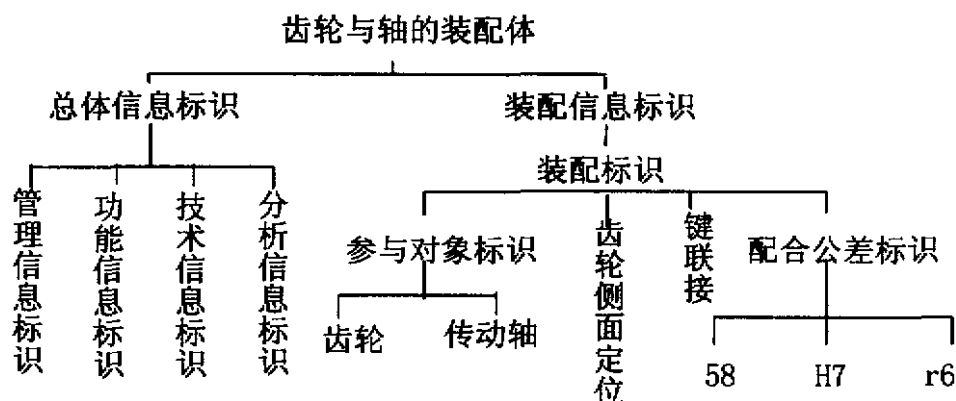


图 4-5 装配体数据模式描述实例

4.6 小结

本章将数据库技术和面向对象技术相结合，给出面向对象数据模型的定义和相关概念：类、继承、对象、对象的属性、对象的操作和对象的标识，并对这些概念进行了定义和说明。根据对象的定义研究了对象的构造方式以及对象的操作，最后利用这些基本概念和对象的构造方式描述了轴类零件和装配体的数据模式，并应用相应的数据模式对实例进行了描述。对于其它的所有零件和装配体都可以利用同样的方法进行描述，不管零件有多复杂都可以通过三种基本构造方法及其组合进行构造。在构造过程中遇到没有相应类进行定义的对象，可以根据类的继承操作在现有类的基础上进行定义。对象的各种管理操作可以通过各种外部操作完成，对象属性的管理可以通过对象的内部操作完成。因此可以得到这样的结论：面向对象数据模型（OODM）能够描述工程领域中的所有工程数据，是适合 CAD 系统中工程数据管理的有效模型，为工程数据库管理系统的建立打下了良好的基础。

第 5 章 EDBMS 的系统设计

本章首先研究 EDBMS 的开发策略及工程数据库的体系结构，而后探讨工程数据库管理系统的总体设计方法。

5.1 EDBMS 开发策略研究

开发策略的研究和选择在很大程度上决定了工程数据库管理系统的功能和开发的工作量，因此本节首先研究 EDBMS 的开发策略。纵观所有 EDBMS 的开发策略可以将其分为三类。

5.1.1 开发专用的工程数据库管理系统

这种策略的特点是只针对某个专门的应用领域开发专用的工程数据库管理系统，针对性特别强，与应用系统间是一对一的关系。它只考虑该专用领域的要求，一般不提供与该领域无关的功能。有的提供专用的数据库数据描述语言（DDL），有的是固定的模式和结构，不可改变。由于是针对单一系统，一般功能较单一，因此采用这种策略开发的工程数据库管理系统结构较简单，实用性特别强，工作效率高，开发的工作量小，周期短。其不足之处是缺乏通用性，局限性很大，系统常与应用程序紧密结合，修改和维护困难。

如计算机辅助工艺设计管理系统（GS-CAPP）中用到的工程数据库系统。该系统采用 STEP 标准建立全局产品数据模型，自主开发专用的工程数据库，重点解决零件工艺规划过程中的工程数据管理。又如为设计土建施工图而开发的工程数据库管理系统（ADBMS）。它提供许多标准件描述文件用于描述土建中常用到实体的几何与非几何数据。这些标准件描述文件描述的实体不能应用到机械领域，因此这个数据库管理系统不具有通用性。还有为工业产品外观设计而开发的工程数据库管理系统（OOEDBMS），它只适合管理产品外观设计过程中出现的工程数据。

5.1.2 扩展商品化的 DBMS 适合工程应用

扩展某些较优秀的商品化数据库管理软件用于工程设计是目前国内外开发工程数据库管理系统较普遍采用的开发策略。它们都是以原系统为基础,在某些功能上进行加强,但所采用的实现方式有较大的区别,可分为两种。

1. 在商品化 DBMS 基础上建立工程应用接口

利用已经商品化的 DBMS 软件,如关系数据库管理系统 Visual FoxPro, Oracle, Informix 等,充分利用这些系统的核心功能模块:存储管理、事务管理、数据库控制以及维护工具等。在这些系统之上,开发一层面向工程应用的工程数据库管理系统的界面作为与应用程序的接口。由于商用数据库系统可描述的数据模型与工程应用的数据模型不匹配,因此在接口部分要实现两种模型中不匹配部分的转换。这种方法的优点是避开了实现难度较大的 DBMS 核心模块的开发,直接利用 DBMS 的核心,实现容易,缩短研发周期。其不足之处是工程数据库管理系统的功能受所选底层 DBMS 功能的约束和限制,缺乏灵活性。

2. 扩展商品化 DBMS 功能

这种方式多为在具有 DBMS 源程序的情况下采用,一般都是商品化数据库管理系统的开发商在原系统的基础上推出新的版本以适应新的需求。这种方式开发的系统功能较强,应用范围较广泛,具有很大的适应性。但是这种系统由于受原系统的影响,功能有一定的局限性,效率较低,同时它的开发者受限制,并不是所有人都能采用。如在 System R 基础上扩充复杂数据类型、表结构等开发的 XSQL 系统。

5.1.3 开发全新的工程数据库管理系统

开发全新的 EDBMS 是目前国内外学者研究的主要目标。研究的内容包括^[17]:工程数据模型、动态模式管理、长事务处理、版本管理、存储技术和缓冲技术以及分布式体系结构,以期建立满足工程设计要求的全新的 EDBMS。采用这种开发策略都要研究工程数据模型,如面向对象数据模型、语义数据模型等,同时考虑多媒体技术和智能知识库技术。采用这种方式开发的工作量、难度都较大,开发周期特别长。目前这类 EDBMS 商品化的软件不多,已商品化的 EDBMS 软件大多局

限于一定范围内的小系统, 通用的 EDBMS 还处在原型的研制之中。

5.1.4 本文采用的开发策略

根据上述开发策略的研究, 本文将综合前述三种开发策略的优点, 着重探讨适合于机械 CAD 需要的 EDBMS, 而且采用第二种和第三种策略相结合的方式开发 EDBMS。根据机械 CAD 的需要, 设计面向对象数据模型描述 CAD 领域中的工程数据。在考虑机械 CAD 需要的同时, 兼顾后续环节如 CAM 的要求, 支持并行工程的需要。同时为了缩短开发周期, 化简开发的难度, 充分利用现有的关系型数据库。借助开放数据库连接 (Open Data Base Connected——ODBC), 将现已存于关系数据库中的工程数据, 转化为面向对象数据模型的描述呈现给用户。这样可以充分利用商品化关系数据库系统的一些有效功能处理管理型数据, 将关系数据模型、层次数据模型以及网状数据模型统一到面向对象数据模型之中。在此基础上开发设计面向对象的工程数据库管理系统, 可以充分利用现正在使用的以关系数据模型为基础开发的工程数据库, 可以方便地与现存系统进行对接, 便于推广使用, 而不浪费现有资源。采用这种策略可以降低开发设计的难度, 缩短周期, 而功能上完全能满足工程领域的需要。

5.2 面向 CAD 系统的数据库体系结构

数据库是存放数据的集合。按照数据的组织关系可以有多种体系结构, 如多层结构、三级结构、分布式结构等等。根据机械 CAD 系统的需要, 数据库可以存储在不同的介质中形成不同的物理数据库, 本文吸取多层结构和三级结构的优点, 尽可能的避免它们的不足, 设计三层次数据库体系结构。

5.2.1 三层次数据库体系结构

在机械 CAD 系统中要存储的数据种类繁多 (第二章已进行了说明), 数据的稳定程度也各不相同, 特别是长事务的管理, 因此不能将所有的数据全部存储在一个库中。如只建一个库, 由于长事务会使数据的共享性降低, 失去数据库的特征。同时由于工程数据种类较多使库的管理非常复杂, 更不用说处理数据的动态性。工程数据中有一部分数据是静态数据, 一般不会改变, 大多只用于查询; 另

一部分是动态数据，它不仅要能查询，而且在必要时可以进行修改等。为了方便数据库的设计和的操作，将数据库分为三个层次，如图 5-1 所示。

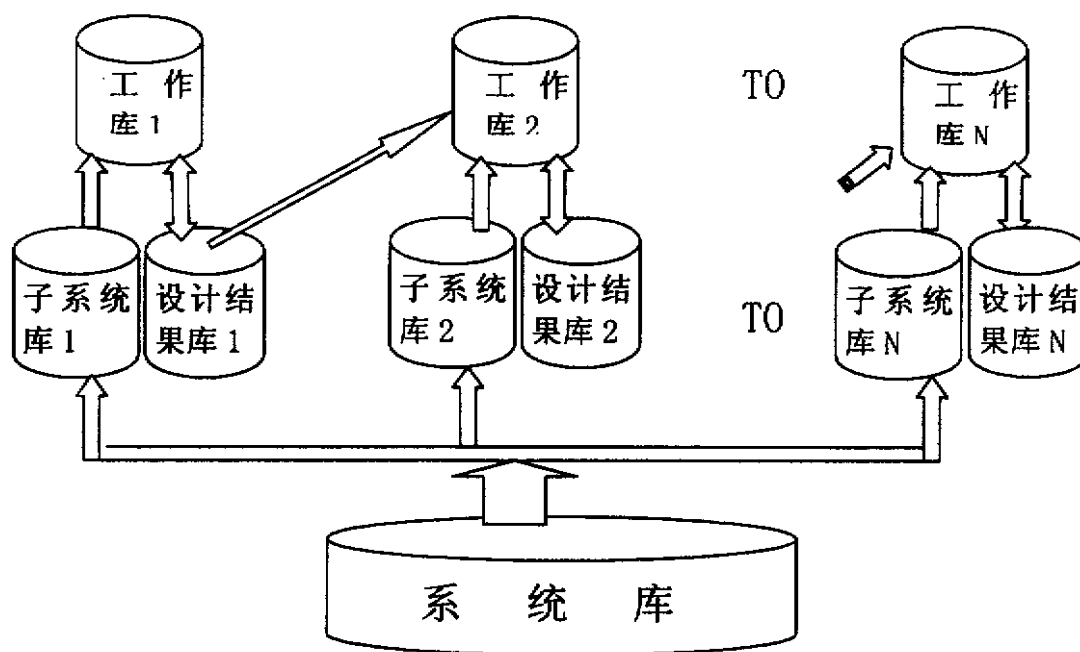


图 5-1 数据库体系结构

它的最底层是系统库，存储各种静态数据，主要采用关系数据库管理系统进行管理。次底层包括子系统库和设计结果库，统称为二级库。子系统库存储设计者从系统库中取得的与设计项目相关的数据。设计结果库存储各种设计结果数据（包括不同版本）。最上层是工作库，存储每个设计者在设计过程中产生的中间结果数据（具有动态性）。这样可以将动态数据和静态数据进行分开管理，同时将需要进行版本管理的数据与不需要版本管理的数据分开，简化版本管理的复杂性。整个工程数据库由多个子库组成，在同一时刻不同的设计者可以在不同的子库中进行操作，便于工程长事务的管理。每个设计项目在开始时，如果需要可以从系统库中取得标准数据放入子系统库中，以后基本上不需要再访问系统库，从而减少系统库访问冲突，提高共享性。当然这是以牺牲物理存储空间为代价的。由于海量存储器的出现并且价格非常便宜，这样作是值得的。由于系统库与子系统库采用的数据模型不一样，因此在子系统库和系统库之间采用开放数据库连接

(ODBC) 进行转换, 各个库之间的数据传递关系如图 5-1 中的箭头所示。数据从系统库传递到子系统库, 最后到达工作库, 而不能反向传递。工作库和设计结果库之间数据可以双向传递, 同时后续设计项目可以从前面的设计结果库中取得数据。需说明的是: 这里的系统库主要是为了与现行的基于关系的工程数据库兼容, 充分利用现有资源, 同时也是为了简化工程数据库管理系统的设计。

5.2.2 系统库

上面已提到系统库是存储静态数据的工程数据库, 一般存储那些在设计过程中要用到的各种标准数据 (包括各种设计手册上的设计规范数据、设计系数、实验数据等等)。这些数据有的已经利用工程数据库管理系统进行管理, 如北京理工大学开发的齿轮精度标准数据库; 有的可以通过适当的变形利用关系数据库管理系统进行管理, 如设计中用到的参数图可以通过离散化变为关系型数据进行管理。系统库可以直接利用关系数据库管理系统进行设计, 也可以包含现存的基于关系数据模型的工程数据库。这样可以简化工程数据库管理系统的设计, 同时能充分利用现有资源, 增加系统的适应性和开放性, 便于推广使用。另外系统库中存储的数据在设计过程中都是不会修改的静态数据, 对它们的操作只有访问, 从而总体上缩短了每个设计者独占系统库的时间, 便于处理工程长事务, 容易保证数据的完整性和一致性。

5.2.3 子系统库和设计结果库

子系统库中存储的是从系统库中取出的、与设计项目相关的标准数据, 在需要时设计者能够对它们进行查询。设计结果库存储的是设计过程中产生的、有多种版本的设计结果数据, 要求能够进行版本管理。另外在需要时后续设计项目可以从前边设计项目的子系统库中取得设计结果数据, 如参考相关零件尺寸等。设计结果库最重要的功能是对同一设计对象的不同版本进行管理, 并提供相应的当前版本供查询。在需要时其它设计者可以取得设计对象当前版本数据而不是旧版本数据, 除非特别说明。在子系统库和设计结果库中, 数据是按照面向对象数据模型进行存储和提供给用户使用。

5.2.4 工作库

工作库是设计者直接使用和接触的特殊数据库。工作库能够存储设计者在设计过程中产生的各种工程数据,包括图形数据和非图形数据,它是一个动态库。在工作时工作库被装入内存,因此数据输入和输出效率高,但它不具有永久性。工作停止时系统自动将工作库中的所有数据转存到硬盘,当然用户也可以根据需要手动转存。在工作库中,所有工程数据只有一个版本,因此不需要进行版本管理。每个工作库主要负责与自己的子系统库交换数据,在极少数的情况下它也能取得其它子系统库的设计结果数据。这样使它的功能较单一,可以集中有效地对动态数据进行管理,便于工作库的设计和维护。

5.3 工程数据库管理系统的总体设计

5.3.1 设计工程数据库管理系统面临的问题

目前在商业领域中广泛应用的数据库管理系统,对于处理 CAD 系统中数量大、形式多样、结构繁琐、关系复杂、规律性差、动态的工程数据,还不能完全满足要求。工程数据库管理系统的研究和开发工作始于 70 年代末期,现阶段研究和开发适合于 CAD 的新一代工程数据库管理系统,还面临着后述的待解决问题。工程数据库管理系统不仅要支持用户定义复杂的数据类型,而且还要支持多对多关系、递归关系等复杂数据结构的描述。对数据模型而言不仅要能统一传统的数据模型,而且还要使数据模型具有更广泛的语义,全面支持 CAD 系统。工程数据库管理系统要能提供模式的动态修改和工程长事务的管理,同时还要求能够进行版本管理,并记录设计过程中的历史数据,使设计能回溯到一个合理的阶段。在分散环境下,设计层次化的数据库是工程数据库管理系统的一个重要功能。为了维护数据的安全,应能对使用者进行有效的管理,可以赋予各类设计人员一定的访问权限,控制一些非法用户访问或修改数据库中的数据。

5.3.2 工程数据库管理系统的功能设计

针对工程数据库管理系统面临的问题,为满足 CAD 系统的需要,对工程数据

库管理系统进行功能设计,以决定所开发的工程数据库管理系统应具有下述功能。

1. 支持面向对象数据模型

第四章已讨论面向对象数据模型具有许多优点,能统一传统的三种数据模型。因此为了实现对 CAD 系统所有工程数据的管理,本文开发的 EDBMS 采用面向对象数据模型。

2. 支持数据模式的动态修改

由于在 OODM 中类具有继承性,因此在设计过程中用户可以根据设计对象的需要,在现有类的基础上设计新类以满足描述产品的需要,或扩展现有类的成员变量。

3. 提供丰富的数据操作

在 EDBMS 中,能提供丰富的操作,包括类的定义、对象的生成、对象的修改以及对象属性的处理等操作。同时针对不同类的对象,还可以定义不同的操作,以完成此类对象的特殊处理。

4. 支持版本管理

在子系统库中,存在设计对象的多个版本。用户可以根据需要对这些不同的版本数据进行处理,如确定当前版本、删除旧版本、添加新版本、查询版本信息等等。

5. 权限管理

为了维护数据的安全性,EDBMS 提供一定的权限管理。对每个设计者限定一定的操作权限,它们只能在权限范围内对数据进行操作,以保证数据的完整性和一致性。

6. 支持长事务

EDBMS 通过对三层次库的管理和维护,提供长事务管理,不仅是本地子系统库中的长事务,而且包括跨库的长事务管理。

5.3.3 工程数据库管理系统的总体结构

为了开发设计三层次体系结构的工程数据库,必须先设计相应的工程数据库管理系统。根据确定的开发策略以及工程数据库管理系统的功能,采用面向对象的方法,针对机械 CAD 的需要设计如图 5-2 所示的工程数据库管理系统。它由四

个功能模块组成：用户接口子系统、对象管理子系统、事务管理子系统和 ODBC 转换子系统。这四个模块分别承担不同的功能并相互协作配合，共同完成面向对象的工程数据库管理系统的所有功能。

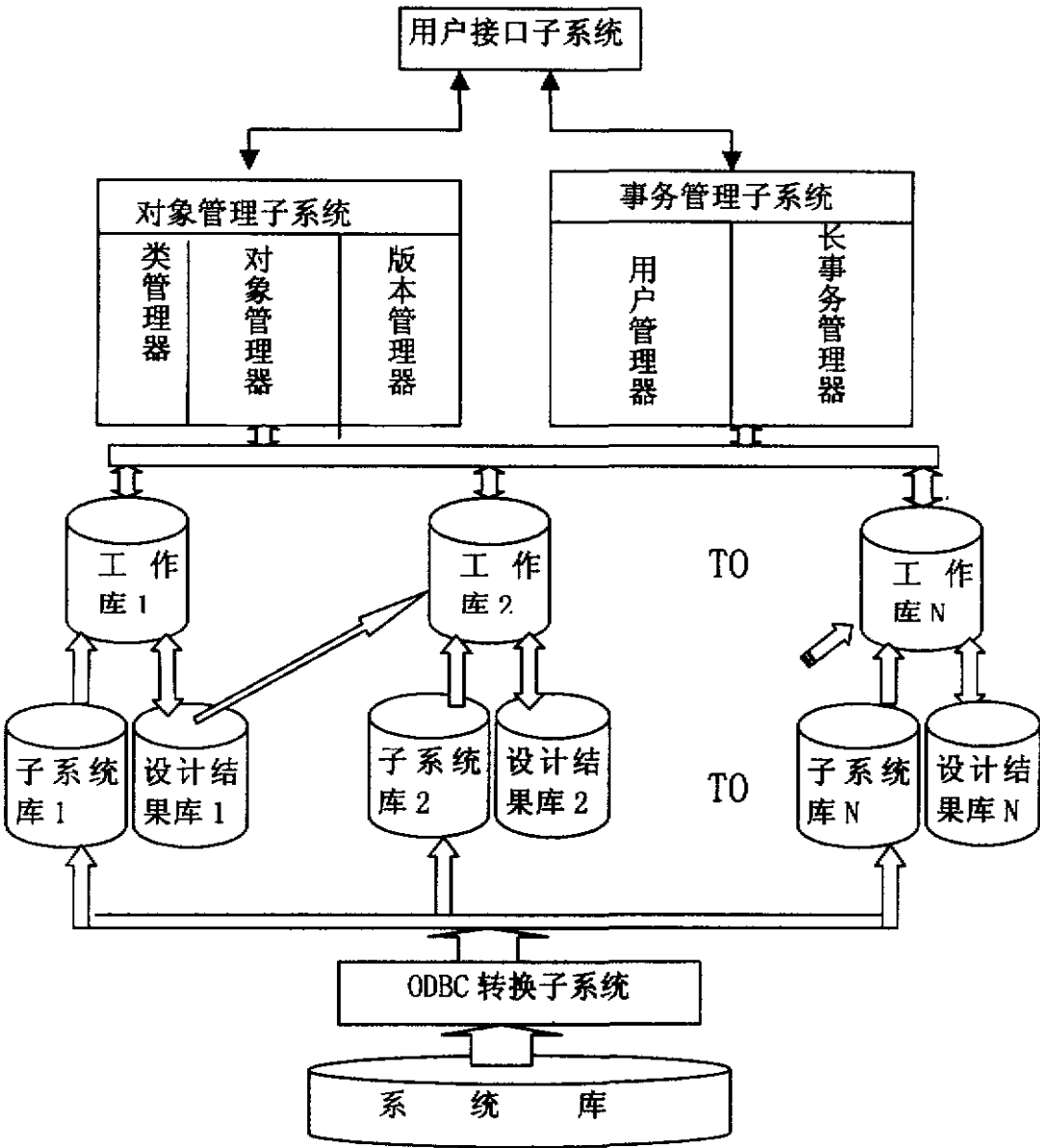


图 5-2 工程数据库管理系统总体结构

5.3.4 用户接口子系统

用户接口子系统是各种用户与工程数据库管理系统交换信息的场所，同时也只有通过用户接口子系统才能使用这个系统，因此用户接口子系统的好坏直接影响着使用 EDBMS 的方便性。本着方便用户的原则，系统提供三种形式的用户接口：图形用户接口、命令语言接口、程序语言接口。图形用户接口是通过图形菜单将各种操作显示给用户，用户通过鼠标或键盘输入命令。这种方式直观、方便，易于操作，是各种软件常提供的基本接口。命令语言接口是通过在特定命令窗口输入系统给定的命令使用 EDBMS 的接口。这种方式对使用者要求较高，需要一定的培训才能使用，但熟悉后它的操作效率特别高。程序语言接口是提供给编程用户通过编写符合语法的程序，由 EDBMS 编译得到各种有关数据库的应用程序。

5.3.5 对象管理子系统

对象管理子系统是工程数据库管理系统的核心，提供类的定义、对象的生成以及对象的版本管理。这个子系统由三个部分组成：类管理器、对象管理器和版本管理器。类管理器负责系统类的管理、新类的生成并解决类继承过程中对象属性以及操作的冲突。在工程数据库设计时，设计者可以根据不同的需要，在系统类的基础上由类管理器通过继承或添加、删除类成员变量的方法定义不同的新类，从而较好地满足实际需要。对象管理器完成由某个具体类生成实例对象的过程，同时对生成的多个对象进行有效的管理，包括对象的查询、存储等操作以及对象属性的各种处理。版本管理器负责不同版本对象的管理，包括对象版本的修改、删除、查询等操作。版本管理是工程数据库管理系统的一项重要功能，是区别于传统数据库管理系统的一个方面，是工程数据管理所必须的功能。

5.3.6 事务管理子系统

事务管理子系统负责 EDBMS 中各种事务的管理。它由两个部分组成：用户管理器和长事务管理器。用户管理器负责管理参与设计的所有用户，包括用户的姓名、权限等属性以及设计任务，同时还要负责管理不同用户工作库与子系统库间建立的联系，为它们之间的数据交换提供管理。长事务管理器提供对各种长事务的处理，包括处理时间长的任务和跨多个子系统库的事务。

5.3.7 ODBC 转换子系统

由于现存的一些工程数据库是用非面向对象数据模型开发的，并在机械 CAD 系统中大量使用，为了充分利用这些数据库资源，设计 ODBC 转换子系统。它负责将非面向对象的工程数据库中的数据转换成 OODM 描述的数据，按用户的要求进行查询。虽然这样做会降低数据的访问效率，但是能够利用现存的各种工程数据库以及成熟的关系型数据库管理系统建立系统库，可以简化工程数据库管理系统的开发难度，同时便于与现有系统进行对接，并大量地推广使用。设计 ODBC 转换子系统是对现实情况和理想状态的一个折衷。在不久的将来，所有的工程数据库系统都采用面向对象数据模型时，则不需要这个子系统就能自由方便地在不同的工程数据库间交换数据。

5.4 小结

本章详细研究工程数据库管理系统的开发策略，比较各种开发策略的优点和缺点，确定本文采用的开发策略：综合多种策略的优点，充分利用现存的各种工程数据库资源，通过 ODBC 转换，以 OODM 将各种不同数据模型进行统一。根据机械 CAD 的需要和开发策略设计数据库的三层次体系结构：系统库、子系统库、工作库。这种体系结构可以将动态数据和静态数据分开，便于的动态数据的处理。另外工作库中动态对象只有一个版本不需要进行版本管理，而子系统库中的静态设计对象可能存在多个版本，需要进行版本管理，这样可以简化版本管理的实现。最后由工程数据库的三层次体系结构完成工程数据库管理系统的总体设计和功能分解，为下一章的工程数据库管理系统的实现技术研究打下基础。

第 6 章 EDBMS 的实现技术研究

EDBMS 的实现技术是基于面向对象数据模型, 利用 VC6.0 程序设计语言对 EDBMS 的体系结构进行编程实现, 使之成为有效的工程数据库管理系统。本章围绕用户接口子系统、工作库、子系统库、设计结果库、系统库的建立和管理介绍 EDBMS 的实现技术。

6.1 EDBMS 用户接口子系统的设计方法研究

用户接口子系统是系统接受用户输入和显示程序运行结果的系统程序。系统使用的方便与否, 很大程度上决定于用户接口子系统的设计。当前较先进的人机接口是图形用户界面, 因此在此仅讨论图形用户界面, 对于其它接口: 命令接口和程序语言接口在这里暂不进行研究。

6.1.1 EDBMS 主界面的设计

根据 EDBMS 的总体结构以及设计过程中数据处理的顺序安排主界面菜单。在主界面上分布有文件、系统库、二级库、类管理器、总体特征、形状特征、事务管理等菜单及工具栏。

1. 主界面菜单功能设计

文件菜单负责项目文件的建立、打开、存贮以及在需要时按一定格式进行打印。系统库菜单负责系统库的建立以及标准工程数据的添加、修改、删除等维护操作。二级库菜单负责利用 ODBC 转换子系统实现数据的转换, 建立子系统库以及子系统库中数据的查询、关闭等操作。同时二级库菜单还负责设计结果库的维护和管理, 包括设计对象的版本管理。类管理器菜单负责根据设计需要定义新的类, 并处理类与类之间的相互关系。总体特征、形状特征及绘图菜单共同完成设计对象的有关工程数据在工作库中的存储和管理。事务管理菜单负责对设计过程中的各种事务进行管理, 包括用户管理和长事务管理。在主界面中各个菜单相互配合

共同完成设计过程中工程数据的管理。

2. 主界面的实现

主界面采用单文档视图结构,利用 VC6.0 的 MFC 应用程序设计向导进行设计。首先在 VC 6.0 的主菜单中选择“File”菜单,在弹出菜单中选择“New”,新建应用程序。紧接着在模式对话框中选择“MFC AppWiard(Eexe)”,在“Project name”编辑框中输入项目的名称,在此输入的是“Management”,选择或输入存放该应用程序项目的文件夹,最后点击“OK”即可。紧接着在第二个模式对话框中,选择单文档结构“Single document”,并在语言框中选择中文,点击下一步“Next”。在随后出现的数据库选项对话框中,选择仅头文件支持“Head file only”,因为该数据库管理系统要利用 ODBC 技术与其它数据库进行连接,然后点击“Next”。在紧接的对话框中都取默认值,点击“Next”。此时在出现的对话框中点击“Advanced”,在“Main Frame Caption”编辑框中输入“工程数据库管理系统”,关闭高级对话框,点击完成即可。到此完成应用程序框架的设计。然后在 VC6.0 的工作室中点击“ResourceView”,再双击“Meun”中的“IDR_MAINFRAME”ID 号,即在右边编辑客户区出现主菜单,此时就可以根据前边的设计,添加相应的菜单项,完成主界面的设计。

6.1.2 文件菜单

文件菜单包括新设计、打开、保存、另存为、打印、打印浏览、打印设置等菜单项,它们各自具有不同的功能。“新设计”负责选择设计项目,建立一个空的工作库,开始一个新项目的设计过程,并且将此设计对象作为第一个版本。“打开”是将一个设计项目(对象)的当前版本装入内存的工作库中,对该对象进行设计或修改。“保存”是将该对象的相应数据从内存工作库存储到硬盘的原文件中,以备今后能够得到该对象的当前状态。“另存为”是将设计对象以另外的文件名进行存储。打印、打印预览等负责对象在打印机上按照用户的要求进行输出。最后是“退出”菜单,结束应用程序。这些菜单的可视化框架 VC6.0 应用程序向导已经生成,可以不必进行修改,只需添加相应的消息处理函数即可。

6.1.3 系统库菜单

系统库菜单是弹出菜单，包括新建、查询、添加和删除四个子菜单项。新建菜单负责建立一个新的系统库。查询菜单负责按用户要求在系统库中查找标准数据，并按一定格式进行显示，供用户参考。添加菜单负责将系统库中没有的标准数据加入系统库中。删除操作负责将系统库中多余的没用数据删掉。它们的实现是利用 VC6.0 的可视化技术。在该菜单的属性框中选择“Popup”属性，出现子菜单编辑框，此时可加入相应的子菜单。在子菜单属性中分别为这些菜单加入相应的标题；新建、查询、添加、删除和相应的 ID 号即可。ID 号可以是默认的，也可以是进行自定义。然后应用类向导，为这些菜单在主框架类中加入相应的消息处理函数，完成系统库菜单的设计。

6.1.4 二级库菜单

二级库菜单包括子系统库和设计结果库。子系统库菜单包括新建、查询、添加、删除和关闭菜单选项。新建菜单负责为一个新项目初始化一个子系统库。查询菜单负责子系统库中数据的查询。添加菜单负责从系统库中将需要使用的标准数据加入子系统库中。删除菜单负责从子系统库中删除用户指定的数据。关闭菜单负责关闭子系统库，以免子系统库中的数据遭到破坏。设计结果库菜单包括查询结果、查询版本、修改版本、设置版本和删除版本，分别负责设计对象结果按要求显示和对象的版本管理。它们的设计方法与系统库中子菜单设计方法相似，只是相应的消息处理函数不同而矣。

6.1.5 类管理器菜单

类管理器负责启动 VC6.0，根据设计的需要在现有系统类和 MFC 类库的基础上通过继承定义新的类或通过添加、删除类的某些属性以满足设计对象描述的需要。同时类管理器还负责现有类的管理。它的实现只需在应用程序框架类中加入相应的消息处理函数即可。

6.1.6 总体特征菜单

总体特征菜单完成设计对象总体特征的设计、修改和查询等操作。它包括管

理特征、技术特征、分析特征、功能特征等子菜单，分别负责对象相应特征的设计、修改和查询等操作，完成工作库中相应数据的管理。

1. 管理特征菜单

管理特征菜单包括标题栏和明细栏菜单。标题栏菜单设计了一个近似零件图中标题栏的对话框，用于设计者输入和输出相应的管理信息。明细栏菜单也可以用同样的方法进行设计。

2. 功能特征菜单

功能特征菜单完成零件功能的设计以及功能特征数据的管理。例如根据设计传动轴的需要，选择合适的电动机型号，由相应的电动机的型号可得到轴的额定转速、额定功率、额定转矩等功能特征。最后利用对话框进行显示，让用户对该系列参数进行了解和查询。

3. 分析特征菜单

分析特征菜单对设计对象的功能、特性进行分析，并判断设计结果是否满足要求。例如在试验系统中按许用切应力对传动轴的最小轴径进行了分析，其它如弯曲应力校核和安全系数校核等可以用同样的方法进行分析，并将计算分析结果进行显示和存贮等操作。在这里分析结果不能直接进行修改，因为这个结果是在各种参数控制下得到的。

4. 装配特征菜单

装配特征菜单可以仿照管理特征菜单利用对话框的形式输入装配的配合精度、定位方式等数据，并可对这些数据进行查询和修改等操作。

5. 材料特征菜单

对于材料特征菜单，设计了一个对话框，让其从子系统库中查询各种材料的属性。这些属性是标准数据，只能查询而不能修改。设计者可以在材料牌号的弹出列表框中，选择不同牌号材料进行查询，直到找到一个符合要求的材料点击确定为止。这样就可以将该种材料的各个属性存入设计对象的材料特征，如点击取消则在工作库中不保存该材料的属性。

6. 技术特征菜单

技术特征菜单是负责设计对象技术要求的设计和修改。在对话框中加入一个可自动换行和接受回车操作的编辑框即可完成它的文本编辑、修改、删除等操作。

同时设计者点击该菜单时，它首先显示原有的技术特征供设计人员参考，如果设计者满意则可保留，否则可删可改。

6.1.7 形状特征菜单

形状特征菜单负责对象形状特征以及精度特征的设计、修改等操作，它由主要特征菜单和次要特征菜单组成。

1. 主要特征菜单

主要特征菜单的设计采用特征表的形式。它能完成主要特征的设计、查询、添加和删除等操作。例如针对轴类零件的设计，它有五个属性页。前三个属性页分别负责对象主要特征的三个几何要素：左端面、右端面、圆柱面的精度特征设计、修改和查询等操作。后两个属性页分别负责次要特征的添加和删除以及依附于该主要特征的辅助特征的设计、修改等操作。

2. 次要特征菜单

由于主要特征和次要特征在组成上有许多相似之处，因此两个子菜单的界面非常相似。例如对传动轴的设计，唯一区别是将主要特征设计中次要特征属性页改为该次要特征所依附的主要特征属性页，就可以完成次要特征的设计以及次要特征数据的处理。

6.1.8 绘图菜单

绘图菜单用于在设计过程中绘制各种草图和显示设计对象的零件图，是工程数据的图形表示。它包括绘制草图、显示零件图以及启动 AutoCAD 等菜单，能够完成设计过程中各种图纸的绘制和图形的显示。

6.1.9 事务管理菜单

事务管理菜单完成事务管理子系统的功能，包括用户管理和长事务管理两个子菜单。用户管理子菜单负责用户的各种权限的管理以及用户其它属性的管理。长事务管理子菜单负责设计过程中跨设计结果库的查询以及长时间设计事务的管理。

6.2 类管理器的实现技术

由于 VC6.0 已经成为大量应用程序的开发语言, 在本数据库管理系统中为了实现阻抗匹配, 以 VC6.0 作为其数据库语言。在该数据库管理系统中内嵌 VC6.0 的编译器作为该系统的类管理器, 因此 VC6.0 的所有语言设施, 如 MFC 类库、Active 控件等都可以应用于工程数据库的设计。在数据库设计过程中, 根据描述工程数据的需要, 可以通过点击界面上的“类管理器”菜单, 启动 VC6.0 编译器。利用 VC6.0 的类设计向导, 可以在 MFC 类库和本数据库管理系统提供的各种类的基础上设计需要的类或对现有类进行修改。对于 VC6.0 的功能和详细使用方法在此不进行具体介绍, 如果需要可以参考有关的资料学习。

6.3 事务管理子系统的实现技术研究

事务管理菜单包括用户管理和长事务管理子菜单。在此仅讨论用户管理, 长事务管理涉及到数据的分布本文暂不进行研究。

为了实现用户管理, 设计一个数据结构, 如下所示:

```
struct UserManagement{  
    CString  User;  
    CString  Mima;  
    CString  Thority;  
}
```

其中 UserMangement 指该数据结构的名称, User 指用户名, Mima 指用户密码, Thority 指用户拥有的权限标识。权限标识可以是多个字符串的组合, 每个字符串在系统中都预定义了它的具体意义。数据库系统管理员通过这个数据结构对各个用户进行管理。每个用户使用该数据库时必须经过如下几个过程。首先向系统管理员提出申请, 给出用户名、将要使用的密以及申请的使用权限。系统管理员根据需要赋予申请者一定的权限。此时该用户就成为这个系统的合法用户, 通过用户名和密码就可以访问相应权限的系统资源。当登陆时要检查用户名和密码, 然后根据申请时赋予的权限让用户使用系统资源。当然用户可以修改他的密

码、用户名等属性。

6.4 系统库的建立技术

系统库是 CAD 系统的基础库,它存储各种机械设计的标准数据(即各种设计手册数据、国家标准、设计规范等)。因此要将标准数据进行适当的处理,以计算机能够接受的方式存储起来,在需要时能够进行查询。系统库的建立一般采用关系数据库管理系统。由于关系数据库的建立技术在许多资料上都作了介绍,因此本节对系统库的建立技术进行简单说明,重点介绍机械 CAD 系统涉及到的各种标准数据的处理。

6.4.1 关系数据库设计概述

数据库设计的基本任务是根据用户需求,处理需求和数据库支撑环境之间的关系,设计出数据库的外模式、概念模式、内模式。它分四个阶段完成:需求分析、概念设计、逻辑设计和物理设计^[41]。

1. 需求分析

要设计一个有效的数据库,必须从系统的观点出发,从多方面对整个组织的需求进行调查和分析,明确数据库的用户和用途。它包括如下几个方面:

(1)信息需求:分析用户需要从数据库中获得的信息内容。在试验系统中指子系统库需要从系统库中获得的标准数据,它包括机械 CAD 系统涉及到的所有标准数据。

(2)处理需求:分析用户需要的各种数据处理功能以及数据处理方式。在试验系统中,数据库的处理需求主要是利用 ODBC 技术进行数据的查询等操作。

(3)安全性和完整性需求:分析用户对数据安全性要求的高低、完整性要求的严格与否,为以下各个阶段的设计提供依据。

2. 概念设计

概念设计是在需求分析的基础上,应用概念设计模型(E—R 模型)表示数据及其相互间的联系,完成外模式的设计。概念数据模型是面向用户描述数据,与具体的 DBMS 无关,是与用户进行交流和数据库移植的依据。

3. 逻辑设计

逻辑设计是将由概念设计得到的数据模式转换成以 DBMS 逻辑数据模型描述的逻辑数据模式, 同时解决数据模式的规范化、安全性和完整性等实现技术。

4. 物理设计

物理设计是根据逻辑数据模式、DBMS 以及计算机系统所提供的手段和施加的限制, 设计数据库的内模式, 即文件结构、组织方式、存取方式等。

系统库的具体设计在这里不进行讨论, 需要时可以参照有关资料学习。

6.4.2 标准数据的来源

为了较好地解决各种标准数据的存贮和管理, 必须了解它们的来龙去脉。只有这样, 才能从本质上掌握各种标准数据间的关系, 才能有效地利用这些关系将它们进行程序化, 通过计算机进行管理。标准数的来源可分为两大类: 一类是通过解析式计算所得的结果。如在齿轮承载能力计算中, 国标提供了大量解析式用以计算, 同时为了简化设计过程, 也提供了图表。另一类是经过有限次试验得到的离散数据, 包括该行业的各种经验数据。对于这些不同数据, 可以根据它们来源和特点采取不同的技术进行处理, 使之利于计算机进行管理。

6.4.3 基于关系数据模型的数据处理技术

关系数据库技术经过三十多年的发展, 积累了大量的成功经验, 并在许多领域得到广泛应用, 它能够对一些标准数据进行有效处理。

1. 数表的处理

针对标准数据的不同来源和特点, 可以采用不同的方法进行处理。

(1) 有限次试验所得到的离散数据的处理

这些数据之间的相互关系仅仅是各组离散数据之间的对应依赖关系, 不存在插值问题。因此可以将它们直接组织成关系表格的形式, 其函数关系如下:

$$f(x) = \begin{cases} a & x_0 \\ b & x_1 \\ c & x_2 \\ \vdots & \vdots \\ \vdots & \vdots \end{cases}$$

其中 $f(x)$ 指函数关系式, a 、 b 、 c 分别指自变量 X_0 、 X_1 、 X_2 对应的函数值。这种数据只在某些点存在对应关系, 不存在连续性, 可以直接利用关系数据库管理系统进行管理。例如电动机系列标准数据, 仅仅依赖其型号可以得到各种具体参数, 而各个型号之间不存在联系, 可以利用 Microsoft Access 2000 建立数据库表格, 如表 6-1 所示。

表 6-1 Y 系列电动机的技术数据

ID	型号	额定功率	满载转速	额定堵转转矩	额定最大转矩	同步转速	极数
1	Y801-2	0.75	2825	2.2	2.2	3000	2
2	Y90L-6	1.1	910	2.0	2.0	1000	6
3	Y90L-4	1.5	1400	2.2	2.2	1500	4
...	...	...	...	...	...	...	...

(2) 利用解析式得到的各种标准数据处理

这种数据间存在函数关系: $Y=F(X)$ 或 $Z=F(X, Y)$, 函数值和自变量之间在本质上存在连续性。在国标中, 为了便于查询这类数据, 将它们进行了离散化处理, 成为函数关系: $Y_i=F(X_i)$ 或 $Z_i=F(X_i, Y_i)$ (其中 $i=0, 1, 2, \dots$)。这种数据在使用时, 如果给出的自变量值不满足要求, 则需要进行适当的插值计算, 而直接利用关系数据模型很难解决这个问题。由于计算机具有处理速度快的特点, 它可以在非常短的时间内检索大量的记录, 比人工检索速度快许多倍。因此可以充分利用计算机的高速性, 将具有解析式的数表进行更小级数的离散化, 构成可以直接满足需要的查询表格。这种细化的离散过程可以有两种方法。一种方法利用原始解析式取微小增量 Δx , 计算相应的 $F(X+\Delta x)$ 作为 $(X+\Delta x)$ 对应的表格数据。函数式 $Z=F(X, Y)$ 也可以用同样的方法进行处理。这种方法对于能找到原始解析式的表格, 数据最精确。另一方法种是在现有表格数据的基础上模拟人工插值过程, 在 X_i 和 X_{i+1} 之间插入各种近似值作为表格的一部分。在使用时, 可以直接进行检索得到相应的值, 而不再进行插值计算。经过这样处理后就可以直接利用关系数据库管理系统对这些数据进行管理和维护。这种方法得到的数据存在一定的误差, 但能满足使用要求。

2. 线图的存贮和管理

线图是函数关系的一种常用表示方法。图示法的特点是直观形象，并能一目了然地掌握函数的变化规律。因此在机械设计标准数据中数据间关系除用数表的形式表示外，还经常用线图来表示。有的采用一般直角坐标表示，也有的采用对数坐标进行表示，采用何种坐标视具体情况而定。这种数据不能直接利用关系数据库管理系统进行管理，必须经过一定的处理，才能以关系表格的形式进行存储。因此对于具有一维函数关系的线图按实际需要可以取非常小的 Δx ，按函数关系 $Y_{i+1}=f(X_i+\Delta x)$ (其中 $i=0, 1, 2, \dots$) 计算相应的 Y_i 值，形式表格如表 6-2 所示。

表 6-2 线图变换表格

自变量	$X_i+\Delta x$	$X_{i+2}\Delta x$	$X_{i+3}\Delta x$	...	$X_{i+n}\Delta x$	...
函数值	Y_{i+1}	Y_{i+2}	Y_{i+3}	...	Y_{i+4}	...

对于二维线图可以用同样的方法进行处理，只是分别取微量 Δx 和 Δy 进行增量计算，形成二维表格。

3. 利用现存的基于关系数据模型的数据库

在机械行业中，存在许多基于关系数据模型的数据库系统，建立了一些存储工程标准数据的数据库。为了简化工程数据库管理系统的设计可以将它们直接引入作为系统库的一部分，这样可以有效地利用现有资源，提高开发的效率。同时也提高了系统的兼容性和开放性，便于该系统的推广使用。

6.5 二级库的实现技术

二级库包括子系统库和设计结果库，子系统库存储与设计项目相关的标准数据，设计结果库存储设计对象不同版本的结果数据。在设计过程应用程序只从子系统库中获取数据，不是系统库，从而减少对系统库的访问次数，有力地支持工程长事务。设计结果库存储和管理具有不同版本的对象，可以集中对设计对象进行版本管理。

6.5.1 二级库的物理组织

数据库的存储结构是工程数据库的物理基础，对数据库系统的性能影响很大。

二级库的物理组织要解决的主要问题包括工程数据对象的物理表示和存储方法、对象文件的组织以及多版本对象的物理组织等方面。

1. 工程数据库常用的物理组织方式^[18]

(1) 记录的存储结构

在一般的工程数据库中,工程数据仍以记录为基本单元进行物理表示和存储,而记录又由若干字段构成。记录的物理表示有三种方法:定位法、相对法和记数法。定位法是按字段的最大可能长度分配固定长的存储空间,不足的字段补以空格的表示方法,一般用于表示固定长度的记录。相对法是将记录的各个字段通过特性字符分开,利用这些特殊字符定位各个字段的方法,它一般能表示变长记录。记数法是在每个字段开始加上一个定长的字段表示该字段的长度的方法,这种方法常用于表示大字段的记录。记录在磁盘上可以分布在一个物理块中,也可以跨多个物理块分布。

(2) 文件的组织

常用的文件组织有如下几种:顺序文件、散列文件、索引文件、树结构文件和簇集索引文件。顺序文件是最简单的文件组织形式,通常用于记录不太多的情形。文件中记录一般都有一个主关键字,通过主关键字,各个记录按顺序进行存储。散列文件是通过某种映射函数 $I(k)$,使任意两个不同的关键字 k_i 和 k_j 之间存在 $I(k_i) \neq I(k_j)$ 文件组织方式。索引文件是在顺序文件的基础上外加若干级索引文件的文件组织方式。树结构文件是将各个记录分布在树结构的结点上的文件组织方式。簇集索引文件是把键值相同的记录集中存放在一个或相邻的多个物理块中的文件组织方式。

2. 对象的物理存储方式

在第四章中已讨论过,对象分为原子对象、简单对象和复合对象。对于原子对象它具有自我标识的能力,一般操作系统能够分配相应的存储空间对它进行存储。对于简单对象和复合对象必须设计一定的存储结构才能对它进行存储。由于简单对象和复合对象具有层次结构,有时多个对象之间还存在网状关系。为了能对它们进行有效的存储,采用分布式存储,即按对象进行存储。每个对象只存储在该类中定义的原子对象属性以及父类中定义的原子对象属性,对于简单对象和复合对象只存储它们的标识,而不存储具体的值。这样既可以减少数据的冗余,

又能保证对象的独立性,可以对单个对象进行修改等操作而不影响其它对象,实现封装性。每个对象存储的内容包括对象的标识、所占存储空间的大小、对象对应的类名、属性 1、……属性 N。其中属性又包括属性的类型、属性值或属性标识,对象的存储格式如图 6-1 所示。

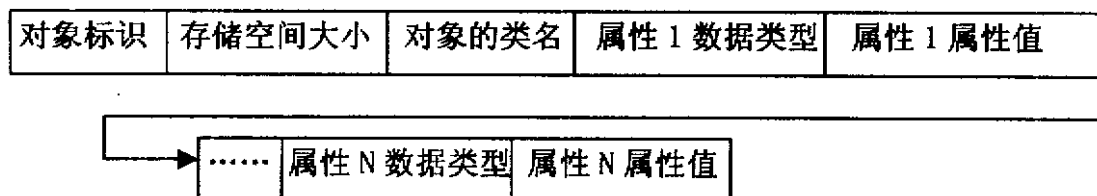


图 6-1 对象的存储格式

对象标识、存储空间大小、对象名具有固定大小的存储空间。每个属性的数据类型与对象数据类型一样,所占存储空间也是固定的。属性值要么是原子对象要么是对象标识,原子对象存储空间固定,对象标识一般是字符型,即标识本身是原子对象,也具有固定的存储空间。因此在对象存储中,每项分配固定大小的存储空间。由于每个对象属性的项数不固定,因此可以利用存储空间大小求得属性得项数。如规定对象存储格式中每项所占空间为 32 个字节,这样只要知道对象存储空间的起始位置,就可以求得各个项目的位置。为了简化问题,如把对象的起始位置定为 0 字节处,则第 32 字节开始的 32 字节标识对象所占存储空间大小。可以将此 32 字节中的数据取出,假设为 280,则可知道此对象有 3 个属性,计算方法为: $(280-32-32-32)/(32+32)=3$,并可求得各个属性的起始字节。

3. 版本管理

在设计结果库中存储的是设计对象的结果数据。这部分数据有不同的版本。为了能对它们进行管理,可以构造一个链表。由于链表具有查询速度快、操作方便的特点,因此能够减少查询版本信息的时间。链表中的每个结点代表一个对象的版本信息,它包含有类对象的指针等相关信息,如存放的文件名称等。利用链表的有序性,可以相应地标识版本的产生顺序,如图 6-2 所示。

图中“Head”指版本链表的头结点,每个结点存储一个对象的版本信息。在 VC 6.0 中, MFC 提供了一个 CObjectList 对象链表类,它能存放 CObject 类及其派生类对象的指针。因此它可以存放设计对象的指针,即设计对象。

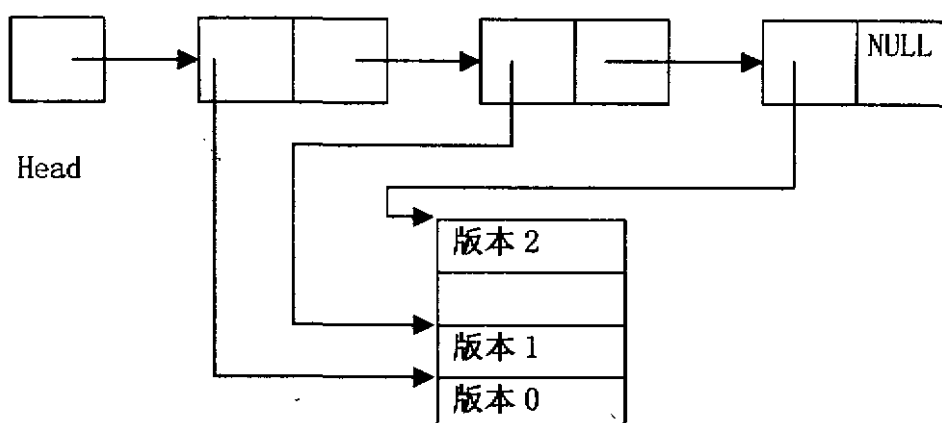


图 6-2 版本管理数据结构示意图

在项目开始初期，由于子系统中没有设计结果对象，因此版本结构数据是只有头结点的空对象链表。当有一个对象存入子系统库中时，就增加一个链表结点对这个对象的版本进行管理，即将对象的指针送入结点的数据域。随设计过程的推进，产生的不同设计结果都可以作为新的版本送入子系统库中。当要进行查询时，只需给出相应的版本号，就能通过链表找到相应的对象。对于不需要的版本，可以方便地从版本链表中删除，即删除该结点，同时释放对象的存储空间。

4. 文件的组织

在二级库中，不同的对象分别用不同的文件进行存储。通过子系统库的数据字典或设计结果库的版本链表查找对象的相应存储文件名，利用操作系统的功能完成文件的存取和管理。

6.5.2 ODBC 转换子系统的实现技术

子系统库中的数据来源于系统库，而系统库中的数据是用关系数据模型进行描述和管理。这就需要利用 ODBC 在两个基于不同数据模型的数据库之间进行数据转换。

1. ODBC 技术简介

ODBC 全称开放数据库连接(Open Data Base Connectivity)，是 Microsoft 公司研制的客户 / 服务器环境下的数据访问标准方法。它是在由 SQL Access Group 提出的调用级接口规范基础上建立的，是一种开放的，能从任何一种数据库访问

另一个支持 SQL 数据库的通用方法。

ODBC 具有许多优点。它是中性的，不依赖于具体的数据库，因此可以方便地实现异构数据库之间的互访。其次 ODBC 具有开放性，并得到大量的支持。大多数数据库都提供了 ODBC 驱动器，从而可以利用这个驱动器访问数据库中数据。大多数编程语言如 VC、VB 等开发工具都支持 ODBC 数据库访问操作，从而可以方便应用程序的开发。另外 ODBC 功能非常强大，了解它的原理后开发人员可以用很少的代码完成复杂的数据库访问，而开发人员不必了解被访问数据库的特性。因此可以从同一应用程序中访问不同的数据库，在试验系统中实现了将系统库中的关系型数据转化为子系统库中面向对象数据模型描述的数据。

2. 应用 ODBC 技术开发转换子系统

(1) ODBC 相关类的介绍

在 VC6.0 中提供了两个类用于 ODBC 联接：一个是 CDatabase，另一个是 CRecordset。CDatabase 类含有进行数据库连接的专用操作，能够连接一个数据库并控制数据的传输，执行不同的 SQL 命令。CDatabase 能够执行大多数 SQL 命令，返回一系列相关的记录，而 CRecordset 类的功能就是对这些记录进行管理。CRecordset 称为记录集，是在内存中形成的内部表，它包含有数据查询的结果。MFC 利用 CRecordset 类构成表并处理记录集，在每个记录集内部声明了多个字段。记录集对象的字段数据成员组成显示和编辑缓冲区，该缓冲区中保存已被选择的某些记录。

(2) 转换子系统的开发实例

为了使用 ODBC 技术，必须先从 CRecordset 类中派生出相应的关系数据表格记录集类。在 VC6.0 开发环境中，利用“Class Wizard”进行添加。点击“New”菜单中的“Class Wizard”，在模式对话框右边点击“Add Class”，然后再点击出现的“New”，此时弹出添加类的对话框。选择基类为 CRecordset，并输入类名确定，即弹出选择数据源的对话框。这时在选择框中选择“MS Access 97 Database”，其它以默认的方式即可。确认后即弹出选择数据库中表格的对话框。此时可根据需要选择相应的表格，如试验系统分别选择“Y 系列电动机参数”等建立了记录集类。在试验系统中电动机参数记录集类的定义如下：

```
class CDianJiSet : public CRecordset
```

```
{  
public:  
    DECLARE_DYNAMIC(CDianJiSet)  
    long m_ID;  
    CString m_column1;  
    double m_column2;  
    long m_column3;  
    double m_column4;  
    double m_column5;  
    long m_column6;  
    long m_column7;  
public:  
    CDianJiSet(CDatabase* pDatabase = NULL);  
    virtual CString GetDefaultConnect();  
    virtual CString GetDefaultSQL();  
    virtual void DoFieldExchange(CFieldExchange* pFX);  
    Virtual void AssertValid() const;  
    virtual void Dump(CDumpContext& dc) const;  
};
```

其中的成员变量描述电动机参数表中各个字段，CDianJiSet(CDatabase* pDatabase = NULL) 是构造函数，其缺省参数是 NULL。这个参数可以是一个 CDatabase 类对象的指针，从而将数据库对象与记录集对象进行关联。在有了这个记录集类的定义后，就可以直接利用这个类生成对象。但是生成的对象还不能直接使用，必须执行数据库对象和记录集对象的打开操作才能读取记录集中的数据。数据库对象的打开操作有两种方式 Open(LPCTSTR lpszDSN, BOOL bExclusive = FALSE, BOOL bReadOnly = FALSE, LPCTSTR lpszConnect = "ODBC;", BOOL bUseCursorLib = TRUE) 和 OpenEx(LPCTSTR lpszConnectString, DWORD dwOptions = 0)。后一种方式要求给出字符描述方式的同时还要求给出打开的方式，可以有只读等方式。由于是打开系统库，只需进行查询，因此选择只读打开方式。这

时的 CDatabase 对象就相当于本地的一个数据库。记录集对象的使用同样要求进行打开操作, 它也具有只读等多种打开方式, 同样只需使用只读方式即可。另外它的打开方式还规定快照和动态集两种使用方式: 快照指打开数据库, 直接将数据读入记录集, 数据库以后的变化不会波及到此记录数据, 而动态集方式表示以后数据库对象中数据的变化会波及到记录集数据。由于是打开系统库, 系统库中的标准数据一般不会改变, 同时为了减少对系统库的占有时间, 在试验系统中选择快照方式。这样就可以通过记录集对象读取关系数据库中的数据进行使用。例如在轴的设计过程中, 试验系统能够利用所定义的记录集成功地查询电动机的参数、材料特性和键槽的相关参数。

6.6 工作库的建立技术

工作库存储在设计过程中产生的各种动态数据, 它只有一个版本。为了对这些动态数据进行管理, 必须根据需要设计相应的类进行描述。在本节以支持传动轴设计数据的描述为例介绍类的设计, 对于其它零件的设计只需再定义相应的类即可, 它们的方法相同。

6.6.1 总体特征描述类的设计

由轴类零件信息模型知道, 轴类零件具有总体特征, 而总体特征又包括管理特征、材料特征等, 因此可以定义相应的类进行描述。首先设计管理特征、材料特征等对应的类, 然后在这些类的基础上设计总体特征类。限于篇幅在此只给出材料特征类和总体特征类的定义。

1. 材料特征类

```
class CCaiLiaoTZ : public CObject
{
public:
    CString YingYong;
    int ShuX1;
    int ShuX2;
```

```
int ShuX3;  
int ShuX4;  
int ShuX5;  
int ShuX6;  
int ShuX7;  
int ShuX8;  
int ShuX9;  
int ShuX10;  
int ShuX11;  
int ShuX12;  
int ShuX13;  
void Show();  
CCaiLiaoTZ();  
virtual ~CCaiLiaoTZ();  
};
```

该类是从 CObject 类派生出来的。其中 YingYong 表示材料的牌号，其它的 ShuXi (i=1 到 13) 表示材料的属性。Show() 函数是材料属性的显示操作，CCaiLiaoTZ() 是材料特征类的构造函数，~CCaiLiaoTZ() 是材料特征类的析构函数。

2. 总体特征类

```
class CZhongTiTZ : public CObject  
{  
public:  
    CGongNengTZ GongNengTZ;  
    CCaiLiaoTZ CaiLiaoTZ;  
    CFenXiTZ FenXiTZ;  
    CZhuangPeiTZ ZhuangPeiTZ;  
    C JiShuTZ JiShuTZ;
```

```
CGuanLiTZ GuanLiTZ;  
CZhongTiTZ();  
virtual ~CZhongTiTZ();  
};
```

其中从上到下分别表示功能特征、材料特征、分析特征、装配特征、技术特征和管理特征。由这个类就能描述设计对象的总体特征信息

6.6.2 形状特征类的设计

为了能进行串行化,所有的类都从 CObject 类进行派生。为了说明类之间的相互关系,从自定义的基础类开始进行说明,复杂类是在基本数据类型、自定义类和 MFC 类库的基础上进行定义的。

1. 线段类

为了描述几何要素中心线,定义 CLine 类,它包括两个属性:开始点和结束点。它的具体定义如下:

```
class CLine :public CObject  
{  
public:  
    CPoint EndPoint;  
    CPoint StartPoint;  
    CLine(int X1, int Y1, int X2, int Y2);  
    CLine(CPoint Point1, CPoint Point2);  
    CLine();  
    virtual ~CLine();  
};
```

2. 圆面类

为了描述圆柱的左端面 and 右端面,定义圆面类。它由相应的直径和精度特征进行描述。类的定义如下:

```
class CYuan : public CObject  
{
```

```
public:
    int Diameter;
    CCanShu ChueiZhiDu;
    CCanShu PingXingDu;
    float PingMianDu;
    float BiaoMianCCaodu;
    CPoint Point;
    CYuan(CPoint pnt, int d);
    CYuan();
    virtual ~CYuan();
};
```

其中 CcanShu 数据类型的 ChueiZhiDu 和 PingXingDu 分别描述精度特征平行度和垂直度, 包括相应的参考几何要素和精度值。浮点数类型的 PingMianDu 描述该圆面具有的平面度。整数类型的 Diameter 描述该圆面的直径。点类型 (Cpoint) 的数据 Point 描述圆心。

3. 柱面类

为了对几何要素圆柱面进行描述, 定义柱面类如下:

```
class CZhuMian : public CObject
{
public:
    float XiaPianChaL;
    float ShangPianChaL;
    float XiaPianChaD;
    float ShangPianChaD;
    int Length;
    int Diameter;
    CCanShu MianLKuoDu;
    CCanShu YuanDu;
    CCanShu ZhiXianDu;
```

```
CCanShu BiaoMianCCaoDu;  
CZhuMian(CYuan ZYuan, CYuan YYuan);  
CZhuMian(int Le, int dia);  
CZhuMian();  
virtual ~CZhuMian();  
};
```

其中整型数据类型的 Length 和 Diameter 分别描述柱面的长度和直径, 浮点数据类型的 XiaPianChaL、ShangPianChaL、XiaPianChaD 和 ShangPianChaD 分别长度的上偏差、下偏差和直径的上偏差、下偏差。CCanShu 数据类型的 MianLKuoDu、YuanDu、ZhiXianDu 和 BiaoMianCCaoDu 分别描述柱面的面轮廓度、圆度、直线度和表面粗糙度。

4. 圆柱类

圆柱类是构成传动轴主要特征——轴段对象的类, 是在上述类的基础上构成。它的定义如下:

```
class CYanZhu : public CObject  
{  
    BOOL BiaoShi;  
public:  
    CLine Line;  
    CCanShu GuoDuYuan;  
    CDaoJiao DaoJiao;  
    CString CiYaoTZ[10];  
    CZhuMian ZhuMian;  
    CYuan YouDuanMian;  
    CYuan ZuoDuanMian;  
    CYanZhu(CZhuMian ZhuM);  
    CYanZhu(CYuan ZDM, CYuan YDM);  
    CYanZhu();  
    virtual ~CYanZhu();
```

};

其中 ZhuMian 描述圆柱的几何要素柱面, ZuoDuanMian 和 YouDuanMian 分别描述圆柱的左端面和右端面, Line 描述圆柱的中心线。字符串数组 CiYaoTZ[10] 描述依附在该圆柱上的次要特征标识, 通常一个圆柱上的次要特征数不会超过 10 个, 故定义为 10 维的数组, 特殊情况可以增加数组的维数。GuoDuYuan 和 DaoJiao 分别描述圆柱上的辅助特征: 过渡圆和倒角。私有逻辑型变量 BiaoShi 用于描述圆柱的状态, 是为了对圆柱对象进行操作而定义的。当圆柱对象的 BiaoShi 为真时表示圆柱对象有效, 否则表示该圆柱对象处于删除状态。

5. 轴类

轴类是生成设计对象的类, 它不仅具有相应的属性变量, 而且具有较多的操作, 它能完全支持轴的设计过程。在试验系统中, 为了说明类设计的方法, 定义了轴的一些主要属性变量和方法, 有部分属性变量和方法未进行定义和说明。轴类的定义如下:

```
class CAxis : public CObject
{
private:
    int Now;
public:
    CYanZhu ZhuYaoTZ[6];
    CZhongTiTZ ZhongTiTZ;
    void DesignXingZh();
    void GetZuiXZJ();
    void GetXiaoLu();
    void GetJianShuB();
    void SelectDianJi();
    void GetCaiLiaoTZ();
    CAxis();
    virtual ~CAxis();
};
```

其中的总体特征类型(CzhongTiTZ)的变量 ZhongTiTZ 描述该轴类对象的总体特征,圆柱类型(CyanZhu)的数组变量 ZhuYaoTZ[6]描述该类对象的主要特征,即轴段对象。整型变量 Now 是辅助轴类对象设计的属性变量。在轴类对象中定义了许多操作,如 DesignXingZh()、GetZuiXZJ()、GetXiaoLu()、GetJianShuB()等用于完成轴类零件的设计。

6.6.3 工作库的建立

有了这些类的定义,就可以利用这些类描述具体的设计对象(传动轴)。根据设计过程的需要,对相应的类进行实例化,生成对象。然后通过具体的设计对各个对象以及属性进行赋值,并存入工作库中。在设计过程中可以随时访问对象,同时对对象的属性可以进行修改、删除等操作。从而实现工程数据的动态管理。

6.7 小结

本章研究了用户接口子系统、系统库、子系统库、设计结果库以及工作库的建立技术、ODBC 的使用技术和类库的设计技术。通过这些技术的介绍能够建立基于面向对象数据模型的工程数据库管理系统的体系结构,能够了解应用面向对象数据模型建立工程数据库的过程和方法。同时说明面向对象数据模型能够有效地处理机械 CAD 中的工程数据,并具有较强的兼容性,便于大量推广使用。

第 7 章 EDBMS 试验系统及运行实例

为了说明面向对象数据模型的正确性和工程数据库管理系统体系结构的有效性以及实现技术的可行性,本章将简要介绍 EDBMS 试验系统的功能及组成部分,而后利用所开发的试验系统说明减速箱传动轴设计过程中大部分工程数据的处理过程。

7.1 EDBMS 试验系统简介

该试验系统采用微软的 VC 6.0 作为开发工具,内嵌 VC6.0 编译器和 Microsoft Access 2000,在 Windows 98 平台上进行开发,其中的系统库部分利用关系数据库管理系统 Microsoft Access 2000 进行设计。

该试验系统由用户界面、类管理器、系统库、ODBC 转换子系统、二级库、工作库、对象管理子系统、事务管理子系统八个功能模块组成。用户界面能够接受用户命令的输入以及工程数据库和设计过程中工程数据的显示。类管理器根据设计对象描述的需要通过继承或添加、删除属性变量产生新类。系统库模块通过启动关系数据库管理系统 Microsoft Access 2000,能够完成系统库的建立、查询、添加和删除等操作。ODBC 转换子系统完成从系统库中取出关系型标准数据存入面向对象的子系统库中。二级库模块能将从 ODBC 转换子系统得到的数据利用面向对象数据模型重新描述存入子系统库,并进行管理,同时它能完成设计对象结果数据的管理。工作库能够完成设计过程中动态数据的存储和管理。对象管理子系统通过总体特征和形状特征子模块完成对象的生成及管理。事务管理子系统能够完成用户管理和长事务管理的功能。

7.2 EDBMS 试验系统的运行实例

7.2.1 验证实例分析

为了说明面向对象数据模型描述工程数据的有效性、EDBMS 及其实现技术的可行性,必须应用实例进行验证。在机械 CAD 系统中,设计对象复杂多样,使试验系统的建立比较困难。为了解决这个问题,试验系统选择减速箱主传动轴设计过程中工程数据的管理为例进行验证。验证过程可以按两种方式进行:按设计过程或按系统结构,本节是将二者组合进行验证。众所周知,减速箱主传动轴设计时,首先是根据需求选择电动机型号,然后根据电动机的参数计算主传动轴的功能特征参数以及最小轴径,接着由最小轴径设计传动轴的结构,并根据传动轴的结构进行校核,最后绘制零件图等。在这个过程中涉及到的工程数据包括管理数据(各种标准数据)、设计数据、图形数据以及软件包数据。能够对传动轴设计过程中的工程数据进行管理,照样也能对其它零件设计过程中的工程数据进行管理,从而验证 OODM 的有效性、EDBMS 体系结构以及实现技术的可行性。

7.2.2 系统的启动

在 Windows 下双击 EDBMS 图标,启动试验系统,得到如图 7-1 所示的主界面。此时用户可以点击文件菜单中的“新设计”,选择设计的项目,在这里可以选择“传动轴”,确认后就可以开始传动轴设计过程。这时在标题栏显示:传动轴设计——工程数据库管理系统。

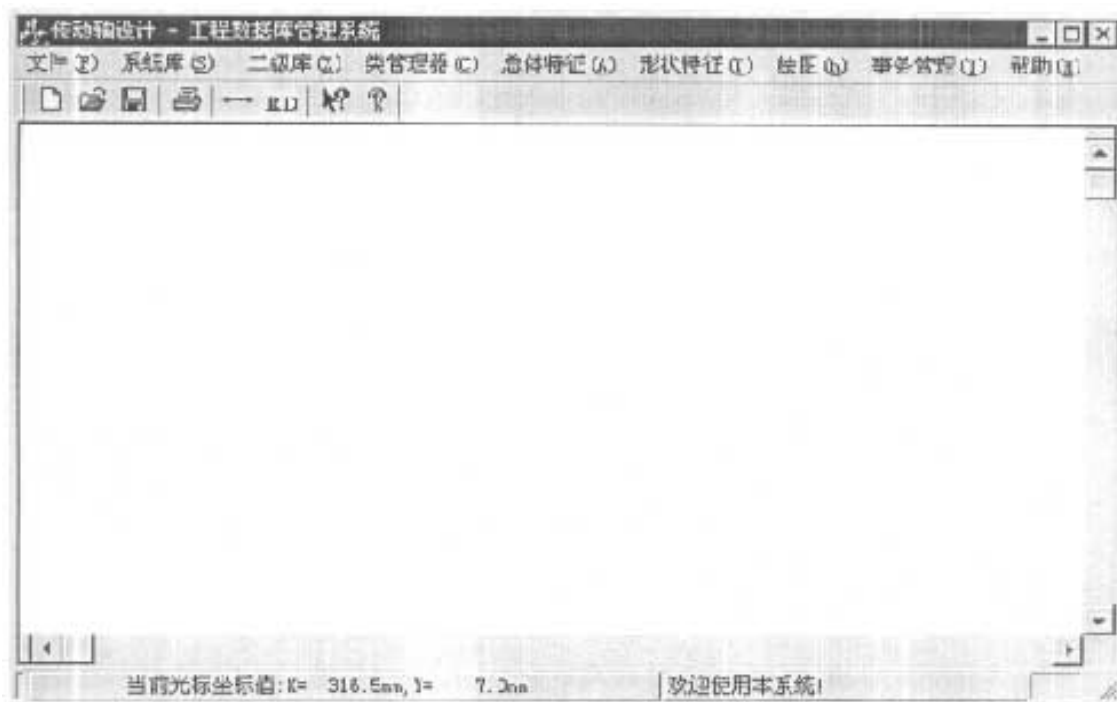


图 7-1 工程数据库管理系统主界面

7.2.3 系统库的建立和管理

一个设计过程必须要有系统库的支持，因此首先验证系统库的建立和维护。如果要新建系统库则点击“系统库”菜单中的“新建”子菜单，启动 Microsoft Access 2000 出现如图 7-2 所示的界面。这时可以开始一个新系统库的建立过程，其方法参阅有关 Access 2000 的使用说明。如果要添加、删除和修改系统库，可以用相似的方法启动 Access 2000 进行相应的操作。这样可以完成系统库的建立及维护等工作。

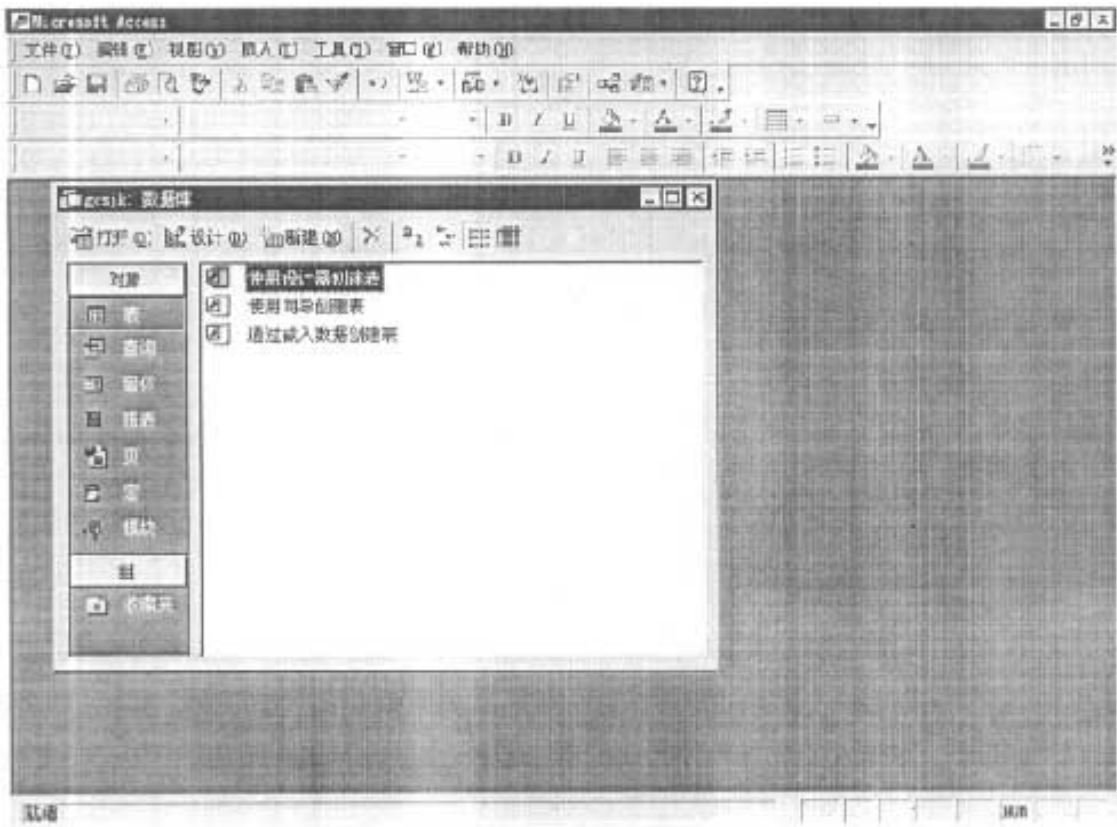


图 7-2 系统库的建立界面

7.2.4 子系统库的建立和维护

通过子系统库的建立和维护可以验证 ODBC 转换子系统的有效性以及子系统库设计的可行性。在主界面的“二级库”菜单中点击“子系统库”的“新建”菜单，则可以初始化一个子系统库，建立相应的数据结构，并可选择一些标准数据通过内置的 ODBC 转换子系统，将这些数据装入子系统库。在建立子系统库后，可以通过查询菜单选择该库中的标准数据进行查询。图 7-3 表示关系型的系统库中电动机参数。图 7-4 表示利用 ODBC 转换子系统将系统库中的数据装入子系统库后的结果。通过该运行过程说明 ODBC 转换子系统切实可行、有效；子系统库能够正常工作，完成相应的功能。

Y系列电动机参数: 表							
额定功率	ID	同步转	额定量	额定量	满载转速	电动机型	极数
.75	1	3000	2.2	2.2	2825	Y801-2	2
1.1	2	3000	2.2	2.2	2825	Y802-2	2
1.5	3	3000	2.2	2.2	2840	Y90S-2	2
2.2	4	3000	2.2	2.2	2840	Y90L-2	2
3	5	3000	2.2	2.2	2880	Y100L-2	2
4	6	3000	2.2	2.2	2890	Y112M-2	2
5.5	7	3000	2.2	2.0	2900	Y132S1-2	2
7.5	8	3000	2.2	2.0	2900	Y132S2-2	2
11	9	3000	2.2	2.0	2930	Y160M1-2	2
15	10	3000	2.2	2.0	2930	Y160M2-2	2
18.5	11	3000	2.2	2.0	2930	Y160L-2	2
22	12	3000	2.2	2.0	2940	Y180M-2	2
4	13	1000	2	2.0	980	Y132M1-6	6

图 7-3 系统库中的电动机参数

无标题 - 工程数据库管理系统					
文件(F) 系统库(S) 二库库(L) 类管理器(C) 总体特征(A) 形状特征(E) 绘图(D) 事务管理(M) 帮助(H)					
□ □ □ □ □ RD ? ?					
序号	电动机型号	额定功率 (kW)	满载转速 (r/min)	额定满载转速	额定量
第 1 种型号	Y801-2	0.8	2825	2.2	2
第 2 种型号	Y802-2	1.1	2825	2.2	2
第 3 种型号	Y90S-2	1.5	2840	2.2	2
第 4 种型号	Y90L-2	2.2	2840	2.2	2
第 5 种型号	Y100L-2	3.0	2880	2.2	2
第 6 种型号	Y112M-2	4.0	2890	2.2	2
第 7 种型号	Y132S1-2	5.5	2900	2.0	2
第 8 种型号	Y132S2-2	7.5	2900	2.0	2
第 9 种型号	Y160M1-2	11.0	2930	2.0	2
第 10 种型号	Y160M2-2	15.0	2930	2.0	2
1					
4					
当前光标坐标值: X= 200.7mm, Y= 4.0mm 欢迎使用本系统!					

图 7-4 子系统库中电动机参数对象

7.2.5 对象管理子系统以及工作库、设计结果库的运行分析

通过在主界面中点击“类管理器”，可以启动 VC6.0 编译器。根据设计过程中工程数据描述的需要，利用 MFC 类库以及系统提供的类等可以定义各种类。如在这里定义了线段类、圆类、圆柱类等描述形状特征的类和管理特征类、材料特征类等描述总体特征的类。有了这些类就可以用于描述设计过程中的工程数据。如图 7-5 所示的电动机参数，可以用功能特征类的相应参数进行描述，图 7-6 中的材料参数可以用材料特征类进行描述。图 7-7 中的标题栏信息属于管理特征可以用管理特征类进行描述。其它总体特征类的设计不再累述。形状特征和精度特征也可用相应的类进行描述，如图 7-8、7-9、7-10 中所示的主要特征（轴段 1）、辅助特征以及精度特征可以用相应的主要特征类（圆柱类）、辅助特征类（倒角类和过渡圆类）以及不同的精度特征类进行描述，并将描述的各个对象存入工作库中。次要特征的设计和描述与主要特征非常相似，限于篇幅在这里不进行说明。在需要查询和修改时，可以用同样的方法点击相应的菜单，显示各个对象的值或标识，从而可以在各个编辑框中进行修改，包括删除和添加，最后点击确认即可使对对象的改动有效，即动态改变对象的值。对设计对象型的改变，通过启动 VC6.0 增加相应的类或增加、删除原有类属性完成。在设计过程中可以直接使用新得到的类描述工程数据，从而完成工程数据型的动态改变。设计完成后，可以将设计结果存入设计结果库中，并进行版本管理。通过对上述过程中工程的管理，说明该试验系统能够实现数据模式的动态修改、对象的描述和管理，同时也说明工作库和设计结果库的设计是有效的，它能够完成工程数据的管理。

电动机型号选择

电动机有关参数

电机型号	Y160L-6	堵转额定转矩	2	N.m	
额定功率	11	KW	最大额定转矩	2	N.m
满载转速	970	r/min	极数	6	
同步转速	1000	r/min			

选择电动机型号

Y160L-6

查 询

确 认

选择电动机型号，然后单击查询，上面会显示该电机的各种参数，直到满意后确认

图 7-5 电动机参数

选择合适的材料

推荐热处理 (摄氏温度)			试样毛坯尺寸 [mm]	机械性能				
正火	淬火	活火		抗拉强度	屈服强度	延伸率	收缩率	冲击功
050	040	600	25	600	355	16	40	39

材料交货状态硬度HDI不大于

未热处理

退火处理

强度系数

材料牌号

229

197

112

45

应用举例

用于制造齿轮、蜗轮、链轮、轴、键、销、蒸汽透平机的叶轮、压缩机及泵的零件、轧辊等。可代替渗碳钢做齿轮、轴、活塞销等，但要经高频或火焰表面淬火

选择材料牌号

45

查 询

确 定

图 7-6 材料特性参数

标题栏信息

输入标题栏的有关信息

	日期	零件名称	材料	图号	
设计	zzy	5.1	传动轴	45	01
校核	czy	5.2	比例	数量	设计单位
审核	ggan	5.4	1:1	1	西南交通大学

确 认 取 消

图 7-7 管理特征中标题栏属性

主要特征

左端面 | 右端面 | 圆柱面 | 次要特征 | 辅助特征 |

☐ 第一系列 相应值

☒ 第二系列 表面粗糙度

平面度 0 参考要素

平行度 0 参考要素

垂直度 0 参考要素

选择对象 输入直径 45 上偏差 -0.0600 下偏差 -0.0400

轴段1 输入长度 65 0 0

查 询 删 除 添 加 结 束

图 7-8 主要特征及几何要素（左端面）设计中的工程数据

主要特征

左端面 | 右端面 | 圆柱面 | 次要特征 | 辅助特征 |

输入次要特征

添加

删除

选择对象

轴段1

输入直径

45

输入长度

65

上偏差

+0.0300

下偏差

-0.0400

查询

删除

添加

结束

图 7-9 轴段 1 上添加和删除次要特征

主要特征

左端面 | 右端面 | 圆柱面 | 次要特征 | 辅助特征 |

倒角

位置选择

左端

角度

45

宽度

2

过渡圆

位置选择

右端

半径

15

选择对象

轴段1

输入直径

45

输入长度

65

上偏差

+0.0600

下偏差

-0.0400

查询

删除

添加

结束

图 7-10 轴段 1 上辅助特征的设计数据

7.2.6 草图的绘制

为了说明图形数据的生成,以主要特征的简单绘制为例进行说明。试验系统根据不同的目的提供了两种方法绘制草图:一种是通过屏幕点击手工绘制草图,另一种是通过形状特征的描述直接在屏幕上显示草图。前一种方法用于在确定最小轴径后试绘零件结构图。后一种方法直接根据对象的参数显示零件草图,用于让设计者检查设计结果的正确性。通过输入第三章传动轴的属性参数,可以得到如图 7-11 所示的界面。通过这种方式说明该 EDBMS 支持设计过程以及图形数据的管理。

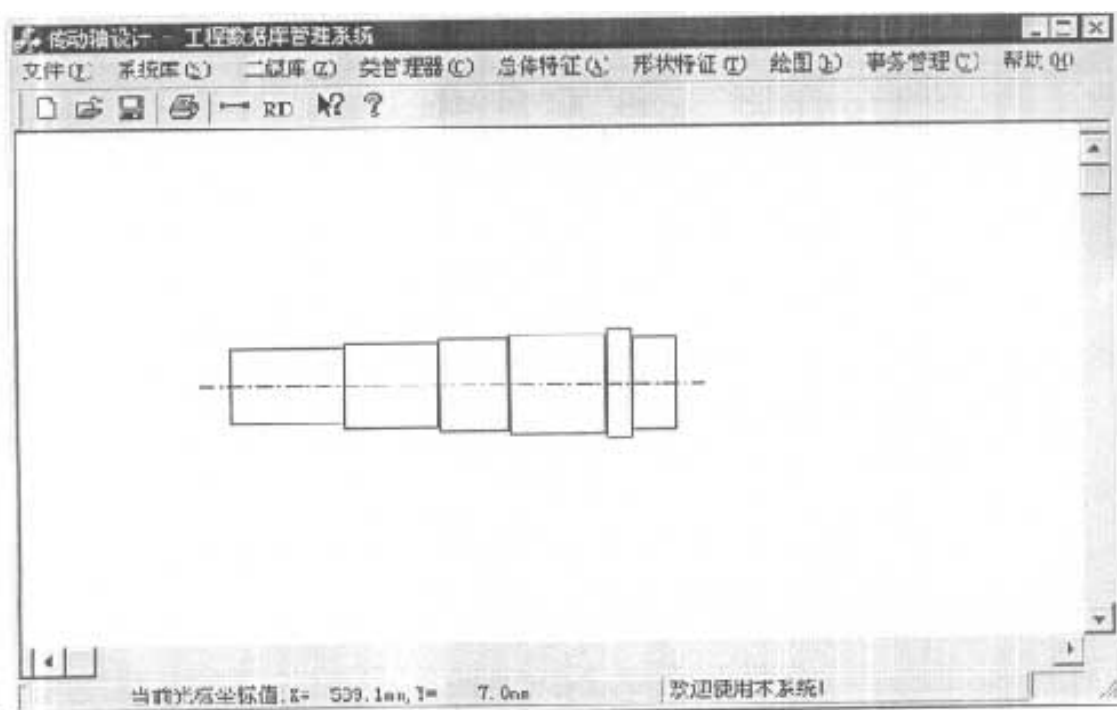


图 7-11 零件草图的绘制

7.2.5 其余功能模块的说明

由于工程数据库管理系统的研究和开发是一项工作量大的课题,限于时间的仓促,其余的功能模块未完全编码实现,但是用相似的方法可以完成其它功能模块的实现。

7.3 小结

通过试验系统的运行，并对传动轴设计过程中各种工程数据的管理，验证了 OODM 数据模型的正确性，能够描述设计过程中的所有工程数据，同时也说明了 EDBMS 体系结构和实现技术的可行性。该试验系统能够建立系统库、子系统库、设计结果库以及工作库，能够实现不同数据模型描述数据的转换。同时利用类管理器能够动态的产生新的数据类型，实现数据模式的动态定义和修改。

结 论

本文为了建立工程数据库管理系统,研究了机械 CAD 系统中的工程数据类型,并根据零件特征的定义研究零件以及装配体信息模型,完成工程数据库管理系统建立的基础研究。在此基础上研究面向对象数据模型及对象的构造方式和操作,然后过渡到工程数据库管理系统的体系结构的设计,最后研究工程数据库管理系统的实现技术。通过这些研究,取得如下几个方面的进展。

1. 对工程数据进行了深入分析,掌握了其本质特征

本文将工程数据分为四类:管理数据、图形数据、设计数据和软件包数据,并对各类数据的本质特征进行了深入的研究。在此基础上归纳了工程数据的总体特征以及与商用数据的区别和联系。

2. 以零件特征为基础建立了零件信息模型以及装配体信息模型

根据信息模型建立的需要,本文给出了零件特征的定义。根据不同类型零件的特征设计了零件信息模型以及装配体信息模型,并对具体零件进行了描述。证明设计的信息模型能够面向用户描述所有的工程数据,可以辅助数据模型的设计。

3. 研究了 OODM 的一系列概念以及对象的构成方式和操作

探讨了面向对象数据模型的基本概念,并给出了相应的定义和说明。研究了对象的构造方式,在所有的构造方式中以关系构造为主。对象具有外部操作和内部操作,分别完成对象的不同管理。面向对象数据模型(OODM)能够描述工程领域中的所有工程数据,是适合 CAD 系统中工程数据管理的有效模型,为工程数据库管理系统的建立打下了良好的基础。

4. 研究了工程数据库管理系统的体系结构

设计了工程数据库的体系结构:系统库、子系统库、设计结果库和工作库,分别具有不同的功能。设计了工程数据库管理系统的总体结构,它由用户接口子系统、对象管理子系统、ODBC 转换子系统等组成。

5. 实现了 EDMBMS 的几个关键技术

研究了用户界面的设计方法、系统库、子系统库和设计结果库及工作库的建

立技术、ODBC 技术等，并用 VC6.0 编码实现了试验系统。

通过对传动轴设计过程中工程数据的管理，验证了 OODM 的有效性和工程数据库管理系统体系结构及实现技术的可行性。

工程数据库管理系统是一个复杂的大型软件系统，涉及的面广、领域问题多，并且国内外参考的资料非常少。工程数据库管理系统的建立是一个逐步完善的发展过程，不可能一蹴而就，更不可能一步到位，需要大量的科研工作者的共同努力，逐步完善。限于作者的知识面、精力和时间，未能对一些问题作深入细致的研究，而仅仅是作了一些探索，希望能对所有工程数据库系统的研究者起到抛砖引玉的作用，同时也希望各位专家能够对本论文的不足之处进行批评和指正，使作者在以后的研究中能够加以改进。

致 谢

感谢导师高国安教授这几年来对我学习的指导和生活帮助,特别是论文工作的完成,花费了导师不少的心血。在这几年的学习中,导师广博的知识、严谨治学的态度、宽以待人的生活作风给我留下了深刻的印象,指引我人生的道路,使我终生受益。

感谢本方向的师兄弟妹这几年来对我的帮助和支持,在此致以诚挚的谢意和祝福:愿他们好人一生平安,事业有成。

感谢父母及所有长辈对我的培养和教育,愿他们身体健康、事事顺利。

特别感谢汪志华女士对我学习和论文工作的帮助和支持。在论文期间,她为文稿的录入花费了大量的心血,并提出了一定的改进意见,在此表示诚挚的谢意。

在此一并感谢在研究生期间给我帮助和支持的所有老师、同学和亲戚朋友。

最后感谢西南交通大学对我的培养和教育。

左 朝 树

2001 年 5 月

参考文献

- [01] 卢跃平. 面向对象工程数据库开发工具的研究与实践. 硕士论文. 成都: 西南交通大学, 1993
 - [02] 张爱军. 机械 CAD 系统设计中的数据支持技术研究. 硕士论文. 成都: 西南交通大学, 1995
 - [03] 邵宇开. CAD/CAM 集成技术的研究与实现. 硕士论文. 成都: 西南交通大学, 1998
 - [04] 周宇戈. 蜗轮滚刀参数化 CAD 系统研究. 硕士论文. 成都: 西南交通大学, 2000
 - [05] 邓建华. 面向对象的智能决策支持系统理论与实践. 硕士论文. 成都: 西南交通大学, 1997
 - [06] 王先逵, 李志忠, 刘成颖, 蒲建, 黄新刚. 用信息元构造 CAPP 系统框架. 机械工程学报, 1999; 35 (3)
 - [07] 张和明, 熊光楞, 张玉云, 孙喜杰. 基于 STEP 的并行产品开发信息集成及设计版本管理技术. 机械工程学报, 1999; 35 (2)
 - [08] 张凤荔, 葛晓峰, 卢显良. 基于数据仓库的综合查询系统的设计与实现. 电子科技大学学报, 1999; 28 (2)
 - [09] 邵家玉, 陆平, 周伯鑫, 徐南荣. 一个演绎工程数据库语言 DEL. 计算机研究与发展, 2000; 37 (2)
 - [10] 邓铁清, 吴永源. 一个基于逻辑程序设计范例的专家数据库开发工具. 计算机研究与发展, 1999; 36 (4)
 - [11] 刘云生, 李国微. 实时数据库数据特征对事务处理的影响. 计算机研究与发展, 1999; 36 (3)
 - [12] 李垒, 李昌贵. PROLOG-DBMS 系统的数据库维护. 计算机研究与发展, 1999; 36 (3)
 - [13] 马万太, 王宁生. 工程设计数据库管理中版本模型的研究. 计算机研究与发展, 1999; 36 (3)
 - [14] 沈建强, 夏耘. 工程数据管理系统中的工程图挡检索. 计算机应用, 2000; 20 (3)
 - [15] 林毅, 沈军营, 严隽琪. 几何对象与特征对象的关联方法. 计算机应用, 2000; 20 (3)
 - [16] 李伯虎. 计算机集成制造系统 (CIMS) 约定、标准与实施指南. 兵器工业
-

出版社, 1994

- [17] 赵致格, 殷人昆. 实用工程数据库技术. 机械工业出版社, 1999
- [18] 宛延闾. 工程数据库系统. 清华大学出版社, 1999
- [19] David M. Kroenke, 施伯乐. 数据库处理. 电子工业出版社, 1998
- [20] 杨德元. 分布数据库管理系统概论. 清华大学出版社, 1987
- [21] 董逸生. CIMS 中的数据库技术. 机械工业出版社, 1997
- [22] 张建沛, 戴宗荫, 李玉忱, 杨静. 数据库系统教程. 中国铁道出版社, 1995
- [23] 蔡颖, 薛庆, 徐弘山. CAD/CAM 原理与应用. 机械工业出版社, 1998
- [24] 唐荣锡. CAD/CAM 技术. 北京航空航天大学出版社, 1999
- [25] 《CAD/通用技术规范》编写组. CAD 通用技术规范. 中国标准出版社, 1995
- [26] 郑怀远. 工程数据库技术. 机械工业出版社, 1995
- [27] 陈余年, 方美琪. 信息系统工程中的面向对象方法. 北京: 清华大学出版社, 1999
- [28] 何新贵, 唐常杰, 李霖, 刘云生. 特种数据库技术. 科学技术出版社, 2000
- [29] 刘云生, 卢正鼎, 卢炎生. 数据库系统概论. 华中理工大学出版社, 1992
- [30] 何耀钦, 薛华成. 数据组织与管理. 机械工业出版社, 1992
- [31] 周胜奎, 严嵩芬, 张紫夏. 网状数据库设计与维护. 地震出版社, 1992
- [32] 李也白, 范春晓. 面向对象数据库. 高等教育出版社, 1998
- [33] 裘春航, 程耿东. 工程数据库. 人民交通出版社, 1990
- [34] 陈学俭, 陈洪亮. 数据库原理与工程应用. 中国科学技术大学出版社, 1996
- [35] 陈建新, 叶飞跃, 马玉书, 全兆岐. 数据库技术. 石油工业出版社, 1995
- [36] PETER, EDWARD YOURDON, 邵维忠, 廖钢城, 李力. 面向对象的分析, 1992
- [37] 陈其明. 工程数据库原理. 测绘出版社, 1991
- [38] Michael Stonebraker, Dorothy Moore, 杨冬青, 唐世渭, 裴芳. 对象—关系数据库管理系统——下一个浪潮. 北京大学出版社, 1997
- [39] 施伯乐, 丁宝康, 楼荣生. 数据库系统导论. 高等教育出版社, 1994
- [40] 李幼文. 数据库系统原理和技术. 科学技术出版社, 1995
- [41] 王能斌. 数据库系统. 电子工业出版社, 1995
- [41] 程宝义. 计算机辅助设计基础. 国防科技大学出版社, 1999
- [42] 邱宣怀等. 机械设计. 高等教育出版社, 1997
- [43] 吴宗泽, 罗圣国. 机械设计课程设计手册. 高等教育出版社, 1992
- [44] 康师友. C++面向对象程序设计简明教程. 西安电子科技大学出版社, 1998
- [45] David J. Kruglinski, 北京希望电子公司译. Visual C++ 6.0 技术内幕. 北京希望电子出版社, 1998
- [46] Keith Bugg, 常小刚. Visual C++ 除错手册. 中国电力出版社, 2000

- [47] Chuck Wood, 梁普选, 梁津, 刘玉芬. Visual C++ 6.0 数据库编程大全. 电子工业出版社, 2000
- [48] Bertino E.. Index Configuration in Object-Oriented Database. The VLDB Journal Vol.3 No.3, July 1994
- [49] Distributed Component Object Model Protocol_DCOM/1.0. Microsoft Corporation, Nov. 1996
- [50] Kemper A., Moerkotte G.. Object-Oriented Database Management. Prentice Hall International, Inc. 1994
- [51] Cattel R.G.G.ed.. The Object Database Standard:ODMG-93. Morgan Kaufmann Publishers, 1994
- [52] T.H.Cormen, C.E.Leiserson and R.L.Rivest, Introduction to Algorithms, the MIT Press, 12th printing, New York, McGraw_hill, 1994
- [53] Gero. J.S., Roseman MA. A conceptual framework for knowledge-based design research at Sydney University's Design Computing Unit in J.s. Gero., Artificial Intelligence in Design, CMP/Springer Verlag, Southampton and Berlin, 1989
- [54] Lenart M., Padaeitz P., Pastor A. Formal specification for design automation. In Gero. J. S., Tyugu E: Formal design methods for CAD, Volume B-180^r IFIP Transactions. North-Holland: Elsevier Science B. V., 1994
- [55] HE B., Georgew C. Formally describing intelligent CAD. UNU/IISP Report No. 159, Macau: UNU/IISP, 1999
- [55] Bancilhon, F., S. Cluet, C. Delobel. "A Query Language for the O2 Object-Oriented Database System," Altair Technical Report: 35-89, August 1989
- [56] G. Wiederhold, Knowledge and Database Management, IEEE Software 1984: 1(1)
- [57] F. Bancilhon, Object-Oriented database Systems, In Proc. 7th ACM -SIGMOD Symp., on Principles of Database Systems, Austin, Tex., March 1988
- [58] J. D. Ullman, Principles of database and Knowledge-base Systems, Vol. I, II, Computer Science Press, 1988
-