

微机 MIS 集成构造系统的研究与实践

华资总公司 周堤基 何久田 李德彩

摘要: 微机 MIS 集成构造系统 (Microcomputer MIS Integrated Construction System - ICS) 是基于软件重用理论, 采用应用描述语言、扩展数据字典和标准预制构件快速构造实用 MIS 的一种开发工具, 本文主要介绍它的设计思想、基本结构、构造原理、设计原则和实际使用情况。

一、问题的提出

近几年来, 随着我国现代化建设的发展和微机技术的广泛应用, 各部门对以数据处理为主的 MIS 的需求日益增加, 而应用系统的开发主要靠为数不多的计算机专业人员以手工方式来做, 其生产效率很低, 难以满足实际工作的需要。随着应用系统的不断投入使用, 其维护工作量急剧增加, 给软件开发人员带来了很大的后顾之忧, 有人这样形容: 开发一个应用系统等于给自己背上了一个包袱, 有些系统由于缺少必要的维护而不得不中止使用, 造成不必要的浪费; 一些应急任务的执行, 一般都具有很强的时间性和指令性。为了完成这些任务, 不得不临时抽调很多人加班加点突击, 但在时间和质量上仍难以使用户部门满意。

为了解决这些软件危机, 多年来国内外的计算机专家们正在寻求新的途径。从当前国内外研究现状看有两种发展趋势: 软件生产自动化 (如程序自动生成技术) 和软件重用技术。程序自动生成技术是基于一种事先规定的规范说明和模式, 通过一系列的对话完成定义操作。然后生成出所需的应用程序; 软件重用技术是依靠标准化的软构件组装应用系统, 这是计算机软件人员多年梦寐以求的事, 近几年来, 国内外的专家们进行了大量的研究, 并在工程计算领域取得一些可喜的成果, 但在 MIS 领域, 尚未见到具有实用价值的系列产品推出。是否有可能建立一个具有重用性的软构件库, 然后根据需要部分或全部地构造应用系统呢?

根据我们近十年开发的各种 MIS 实践, 一般的 MIS 均包括系统控制、数据录入与维护、数据查询、数据统计汇总、数据分析、数据输出 (包括显示和打印)、系统

管理等几大部分, 如果抽出具体的数据, 其基本功能是共同的。因此研究一个基于软构件的微机 MIS 集成构造系统是有可能的, 而且也是非常必要的。

二、设计思想和理论基础

1. 设计思想

用标准构件进行组装生产是近代大工业中广泛采用的一种模式, 在建筑、机械、电子以及计算机硬件等领域, 以这种方法进行设计生产, 极大地提高了生产效率和产品质量。将这种模式应用于 MIS 设计就是本集成构造系统的设计思想。其关键问题是如何的提取 MIS 处理过程为可操作的通用构件。图 1 是这种思想的示意图:

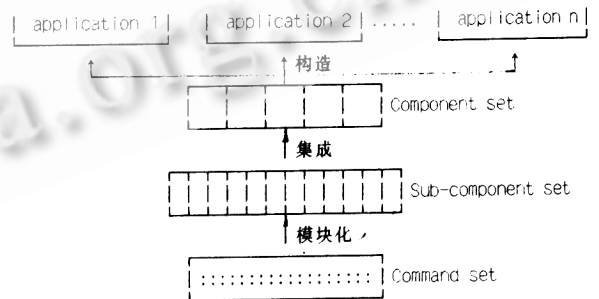


图 1 应用系统集成构造示意图

从图中可以看出: 首先将程序设计语言中的赋值语句、条件语句、函数调用等按某种逻辑模块化为一个个积木块 (building block), 也可称子构件; 然后将这些积木块按照某种功能的要求集成为一个个功能完整的构件, 有人则称之为“软插件” (Software Connector); 最后可以根据用户的实际需要, 选择一组构件, 并做适当的剪裁, 再构

成一个个应用系统的原型,这些原形可以帮助进一步明确用户的需求,经过适当修改即可形成实用 MIS。

2.理论基础

集成构造系统的理论基础是结构化程序设计中的模块化理论和近几年发展起来的软件重用理论。

模块化是结构化程序设计的核心,其基本思想是把程序划分成若干个模块,每个模块是相对独立,完成单一功能的一段程序,例如,过程、函数、子程序、宏等。采用模块化原理可以提高程序的可读性、可靠性,有助于组织软件的开发。另一方面标准化的模块可以实现模块级的重用,提高软件的开发效率。

软件重用理论是目前国内外计算机界研究的热门课题,其研究的重点是:软构件的定义、属性说明、构件库的建立以及描述语言等。软件重用分为:不作任何修改地用于另一个新系统的上下文中(黑箱应用);作适当修改才能用于另一个新系统的上下文中(白箱应用)。所有可重用的软件组成一个开放型的软件库,该库可易于修改和扩充。这种理论是面向对象的程序设计语言的一个重要特征。可提高软件的开发效率,提高软件的可靠性和可维护性。专家们认为这是一种提高软件生产力的有效手段,并预言这一理论在九十年代将有突破性的进展。

三、集成构造系统的构成

ICS以软件重用技术为基础,采用应用描述语言、扩展数据字典和标准预制软构件三位一体的设计方法,构成了一种积木式 MIS 集成开发环境。使用时只要定义出数据结构,用应用描述语言描述出系统的控制机制,并选择一组适当的预制软构件,即可形成一个高质量的实用 MIS。系统的基本结构如图 2 所示:

其中:

1.应用描述语言 (Application Description Language),作为一种规范描述语言,ADL 为应用系统模型的构造提供了形式化手段。它以功能路径为主线,对应用系统的结构体系(树状分层菜单)和控制机制(叶结点对构件的调用)进行一体化描述,能方便地生成应用控制部件。ADL 应用描述语言具有非过程化的特点,从软件工程角度看,其应用描述文本本身就是一份易于理解、便于交流的文档。

2.扩展数据字典(Extended Data Dictionary),将各类应用处理对象的锚述信息用结构化形式进行定义。ICS 从外部特征、使用特性和处理特性的角度将应用系统的基本成份分别归纳为若干类应用处理对象,打印报表、统计分析图表等,它们分别被不同的字典表所定义和描述。作为实现软件重用的基础,ICS 扩展数据字典将应用对象的结构信息和语义信息(关联关系,操作特性和完整性约束等)用数据化的形式进行定义描述,具有良好的可操作性,同时 ICS 各类字典根据应用对象间的相互联系保持内在的协调统一,具有集成性和完整性。下表是项目字典将存放的具体数据项(Fields)信息,它包括的基本内容有:

项目	类型	说明
表名代码	字符	字段中文名
项目名代码	字符	字段名
项目中文名	字符	字段中文名
类型	字符	(N、C、D)
长度	数字	
小数位	数字	
约束条件	表达式	取值范围
参照项	枚举	辅助用户录入数据
关联项	字符	与此项有关联的项目
:	:	:

上述字典包含的是一些基本内容,对于不同功能的构件可视具体需要进行必要的选择和扩充。

3.构件库 (Software Component Library),---对应用对象进行各种加工处理的通用程序块的集合。本集成环境提供了一组功能齐全的软构件,它包括:开发级软构件和应用级软构件两大类。用于开发阶段的有:

- 字典定义 : 数据库定义、创建、录入画面生成等。
- 菜单定义 : 应用主控生成(下拉式、对角式)、应用系统组装的有:
- 数据录入构件: 键盘录入、多表组合查询等。
- 数据统计构件: 统计汇总。
- 报表打印构件: 二维表和复杂表打印。统计图形处理构件: 直方图、圆饼图、折线图、曲线图。
- 文字处理构件: 文本编辑、查询、文献库管理。

此外 ICS 提供了一套使用说明,其中包括每个构件的基本功能、使用方法、约束条件及支持环境等。

基本功能:描述构件能完成的基本功能及特点。

定义方法:构件所需的数据字典或其它信息的定义方法、步骤及其注意事项。

使用格式:构造系统时的调用命令格式以及所需参

数的说明。

约束条件:使用构件的制约因素。

支持环境:描述构件所依赖的设备和软件条件。

除上述基本内容外还需提供必要的检索信息,以便构件存入构件库后方便查找。如构件分类码、特征等。

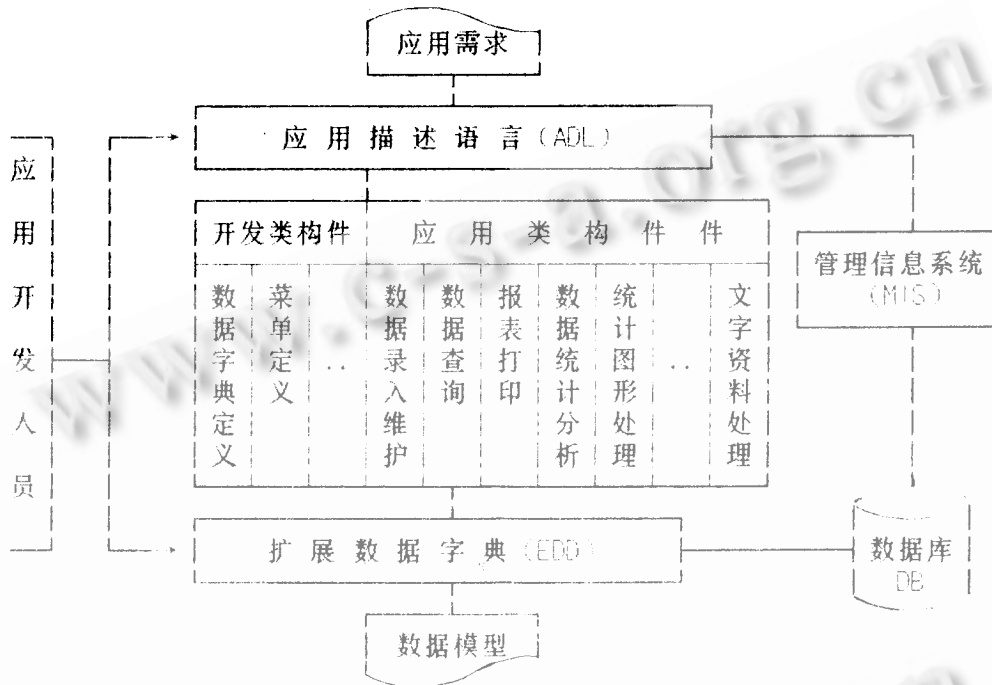


图 2

四、集成构造系统的构造原理

ICS 是生命周期法和原型法相结合的产物,它可以辅助程序员完成详细设计阶段以后的各项工作。其设计思想是以工程化的扩展数据字典为核心,以各种通用性强的标准预制构件为依托,用它开发应用系统,需要在系统分析阶段理清系统中的数据元及其之间的关系,在系统设计阶段完成数据库设计和功能层次结构的设计,经过测试调整即可组装出高质量的实用 MIS。下页图 3 是用 ICS 建立 MIS 的工作过程:

五、软构件的设计方法与设计原则

1. 构件的设计方法

构件设计应当由有经验的设计人员来完成。首先制订设计规范,确定构件的功能及其接口;然后进行构件的

设计,构件设计包括方案设计和编码;设计出的构件需经严格的验证,验证包括功能测试和性能测试等,不合格的需进一步进行修改,合格的则放入构件库;最后一步是实际使用。

设计好的构件包括以下内容:

- 构件标识—构件的代码和表达构件功能的中文名;
- 构件说明—构件的功能、属性、复用、定义方法、调用接口及测试环境等说明信息;
- 构件源代码—构件的源程序代码包括子构件的程序代码;

构件目标代码—构件的目标程序。

2. 构件的设计原则

构件库中的构件是由众多的技术人员合作研制的,要使功能各异的一组构件能方便地组装成不同的应用系统,我们在设计时提出了以下设计原则:

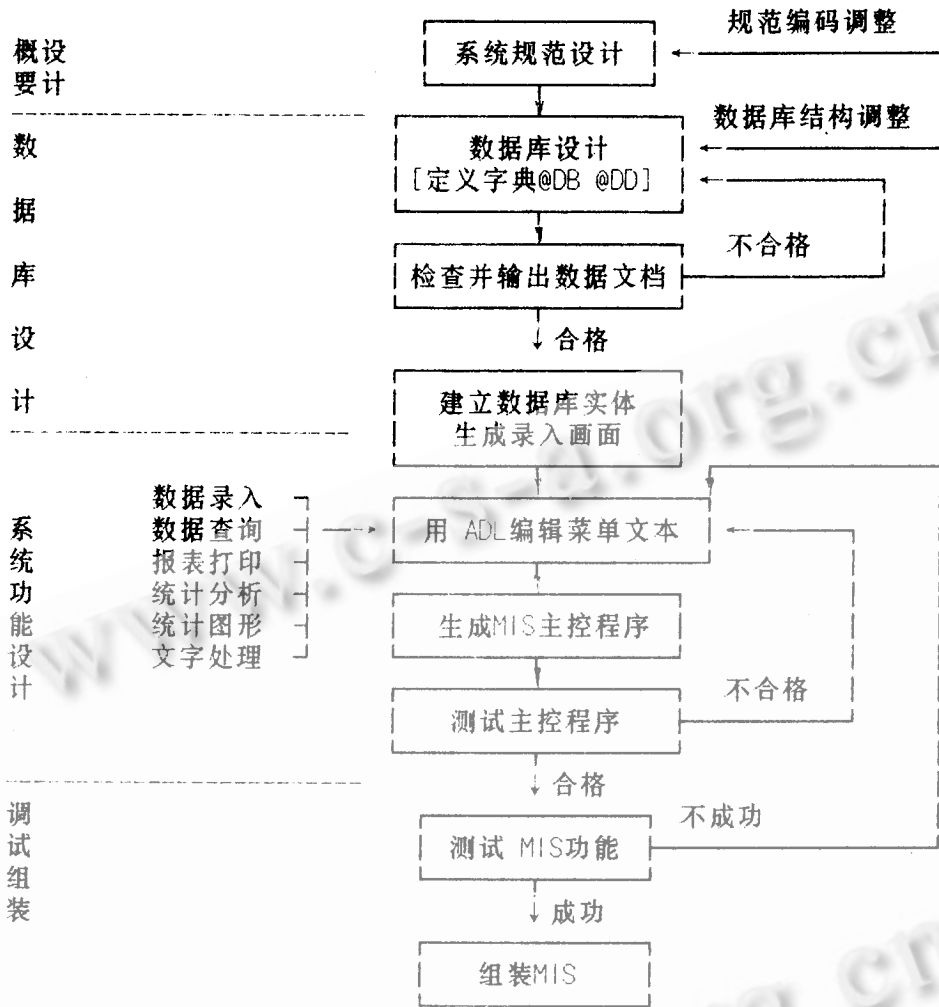


图 3

• 与对象、实体分离的原则

只有与对象、实体完全分离,才能保证构件的通用性。一些数据库管理系统都是与数据完全分离的,它与数据之间的联系是通过内设的数据字典来实现的。又例如:一些标准数学函数,它与数据之间的联系是通过参数的哑实结合来实现的。ICS的构件设计均以扩展数据字典来定义各类不同对象的结构信息、语义信息和操作信息,并以此为依据实现动态参数处理和命令动态执行。从而保证对象、实体与操作构件的分离。

• 规范化、标准化原则

对每个应用系统,用户一般都希望显示画面统一,操作方式一致,名词术语规范。因此 ICS 集成构造系统内的各构件的设计必须有统一的设计规范或约定,才能保证其风格基本一致。这种约定主要包括:编码规则、数据

字典的基本内容、接口形式、屏幕格式、操作方式、功能键的定义、错误信息编码以及所依赖的软硬件环境等。它是每个 ICS 构件设计的依据。

• 完整性原则

主要是指构件功能的完整性,每个构件应是相对独立的一个整体,可以独立使用,可以独立完成某一功能。因此,每个构件必须作精心的设计,以尽可能地达到完成某一目标所需要的全部功能,否则难以满足众多用户的不同需要。例如一个统计图形构件,它不仅要具有制作常用的直方圆、园饼图、折线图、曲线图的功能,还要提供三维直方图。以及几种不同统计图同时使用的功能。

• 可剪裁原则

由于每个构件具有很强的功能,多个构件构成一个

应用系统后,将使系统具有强大的功能,但另一方面却可能带来功能冗余,即包含某些用户不需要的功能,使系统过于庞大,降低系统运行速度,为了使构造的系统得到优化,因此系统个构件必须能进行适当的剪裁,以除掉那些多余的模块。

· 易插接原则

ICS 集成构造系统的基本框架是基于树型结构菜单构件,作为菜单树的叶节点,每个构件就当接口简单、调用方便、易于在这种基本框架上进行摘挂。

六、集成构造系统的实际使用情况

在最近一两年中,这套环境分别在几十个单位推广使用,收到了明显的效果,主要有以下几点:

1.设计思想先进,功能完整配套

以软件重用理论为基础,采用应用描述语言、扩展数据字典和可重用预制软件构件三位一体的设计方法,构成一整套积木式的 MIS 集成开发环境,该成果荣获军队科技进步二等奖。

2.开发应用系统的速度快、周期短

开发应用系统只需根据需求选择一些构件,并利用工具填写相关的数据字典,而不必靠手工一条一条的编写程序。用此环境建立一个中等规模的 MIS 原型,只需约一周的时间,经过与用户交换意见,再作适当的调整、修改,即可形成实用的应用系统。我们在开发全军房地产清查整顿信息处理系统时,其中企业化工厂部分,用一个半月三个月的时间就完成了系统的开发任务,这样的工作量若用传统的手工编程约需二、三个人年,粗略估计,提高工效 6~8 倍。

3.用 ICS 构造的应用系统运行可靠性高

系统的可靠性是用户十分关心的一个问题,用传统的手工编程的专用系统,在投入使用之前,一般要经过长时间的调试,但对于一个复杂的系统是无法用穷举法对每一种可能进行遍历,因而在投入实际使用后仍可能潜在很多的错误。而该环境中各个软构件都是精雕细琢而成,并在众多系统中重用,经受过比较复杂环境的考验,使运行的可靠性大大提高。

4.生成的 MIS 维护简单

当 MIS 需扩充或维护时只需修改数据字典和应用功能描述即可重新生成,而不必修改程序,避免了修改程

序可能引起的连锁反应,使最终用户可独立进行维护。

5.生成的 MIS 的优点

界面友好、操作方便,ICS 提供的软件用户界面、操作方式统一规范,揭示信息丰富,给使用提供了很大方便;系统提供的选项查询,条件组合查询,选页报表打印等功能给用户使用以相当大的灵活性。

6.系统开放性好

它不仅可以直接连接用户自己用 Foxbase 开发的功能模块,而且也可以接纳用户用其它语言开发的专用功能模块,并且均可作为软部件参加 MIS 组装。这为继承用户原有的程序和开发新的专用功能提供了途径。

7.用 ICS 构造的系统培训用户的时间短、费用少

该环境中各软构件均按统一的规范设计,其操作方式、功能键的设置、屏幕格式一致,信息揭示丰富,稍有计算机知识的人经过较短培训,即可学会使用,减少了培训的时间和费用。

七、结束语

《微机 MIS 集成构造系统》着眼于应用开发手段的改善和应用软件质量的提高,采用应用描述语言、扩展数据字典和标准预制构件三位一体的设计策略,在软件重用方面闯出了一条新路,为软件产业走向工程化探索了一定的经验。

参考文献:

- 1.M.Lenzi, et al, "Software Reuse Through Building Blocks",《IEEE Software》pp34, Jul.1987
- 2.Ledbetter and Cox, "Software-ICS: A Plan for Building Reusable Software Components",《Software State-of-the-art: Selected Papers》,1990
- 3.G.Lanergan and a.Grasso, "Software Engineering with Reusable Designs and Code",《Software State-of-the-art: Selected Papers》,1990
- 4.T.Mittermeir and M.oppitz,《IEEE Software》,1987
- 5.董枢美、李开德, "一种面向可理解性的可复用软件开发方法"《软件学报》,PP31.1,1990
- 6.朱海滨、陈火旺, "从语言的结构特征看'软插件'的形成",《计算机科学》,PP19.3,1990
- 7.蔡德明, "可复用软件库管理系统 SBMS",《计算机学报》,PP445.6,1990

