

基于组件结构的企业信息管理系统初探

陆 伟

(武汉大学图书情报学院, 武汉 430072)

摘要 本文针对我国当前 MIS 建设存在的问题,就软件工程领域提出的最新模式——软件组件结构进行了初探,并阐述了其在 MIS 应用中存在的问题。

关键词 管理信息系统 软件工程 软件组件 软件组件结构

Preliminary Approach to the Management Information System Based on Component Infrastructure

Lu Wei

(College of Library and Information Science, Wuhan University, Wuhan 430072)

1 问题的提出

管理信息系统(MIS)的开发在我国已有十几年的历史了,至今仍是风起云涌,方兴未艾。客观的说,我国 MIS 的建设确实取得了一定的成绩,在各行各业也正有一些 MIS 逐渐走向成熟,为我国企业的现代化发挥着应有的作用。但是,回过头来,我们却发现这些成功的案例与整个 MIS 开发的规模相比,却显得微乎其微,可以说,我国 MIS 开发的现状并不容乐观:人们一方面抱怨“低水平重复”,一方面却又对这种行为乐此不疲。这就导致同是一个行业,大家使用的 MIS 常常五花八门,甚至同一个企业的分厂、同一个商场的不同楼层,使用的 MIS 也是“龙生九种,各不相同”,而且这些 MIS 往往都是简单的事务、报表处理,真正能够体现管理水平,提高效率的却很少见到。究其原因,笔者认为,主要有如下几点:

(1) MIS 的规模与复杂程度与以前相比,有了数量级的增长。

(2) 需求复杂,规模大。现今的 MIS 系统往往覆盖了企业生产管理的各个层面,因而规模比较庞大,一个中型 MIS 系统一般需要大约一年时间才能完成,而大型系统往往需要几年。

(3) 缺乏规范性。这其中包括两方面的内容,一方面是软件开发人员不能严格按照软件工程的标准去做;另一方面是企业缺乏规范性,国外企业的管理模式有一种“八二原则”,即不同企业的管理信息虽然多种多样,但只要仔细归类,总可以找出 80% 左右的共性。剩下的那 20% 则是企业的个性特征,软件开发商可以为这些具有个性特征的东西设置众多供选择的相关参数,最终组合成适合各个企业的 MIS。这种“八二原则”不仅为软件开发商带来了方便,也使企业管理者得以“用 20% 的精力处理 80% 的常规业务,用 80% 的精力开拓新业务”。

收稿日期:1998年11月30日

作者简介:陆 伟,男,1974年生,武汉大学图书情报学院硕士研究生,研究方向为信息系统工程(软件开发环境与工具)。

我国的情况恰恰相反,专家们称之为“倒八二”,即企业运作中 80%的做法是“独具特色”,共性的东西少之又少。这种“倒八二”导致 MIS 软件只能专用不能通用。所以,细究起来,我国一些企业自行开发 MIS,也是不得已而为之。

(4)需求变化快。需求的变化是主要威胁 MIS 系统生命的因素。以往开发的系统,一般缺少对变化的适应性,每个变化都需要反馈到开发人员手上,开发人员再把可能已经是几年前的程序重新更新一次。这种形式的工作将花费技术人员很大的精力,也为 MIS 系统的维护带来了极大的困难。

(5)领导重视及各单位的配合问题。MIS 工程是“一把手工程”,需要企业领导的重视与亲自参与,企业领导必须转变 MIS 建设是信息中心的事,只要信息中心抓就足够的错误想法,没有企业领导及员工的大力参与,要开发出实用的 MIS 是不可能的。

(6)软件开发人员与企业用户之间的沟通问题。软件开发人员是技术专家,但对企业的管理可能不是很精通,这样双方在沟通上的误解也导致了 MIS 开发的不确定性。

解决上述问题除了要规范企业管理,发挥各种人为的因素外,在软件开发的方法上探索出一条新路也是需要解决的燃眉之急。笔者认为,目前在软件工程领域的热门话题软件组件结构将是 MIS 开发的方向。

2 软件组件结构(SCI)

2.1 软件组件结构提出的背景

软件开发迄今为止已有近五十年的历史了,期间大体上经历了如下发展阶段:原始开发——函数的使用——结构程序——模块化——面向对象。然而,随着软件规模的逐渐扩大,软件复杂性的与日俱增,传统的开发模式越来越难以满足人们的需要。据美国国家标准与技术研究院的一份报告显示,占据世界软件销售额 85% 的是大型的专用软件,而其开发的失败率却高达 70%。如何才能提高软件生产效率,保障软件产品质量,减少软件开发的重复劳动,亟需对软件的生产方式做一次根本性的变革。软件组件结构(Software Component Infrastructure, SCI)就是因应这一需要而产生的,其目的是彻底改变软件生产方式,从根本上提高软件生产的效率和质量,提高开发大型软件系统尤其是 MIS 系统的成功率。有了软件组件之后,应用开发人员就可以利用现成的软件组件装配成适用于不同领域、功能各异的应用系统。

2.2 软件组件

2.2.1 软件组件的概念

软件组件(Software Component)可以说是跨世纪软件技术发展的一个趋势,也是软件工程界的一个热门话题。关于软件组件,至今也没有一个统一明确的定义。目前典型的有四种说法,分别从不同角度对软件组件加以说明:第一种定义认为软件组件是一个并非不重要的、相对独立的而又能够被替换的系统中的一部分。第二个定义从动态的角度出发,指出处在运行状态的软件组件是一个将若干个程序链接在一起的程序包。第三种定义认为软件组件是可以独立使用的,并且不是有系统开发人员和系统最终用户而是由第三方提出的。第四种定义则从商用角度出发,认为所谓商用组件是一个能够自动化的商业过程的软件实施办法。这几点定义虽然角度不同,但都强调软件组件的独立性、互换性和功能性,即软件组件不依存于某一个系统,它可以被相同的组件替换,并且具有实际的功能意义。笔者认为,如下对组件的描述似乎更为贴切:软件组件是软件的基本量子(单元),这种单元应当足够大,以使组件具有一定的功能,并能被打包和插用。但是,这种单元又应当足够小,以便于它的维护和升级。

2.2.2 软件组件的属性

作为软件的基本量子单元,软件组件具有如下属性:

(1) 可插用性——组件是现成的可打包的软件部件,它可以从经销商处购得。

(2) 可维护性——组件是足够小的,因而易于维护和升级。

(3) 有功能性——组件是足够大的,使它以希望的有用方式动作。

(4) 不存在直接与其它组件通信的路径——组件不直接与其它组件通信,这对于组件的维护和升级是必不可少的。该属性并不意味着一个组件不能请求另一组件的服务,相反,它提供了为其它组件所调用的接口界面。

(5) 自我可描述性——组件能向系统的其余部分描述它提供的服务。这种描述通常是由通过说明性的、与实现无关的接口定义语言提供的。

此外,组件还具有具体化,专业化,可改编,可装配等属性。

2.3 软件组件结构模型

软件组件结构是软件开发的最新模型,与传统的面向过程的模型和面向对象的模型相比,具有更易维护、更易封装、重用率高、开发周期短等优点。通过对象总线,这种模型使跨越不同操作系统的、硬件架构、编程语言和编译程序的异类环境上的应用程序开发成为可能。同时,它通过引入框架使软件工程从过程化编程向类库编程再前进了一步。组件是可以重用的现成的软件部件,它可以从软件经销商处买到,组件被安装到框架,框架和组件通过对象总线与别的框架和组件连接。组件、框架和对象总线组成了软件的应用程序。图1是软件组件结构的基本框架。

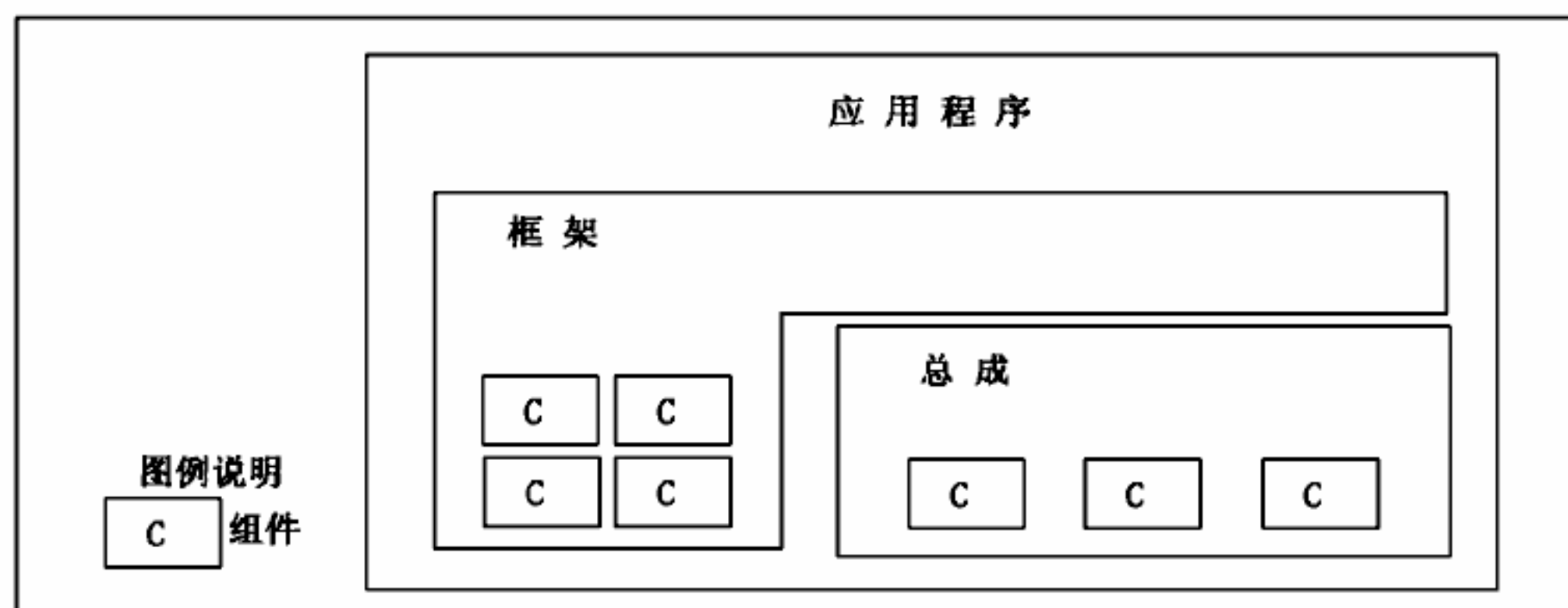


图1 基于框架的应用程序

专业领域内的一个应用程序由两部分组成:不变部分和变化部分。不变部分是一个应用程序和同一领域内其他应用程序分享的部分,变化部分是使一个应用程序在其领域内成为与众不同的那一部分。图中框架部分把握了某个领域内全部问题集中解决方案的不变部分,该部分由领域内的专家所设计,并已编码和调试,开发者只需向框架添加部分代码以把握其动作,使该应用程序成为所在领域内特定的应用程序,而这些附加代码只占整个应用的20%,其余80%都由框架来把握,这样开发者的工作量就大大降低了。图中总成代表了应用程序的变化部分,它是综合了领域的知识、专长、规则和特殊解法的策略,是解决方案中那部分随领域中问题不同而变化的部分,与框架正好相反,框架把握该领域中解决方案的不变部分。总成的代码符合框架建立的协议,并将它扩充或完善成为特定的解决方案。组件经过框架安插在一起就建立了许多应用程序,然而,这些应用程序仍然限定在单台机器上,对象总线把组件和框架

的能力扩展到开放网络。它使成百上千独立的软件单元在由不同操作系统、语言、硬件平台和编译程序组成的环境下互相无缝的交互运行,从而组成为大型的软件系统。对象总线著名的例子是 Object Management Group 的 CORBA(Common Object Request Broker Architecture, 公用对象请求代理架构) 以及 Microsoft 的 DCOM(Distributed Component Object Model, 分布式组件对象模型)。

3 基于软件组件结构的 MIS 开发所面临的问题

(1) 制定行业标准, 规范企业管理 这是进行基于组件的 MIS 开发的先决条件。我国正处于新旧体制的转轨时期, 行业标准与规范还很不成熟, 还有待进一步完善; 企业管理模式还有待进一步调整。只有形成统一的行业标准, 成熟的企业管理模式, 我们才能走出倒“八二”的怪圈, 这样, 专业标准组件的开发才有意义。

(2) 合理组件的定义 对于软件组件的开发人员来说, 如何在应用领域的模型中找出有共性、可通用的部分做成软件组件是他们要面对的首要问题。要解决这一问题, 光有软件技术是不够的, 还要求有应用领域的相关知识, 只有两者结合才能够解决应用问题。因此, 做构件之前首先要明确它将适用于哪个应用领域, 然后在根据这个领域的知识和应用模型抽取出最合理的组件定义。

(3) 组件相互之间的集成和装配 为了能够用组件集成应用系统, 使组件能够装配互换, 必须要有一个统一的标准, 如前文提到的 CORBA 等。

(4) 组件之间的相互通讯与数据共享。

(5) 对传统 MIS 数据的继承 软件组件技术才刚刚起步, 而在此之前已有成千上万 MIS 系统在运行, 如何继承传统 MIS 的数据, 实现其向组件技术的转变, 也是我们所要面临的问题。

4 结 束 语

目前, 基于软件组件的 MIS 开发还很不成熟。虽然在理论上已取得了一些进展, 但在 MIS 的实际应用中仍有不少问题要解决。尤其是我国的企业还很年轻, 管理的规范化、标准化还有很长的路要走, 但是, 随着通过 ISO 9000 国际标准认证企业的增多, 我国企业的管理水平必将迈上一个新的台阶, 基于软件组件结构的 MIS 建设也必将有自己的用武之地。

参 考 文 献

- 1 Rudolf K. Keller 等. Design Components: Towards Software Composition at the Design Level
- 2 L. B. S. Raceoon. Fifty Years of Progress in Software Engineering. Software Engineering Notes, 1997, 22(1)
- 3 Andert 等. Object Frameworks in the Taligent OS. Proceedings of the IEEE COMPCON, Spring 94
- 4 微电脑世界主编. 可恶的信息技术. 微电脑世界, 1998, (37)
- 5 冯洁. 软件工程的演进. 微电脑世界, 1998, (35)
- 6 贺莉. 规范化管理是商业 MIS 成功的保障. 信息与电脑, 1998, (4)
- 7 刘庆波. 试论在中小型外贸企业中建立管理信息系统. 计算机世界, 1997, (2)
- 8 张勇. MIS 开发工具的产生原因及设计思想. 计算机世界, 1997, (14)
- 9 张海藩. 软件工程导论. 北京: 清华大学出版社, 1998

(责任编辑 芮国章)



知网查重限时 **7折** 最高可优惠 **120元**

本科定稿，硕博定稿，查重结果与学校一致

立即检测

免费论文查重: <http://www.paperyy.com>

3亿免费文献下载: <http://www.ixueshu.com>

超值论文自动降重: http://www.paperyy.com/reduce_repetition

PPT免费模版下载: <http://ppt.ixueshu.com>
