

一个模板重用的实例 ——报表打印生成器的设计与实现

郭江 廖越虹 (北京航空航天大学软件工程研究所)

摘要: 本文从软件重用的方法——模板重用的角度出发,讨论了一个实用的应用生成器——通用报表打印应用生成器的设计,并详细给出了它的实现技术,以使其具有广泛性和通用性。该应用生成器是一个实现了的软件工具,具有很好的可移植性。

一、引言

软件重用的方法可以分为两大类:重用构造块方法和重用模块方法。应用生成器是软件重用的重要手段之一。它可以认为是以模板重用为核心的软件重用,也就是用特殊领域产生的知识进行编码,以便在该领域产生同样的系统时进行重用。

报表打印是业务处理中最常见的一种数据处理方式,它能给用户以清晰的文档形式,而且还便于汇报、整理。正因如此,报表打印用途非常广泛;也正因为用途广泛,使得报表各不相同,各有各的特色。但这不过是表面现象,各种报表在总体上基本是一样的。因此,如要为每一报表编制一个打印程序,既无必要,工作量又太大,而且开发出的打印程序可靠性又不一定好,错误再所难免,交付的时间也成问题。在这种情况下,应用生成器这种软件重用的手段就成了一种很自然的选择。

本文将介绍通用报表打印应用生成器的设计与实现。该软件是一个实用系统,它能生成各种各样的打印程序,所生成的打印程序可打印宽度达 240 个英文字符,表头结构达十层(从理论上讲,表头层数是不受限制的;而在实现时,过多的表头层数对用户没有什么实际意义)的报表,对任何报表来说都是足够的了。并且每次生成打印程序的同时,也为它生成了相应的表头文件和参数文件,通过对它们的修改可以简单地修改打印程序。考虑到使用的方便性,该软件系统采用多级菜单选择方式。从这个角度上来讲,用户只需将有关报表的信息输

入描述字典和结构字典,在以后使用时菜单中将自动显示有关该报表的信息,因而只需选择相应的选择项即可。用户只要选择他想要的,而生成器就会自动生成打印程序。该软件系统还提供了一个专门编辑表头的全屏幕编辑器,用户可以方便地对表头进行修改。

该软件系统是用 C 语言编写的,在各种中文环境下均可以运行,对机器无任何特殊要求,具有很好的兼容性。而生成的程序则给出了完整的程序清单,可不加修改地在 dBASE III 或 FoxBASE 环境下运行,打印出令人赏心悦目的报表。

二、涉及到的几个数据库

在设计应用生成器时,我们首先设计了它所用到的几个数据库:描述字典(SYSTEM_DB)、结构字典(STRUCT_DB)和当前使用数据库(REAL_DB)。下面分别加以说明。

1. 描述字典

库结构如下表:

字段	字段名	类型	宽度	小数
1	SSYS	Character	30	
2	DQK	Character	4	
3	DYCXM	Character	6	
4	LSK	Character	6	
5	SBK	Character	8	
6	JGK	Character	5	
7	SJTX	Character	6	

8	BBLX	Character	2
9	BBH	Character	12
10	JSGS	Character	4
11	HZFS	Character	4
12	FJCL	Character	2
** 总计 **			90

这个库中存储了对各个报表的总体描述信息。这些信息在生成报表打印程序时显示是不可缺少的,描述字典的存在也是使用本应用生成器的不可缺少的条件之一。

2. 结构字典

库结构如下表:

字段	字段名	类型	宽度	小数
1	FIELD_NAME	Character	10	
2	FIELD_TYPE	Character	1	
3	FIELD_LEN	Numeric	3	
4	FIELD_DEC	Numeric	3	
5	FIELD_REC	Character	30	

** 总坟 ** 这个库与相应的当前库对应,即有一个当前使用数据库,就有一个结构字典与它对应。

在本系统中,这个库的作用是存储报表各阶段的相关信息,以便在生成报表打印程序时使用。另外可用来显示各阶段的中文含义,以使用户可以方便地确定每栏的内容。

3. 当前使用数据库

库结构由用户确定,与相应结构字典完全一致。在生成的报表打印程序中,所有输出表格中的数据都来自这个库。但事实上,在本软件系统执行时并不涉及到这个库,它只在所生成的打印程序执行时才使用。

三、系统总体设计思想

本系统用于生成报表打印程序,因此在设计时要先从报表下手。

一个报表主要由标题、表头、内容及一些诸如表号、日期等的附加内容组成。内容是取自当前库中的信息,其余部分基本上都由用户自行确定。标题、日期等信息都是以固定模式打印出来的,其内容也比较简单,容易确定。相比之下,表头比较复杂,模式也不固定,因此处理表头就成为设计这个应用生成器时要考虑的主要问题。

本系统一次只生成一个报表打印程序,所以首先必须确定要生成哪一个报表打印程序,这只要由用户通过菜单式选择所使用的结构字典就实现。

结构字典确定后,先生成表头才能进一步生成报表打印程序,而关于表头的各种信息都要由用户输入。在这里采用对话式输入,这种输入方式简单易掌握,报表的内容与表头是一一对应的,所以在输入建立表头的有关信息时,也同时确定报表各栏的内容是很方便的。

由于表头形式比较多变,为了弥补表头自动生成时可能造成的缺陷,我们设计了一个全屏幕编辑器,以使用户可以对表头中不满意的地方进行修改。

对表头进行全屏幕编辑后,生成报表打印程序所需的条件就已基本具备了,只要按已选定的模板生成报表打印程序并将之写入磁盘文件就可以了。在系统中所采用的模板是经过实践反复考验的,并且很好地处理了打印出的报表格式,设置了标题、日期、表号、页号及填表人等项的字体和放置位置。

系统的总体设计思想基本如此,系统的总体设计结构如图 1 所示。

四、系统实现

按照前面的系统总体设计结构图,本系统可分为四部分,下面将详细讨论在具体实现时遇到的一些技术问题。

1. 如何直接读取数据库中的信息

正如前面所交待过的,要使用的所有数据都存储于 dBASE III 或 FoxBASE 数据库中,而本系统是用 C 语言编制的,于是怎样直接从数据库中读取信息就成了必须解决的一个技术问题。

dBASE III 或 FoxBASE 数据库文件都分为如下两部分,第一部分是文件结构信息,第二部分是已写入文件的实际数据。

文件结构信息对文件进行了总体描述,并对数据库的各字段进行了描述。在这一部分中,提供了当前记录条数,文件结构长度、每条记录长度等几个重要数据,利用 C 语言提供的 fseek、fgetc 等函数读出感兴趣的数据,根据这些数据就可以对数据库的内容进行直接读取了。

2. 如何让用户方便地输入有关表头的信息

首先,对表头的结构进行一些分析,事实上,我们发

现不论表头结构看上去如何复杂,它的基本结构都是如图 2 所示的那样。

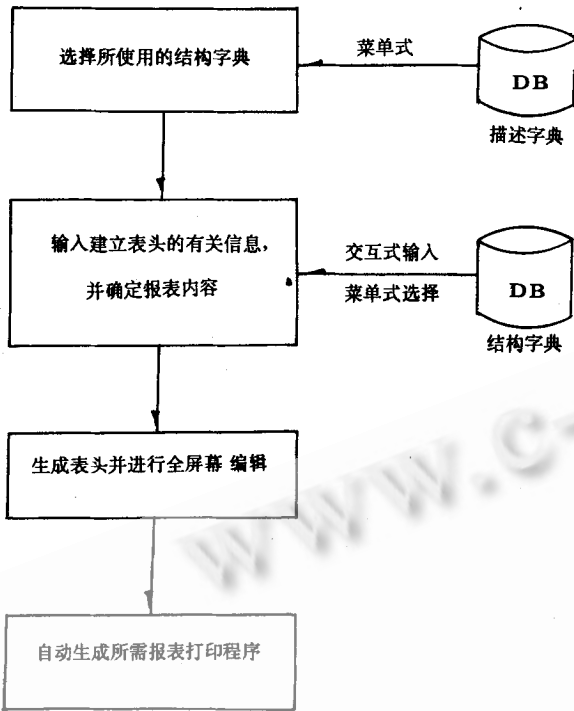


图 1 总体结构设计

多层表头的变化仅仅表现在图 3 所示的那些虚线上,只要在生成表头时将这些虚线去掉,并将分隔符做相应的变化就可以得到图 4 所示的复杂的表头。



图 2

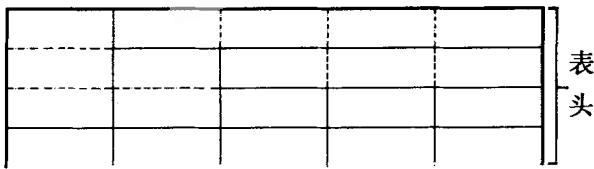


图 3

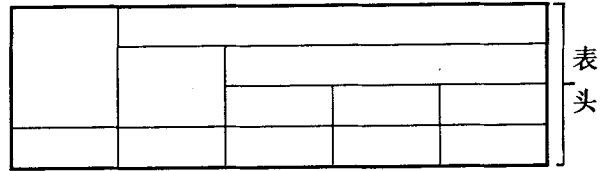


图 4

经过进一步的分析发现,一个报表的内容一经确定,不论表头如何复杂,如何多变,报表的内容也不会再改变了。这就是说,表头只是表现报表内容的形式,它之所以成为报表的一个必不可少的部分,是因为它是报表内容的必要说明,没有它报表就只是一堆难于理解的文字或数字,这和注释使程序具有良好的可读性的道理是一样的。因此,表头处理的主要任务是为内容加上一个清晰易懂、整齐美观的表头,而非像我们平常所习惯的那样按照表头去填表。这个概念对于我们的处理是很有用的。在明白了这一概念之后,还要注意下面这一点,那就是表头最下面一层是与报表内容逐栏对应的,多层表头的出现只是为了使表头更清晰、更美观。

经过上面的一系列分析之后,把表头作为一个独立部分来处理就是很自然、很容易理解了。而将表头和报表内容联系起来,也就是与报表内容逐栏对应的表头最底层了。众所周知,一个整齐的报表,它的每一栏的宽度从上至下是完全一样的,这样才能形成一条条笔直的分隔线,这个宽度对于形成报表是至关重要的。在本系统中,将报表各栏内容所对应的数据库相应字段的宽度作为报表各栏的宽度,同时也是相应表头底层各栏的宽度。既然表头底层具有如此重要的作用,而它的每一栏又都需要单独处理,因此我们确定了由下至上、逐层逐栏地让用户输入有关表头信息的方法。事实上,由于各栏的宽度与相应栏报表内容对应,在输入有关表头信息的同时,也就确定了报表内容。

要形成形式复杂的表头,只是在简单形式的表头上做一些修改。这些修改是按用户的意志完成的,也就是说在什么地方做什么修改是用户决定的,这些信息都要由用户来输入。怎样才能让用户只输入极简短而又容易识别的信息就能达到自己的目的呢?我们采用了代码方式。

这里,我们所使用的代码是由两位阿拉伯数字组成的,表头的每一层的每一栏都对应一个代码,系统将根据这些代码来形成表头的框架。为保证能够正确理解用户意图,系统规定了三种代码形式,用于用户输入。

表头的最底层各栏代码由系统给出,从第一栏的 01 开始依次增大,而非最底层的代码则由用户输入,它们是在最底层代码基础上稍加变化形成的,我们以下要说明的也正是用户输入的合法代码形式。

(1)XX型:此型代码只由两个数字组成,不加任何扩展符号。这种代码表示本栏与它所对应的最底层子栏宽度相同,且不做任何修改。事实上,这种代码在实际应用中没有什么意义。

(2)XX/型:此型代码是由两个数字外加扩展符“/”构成的。它表示此栏与它所对应的下层一栏相通,它们中间没有分隔线。这样,由于每个制表符分隔线占一个字的宽度,就可在原分隔线处写上字来构成此联通字栏内容的一部分,这就使表头更灵活。

(3)XX-YY型:此型代码是由两个代号中间加扩展符“-”构成的,它表示本栏由 XX 到 YY 之间的几个子栏联通而成,它们之间无分隔线,构成一个长条的横向栏。这样表示本栏包括了若干子栏,表达了一种从属关系,而且内容也可因栏宽变宽而增加。

图 5 示意出了这三种代码所表示的子栏框架。

对这些代码进行翻译、解释之后,把抽象的代码变为具体的符号,即画出报表框架的制表符,就可以画出用户需要的表头框架。

在表头处理中还必须注意一个问题,就是表头的宽度。因为打印机能打印的最大宽度是有限的,所以若用户要打印的表太宽,即比打印机的最大打印宽度还要宽,打印时超宽的部分就会丢失。因此我们设计了一个标尺,用户每输入一栏就判断一下加上这一栏后报表有没有超宽。如果太宽,就给用户一些提示,并采取相应的补救措施。

3.如何形成整齐的表头

当用户输入了有关表头的所有信息后,依照正常情况,就可以将表头画出来。

01	021	03-05			} 表头
01	02	03	04	05	

图 5

但是,也有可能出现这种情况,就是表头某层某栏的名称过长,比它对应字段的宽度还要宽。这时,由于表是分层处理,生成的表头就可能会参差不齐,要使各层对齐是必须对它相关的许多栏进行分析后才能确定如何修改的。这个工作对于手工作业来说是直观而简单的,但对于机器,由于它远不如手工那样灵活,处理起来就相当复杂,所以这个工作由手工来做显然更为合适。

另外,尽管在输入各种信息时均要经过用户确认,但仍然可能出现错误,也可能在事后发现当时输入的东西要做一些修改。做这一工作显然不可能将所有信息重新输入一遍。为解决这个问题,我们想到了另一个工具——编辑器。

在本系统中,将生成的表头直观地展示给用户是很必要的。因为只有将所有有关的信息输入之后,表头才能形成,所以在输入期间用户是不可能看到具体的表头的。在输入结束后,利用一个编辑器将表头展示给用户,一来可以作为一个整体显示出来,使用户掌握全局;其次可以进行必要的修改。

由于主要是用来编辑表头,所以这个编辑器的功能比较简单,只要自动将表头显示在屏幕上,并给出必要的操作提示,待用户感到满意后在退出这一状态的同时将表头及有关信息存盘即可。

参考文献:

- [1]郭江,“软件重用的模板方法”,计算机应用研究,1993.3.
- [2]廖越虹,学士论文,报表打印程序应用生成器。