



通用报表系统实现技术探微

深圳信息职业技术学院 陈新林

一、问题的提出

通用报表系统主要用来灵活实现企业报表的呈现与打印。目前,我国的通用报表系统主要用于财务与会计报表的呈现与打印,都表现为类似 EXCEL 的风格特点,可用操作 EXCEL 表格的方法(或类似方法)操作这些系统。这类系统都存在着不能真正实现灵活的数据源取数的问题。目前的通用报表系统是不可以自定义取数源与取数函数或公式的,而只能选择系统提供的有限的取数函数或公式来自定义报表,如果需要对这些取数函数或公式以外的数据资源的访问,如取非财务数据制作管理报表或者需要更灵活地进行取数就不方便了。一个真正灵活的报表系统应该可以自定义取数源、取数函数或公式以及变量,且能与 EXCEL 兼容。

二、实现条件与需要解决的问题

市面上一直流行一个叫 Formula One 的 ActiveX 的专业控件,其不同版本可以实现与 EXCEL 不同版本的完全兼容

(除 MACRO 和 VBA 外)。通过开发基于 Formula One 的 ActiveX 通用报表系统,即可实现 EXCEL 报表的通用功能,这样凡会使用 EXCEL 的用户即可使 EXCEL 的简单、方便、灵活在设计通用报表系统中得以体现。但要获得真正灵活的通用报表系统,在上述系统的基础上还需要解决以下问题:①可以通过一种方式自己定义取数函数或公式;②可以通过一种方式自己定义部分变量;③设计通用报表系统时可以利用这些自定义的函数或公式、自定义的变量及像 EXCEL 中可以通用的公式、变量、函数等组成表达式;④上述自定义的表达式可以被系统认识(即系统本身可翻译与识别),并取得需要的数据。

三、实现思路

要解决上述问题,必须在一个与 EXCEL 兼容的报表系统中建立一个编译器来解决其中的核心问题,即上述问题四。如何建立一个这样的编译器则是问题的关键。笔者假设有这样

“待转资产价值”科目,如果接受捐赠企业自行支付固定资产增值税进项税额,则应按支付的固定资产增值税进项税额,贷记“银行存款”等科目,按接受捐赠固定资产的价值,贷记“资本公积”、“待转资产价值”科目。如果接受捐赠企业另支付其他费用如运输费等,还应贷记“银行存款”等科目。

5.销售本企业已使用过的固定资产的核算。企业销售自己已使用过的固定资产,如该项固定资产取得时,其增值税进项税额已记入“应交税金——应抵扣固定资产增值税(固定资产进项税额)”科目,则销售时计算确定的增值税销项税额,应借记“固定资产清理”科目,贷记“应交税金——应交增值税(销项税额)”科目;如该项固定资产取得时,其增值税进项税额未记入“应交税金——应抵扣固定资产增值税(固定资产进项税额)”科目,则按税法规定在销售时允许抵扣的增值税进项税额,应借记“应交税金——应抵扣固定资产增值税(固定资产进项税额)”科目,贷记“固定资产清理”科目,销售时计算确定的增值税销项税额,应借记“固定资产清理”科目,贷记“应交税金——应交增值税(销项税额)”科目。

例 2:某企业出售一台使用过的设备,原价 468 000 元(含增值税),已使用 3 年,计提折旧 120 000 元,售价为 351 000 元(含增值税)。假设该项固定资产取得时,其增值税进项税额未记入“应交税金——应抵扣固定资产增值税(固定资产进项税额)”科目,则有关账务处理如下:

(1)固定资产转入清理时,借:固定资产清理 348 000 元,累计折旧 120 000 元;贷:固定资产 468 000 元。借:应交税

金——应抵扣固定资产增值税(固定资产进项税额)68 000 元;贷:固定资产清理 68 000 元。

(2)收到价款时,借:银行存款 351 000 元;贷:固定资产清理 351 000 元。借:固定资产清理 51 000 元;贷:应交税金——应交增值税(销项税额)51 000 元。

(3)结转固定资产清理净损益时,借:固定资产清理 20 000 元;贷:营业外收入 20 000 元。

6.固定资产购入后发生不得抵扣进项税额情况的核算。企业购入固定资产时已按规定将增值税进项税额记入“应交税金——应抵扣固定资产增值税(固定资产进项税额)”科目的,如果相关固定资产专用于非应税项目、免税项目、集体福利和个人消费,以及将固定资产供未纳入抵扣适用范围的机构使用等,应将原已记入“应交税金——应抵扣固定资产增值税(固定资产进项税额)”科目的金额予以转出,借记有关科目,贷记“应交税金——应抵扣固定资产增值税(固定资产进项税额转出)”科目。

另外,企业将应抵扣的固定资产进项税额抵减未交增值税时,应借记“应交税金——应交增值税(未交增值税)”科目,贷记“应交税金——应抵扣固定资产增值税(已抵扣固定资产进项税额)”科目。期末,企业以当期新增增值税税额抵扣固定资产进项税额时,应借记“应交税金——应交增值税(新增增值税税额抵扣固定资产进项税额)”科目,贷记“应交税金——应抵扣固定资产增值税(已抵扣固定资产进项税额)”科目。☐

的表达式: $\text{Formula} = \text{Func1}(\text{var1}, \text{var2}, \text{CONST}) + \text{A3} + \text{B4} + \text{SUM}(\text{A1}; \text{A5}) + \text{Func2}(\text{var1}, \text{CONST}, \text{var3}, \dots) + \text{Var4} + \text{Var5} + \text{Func3}() \times (-1)$ 。

对上述表达式笔者分析如下: 表达式中有多个函数, 其中有三个自定义函数(Func1, Func2, Func3), 有一个系统函数(SUM), 有单元格变量(如 A3, B4)。表达式可以有多个函数, 自定义函数可以有零到多个参数, 其中应该有变量和常量, 这是上述表达式的基本情况。

设计出的编译器至少要能实现对上述表达式的编译与识别。由于 Formula One 与 EXCEL 兼容, 因此通过 Formula One 实现这样的表达式识别有一定的基础, 如系统函数、单元格变量、系统常量、系统变量及由此组成的表达式系统是可以识别的。

现在的问题是, 如何识别用户定义的各类常量、变量、函数以及这些自定义函数参数的自定义变量等, 这些特性的识别是我们解决的关键问题。在实现时, 笔者在数据库中建立了两个表, 一个用于存放用户定义的各类常量和变量, 另一个用于存放用户定义的各类函数。这样通过系统设置, 这些信息即可进入系统, 得到系统的统一管理。然后, 在上述表达式中, 分别在用户定义的常量、变量和函数前面加一特殊符号(如“\$”)加以鉴别, 这样当我们识别并替换完这些带“\$”的表达式后, Formula One 即可完成对整个表达式的识别, 通过提取数据库中的数据, 即可把表达式转换为数量信息, 达到报表数据呈现的要求。

四、算法实现分析

一方面, 对于会计报表数据取数有一些特殊要求: ①历史数据的获取; ②表间取数(或称账上取数); ③数据一致与校验; ④报表设计与数据呈现分离; ⑤报表生成自动化。

另一方面, 具体分析表达式时有如下难点: ①表达式语法可能有错误, 需要做语法检查。②变量有多个, 且有多种类型, 必须只处理用户定义的系统级和用户级变量, 且用户定义的各类变量都要求被处理, 并要处理好变量的数据类型。③函数也有多个, 且有多种类型, 必须只处理用户定义的系统级和用户级函数, 且用户定义的各类函数都要求被处理。另外, 函数中还含有各类变量, 这些变量也需要识别与处理。④对表达式中各类用户级变量、函数的分析和识别以及数据提取变换后的结果依然是表达式, 且要完全被 Formula One 本身识别和转换为最终结果数据。

为了解决上述问题, 建立如下系列函数以实现对表达式的分析: $\text{FR}(\text{string}); \text{string}$ ——表达式分析入口, 由此函数完成对整个表达式的识别工作。由它调动以下函数来实现。 $\text{FF}(\text{string}); \text{string}$ ——分析与发现表达式中的变量, 包括非法变量的识别。 $\text{FG}(\text{string}); \text{string}$ ——到数据库中获得此变量的数字值或字符串。 $\text{FR}(\text{string}); \text{string}$ ——用数字值或字符串替换变量或函数。 $\text{FF}(\text{string}); \text{string}$ ——分析与发现表达式中的内部函数, 包括非法函数的识别。 $\text{FG}(\text{string}); \text{string}$ ——到数据库中获得此内部函数的数字值。 $\text{FF}(\text{string}); \text{string}$ ——分析与发现表达式中的用户定义函数, 包括非法函数的识别。 $\text{FG}(\text{string}); \text{string}$ ——到数据库中获得此用户定义

函数的数字值。

五、编译器的实现步骤

第一步, 变量的识别与替换: ①分析与发现表达式中的变量, 包括非法变量的识别; ②到数据库中获得此变量的数字值或字符串; ③用数字值或字符串替换变量或函数。

第二步, 内部函数的识别与替换: ①分析与发现表达式中的内部函数, 包括非法函数的识别; ②到数据库中获得此内部函数的数字值; ③用数字值替换变量或函数。

第三步, 自定义函数的识别与替换: ①分析与发现表达式中的用户定义函数, 包括非法函数的识别; ②到数据库中获得此用户定义函数的数字值; ③用数字值替换变量或函数。

需要注意的是, 以上步骤不可以颠倒。

六、算法实现实例分析

表达式: $\text{FormulaSource} = "=" \& \text{LJYUE}(\text{AccSet}, '101', '', \text{ThisMonth}) + \text{A8} + \text{B4} + \text{SUM}(\text{A1}; \text{A5}) + \& \text{FaShengE}(\text{AccSet}, '102', '', \text{ThisMonth}) + \& \text{LJYUE}(\text{AccSet}, '121', '', \text{ThisMonth}) \times (-1) + \& \text{SQ}(\text{AccSet}, \text{表名称}, \text{D5})"$ 。以上表达式的含义为: 取本月 101 科目 AccSet 账套累计余额 + A8 + B4 + A1 至 A5 的合计 + 本月 102 科目 AccSet 账套发生额 - 本月 121 科目 AccSet 账套累计余额 + 上期某表(如 \$ 表名称 = '资产负债表') D5 单元格的数据。

按上面的三个步骤, 经过第一步扫描和替换, 表达式将变为: $\text{FormulaSource} = "=" \& \text{LJYUE}('02', '101', '', '200410') + \text{A8} + \text{B4} + \text{SUM}(\text{A1}; \text{A5}) + \& \text{FaShengE}('02', '102', '', '200410') + \& \text{LJYUE}('02', '121', '', '200410') \times (-1) + \& \text{SQ}('02', '资产负债表', \text{D5})"$ 。这里假设账套 AccSet = '02', 月份 ThisMonth = '200410'。

经过第二步扫描和替换表达式将变为: $\text{FormulaSource} = "=" \& \text{LJYUE}('02', '101', '', '200410') + \text{A8} + \text{B4} + \text{SUM}(\text{A1}; \text{A5}) + \& \text{FaShengE}('02', '102', '', '200410') + \& \text{LJYUE}('02', '121', '', '200410') \times (-1) + 2 \ 334.56"$ 。这里假设上期资产负债表 D5 单元格的数据 = 2 334.56。

经过第三步扫描和替换表达式将变为: $\text{FormulaSource} = "=" \& 23 \ 210 + \text{A8} + \text{B4} + \text{SUM}(\text{A1}; \text{A5}) + 150.1 + 300 \times (-1) + 2 \ 334.56"$ 。这里假设累计余额、发生额取到了相应的数据。

七、部分函数实现的关键技术

1. 用户自定义函数如 $\text{LJYUE}(\text{AccSet}, '101', '', \text{ThisMonth})$ 的定义问题。设计时通过数据库表来实现, 表结构包括函数名、函数的功能与参数提示、对应的 SQL 语句、参数个数、返回数据个数等。如:

函数名 = LJYUE; 函数的功能与参数提示 = 取? 账套? 一级科目? 明细科目? 年月的累计余额; SQL 语句 = $\text{select debit_lj} - \text{credit_lj from report_sum where acc_set} = ? \text{ and acc_code} = ? \text{ and sub_acc_code} = ? \text{ and year_month} = ?$; 参数个数 = 4; 返回数据个数 = 1。步骤: ①对实际 LJYUE 参数的替代; ②查数据库表得到其对应的 SQL 语句; ③替换 SQL 语句中带? 的参数; ④执行 SQL 语句, 即可得到结果。

2. 表间取数如 $\text{SQ}('02', '资产负债表', \text{D5})$ 的定义问题。该函数的含义是: 取 02 账套上期资产负债表 D5 单元格中的数据。□