基于 TCL 的电路板原理图自动审查平台建设

田 野 赵保华 屈玉贵
（中国科技大学计算机系 合肥 230027）

摘 要：该文提出了一种用 TCL 语言作为审查脚本语言构造原理图审查系统的系统框架。在该系统中，利用 TCL 语言的扩展性，制定了一套应用于原理图通用数据结构的 TCL 扩展命令，并在此基础上构建原理图规则库和自动审查执行器，实现了原理图的自动审查。

关键词：TCL(Tool Command Language)，电路板原理图设计，通用数据结构，自动审查
中图分类号：TP311.52 **文献标识码**：A **文章编号**：1009-5896(2005)01-0158-03

An Auto-verification Platform for Schematic Design Based on TCL

Tian Ye Zhao Bao-hua Qu Yu-gui
(*Dept of Computer Univ. of Sci and Tech of China, Hefei* 230027, *China*)

Abstract In this paper, an auto-verification platform for schematic design is presented based on the extended TCL as a script language. Using the extensible property of the TCL, constructed a set of extended TCL command on the general data structure of the schematic design, and on this base, a rule library and an auto-verification runner are built to make the auto-verification of the schematic design possible.

Key words TCL, Schematic design, General data structure, Auto- verification

1 引言

在电路板的原理图设计阶段，常常有一些固定的模式，比如某个器件的某个典型应用或某个器件的使用限制。把这些规则用某种形式记录下来，就形成了原理图规则。在本文描述的系统中，我们采用扩展的 TCL 脚本和原理图通用数据结构文件来实现对规则的记录和自动执行。

我们实现的系统应用于 EDA 工具 ViewDraw 中，具体方式是在 ViewDraw 中采用 AddIn 方式，加入菜单，激活不同的模块实现。系统使用 TCL 语言作为规则描述语言，通过执行 TCL 脚本对 ViewDraw 的本地原理图工程实行规则审查。系统还包含本地/服务器的规则管理和规则编辑。

2 原理图数据结构

在原理图设计中，设计通常都是分层的、模块化的。一般来说，某个设计的顶层图包含了该设计的所有模块，顶层图可能不止一页，而每个模块又有自己的顶层图和下层实现，同样，每层的实现可能也不止一页。由于原理图设计的这些特点，直接使用原理图文件进行审查将导致规则不具有普适性，也给规则的设计造成困难。

2003-08-26 收到，2004-03-05 改回
国家自然科学基金重大研究计划项目(90104010)，国家自然科学基金科学部主任基金项目(60241004)，教育部博士点基金项目(2000035802)，安徽省自然科学基金项目(01042208)，国家 863 计划项目(2001AA112062 和 2001AA121016)，中国科学院院长基金特别支持项目(院基计字 905 号)和中兴通信股份有限公司基金资助课题

基于上述考虑，在我们的系统中，将先把 ViewDraw 的原理图设计文件转化为以 EDIF 数据结构为基础的原理图通用数据结构文件[1]，再在此基础上进行审查。

3 系统构成

原理图自动审查平台是一个集管理功能和审查功能于一体的软件系统，具体包括：

(1) 规则管理模块 规则管理模块包含两个子模块：服务器规则管理模块和本地规则管理模块。规则管理模块实现规则的添加、删除、修改、查看、上载、下载、导入和导出功能。

(2) TCL脚本编辑模块 TCL脚本编辑模块包含脚本编辑、命令的自动生成和规则脚本强制检查。

(3) 规则审查模块 包括审查控制模块和审查执行模块。

(4) TCL扩展命令集 扩展用于构建TCL审查脚本的扩展命令集。

系统结构见图1。

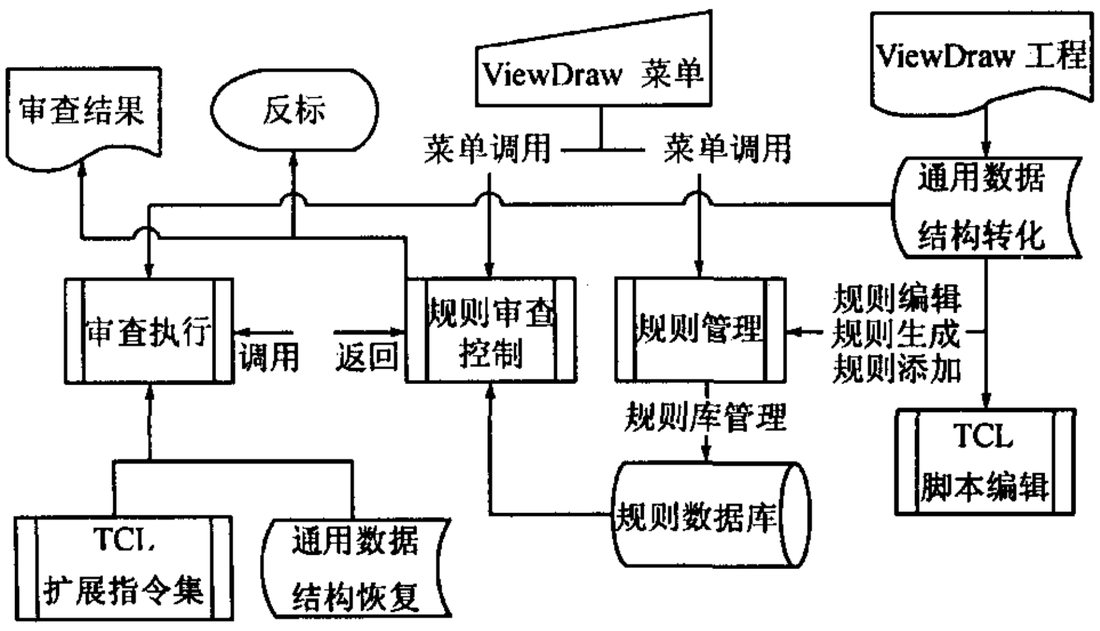

图1　原理图审查平台系统结构图

4　规则审查描述

在我们的原理图审查平台中,我们把规则分为 3 种类型:标准规则、一般规则和标准原理图规则。下面分别说明:

标准规则　规则仅为一个 TCL 脚本。这种类型的规则一般是与具体的器件无关的。例如,“所有标号为 GND 的器件管脚连接的网络必须为地网络”就是一条标准规则。对这条规则的审查就是通过执行规则脚本,检查待审查的原理图设计。

一般规则　规则包括一个器件编号列表和一个 TCL 脚本。这种类型的规则和某一种具体的器件有关。例如,“所有 C 型号电容#1 脚必须接高电平”就是一条一般规则。对这条规则的审查也是通过执行规则脚本,检查待审查的原理图设计。

标准原理图规则　规则包括一个器件编号列表、一个原理图通用数据结构文件和一份 TCL 脚本。这种类型的规则和某种器件的某个固定的设计模式有关,该设计模式就存放在规则的原理图通用数据结构文件中。对这条规则的审查就是通过执行规则脚本,匹配比较待审查的原理图和规则的原理图通用数据结构信息,检查设计是否按照规则包含的设计模式来进行。

由于规则的不同类型和使用方式,我们采用以下流程进行原理图审查:

第 1 步　获取 ViewDraw 当前的设计工程,转化为待审查的标准数据结构存放在内存中。对于一般规则审查或标准原理图规则审查,列出该设计中所有的器件名称和编号,供用户选择;对于通用规则审查,直接进入第 2 步。

第 2 步　对于通用规则审查,列出所有规则供用户选择;对于一般规则或标准原理图规则,列出所有器件编号,列表包含用户在第 1 步选择的器件的规则,供用户选择。

第 3 步　根据用户在第 2 步选择的规则,下载规则(对于标准原理图规则审查,还要下载规则的通用数据结构文件),生成审查控制脚本。审查控制脚本使用 source 命令进入具体的规则脚本,并获取返回值。

第 4 步　执行控制脚本,进行审查。对于通用规则和一般规则,执行器直接进入每个规则脚本执行审查;对于标准原理图规则,在进入每个规则脚本之前,执行器在控制脚本中先通过扩展命令从文件恢复规则所用的标准原理图通用数据结构。

审查过程如图 2 所示:

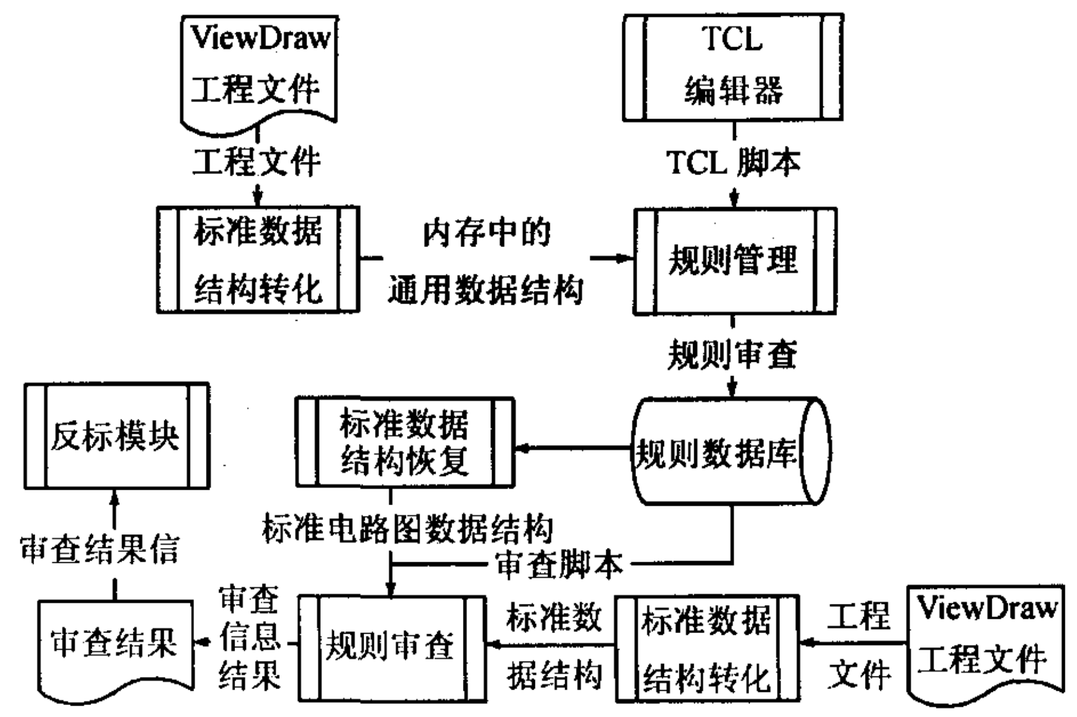

图 2　原理图审查过程示意图

5　扩展命令

在了解了原理图审查平台的系统结构和工作流程后,下面介绍基于原理图通用数据结构扩展的 TCL 命令。规则脚本就是由这些扩展命令构成。

在制订的扩展命令的过程中,遵循如下原则:

(1) 命令格式符合 TCL 语言的通用语法;

(2) 使用 MFC 和 C++实现扩展命令,并力求使循环查找、分析判断等工作在扩展命令中完成,以提高效率。

根据扩展命令不同用途,我们大致实现了以下几类扩展命令:

(1) 对象查找类扩展命令　该类型命令包括:(a) 在整个原理图设计中查找器件、网络或管脚;(b) 在某一设计页中查找器件、网络、总线或管脚;(c) 通过器件查找网络或管脚;(d) 通过网络查找器件、总线或管脚;(e) 通过管脚查找器件或网络;…。

(2) 属性获取类扩展命令　该类型命令包括获取设计页、器件、网络、总线或管脚的 LABEL、VALUE 等各种属性的属性值。

(3) 几何属性类扩展命令　该类型命令用于获取对象的位置、尺寸、方位等几何特性。

(4) 比较匹配类扩展命令　该类型命令检查一些连接特性如相连、串通、上拉、下拉等等特性。

(5) 标准电路描述类扩展命令　该类型命令描述标准原理图规则的各种限制,如指定核心器件、边界网络等等。

(6) 审查结果输出类扩展命令　该类型命令只有一条,用于报告审查信息。

(7) 其它类型扩展命令　包括其它所有命令。

6 审查示例

最后，我们举一个审查的例子来说明原理图审查平台的工作情况。下面是一条通用规则，检查设计工程中的所有器件是否由名称标识，报告设有名称标识的器件，并统计个数。规则脚本如图 3 所示，其中有下划线的的是扩展命令：

```
Report "info" " " "检查所有器件名称标识规则开始"
set count 0
GetCompFirst compId
while { $compId != " " } {
        incr count
        #取得 Device 属性
        GetCompProperty $compId DEVICE deviceVal
        #判断属性值是否为空
        if { [string equal $deviceVal " "] == 1 }  {
            Report  "error"  "$compId"   " 器 件 $compId  的
Device 属性为空"
        }
        #取得下一个器件 ID
        GetCompNext compId
}
set result [concat "审查" $count "个器件"]
Report "info" " " "检查所有器件名称标识规则结束 $result"
```

图 3 规则脚本示例

执行该规则得到的结果如表 1。

7 结束语

在本文中，我们描述了自行开发的基于 TCL 语言的原理图自动审查平台，介绍了系统的基本构成和原理图审查原理。另外，我们还就审查的核心——TCL 扩展命令做了详尽的介绍，并在本文最后给出了一个实际审查规则示例。从本系统的使用情况来看，使用扩展 TCL 语言作为审查脚本可以高效、准确地完成原理图的自动审查工作。

表 1 规则审查结果报告

1	系统信息		用户开始审查... Project = s2catop TopPage = s2catop	2003, 22 March, 19:06
2	错误	$1I95\$5I65	器件$1I95\$5I65 的 Device 属性为空	2003, 22 March, 19:06
3	错误	$1I49\$1I509	器件$1I49\$1I509 的 Device 属性为空	2003, 22 March, 19:06
4	错误	$1I49\$1I374	器件$1I49\$1I374 的 Device 属性为空	2003, 22 March, 19:06
5	系统信息		审查结束： 通过 = 0, 警告 = 0,错误 = 3, 提示 = 0	2003, 22 March, 19:06

参 考 文 献

[1] 蔡固顺，饶勇，李玉山. 电子自动化设计数据的标准格式问题. 电子科技, 1998, (6): 18 – 21.

[2] John Ousterhout. Tcl and the Tk Toolkit. Boston, MA: Addison-Wesley Publishing Company, 1993: 2 – 389.

田 野： 男，1977 年生，硕士生，研究方向：通信软件和协议工程.

赵保华： 男，1947 年生，教授，研究方向：软件工程、通信软件和协议工程.

屈玉贵： 女，1945 年生，教授，研究方向：计算机体系结构、通信软件和协议工程.