Z 规格说明自动生成器①

赵正旭[1], 温晋杰[2]

[1](石家庄铁道大学, 石家庄 050043)
[2](石家庄铁道大学 信息科学与技术学院, 石家庄 050043)

摘 要: 形式化 Z 语言采用严格的数学理论可以有效提高软件的可靠性和鲁棒性, 但是由于其包含的数学理论使得只有少数人能够熟练应用Z语言进行形式化规格说明书的编写. 目前, 多数对于Z语言的研究集中在理论阶段, 还没有相应的工具支持 Z 规格说明的自动生成. 本文中对于 Z 规格说明自动生成器的研究有助于降低 Z 规格说明书的编写难度, 降低了形式化开发的难度及成本, 对于形式化 Z 语言的推广具有重要的意义.

关键词: Z 语言; 形式化; 自动生成器; 规约; 语义分析

Z Specification Automatic Generator

ZHAO Zheng-Xu[1], WEN Jin-Jie[2]

[1](Shijiazhuang Tiedao University, Shijiazhuang 050043, China)
[2](School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China)

Abstract: The formalized Z language can improve the reliability and robustness of the software via using complex mathematical theories. However, only a few people can understand these theories and compile with Z specification. At present, the main research of Z language focuses on the theoretical research. There is no corresponding tools support the automatic generation of Z specification. The research of Z specification automatic generator introduced in this article can help with the compilation of the Z specification and cut the cost of formal development. This automatic generator has great significance for the large-scale promotion of the Z language.

Key words: Z language; formalization; automatic generator; specification; semantic analysis

形式化 Z 语言是一种基于一阶谓词逻辑和集合论的规格说明语言[1]. 可以进行计算机硬件以及软件系统的描述与验证, 尤其适合于极重大安全性系统, 如航空航天项目. 其基本思想是利用一些已知特性的数学抽象来为目标软件系统的状态特征和行为特征构造模型[2]. 在需求规格说明中, Z 语言精确的描述软件系统“做什么”而不涉及“怎么做”, 只对目标软件系统进行功能描述. 通过明确定义状态和操作来建立一个系统模型(使系统从一个状态转换到另一个状态), 具有其他描述方法不可比拟的严谨性、清晰性和无二义性[2]. 但是, 从国内的现状来看, 形式化 Z 语言的应用还有待于进一步的推广和深入, 降低形式化开发的成本是一个重要前提. 利用形式化方法 Z 语言软件开发的成本高居不下的一个重要原因就在于 Z 语言需求规格说明书的撰写环节. Z 需求规格说明书的撰写要求撰写者对数学理论熟练掌握, 包括数据结构、数学抽象思维、数学建模等都有一定的要求, 如果对 Z 语言没有任何的了解或者接触, 想要撰写一份合格的 Z 规格说明书既消耗时间又消耗物力财力, 具有相当高的难度, 很大程度上影响了 Z 语言的推广. 主要有以下两种原因:

(1)Z 语言包含的数学符号和逻辑操作对于软件工程领域的技术人员来说, 是及其陌生和难以理解的. 学习和使用 Z 语言是一个十分困难的过程.

(2)软件设计师和软件工程师对形式化方法的作用有不同的认识, 而对 Z 语言形式化描述中的数学理论缺乏兴趣.

所以, 一套具有良好用户界面、易于学习和操作

① 收稿时间:2015-08-06;收到修改稿时间:2015-10-19

简单的 Z 语言支撑工具, 对于形式化 Z 语言的推广应用是大有裨益的. Z 语言作为目前使用最为广泛的形式化描述语言之一, 它有以下几个主要特点:

首先, Z 语言不是计算机程序编制工具或编程系统. Z 语言是设计规范, 采用了包括集合、序列、包、关系、函数、类型、对象等抽象的数学理论. 所以, Z 语言是一种数学语言. 其次, Z 语言形成的规格说明的形式不是完全类似于 ASCII 码的文字和字符串, 而是包括了规范化的数学符号和演算图形. Z 语言形成的规格说明的内容不是能编译和运行的程序编码, 而是用来进行逻辑推理和数学演算的. 另外, Z 语言没有完全使用数学符号来形成规格说明, 它也使用了自然语言来定义变量和添加批注. 所以, 由 Z 语言生成的规格说明易于读写和解析. 最后, 由 Z 语言所形成的规格说明不仅严谨、清晰、无二义, 而且可以通过形式化方法软件对其进行验证和推理.

1 主要的Z语言编辑工具

一直以来, 软件开发者都期望研发出的软件具有较高的可行性和鲁棒性, 形式化方法使用适当的数学分析以提高设计的可靠性和鲁棒性. 但是, 由于采用形式化方法的成本高意味着它们通常只用于开发注重安全性的高度整合的系统. 所以, 对 Z 语言辅助工具的研究以降低形式化开发的成本就成为一个研究的热点. 针对 Z 语言的辅助工具有很多种, 但是大多数都不支持于 2002 年颁布的 Z 语言 ISO 标准, 关于 Z 语言的工具在文献报告中并不多见, 主流的 Z 语言支撑工具可以分为 Z 独立系统、Z 接口模块、Z 集成插件. Z 独立系统中具有代表性的为“Z\EVES”和 “Community Z Tools”简称“CZT”; Z 接口模块中具有代表性的工具为“Z2HTML”; Z 集成插件中具有代表意义的为“Z Word Tools”; 接下来, 重点介绍一下使用最为广泛的“Z Word Tools”.

“Z Word Tools”是基于 Microsoft Office Word 的 Z 语言的插件, 在计算机上安装“Z Word Tools”后, 可以在 Microsoft Office Word 中进行 Z 语言的编辑、类型检查、文档结构检查等工作, 输出的是 word 文档, 只有自己的计算机上面装有“Z Word Tools”才可以正确地查看该文档. “Z Word Tools”具体功能包括:

(1)向微软 Office Word 提供了了一个 Z 语言的 Unicode 字形库[16];

(2)具有 Z 语言的“所见即所得”风格的人机交互编辑界面并集成在微软 Office Word 系统中;

(3) 拼写检查使用了模糊的 Spivey Z[17]和 ISO 标准 Z[18,19]以及 CZT 功能;

(4)可以在 Z 语言规格说明中设置索引和交叉索引;

(5)为 Z 语言规格说明绘制与显示结构示意图.

通过集成插件系统来实现 Z 语言形式描述克服了上述独立系统的困难, 因此提高了 Z 语言形式描述的规格说明的兼容性, 从而扩展了 Z 语言的应用范围.

图 1 Z Word Tools 菜单栏

严格来讲, 上述主流的 Z 语言辅助工具属于 Z 语言的编辑器, 本文将要介绍的是 Z 规格说明自动生成器. 与上述 Z 语言辅助工具相比, 我们的 Z 规格说明自动生成器具有以下五个特点:

(1)Z 规格说明自动生成器完全是自主研发. Z 规格说明自动生成器的菜单栏、提示信息、工具栏均为汉语, 它们通俗易懂并且便于软件工程和技术人员学习和使用 Z 语言.

(2)Z 规格说明自动生成器是一个独立系统, 它不需要和任何其它应用程序集成使用.

(3)Z 规格说明自动生成器的每一步输入过程都非常简单. 它是通过人机交互的图形界面并且按有序的步骤进行选项式输入. 所以, Z 规格说明自动生成器的操作方便并且 Z 语言形式化的描述的过程容易被软件工程和技术人员所理解.

(4)Z 规格说明自动生成器改变了 Z 语言的使用方法. 它不需要软件工程和技术人员熟悉和理解 Z 语言的抽象演算以及基础概念和数学理论, 它只需要软件工程和技术人员理解状态变量的类型, 例如哪个是属于全局变量, 哪个是属于输出变量, 哪个是属于输入变量.

(5)由 Z 规格说明自动生成器定义并生成的 Z 模式与标准的 Z 模式一致, 并且以图像文件存储, 便于传播.

Z 语言实际上是一种数学表达规范，而 Z 规范中的基础理论和概念对于软件工程领域的技术人员来说，是极其陌生和难以理解的. 学习和使用 Z 语言是一个十分困难的过程，这可能就是 Z 语言没有得到广泛使用和如期发挥它作用的主要原因. 本节介绍了 Z 语言相关的一些常用辅助工具，从这些系统的结构方面论述了 Z 语言的使用方法. 这些实用方法无疑是今后研究和探讨如何进一步推广和使用 Z 语言的关键和焦点.

目前，几乎所有的与 Z 语言相关的形式化描述系统都是注重于 Z 语言的撰写、编辑、检查、验证等过程. 这些系统并没有使软件工程和技术人员从 Z 语言的基础概念和数学理论中完全解脱出来. 在今后的实际应用中，Z 语言应该侧重于方便易懂的用户界面、易于学习和操作简单的形式化方法的辅助工具. 在下文将介绍一个自己编码完成的 Z 规格说明自动生成器. 希望这个自动生成器能够为软件工程和技术人员屏蔽 Z 语言的基础概念和数学理论，帮助他们顺利完成 Z 语言形式化描述的规格说明的撰写任务. 这对促进 Z 语言的应用推广十分重要.

2　设计与实现

Z 规格说明自动生成器的设计宗旨是要面向所有软件过程参与者的，并不只是面向 Z 语言的学习者. 每一名用户利用该自动生成器都可以很方便地生成一份垂直模式的 Z 规格说明.

Z 规格说明自动生成器的界面有两个区域，即输入区域和显示区域，输入区域又分为状态变量的输入区和变量之间操作关系的输入区域. 如图 2 所示.

图 2　Z 规格说明自动生成器首页

在状态变量的输入区内可以输入的内容有以下几个类型.

模式类型: Z 语言的模式一共有四种类型. 它们分别为初始化模式、状态模式、操作模式、报错模式. 当我们点击下拉框时，下拉框里便会显示出这四种模式类型，以供我们选择. 如图 3 所示.

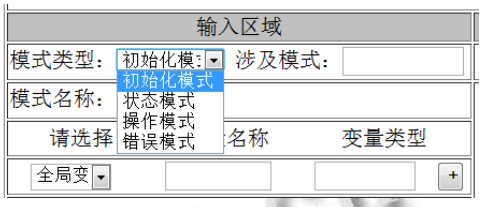

图 3　模式类型示意图

涉及模式: 涉及模式也是关联模式，它表示的是当前模式中所要包含的模式. 例如，假如我们要描述一个软件系统的密码修改操作，我们必须要求该用户成功登录该系统，所以描述修改密码的 Z 模式中就要包含描述成功登录系统的 Z 模式. 描述成功登录系统的 Z 模式就是描述修改密码的 Z 模式的涉及模式.

模式名称: 用于输入用户自行定义的模式名称.

变量种类: 主要是为了区分所定义的变量是属于哪一种变量. 我们一共有三种变量，即全局变量、输出变量、输入变量. 如图 4 所示.

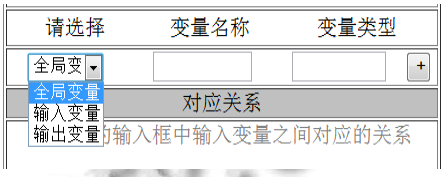

图 4　变量种类示意图

变量名称: 选择变量种类之后，我们在可编辑输入框中输入相应的自定义变量名称.

变量类型: 输入变量名称所属于的类型，可以自定义变量类型. 比如，数字 2 属于自然数 N.

在变量类型一栏的后面有一个“+”按钮，我们可以根据模式中变量的数目来动态地添加“变量种类”，“变量名称”，“变量类型”的输入框.

操作关系就是上面所输入的状态变量之间的对应关系. 因为 Z 语言采用了非 ASCII 符号来表示变量之间关系，所以自动生成器采用了 Unicode 编码符号来处理和显示这些特殊的 Z 语言符号. 我们点击输入区域的下拉框之后，自动生成器就会显示出一系列的表示集合与集合或者是变量与变量之间关系的特殊符号含义以供我们选择.

显示区域要根据输入区域所输入的信息显示相应的 Z 模式框.

在 Z 规格说明自动生成器具体实施的过程中, 主要遇到以下三个问题:

①Z 模式框的显示

Z 模式框是 Z 规格说明中的基本结构, 首先面临的问题就是如何以图形化的方式显示 Z 模式框. 经过分析, Z 模式框是由水平线与竖直线组成, 在构造 Z 模式框的时候, 可以利用水平线和垂直线拼接而成. 显示 Z 模式框水平线和垂直线的主要代码如下所示:

```
<p class="ZSchTop">
        <u><span lang="EN-US"><span
style="mso-tab-count:1">    
        </span></span></u><i
style="mso-bidi-font-style:normal"><span
lang="EN-US">
        SchemaName</span></i><u><span
            lang="EN-US"><span
style="mso-tab-count:1">    &
nbsp;
        </span><o:p></o:p></span></u>
</p>
<div
style="mso-element:para-border-div;border:none;border-l
eft:solid windowtext 1.0pt;
mso-border-left-alt:solid windowtext .5pt;padding:0cm
0cm 0cm 4.0pt;margin-left:
36.0pt;margin-right:-144.0pt">
<p class="ZMid">
        <span
lang="EN-US"><o:p> </o:p></span></p>
</div>
<div
style="mso-element:para-border-div;border:none;border-l
eft:solid windowtext 1.0pt;
mso-border-left-alt:solid windowtext .5pt;padding:0cm
0cm 0cm 4.0pt;margin-left:
36.0pt;margin-right:-144.0pt">
    <p class="ZSchBottom">
 <span lang="EN-US"><o:p> </o:p></span></p>
    </div>
```

②显示区域的保存

显示区域显示最终生成的 Z 规约, 为了降低保存的难度, 采取的方案是将输出结果显示在 panel 控件上, panel 控件主要是作为其他控件的容器, 我们可以根据需求设置其的可见性. 将 panel 区域的内容保存为图片时, 需要获取 panel 控件的源点坐标以及 panel 区域的长宽通过屏幕捕捉函数截取屏幕按照指定的文件路径保存到相应的文件夹当中. 核心代码如下所示:

```
private void button_Click(object sender, EventArgs e){
  Rectangle rect = new Rectangle(0,0,panel1.Width, pan
el1.Height);
  using( Bitmap bmp = new Bitmap(rect.Width, rect.Hei
ght) ){
    this.panel1.DrawToBitmap(bmp, rect);
    bmp.Save(@"c:\temp.jpg");
  }
}
```

我们把自动生成器的输出采用.jpg 图像文件来保存, 这是因为三个方面的因素. 首先是为了避免对 Z 语言的特殊符号和非 ASCII 符号的预处理. 其次是因为.jpg 图像文件的读取和显示不需要在计算机上安装专用 Z 语言形式描述软件工具, 如 CZT 或微软 Office Word 与 Z Word Tools 的集成辅助工具. 最后是考虑.jpg 图像文件便于网络传输并且可以方便地嵌入网页和常用的文字文档之中. 但是, 输出的.jpg 图像文件不能支持 Z 语言规格说明的自动检查, 也不能支持软件工程中对软件设计的自动检验和验证.

③特殊符号的显示

Z 语言中包含为数众多的特殊符号, 经过仔细的考察, 至今, 没有任何一种字体能够将 Z 语言中的特殊符号全部包含其中, 所以, 我们采用 Unicode 编码来实现. Unicode 码是一种国际标准编码, 扩展自 ASCII 字元集. 电脑上普遍使用的每字元有 8 位元宽; 而 Unicode 使用全 16 位元字元集. 这使得 Unicode 能够表示世界上所有的书写语言中可能用于电脑通讯的字元、象形文字和其他符号.

为了更进一步简化该生成器的使用, 在前台界面不会出现 Z 语言中繁杂的特殊符号, 出现的均为特殊符号对应的具体含义, 例如“∧”对应为“合取”、“?”对应为“空集”等. 用户选择特殊符号对应的含义后, 我们就可以判断用户选择的是哪个特殊符号, 然后在 Z

规格说明中的相应位置插入该符号对应的 Unicode 码, 然后根据 Unicode 编码将该特殊符号的符号形状显示在 panel 区域的相应位置[3]. 如图 5 所示.

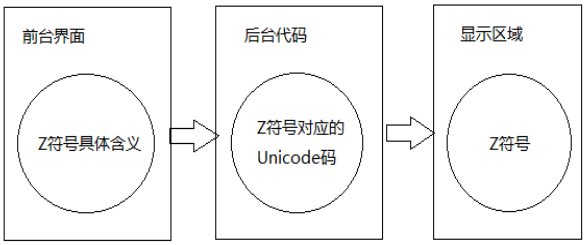

图 5　显示原理示意图

表 1 是 Z 语言中部分特殊符号与 Unicode 编码以及特殊符号与其具体含义三者的相互映射关系, 由于篇幅有限, 没有将所有特殊符号与 Unicode 码的映射给出.

表 1　特殊符号 Unicode 编码表

语义	Z 符号	Unicode 编码
合取	∧	\u2227
析取	∨	\u2228
否定	¬	\u00AC
属于	∈	\u2208
并集	∪	\u222A
交集	∩	\u2229
等价	⇔	\u21D4
蕴含	⇒	\u21D2
真包含	⊂	\u2282
不属于	∉	\u2209
类属于	:	\u003A
整数集	ℤ	\u2124
关系逆	~	\u223C
全函数	→	\u2192
不等于	≠	\u2260
差集	\	\u2216
抽取	↾	\u21BF
过滤	↾	\u21BE
空集	∅	\u2205
自然数集	ℕ	\u2115
队列级联	⌢	\u2040
定义域限定	◁	\u25C1
全入射函数	↦	\u21A3
全满射函数	⇻	\u21A0

另外, 为了规范 Z 规格说明的生成, 还需要基本定义以下规则:

规则 1.　在模式类型下拉框中, 用户选择初始化模式时, 不包含任何已定义的状态模式与操作模式, 所以涉及模式为空; 用户选择模式类型为错误模式时, 由于, 错误模式描述操作操作错误时的状态量的变化及对应关系, 操作失败不引起状态量的变化, 对其所涉及模式进行修饰的时候选择符号“Ξ”描述, 其余情况均会引起状态量的变化, 选择符号“Δ”来描述; 其中, ΔSys≙ [Sys, Sys′]和 ΞSys≙ [ΔSys|θSys=θSys′]

规则 2.　用户在变量种类下拉框选着为输入变量时, 在显示区域后面紧跟“?”进行修饰; 如果选择,输出变量时, 在显示区域后面紧跟“!”进行修饰; 选择为全局变量时, 如果该全局变量表示的是一个集合, 则其所属类型前面加“IP”, 表示是该集合幂集的一个子集; 如果该状态变量表示的是一个有限集合, 则其所属类型前面加“IF”, 表示是该集合幂集的一个有限子集; 否则不进行修饰.

上述两条规则针对用户的不同选择而定义, 基本能够完成简单的 Z 规格说明的生成. 但是, 本规格说明自动生成器尚处于研发应用的初期, 考虑还不够周全, 定义的规则也比较粗糙, 功能还有待于进一步的增加完善.

3　实例研究与工具演示

手机是我们日常生活中必不可少的通讯工具, 本节选择联系人相关操作进行 Z 语言的描述与验证.

首先我们要确定手机通讯录要涉及到的状态变量并对其命名. 通讯录中联系人姓名为 PhoneName、通讯录中联系人电话定义为 PhoneNumber, 定义姓名和电话号码的类型分别为 Name 与 Number. 接下来, 我们要定义操作之后系统要给出的提示信息. 由于操作之后系统要给出的提示信息为具体值, 且比较简单, 所以我们可以通过枚举法来定义, 枚举如下:

RESPONSE::= Success.

描述任何一个客观事物, 首先就是对其进行初始化, 包括对涉及到的状态变量、集合的初始化, 使用 Z 语言的插件“Z Word Tools”在 Microsoft Office Word 中编辑初始化模式如下.

为了验证本课题组开发的 Z 规格说明自动生成器的可行性, 利用 Z 规格说明自动生成器生成 Z 模式的时候, 选择、输入以及结果的输出显示如图 6 所示.

InitPhone

PhoneName : $\mathbb{P}$ *Name*
PhoneNumber : $\mathbb{P}$ *Number*
name ?: Name
number ?: Number
report !: RESPONSE

PhoneName = {}
PhoneNumber = {}

图 6 初始化模式生成界面

增加联系人操作会涉及到初始化的变量, 所以要涉及初始化模式 InitPhone, 增加联系人成功之后, 通讯录中联系人姓名为 PhoneName 会改变、通讯录中联系人电话 PhoneNumber 也会变为原有的号码的集合加上新输入的联系人电话, Z 模式描述如下:

Addsuccess

Δ*InitPhone*
name ?: Name
number ?: Number
*_ name number_*: *Name*↔ *Number*
report !: RESPONSE

PhoneName =dom *name number*
PhoneNumber =ran *name number*
name number′= *name number*∪{ *name*? ↔ *number*?}
report !=Success

利用 Z 规格说明自动生成器生成 Z 模式的时候, 选择以及输入的数据如图 7 所示:

图 7 增加联系人输出界面

当我们要删除联系人时, 我们一般按照这样的操作流程进行, 即输入要删除的联系人的姓名, 然后将对应的联系人信息及其电话号码一起删除. 在这个操作中, 姓名与电话号码满足二元对应关系. 借助于这种对应关系, 我们可以方便地通过所输入的姓名来找到相对应的电话号码. 要删除联系人的操作的 Z 语言描述如下:

Deletesuccess

Δ*Addsuccess*
name ?: Name
*_ name number_*: *Name*↔ *Number*
report !: RESPONSE

---

*d*om *name number*= *PhoneName*
ran *name number*= *PhoneNumber*
*_ name number_*′= {*PhoneName*↔ *PhoneNumber*}\
{ *name*? ↔ *number*?}
report !=Success

我们利用 Z 规格说明自动生成器所生成的 Z 模式 Deletesuccess 以及所做的选择和输入的数据如图 8 所示.

Z 规格说明自动生成器的显示区域所显示的输出结果如图 9 所示. 当我们完成要删除联系人操作的描述时, 通过点击保存按钮, 模式 Deletesuccess 以图像格式保存为磁盘文件.

图 8　删除联系人输入界面

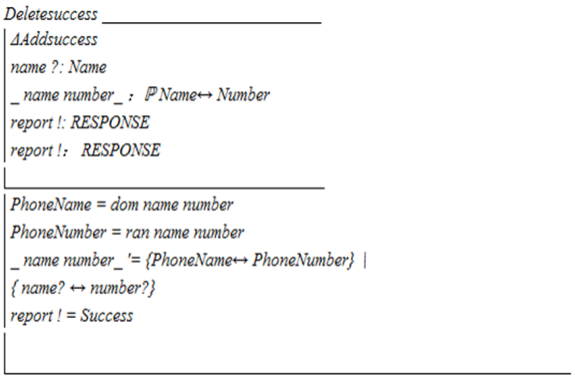

图 9　删除联系人生成图

修改联系人的操作一般是针对该联系人的电话号码的修改. 描述这个与前面描述的删除联系人的操作类似, 它们都是借助于联系人的姓名与电话号码之间的对应关系进行的. 我们通过输入联系人的姓名来找到相对应的电话号码, 然后来进行修改操作. 描述这个操作的 Z 模式如下:

Modifysucces

ΔAddsuccess
name?:Name
number?:Number
_ name number_: Name↔ Number
report !: RESPONSE

---

dom name number= PhoneName
ran name number= PhoneNumber
_ name number_'= {PhoneName↔ PhoneNumber}⊕
{ name? ↔ number?}
report !=Success

图 10 所示的截屏是我们利用 Z 规格说明自动生成器生成的描述修改联系人的操作的 Z 模式 Modifysuccess 的选择以及输入的数据.

图 11 所显示是 Z 规格说明自动生成器输出区域显示并最终所生成的 Modifysuccess 模式以图像格式保存到磁盘文件里.

上述实例介绍和验证了 Z 规格说明自动生成器的功能性和可用性, 通过对一个实际应用程序的 Z 语言描述, 叙述了 Z 语言的一种新的使用方法. 借助于形式化描述的辅助工具, 使软件工程和技术人员能够在不熟悉 Z 语言的基础概念和数学理论的情况下使用 Z 语言来撰写正确的 Z 规格说明书. 实践证明, 该规格说明自动生成器可以有效地生成 BOX 风格的 Z 规格说明, 而且具有很好的可行性. 本项研究是对于 Z 规格说明的自动生成的一次有效尝试, 为 Z 规格说明自动生成器的研究起了带头作用.

图 10　修改联系人输入界面

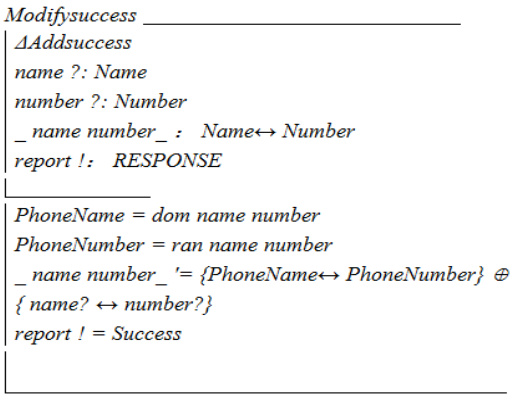

图 11　修改联系人生成图

该规格说明自动生成器可以成功生成小规模系统的 Z 规格说明书, 为了今后在大规模系统中进行实现和推广, 该 Z 规格说明自动生成器还需要做以下改进:

(1)Z 规格说明自动生成器最终输出的为.jpg 图像文件, 不支持 Z 语言规格说明的自动检查、自动检验和验证, 而且每个模式就是一个单独的图像文件, 如果是一个大型系统的 Z 规格说明的话, 就会产生大量的单独的文件, 查看起来耗费时间. 所以, 需要改进输出格式为 PDF 文件, 查看方便, 而且方便读取其中的内容进行形式化自动检查.

(2)在输入区域, 变量之间对应关系的输入框数量难以满足大型系统的需求, 下一步应该实现动态地添加.

4 结语

软件工程是门实用性的学科, 一个国家各个方面的发展离不开软件工程. 基于形式化语言的软件需求规格说明是软件工程学科的大趋势, 与国外软件工程形式化水平相比, 国内软件工程形式化实践任重而道远. 随着形式化方法研究的不断深入, 形式化规格说明技术将会得到更加广泛的应用. 本文主要对自主研发的 Z 规格说明自动生成器进行了简单的介绍, 并对其进行了应用测试, 通过验证表明该自动生成器可以成功地完成小规模系统的 Z 语言描述, 希望本项研究能够为 Z 语言的推广做出贡献. 在本项研究的基础上, 还可以在通过语义分析来生成规格说明、自动生成测试用例等各方面进行进一步的研究.

参考文献

1 温晋杰,赵正旭.OpenGL 图形规范的 Z 形式化描述.河北省科学院学报,2014,31(2):41–48.

2 许庆国,缪淮扣,曹晓夏.Object-Z 规格说明测试用例的自动生成器.软件学报,2011,22(6):1155–1168.

3 赵正旭,温晋杰,赵卫华.Z 规范及其使用方法.北京:科学出版社,2015.

4 羊东昭,缪淮扣.Object-Z 编辑器的分析、设计和实现[硕士学位论文].上海:上海大学,2003.

5 赵晓峰,赵正旭.虚拟制造环境的信息规范及其 Z 描述研究[学位论文].济南:山东大学, 2010.

6 赵晓峰, 赵正旭. 基于 Z 的虚拟加工仿真环境规范技术研究.系统仿真学报,2009,21(22):7143–7146.

7 刘贾贾. 基于小世界网络的数据格式转换研究及 Z 语言描述[学位论文].石家庄:石家庄铁道大学,2011.

8 吴方君,徐升华.用 Z 形式化描述程序切片.小型微型计算机系统,2007,28(8):1444–1447.

9 缪淮扣,李刚.软件工程语言—Z.上海:上海科学技术文献出版社,1999.

10 刘海洋.LATEX 入门.北京:电子工业出版社,2013.

11 李莹,吴江琴.软件工程形式化方法与语言.杭州:浙江大学出版社,2010.

12 古天龙,常亮.离散数学.北京:清华大学出版社,2012.

13 International Organization for Standardization. Information Technology—Z Formal Specification Notation—Syntax, Type System and Semantics. ISO/IEC 13568:2002/Cor.1: 2007, C/SC: ISO/IEC JTC 1/SC 22. 2007

14 Martin AP. Proposal: Community Z Tools Project (CZT). Computing Laboratory Oxford University. Sept. 2001.

15 Hall A. 2014, Community Z Tools Project (CZT): Tools for Editing, Type checking and Animating Z specifications and Related Notations. SourceForge.

16 Jacky J. Z2HTML translator. Department of Radiation Oncology, Box 356043, University of Washington/Seattle, Washington 98195-6043 / USA. 2015.

17 The Unicode Consortium, 2006, The Unicode Standard, Version 5.0 (5th Edition), Addison-Wesley Professional. ISBN 0321480910.

18 Spivey JM. The Z Notation: A Reference Manual. Second edition, Prentice Hall International (UK) Ltd. 1992.

19 International Organization for Standardization. Information Technology—Z Formal Specification Notation—Syntax, Type System and Semantics. ISO/IEC 13568:2002, TC/SC: ISO/IEC JTC 1/SC 22. 2002.