维普资讯 http://www.cqvip.com

Object-Z 规格说明的 SQL 动画模拟*)

周 静 缪淮扣

(上海大学计算机工程与科学学院 上海 200072)

摘 要 软件规格说明的确认在软件开发阶段占有举足轻重的地位。形式规格说明的动画模拟技术是一种规格说明的确认方法。本文研究了 Object-Z 规格说明的 SQL 动画模拟方法,设计了从 Object-Z 到 SQL 的转换规则,并提出了模块封装的思想,即用存储过程表示类、对象和模式等模块,用户通过调用执行存储过程确认规格说明是否满足其需求。

关键词 Object-Z,规格说明,确认,SQL,动画模拟

The Animation of Object-Z Specification in SQL

ZHOU Jing MIAO Huai-Kou

(College of Computer Engineering and Science, Shanghai University, Shanghai 200072)

Abstract The validation of the software specification holds the balance during the software development. The animation technology of the formal specification is a kind of the validation methods. The paper studies the animation of Object-Z specification in SQL, and designs the rules of the transformation from Object-Z to SQL, it advances the idea of the module encapsulation as well, namely denotes class, object and schema etc. modules with storage procedure, the user can validate whether the specification satisfy their requirement through executing the storage procedures.

Keywords Object-Z, Specification, Validation, SQL, Animation

1 引言

随着信息技术的不断发展,计算机已广泛应用于社会的各个领域,人们愈来愈重视软件的质量。大量软件项目开发的事实表明:大部分的软件开发费用是用于纠正在测试阶段发现的各种错误,而这些错误中的很大一部分是由需求分析阶段对规格说明描述不精确引起的。为了克服采用自然语言和程序设计语言描述规格说明的缺陷,人们提出了形式化方法。形式化方法采用形式规格说明语言来描述和验证软件系统。Z 语言是一种以一阶谓词逻辑和集合论为基础的形式规格说明语言。Object-Z 是基于模型的面向对象的形式规格说明语言,它对 Z 进行了面向对象的扩充。

虽然形式化方法有很多的优点,但由于规格说明比较抽象,需要有较强的数学基础,不少开发人员感到不易理解。规格说明一般是不可执行的。如果用户和领域专家对形式规格说明方面的知识了解甚少,就很难理解形式规格说明,也就难以确定规格说明是否与他们的需求相一致。为了清除形式规格说明中的不一致性和模糊性,必须对形式规格说明进行确认(Validation)和验证(Verification)。所谓规格说明的确认就是对描述软件需求的规格说明进行检查,根据检查的反馈结果,确定其所描述的系统功能是否满足最终用户的需求。本文所研究的基于 Object-Z 规格说明的 SQL 动画模拟就是一种规格说明的确认方法。

2 规格说明的动画模拟技术

规格说明的动画模拟就是将形式化规格说明转化成一个可执行但不丢失规格说明所提供的语义的程序,通过执行这个转化后的程序来检查规格说明所描述的软件系统功能的实现情况,以此确定所描述的规格说明是否与用户需求相一致。动画模拟的过程如图 1 所示。

目前,国内外已有一些相关的研究工作,已有几种不同的方法将规格说明转换成用 Prolog、Mercury、Haskell、Java 等语言所编写的可执行程序。针对不同的动画模拟策略已开发出相应的动画模拟工具。例如,

BZ-TT[9]:实现对 B 规格说明的动画模拟,它从初始状态开始模拟操作序列的执行,显示不确定性操作的解决方案,是一种典型的基于值的动画模拟工具。

ProB[10]:一种 B 模型检查器,它是使用 SICStus Prolog 的 Tck/TK 库进行开发,并在其早期 CSP 动画模拟工具的基础上加以构建。

PiZA[11]:一种用 Prolog 实现的 Z 规格说明动画模拟工具,它允许 Prolog 声明嵌入到规格说明中。

Possum[12]:Queensland 大学的 SVRC 小组开发的一种 Sum 规格说明动画工具,亦可以模拟 Z 规格说明。

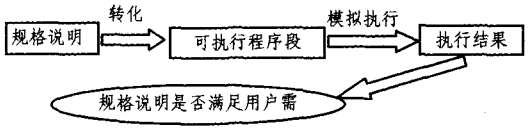

图 1 动画模拟过程

对于 Object-Z,Queensland 大学的 SVRC 小组对该规格说明语言提出了一种动画模拟方法。这种方法的主要思想是在 Z 中建立一个框架来管理对象和对象引用的动态实例,并将所有来自 Object-Z 规格说明的操作转换成 Z 的操作,然后利用 Z 的动画工具 Possum 动画模拟转化后的规格说明。

本课题组开发了一个 Object-Z 的动画模拟工具——OZ-

*)本文的研究工作受国家自然科学基金(60373072)和上海市教委科技发展基金资助。周 静 硕士研究生,研究方向为形式化方法;缪淮扣 教授,博导,研究方向为软件工程、形式化方法。

维普资讯 http://www.cqvip.com

Animator[13]，它利用 Prolog 的谓词库实现 Object-Z 的集合理论、对象特征以及各种复杂的操作复合的动画模拟。

3 Object-Z 的 SQL 动画模拟方法

规格说明的 SQL 动画模拟，即用对应的 SQL 语句表示规格说明语句，执行转化后的 SQL 语句，并将执行结果以清楚易懂的方式显示出来，用户以此判断规格说明是否与他们的需求一致。国外的 Matthew Love 提出用数据表表示数据结构，将谓词进行分组处理的方法，用 Oracle 工具 SQL * Forms3.0 作为动画模拟的环境，实现 Z 规格说明的 SQL 动画模拟[4]。

目前大部分的软件开发都是面向对象的开发。本文在 Matthew Love 对 SQL 动画模拟 Z 规格说明研究的基础上，研究用 SQL 动画模拟 Object-Z 规格说明的方法，对部分基本谓词的转换进行了改进，尽可能地简化转换后的 SQL 语句。本文还提出了对类、对象和模式等模块进行封装的思想，针对每个模块建立一个 SQL 存储过程，通过调用与模块对应的存储过程动画模拟该模块。

在本文提出的使用 SQL 动画模拟 Object-Z 规格说明的方法中：1）利用 SQL 的存储过程进行模块封装，简化了动画模拟的过程；2）将谓词的映射关系和各基本类型变量的值存储在数据库中，使模拟框架和数据完全独立，提高了动画模拟的效率；3）Object-Z 是以一阶谓词逻辑和集合论为基础的，SQL 中的大部分操作也是针对集合的操作，因此两者之间的映射相对较为直观；4）当前数据库操作系统应用比较广泛，一般的机器上都装有数据库操作系统，因此动画模拟器的可移植性较高。

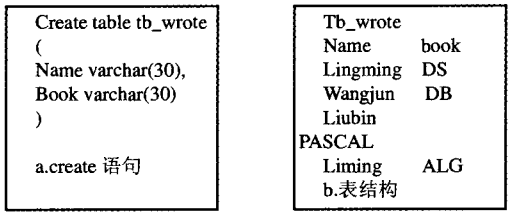

图 2 关系 wrote 的 create 语句和表结构

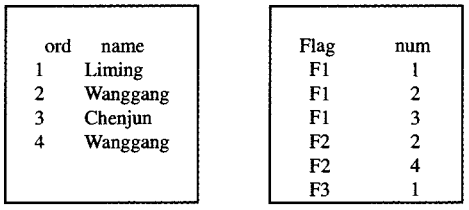

图 3 序列和幂集的表结构

3.1 基本数据结构

在本文的研究中，将数据类型分成两大类，即基本数据类型（整型、字符类型、关系类型等）和类（对象）类型。在动画模拟过程中，用数据表存储基本类型变量的值，对规格说明中每个基本数据类型的变量都建立一个数据表。表名用变量名表示，只有一个字段的数据表的字段名也用变量名表示，方便数据的存取。在将规格说明转化成 SQL 的过程中，每个状态变量（即列表中的变量）都有两个对应的数据表，用来分别存储前、后状态变量的值。第一个数据表存储前状态变量的值，第二个数据表则存储后状态变量的值，两个数据表的结构相同。后状态数据表的表名在前状态数据表的表名后加'1'，与表示后状态的修饰符"'"对应。一般简单类型的变量（整型、字符型等）用一个字段表示，字段名为这个表的表名。关系、函数等多元类型的变量根据这个类型变量的特点用多个字段表示。

关系（主要指二元关系）用含有两个字段的数据表表示，如关系：wrote=={Liming(DS, Wangjun(DB, LiuBin(PASCAL, Liming(ALG}，其对应的 create 语句和数据表如图 2。

函数是特殊的二元关系，对定义域中的任意元素，在值域中只有一个元素和它对应。因此，在建立与函数对应的数据表时，除了满足数据表具有两个字段外，还要使表示定义域的字段具有唯一性（unique）。函数可分为入射、满射和双射等。在建数据表时，按照各种函数的特点对字段进行约束，例如入射函数还要使表示值域的字段也具有唯一性。

序列是一种特殊的函数，它描述了带有次序的对象的集合。因此，序列用具有两个字段的数据表表示，定义域字段表示元素的次序，值域字段表示与定义域字段对应的元素。为了提高模拟的效率，在建数据表时可给这个字段加上 IDENTITY 属性。用户往表中依次插入序列的元素时，其对应的序号也会自动插入。如序列：⟨Liming, Wanggang, Chenjun, Wanggang⟩的表结构如图 3。

包也是一种特殊的函数，它描述了数据集合中每个元素出现的次数。与包对应的数据表的两个字段中，第一个字段表示集合的元素，第二个字段表示这个元素在集合中出现的次数。

幂集类型也用一个具有两个字段的数据表表示，第一个字段用来标识幂集中的每个对象，第二个字段表示幂集的每个对象中的元素。如幂集：{{1,2,3},{2,4},{1}}表示如图 3。

对于给定数据类型，可以先用 SQL 自定义这个数据类型，然后就可以将这种数据类型看成基本数据类型进行操作。

3.2 基类

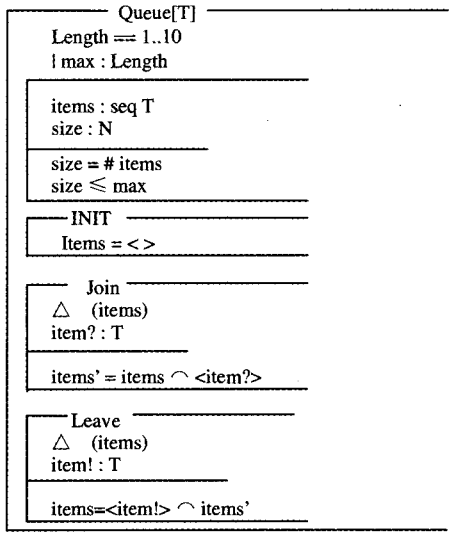

图 4 类 Queue

基类是 Object-Z 中最简单的类，在规格说明转化成 SQL 的过程中，针对基类中的初始状态模式和每个操作模式，都会建立一个对应的存储过程。对状态模式则会建立两个存储过程，分别表示前状态模式和后状态模式，其名称是以类名和模式名连接的字符串。前状态模式的存储过程的名称直接用类名加'_s'表示，后状态在前状态名称后加'1'。每个存储过程有一个字符串类型的参数（其默认值为空字符串），在类进

维普资讯 http://www.cqvip.com

行实例化时，用来标识同一类的不同对象。而存储过程中用到的每个可见变量名都加上这个参数的值形成新的名称，以方便对这个对象性质的引用。

图 4 中的类 Queue，其转化后的基本框架为：

```
前状态模式：
Create procedure Queue_s
  @obj varchar(20)=''
    As
      ……
    Go
```

```
后状态模式：
Create procedure
Queue_s1
@obj varchar(20)=''
    As
……
    Go
```

```
初始状态模式：
Create procedure Queue_Init
@obj varchar(20)=''
    As
……
    Go
```

```
Join 操作模式：
Create procedure Queue_Join
@obj varchar(20)=''
    As
……
    Go
```

```
Leave 操作模式：
Create procedure Queue_Leave
@obj varchar(20)=''
    As
……
    Go
```

3.3 对象

对象是类的实例，对对象进行转化，其实就是执行与该对象所属类对应的各存储过程，并把这个对象的名称传给存储过程。

例如有一个类 MsgDistributor(图 5)，在这个类中定义了类 CQueue(图 4)的两个对象 route1 和 route2。

```
MsgDistributor
route1, route2:  CQueue[MSG]
INIT
route1.INIT∧route2.INIT
Join1≡[route1.count-route2.count<10]
    ∧[route1.size<max]∧route1. Join
Join2≡[route2.count-route1.count<10]
    ∧[route2.size<max]∧route2. Join
Leave1≡[route1.size>0]  ∧route1.Leave
Leave2≡[route2.size>0]  ∧route2.Leave
```

图 5　类 MsgDistributor

在这个类的转化过程中，由于 route1 和 route2 是类 CQueue 的两个对象，因此对这两个对象的声明，其实就是创建与这两个对象对用的各个变量的数据表，如 route1_size，route1_count。对操作 route1. Leave 的执行其实是执行以下的 SQL 语句：Exec CQueue_Leave 'route1'。

3.4 类的继承

在规格说明中，如果存在继承结构，则把继承类展开，把继承类中所有可见的属性复制到子类中，把状态模式合并，把超类和子类中同名的操作模式合并。通过以上操作，这个父类就转化成了基类，然后按照基类的转化方法将其用 SQL 表示。

3.5 模式

在本文研究的 SQL 动画模拟中，模式是动画模拟的基本单位。模式动画模拟分为转化、调用及数据清理 3 个步骤。

3.5.1 转化

每个状态模式都建立两个存储过程，分别表示前、后状态模式。模式是由规格说明的谓词和包含的子模式(子模式指被其他模式调用的模式)组成的。子模式的转化就是执行与这个子模式对应的存储过程。谓词基本可分为两种类型：条件检查和赋值(赋值一定含后状态变量，或输出变量)。将 Object 规格说明转化成 SQL 语句时，首先需要找出两种语言的各操作符之间的映射关系，简化复杂操作符，使之尽量用简单操作符表示，然后按照各映射关系将规格说明转化为 SQL 语句。在转化中，条件检查语句用 SELECT 查询语句表示，以此检查这个谓词是否成立。对赋值谓词的转化则相对复杂，主要用 DELETE、INSERT 和 UPDATE 语句。整个模式的转化主要分为变量声明的转化和谓词转化两个步骤。

假设模式 AddEntry 是类 phone 中的一个操作模式，这个操作模式是往电话记录中插入一个新的 name? 和 newnumber? 记录，见图 6。

```
AddEntry
Δphone
Name? : Person
Newnumber? : Phone
Name? ∈ members
(Name? ↦ Newnumber? )∉telephonedir
telephonedir ' = telephonedir ∪ { Name ? → Newnumber? }
members' =members
```

图 6　模式 AddEntry

将操作模式 AddEntry 转化后的 SQL 语句为：

```
1)Create table in_name (in_name varchar(20))
2)Create table in_ Newnumber (in_ Newnumber int)
3)Create procedure Phone_AddEntry
@obj varchar(20)
  As
  · Select count( * ) from members
    Where name=(select in_name from in_ name)
  · Select count( * ) from telephonedir
    Where name=(select in_name from in_name)
    and num=(select in_Newnumber from in_Newnumber)
  · Delete from telephonedir1
  Insert into telephonedir1 select * from telephonedir
  Insert into telephonedir1 Select * from in_name cross join in_Newnumber
  · delete member1
    insert into member1 select * from members
Go
```

以上的 SQL 语句中，1)、2)表示这个操作模式中两个变量的声明，3)建立表示这个操作模式的存储过程。

3.5.2 调用

将规格说明语句转化为动画模拟语言 SQL 后，要执行转化后的 SQL 语句，建立相关数据表和存储过程，并调用已建立的存储过程来动画模拟规格说明。一个操作模式的动画模拟分为以下几个步骤：

(1)调用前状态模式的存储过程(检查前状态变量是否满足条件)；

(2)调用子模式的存储过程(存在子模式的情况)；

(3)调用执行表示这个操作模式的存储过程；

(4)调用后状态模式的存储过程(检查后状态变量是否满

(下转封四)

维普资讯 http://www.cqvip.com

(上接第 260 页)

足条件);

以上操作模式 AddEntry 的动画模拟通过调用执行存储过程 Phone_s、Phone_AddEntry 和 Phone_s1 实现。

3.5.3 数据清理

对每个操作模式进行动画模拟以后,需要进行扫尾工作,将前状态变量表中的数据更新为后状态表中的数据,并清空后状态变量表,进行新操作模式的动画模拟。

动画模拟操作模式 AddEntry 后的数据清理执行以下的 SQL 语句:

状态变量 telephonedir 的清理:

- Delete from telephonedir
- Insert into telephonedir
  Select * from telephonedir1

状态变量 member 的清理:

- Delete from members
- Insert into members
  Select * from members1

4 工具演示

我们根据以上提出的方法实现了一个动画模拟器,能基本实现简单的规格说明的动画模拟(图 7 略)。以图 4 中的类 Queue 为例,表 1 记录了这个动画模拟器的演示过程(输入 max 的值为 2)。

表 1 动画模拟器的演示过程

Operation	Item?	Item!	Items	Size	Message
Init	-	-	-	0	No error!
Leave	-	-	-	0	Error:items is empty!
Join	Red	-	Red	1	No error!
Join	Blue	-	Red、Blue	2	No error!
Leave	-	Red	Blue	1	No error!
Join	White	-	Blue、White	2	No error!
Join	Green	-	Blue、White	2	Error:size equals max!
Init	-	-	-	0	No error!

以上的演示步骤依次为:(1)队列初始化操作 Init;(2)出队列操作 Leave,队列为空,出错;(3)入队列操作 Join,将 Red 插入队列;(4)入队列操作 Join,将 Blue 插入队列;(5)出队列操作 Leave,Red 出队列;(6)入队列操作 Join,将 White 插入队列;(7)入队列操作 Join,企图将 Green 插入队列,队列元素达最大值,出错;(1)队列初始化操作 Init,队列清空。

总结 本文结合实例介绍了 Object-Z 规格说明的动画模拟方法,使用数据库查询语言 SQL 来动画模拟 Object-Z,用户可以通过简单直观的操作确认规格说明。在这种方法中我们提出,建立 Object-Z 和 SQL 之间操作符的映射关系,用数据表表示规格说明的基本数据结构,以模式为单位建立存储过程,使用存储过程的嵌套来包含模式或引用对象等。我们将进一步改进效率低的映射关系,提高动画模拟工具的工作效率。

参考文献

1 缪淮扣,李刚,朱关铭. 软件工程语言—Z. 上海:上海科技文献出版社,1999

2 Smith G. The Object-Z Specification Language. Kluwer Academic Publishers, 2000

3 Jia Xiao-Ping. An Approach to Animating Z Specification. In: Proc 19th Annual IEEE International Computer Software and Applications Conference (COMPSAC'95), Dallas, Texas, USA, August 1995. 108~113

4 Love W. Animating Z specifications in SQL * Forms 3.0, Z User Workshop. 1992. 294~306

5 McComb T, Smith G. Animation of Object-Z Using a Z Animator. Computer Society, 2003

6 West M M, Eaglestone B M. Software development: two approaches to animation of Z specifications using Prolog. Software Engineering Journal, 1992

7 Chiang Chia-Chu. Automated rapid prototyping of TUG specifications using Prolog. Information and Software Technology, 2004

8 Utting M. BZ-TT Animator Tutorial. 2003

9 Leuschel M, Butler M. ProB: A Model Checker for B. FME, 2003. 855~874

10 Hewitt M A, Halloran C M O, Sennett C T. Experiences with PiZA, an Animator for Z. ZUM, 1997. 37~51

11 Hazel D, Strooper P, Traynor O. Possum: An Animator for the SUM Specification Language. IEEE, 1997

12 朱江,陈怡海,缪淮扣. Object-Z 规格说明的结构模拟动画技术. 上海大学学报,2005,11(6):589~595

计算机科学

(1974 年 1 月创刊)

第 34 卷第 04 期 (月刊)

2007 年 4 月 25 日出版

国际标准连续出版物号 ISSN 1002-137X

国内统一连续物出版号 CN50—1075/TP

定价:30.00 元 国外定价:5 美元

邮发代号:78—68

发行范围:国内外公开

主管单位:国 家 科 学 技 术 部

主办单位:国家科技部西南信息中心

编辑出版:《计 算 机 科 学》 杂 志 社

重庆市北部新区洪湖西路 18 号 邮政编码:401121

电话:(023) 63500828 E-mail:jsjkx@swic.ac.cn

网址:www.jsjkx.com

社 长:牟炳林

总 编:彭 丹

主 编:朱宗元

主编助理:徐书令

印 刷 者:重 庆 科 情 印 务 有 限 公 司

总发行处:重 庆 市 邮 政 局

订 购 处:全 国 各 地 邮 政 局

国外总发行:中国国际图书贸易总公司 (北京 399 信箱)

国外代号:6210—MO