

特定领域建模的研究和实践

徐 锦 夏洁武 吴鸿兵 赵文耘 钱乐秋

(复旦大学计算机系 上海 200433)

摘 要 主要讨论分层系统的DSSA, 获得此领域的参照构架, 并用GenVoca模型进行描述, 最后通过在实际应用中的实例来讨论有关问题。

关键词 特定领域参照构架 GenVoca模型 构件 域

The Research and Practice of Domain-specific Modeling

Xu Jin Xia Jiewu Wu Hongbin Zhao Wenyun Qian Leqiu

(Department of Computer Science, Fudan University Shanghai 200433)

【Abstract】 This paper is mainly about the hierarchical software system, using GenVoca model to describe the architecture. The examples are presented in the paper.

【Key Words】 DSSA; GenVoca model; Component; Realm

软件重用是软件工程界的重要研究课题, 至今, 已有多种方法从不同的角度来解决这个问题。直到DSSA(Domain-Specific Software Architecture)方法的提出, 大规模重用才变得可能。一般的软件构架描述了特定应用中构件间的相互关系。而DSSA通过对某个特定领域的分析, 提出了该领域的典型需求, 这样, 此领域的一个特定需求就变成在典型需求上加上特殊需求, 然后配置合适的构件, 完成相应应用的开发。由于经过特定领域的分析, 设计者能够最大程度地提取此领域的可重用成分, 从而提高软件的生产率(参照构架、应用系统和其构架的关系图略)。

我们对分层系统软件构架的研究是基于DSSA的思想, 寻找一种适应层次系统构架描述的模型, 从而将把参照构架形式地描述出来, 提高系列应用的开发效率, 以求设计重用。

针对层次系统的特征和开发系列应用的要求, 描述这种参考构架的模型应满足: (1)能够定义原始的, 可重用的构件; (2)能够描述构件的组合, 解释不同组合的不同涵义; (3)构件应该满足可拆卸的特性, 以便通过调换构件来实现不同组合; (4)该模式能够用简洁的形式表示复杂的组合。

为适应上述需求, 我们找到了曾经用于描述ADAGE^[2]的参考构架的模型, GenVoca模型。

1 DSSA和GenVoca模型

GenVoca模型主要包括构件和域的概念。构件: 每个构件有其接口和实现, 接口是唯一的外部可见的。每个构件是一个域的成员, 而这个域的每个构件用不同的实现方法来实现在域所定义的公共接口。这样使得域中的成员可以互交换。构件可以用其他构件作为参数来实现其所属域定义的接口, 而不需要了解作为参数构件本身的实现, 所以一个构件可以被看成一层, 而软件系统就是由多个不同的层构成的。参数构件相对于其引用构件是下层对上层的联系。使用GenVoca的设计原则主要表现为以下几点:

(1)可拆卸的构件如同积木一样, 而形成系统的过程就是搭积木的过程; (2)可交换的构件应该有标准接口; (3)构件可以作为参数来传递。参数不同, 则组合不同。

1.1 构件(component)和域(realm)

如果将层次系统中的一层看作一个虚拟机, 则一个构件就是这个虚拟机的一种实现, 而构成这个虚拟机所有实现的构件集合就是域。一个域有一个标准接口, 这个域中的每个构件必须满足这个接口, 但是, 不同构件可以有不同实现。例如域 $R=\{r_1, r_2\}$ 有两个构件 r_1, r_2 , 它们是 R 的接口的不同实现。

1.2 参数

构件可作为另一个构件的参数, 表示两个构件的组合。构件的不同组合, 即不同的参数可以表示不同性能和语义的系统。例如: $S=\{s_1(R), s_2(R), \dots\}$, 则 s_1 表示它通过组合 R 中的构件来实现 S 的接口。 s_1 完成了从 R 到 S 间的映射。另外有些构件组合了自身域的构件, 则这种对称构件可以描述递归的组合, 如 $D=\{d_1(D), \dots\}$, 则序列 $d_1(d_2(d_3)), d_1(d_2(d_4)), \dots$ 就描述了这种构件任意组合的特性, 由此系统的灵活性和多样性就可通过这种不同复合来表达。

1.3 类型等式

GenVoca将系统简单地表达成构件的组合, 如下两个系统: $system_1=s_1(r_1)$; $system_2=s_2(r_2)$; 是 S 的两个可互相交换的实现。当多个表达式使用同一个构件(组合不同, 所以系统不同), 则该构件的重用度就被提高了。

1.4 分层系统

一个复杂系统总可以被表述为构件和构件的联系形式, 如果构件之间的引用基本是单向和无环的, 构件和被引用的构件之间的关系可以被看成分层的, 其中, 被引用的构件相对处于低层。为了能表达更为复杂的关系, GenVoca还支持自身的递归引用的描述。

2 SCD领域的分析和参考构架

"上海市教育考试院的信息管理系统"项目包括对上海市的中学会考, 普通高考, 成人高考的成绩处理, 和普通高校、成人高校录取系统等等。我们分别就成绩处理和其他一些特定领域做了分析和开发。于1997年的运用中, 获得一定的反馈, 因此对这些领域的了解较为深入和详细。在此基

*徐 锦 女, 25岁, 研究生, 主攻软件工程

收稿日期: 1998-05-20

础上,我们进一步按照DSSA的思想,作了特定领域的分析,并使用GenVoca模型对此领域的参照构架作了描述。下面介绍对“成绩处理”领域的分析和描述结果。特别为进一步开发系列“成绩处理”系统,参考构架将“面对多种考试”作为目标。

领域分析:上海教育考试院专门负责上海市的普通高考、中学会考和成人高考的成绩处理,因涉及面广,关系重大,所以,系统处理的数据量大,安全性,正确性要求高。长期以来,在考试院已经形成了一种处理模式:每个考生的每门成绩都是经过两遍录入,并且通过校对两遍录入,报告录入中的错误,并更正错误数据,才形成最终的考生成绩,成绩出来以后,通过计算得到总分,综合分,专业总分等处理分。在正确的成绩基础上,系统必须提供分析数据--对原始数据的统计结果,以供领导做决策。在决策之后,成绩可能还要进行调整。最后,各种“原始分数”,“再处理分数”和统计结果都要打印出来。所以整个处理系统将包括:数据子系统(DL),成绩录入子系统(SCOREIN),成绩校对子系统(SCORECHK),成绩再处理子系统(SCORERED),统计子系统(STAT),打印子系统(PRINT)。其构架如图1所示。在上述的域中,我们将列举某些域的构件,介绍这些构件将完成的功能。

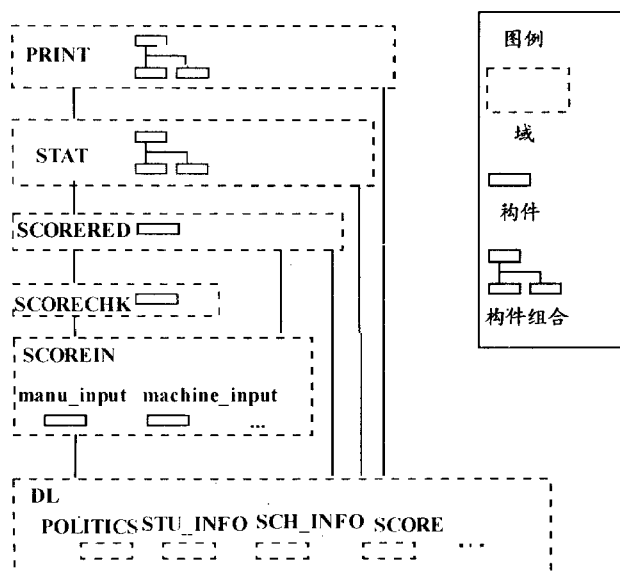


图1 处理系统架构

2.1 DL域

它包括多个子域,如:POLITICS(政策制定和访问域),STU_INFO(学生基本信息访问域),SCH_INFO(学校代码信息访问域),SCORE(成绩库访问域)。所以DL域可以表示成:DL=POLITICS U STU_INFO U SCH_INFO U SCORE U ...

2.2 SCOREIN域

```
SCOREIN={
manu_input(SCORE), /*手工输入*/
machine_input(SCORE), /*机读卡输入*/
...
add({SCOREIN},{SCORECHK}),
select({SCOREIN}) },
```

其中, add构件表示单科成绩可能分为几种方式录入,例如高考外语考试成绩,是读答题卡(machine_input)成绩和手工阅卷(manu_input)成绩的和,所以, add(manu_input, machine_input)就可以表示上述需求(其另一个SCORECHK参数的作用将在SCORECHK域中讲)。另外, select({SCOREIN})表示可以选择参数SCOREIN域中的一种输入方法。例如: select(manu_input, machine_input)构件表示成绩录入将提供两种录入的方式供选择。add和select构件是两个对称构件,用在这里,可以表达系统的可能的多种组合。

2.3 SCORECHK域

以SCOREIN域的构件作为参数,这里不详细说明。一个有趣的现象是,它和SCOREIN域的add构件发生关联--作为其参数。这表示从成绩录入到成绩校对再到成绩加和,可以有个反馈过程。如果将手工录入的两遍数据通过校对后再和机阅数据合并形成某单科的最终成绩,则可以表达为: add(machine_input, scorechk(manu_input, manu_input))。

2.4 STAT域

这个域较为复杂,但从抽象层次上讲,统计总可以表示成图2。

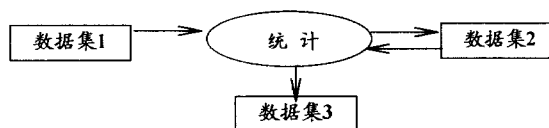


图2 统计

这表示,如果将原始数据作为数据集1输入,则通过统计,可以得到数据集2,而数据集2又可以作为一次中间结果,所以经过再次统计,得到最终所需结果-数据集3。根据这个特点,统计域可定义如下:

```
STAT={
average(SCORERED,DL,interpreter1 /*字段1*/, interpreter2 /*
字段2*/),
/*对最终成绩或中间成绩按字段1分类求字段2的平均*/
total(SCORERED, DL,interpreter1,interpreter2),
/*对最终成绩或中间成绩按字段1分类求字段2的和*/
...
serial({STAT}),
vertical(STAT), ...
}
```

其中, serial构件表示按参数次序来依次统计, vertical构件将一些分类统计结果(竖表)按分类的第一字段汇合成横表。例如, vertical(count(stu_info, interpreter('报名学校','附加分'))表示先按照学校和加分分类统计人数,得到如下表格:(其中stu_info是STU_INFO的构件,用于访问考生信息, interpreter是字段名称解释器, count是统计人数的STAT域中的构件)

```
311 50 2(人)/
311 30 3(人)/
311 20 4(人)/
311 10 5(人)/
312 50 2(人)/
312 30 8(人)/
312 20 5(人)/
312 10 6(人)/
```

(下转第45页)

表1 顾客(数据帧)等待时间随网络发送率G变动情况

G	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Wq1	0.111111	0.25	0.428571	0.666667	1	1.5	2.333333	4	9	∞
Wq2	0.010101	0.020408	0.030928	0.041667	0.05263	0.06383	0.0752688	0.0869565	0.098901	0.111111
Wq3	0.041341	0.085541	0.132888	0.183712	0.23839	0.297553	0.3611	0.430206	0.505344	0.587302
Wq	0.145475	0.329787	0.569859	0.893939	1.35294	2.04878	3.21941	5.57895	12.6986	∞

响预约表,所增加的仅仅是一次冲突,而不会产生连锁冲突。只有在不明优先级产生重构冲突的情况下,才需花费重构预约表的代价,而这一代价又被随后的按预约表无冲突访问的一组帧所分摊。与之相比,传统的CSMA/CD在重负荷的情况下产生众多的冲突,从而使传输几乎停顿。

这个方法的另一个特点是采用分布式优先级控制,在每个节点内维护一个预约表,而不是采用一个中心控制节点来实施优先级控制。这虽然付出了一些内存上的代价,却继承了以太网分布式控制的思想,增加了可靠性、健壮性和实现上的简单性。另外,由于一个以太网(或VLAN)中的节点数目有限,预约表中所需维护的数据量又少得可怜,因此并

出了越来越高的需求,速度与服务质量缺一不可。传统的网络技术已力不从心。为了平滑升级和保留现有投资,有必要在发展新技术的同时,对现有技术进行改造。在这一方面还有许多工作要做,我们提出的方案正是为将优先级引入以太网的一个尝试。希望大家能对这一方案提出宝贵的意见,批评和指正不足之处。

参考文献

- 1 Andrew S.Tanenbaum: Computer Network.Prentice Hall, PTR, 1996
- 2 高传善,张世永,曲海等.计算机网络.上海:复旦大学出版社,1994
- 3 陆传善.排队论.北京:北京邮电学院出版社,1994

不会占用太多的内存。

3 结束语

当今,信息产业迅猛发展,对网络技术提

~~~~~

(上接第25页)

再将此表按照第一个字段('报名学校')变成如下横表:

311 14 2 3 4 5

312 21 2 8 5 6

其中14是(2+3+4+5)的结果,21是(2+8+5+6)的结果。

#### 2.5 简单实例

下面,就上述的几个域,列举一些构件,并介绍一个应用系统的模型是如何使用这些构件组合出来的,同时还可以看到这种描述使相同领域的构件很容易被替换。在DL域,我们为POLITICS子域和SCORE子域制定了如下的构件:

```
POLITICS={
cr_total, /*成人高考计算总分方法*/
cr_compose, /*成人高考计算综合分方法*/
cr_totalsub, /*成人高考计算专业总分方法*/
pt_cal_total, /*普通高考计算总分方法*/
pt_adj_subject, /*普通高考单科调整分计算法*/
cr_stat_zgx, /*成人高考计算资格线方法*/
pt_stat_zgx, /*普通高考计算资格线方法*/ .....
};
SCORE={
tabin1, /*一遍录入库读写*/
tabin2, /*二遍录入库读写*/
scorepart /*部分分读写*/.....
};
```

其他域的构件如上几节所列举。

现在有一个需求是:成人高考考试成绩需要两遍录入,并校对,有些课程的分数为机器阅卷分+手工阅卷分之和,手工阅卷分两遍录入后,经比较合成一致成绩,再和机器阅卷分合并,当成绩完全正确后,对成绩计算总分和综合

分,并且计算资格线。(由于此系统表达式很长,我们用宏来替换一些子表达式)。

```
X=add(machine_input(scorepart),scorechk(manu_input(
tabin1),manu_input(tabin2))
```

```
Y=scorered(X,{cr_cal_total,cr_cal_compose})
```

```
system=cal_zgx(Y, cr_stat_zgx)
```

这样,上述需求的应用系统就可以表达出来了。一般复杂的系统需要分成若干子系统(如X和Y),子系统可被多个构件引用形成其他子系统,最终的系统可以是一个表达式,也可以是多个表达式的并(U)。

#### 3 结束语

DSSA方法对开发特定领域系列应用有着重要的指导意义,其对某个特定领域分析所得到的参照构架既是该领域生成系统的蓝图,也是开发此领域各种特定应用的构架的参考模型。它的作用是指导更高层次的重用,对软件的可升级性和构件级及设计级的重用都作出了贡献。本文强调DSSA方法,并使用GenVoca模型对"成绩处理"这个特定领域进行描述,实践表明,正确分析特定领域和有效选择参考构架描述将提高应用系统的开发效率。

#### 参考文献

- 1 Batory D, O'Malley S, The Design and Implementation of Hierarchical Software Systems with Reusable Components.University of Texas.ACM Transactions Software Engineering and Methodology, 1992, 1(4)
- 2 Deutsch P L.Design Reuse and Framework in the Smalltalk-80 Programming System.ACM Press, 1989
- 3 Batory D, Sinhal V.The GenVoca Model of Software System Generators. University of Texas at Austin IEEE Software, 1994