

文章编号:1001-9081(2005)08-1870-04

奥运赛程编排约束的语义表示法

蒲川江¹, 黄必清², 李春平¹

(1. 清华大学 软件学院, 北京 100084; 2. 清华大学 自动化系, 北京 100084)

(puchuanjiang@tsinghua.org.cn)

摘 要: 奥运赛程编排结果应符合奥运竞赛规程惯例, 各单项比赛的编排规则, 以及各种编排要求, 最大限度满足各种编排约束条件。奥运赛程编排约束具有数量多、种类多、可动态改变的特点。结合关系数据库, 利用约束模板和约束项提出的约束语义表示法具有灵活、易于扩充的优点, 在实现编排约束的表示、解析、交互、存储功能的同时, 解决了编排约束的人机交互和动态变化问题。

关键词: 语义; 约束; 约束项; 约束模板; 约束解析

中图分类号: TP311.52 **文献标识码:** A

Semantic representation method of tournament scheduling constraints in Olympics

PU Chuan-jiang¹, HUANG Bi-qing², LI Chun-ping¹

(1. School of Software, Tsinghua University, Beijing 100084, China;

2. Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: Scheduling results for sport tournament in Olympics should satisfy the tournament routine, programming rules and demands of each sport. It should satisfy all constraints to the max. The tournament scheduling constraints in Olympics had characters of large numbers of quantity and category, dynamic alterability. The semantic representation method, established by constraint templates and constraint items based on the relation database, had flexible and extensible advantages. It realized the functions of representing, parsing, interacting and storing scheduling constraints, also resolved man-machine interactive and dynamic alterable problems.

Key words: semantic; constraint; constraint item; constraint template; constraint parsing

0 引言

针对即将到来的 2008 年北京奥运会(以下简称奥运会), 有许多奥运课题急需解决。其中, 奥运竞赛管理系统就是一个重要的研究课题, 赛程编排子系统作为其关键部分, 其调度水平优劣直接关系到比赛能否公平、顺利地进行。奥运赛程编排(以下简称赛程编排)问题实质上属于时间表问题(Timetabling Problem), 为奥运会所有比赛项目的比赛场次安排参赛者、时间、场地、裁判的同时要考虑多方制约因素, 因此如何有效地表示和管理赛程编排约束, 已成为赛程编排问题的必需研究环节。

目前, 国内集中在田径竞赛的赛程编排研究^[1-3], 国外则集中在团体球类比赛, 如: 篮球^[4,5]、足球^[6]等。较为典型的赛程编排研究是 Nemhauser, Trick 在 1998 年对 ACC(The Atlantic Coast Conference)的 9 个大学队的 1997~1998 赛季篮球联赛(采用主、客场双循环制比赛)需求建模, 采用枚举和整数规划方法^[7], 从 299 738 880 种可能的赋值中, 用 Sun Sparc 工作站连续运算 24 小时找到 17 种可行的模式集。文献[8]针对上述 ACC 篮球联赛编排问题采用约束程序设计方法, 仅在 1 分钟内就找到 17 种模式集, 179 种编排方案。可见该方法在解决体育比赛赛程编排问题上体现出的高效性。

用约束程序设计方法解决奥运会这种大型体育竞赛的赛程编排, 首要问题是合理表示奥运会中的各种赛程编排约束。目前对约束的研究大多采用约束图^[9]、面向对象^[10,11]等方

法, 研究重点在约束表示、约束关系、约束求解方法上, 没有涉及约束的人机交互性、动态可变性。本文针对这方面做了研究, 在关系数据库基础上, 利用约束模板、约束项, 构造出约束语义表示方法, 并对约束解析、约束交互、约束存储进行了论述, 找到一种约束人机交互、可动态改变的约束管理方法。

1 赛程编排约束分类

1.1 赛程编排约束相关对象

约束描述了对对象间保持的关系, 因此在研究赛程编排约束之前, 先要明确其所涉及的对象以及对象之间的联系。与赛程编排约束相关的对象有:

比赛项目对象: 奥运会 Olympics、大项 Sport、分项 Discipline、小项 Event、子项 ChildEvent。

参赛人员对象: 运动员 Athlete、运动队 Team、裁判员 Referee、裁判组 RefereeGroup。

赛程编排单元对象: 小项 Event、赛段 Phase、轮 Round、场 Match。

赛程编排资源对象: 场地类型 FieldType、场馆 Gym、逻辑场馆 LogicGym、场地 Field、时间片 TimeSlice。

图 1 用统一建模语言 UML^[12]中的类图描述了它们之间的联系。

1.2 赛程编排约束分类

根据赛程编排约束自身的特点分成: 金牌数量约束、小项间比赛顺序约束、小项比赛单元比赛顺序约束、子项间比赛顺

收稿日期: 2005-02-17; 修订日期: 2005-04-29 基金项目: 北京市自然科学基金资助项目(4031002)

作者简介: 蒲川江(1974-), 男, 四川乐至人, 硕士研究生, 主要研究方向: 软件工程、任务调度; 黄必清(1966-), 男, 湖北武汉人, 副教授, 博士, 主要研究方向: 网格应用、服务理论、调度理论及算法; 李春平(1965-), 男, 辽宁新民人, 副教授, 博士, 主要研究方向: 信息系统与人工智能、数据库技术、数据挖掘。

序约束、场次约束、场地类型冲突约束、运动员最短休息时间约束、运动员兼项约束^[13,14]。

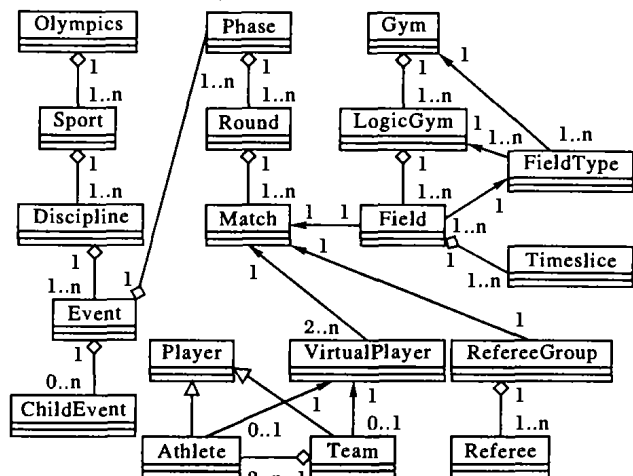


图1 赛程编排约束对象类图

2 赛程编排约束语义表示框架

如图2所示,赛程编排约束语义表示框架由约束交互、约束存储、约束解析三部分组成。约束模板定义了约束的框架,约束交互将自然语言约束转化成语义约束,约束存储再将其永久存储到数据库中,供约束交互、约束解析使用,在赛程编排过程中,约束解析从约束库中提取约束,解释、执行后得出约束断言结果。

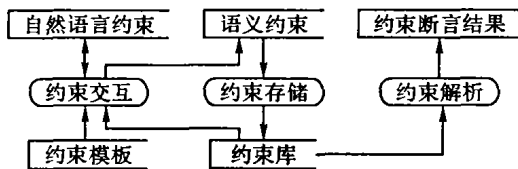


图2 赛程编排约束语义表示框架

3 赛程编排约束语义表示

3.1 赛程编排约束 E-R 模型

为了实现赛程编排约束语义表示,除奥运比赛项目相关对象外,还需建立图3所示的编排约束对象。

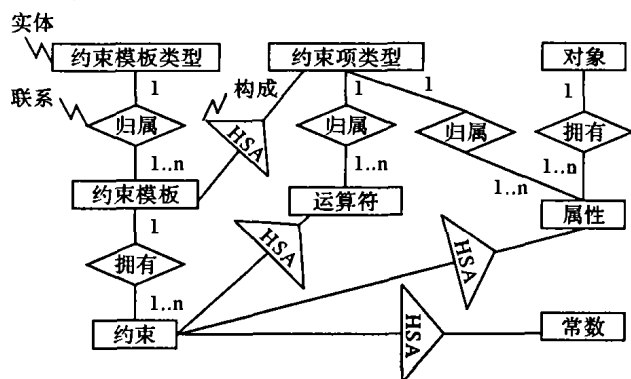


图3 赛程编排约束 E-R 模型

赛程编排约束 E-R 模型说明如下:

(1)约束模板类型:为方便对约束模板的管理,根据约束模板表达的约束主体对约束模板进行分类,包括:金牌约束模板类型、比赛项目约束模板类型、场馆约束模板类型、人员约束模板类型。

(2)约束模板:约束模板由约束项类型组成,定义约束的语义框架。辅助约束交互功能的实现。

(3)约束:约束由约束项组成,描述对象间保持的关系。

从图3可以看出,约束是由对象的属性、常数、运算符三者,按照约束模板定义的约束框架组合而成的。

(4)约束项类型:约束项的语义域。包括:金牌数量项类型、小项项类型、子项项类型、小项比赛单元项类型、场地类型项类型、时间点项类型、各种运算符项类型(如:赋值运算符项类型、算术运算符项类型、次序运算符项类型等)、常数项类型等。

(5)运算符:运算符是对约束中对象之间关系的符号表示。在奥运编排约束中,多个对象之间的关系都可以转化成两个对象之间的关系,因此只需要讨论二元关系运算。如:赋值运算符“=”、加法运算符“+”、先序运算符“>>”。

按照运算符所表示的对象关系的性质对运算符分类,如:算术运算符类型(包括加“+”,减“-”),关系运算符类型(包括大于“>”,小于“<”),次序运算符类型(包括先序“>>”,排斥“><”,交错“×”),括号运算符类型(包括左括号“(”和右括号“)”)等。每个运算符分类都是一种约束项类型。

在定义了运算符所表示的某种约束关系的同时,需要为每个运算符制订相应的运算或操作,以便对约束中的关系做相应处理。

我们定义运算符优先级矩阵,用来明确在同一约束中出现的各运算符的处理优先顺序,见表1。 $OP1 > OP2$ 表示 $OP1$ 的优先级高,做优先处理。表中的“(= =)”表示当左右括号相遇时,括号内运算符已经处理完成。若以后有新的运算符可以补充进来。

表 1 运算符优先级矩阵

OP2	OP1									
	=	+	-	()	>	<	>>	><	x
=		<	<	<	<			<	<	<
+		>	>	<	>	>	>			
-		>	>	<	>	>	>			
(<	<	<	=	<	<			
)		>	>		>	>	>			
>		<	<	<	>	>	>			
<		<	<	<	>	>	>			
>>				<	>			>	>	>
><				<	>			>	>	>
x				<	>			>	>	>

(6)对象:实体的抽象描述,有自己的属性、方法。我们用关系数据库中的表表示对象。

(7)属性:实体的静态性质。用对象属性之间的运算表示对象之间的关系,用关系数据库中表的列表示对象的属性。每一个属性都被人为指定一个数据项类型,标明该属性值表示的语义范围。

(8) 常数: 具体的一个数。约束中经常用到常数。

(9)约束项:组成语义约束的最小语义单位,分为变量、常数、运算符三种处理类型。约束项具有自描述性,用自身信息解释自己,不需要借助附加信息,类似 XML 数据的自解释思想。通过自描述性,可从约束项自身的信息,得知该约束项如何处理,值如何获取,值表示的含义。约束项结构定义如下:

<约束项> ::= <处理类型;值类型;精度;值>

<处理类型> ::= <变量类型|常数类型|运算符类型>。

处理类型决定了该约束项的处理方式,有变量、常数、运算符三

种处理类型。对于变量类型,用关系数据库中的表、行、列定位变量并取值;对于常数类型,不需要定位,直接取出值;对于运算符类型,对该运算符所涉及的所有变量、常数做相应运算或处理。

<值类型>::=<bool|int|float|string|datetime|关系运算符|算术运算符|...>。在明确约束项处理类型的基础上,值类型明确了约束项值的值域。若约束项是变量或常数处理类型,则值类型就是某种数据类型;若约束项是运算符,则值类型就是运算符的类型。

<精度>::=<none|date|time|datetime|day|minute|...>。若约束项是变量或常数处理类型,在明确其值类型的基础上必须指明值类型的精度,以时间点项类型的约束项为例,若值类型是 datetime 型,则用精度指明该 datetime 型的时间精度,比如 day 精确到天,minute 精确到分钟;若约束项是运算符处理类型,该项无意义,设为 none。

表2 编排约束表功能简略说明

中文表名	英文表名	功能
约束模板类型表	TemplateType	对约束模板的分类
约束模板表	Template	构造约束的语义框架
约束表	Constraint	存储编排约束
约束项类型表	ConstraintItemType	对约束项的分类
运算符表	Operator	约束中用到的各种运算符
对象表	Object	存储与比赛项目相关的对象
属性表	Attribute	存储与比赛项目相关的对象的属性

<值>::=<[表 Id; 主键 Id;]字符串>。若约束项是变量处理类型,则值包括表值、主键值、列值,用于定位到关系数据库中的某表某行某列取值;若约束项是运算符或常数处

理类型,则值表示运算符或一个数。

<表>: 关系数据库中的表。只对变量处理类型的数据项有意义。

<主键>::=<列 1Id[,列 2Id[,...]]>。表的主键由 1 列或多列组成,用于表的行定位。只对变量处理类型的数据项有意义。

<字符串>: 若约束项是变量处理类型,字符串完成表的列定位;若约束项是运算符或常数处理类型,则表示运算符或常数的值。

把图 3 编排约束对象 E-R 模型转化成关系数据库中的表,如表 2 所示。

至此,已为编排约束语义表示工作做好了数据库方面的准备。

编排约束语义表示法的关键问题是:如何将自然语言约束翻译成语义层面上的语义约束,如何解释执行语义约束。

我们用约束交互和约束解析解决了这两个问题,下面详细叙述。

3.2 约束解析

参照图 1,约束解析的目的是将数据库中存储的语义约束解释成相应的变量关系式,做相应运算和操作。因为语义约束是由约束项组成的,需要左向右逐一解析每个约束项,对约束中的所有运算符进行优先级比较后,做相应运算或处理。

以约束“男子足球决赛在奥运会闭幕式前一天进行”为例,详细描述其对应的语义约束的解析过程。这里先给出该例子的约束语义表示结果,如表 3 所示。因随后的约束解析、约束交互结果与此表雷同,为引用方便,故将所有相关的约束处理结果预先写入此表。

表3 足球编排约束例子的语义表示

项序	自然语言项(人理解)	约束项类型 (组成约束模板)	约束项(计算机程序理解) (<约束项>::=<处理类型:值类型:精度:值>)	约束解析 临时变量
第 1 项	男子足球决赛比赛日期	<时间点项类型>	<变量类型:datetime:date:场次表 Id; 男足小项 Id,决赛场次 Id;比赛日期 Id>	var1
第 2 项	进行	<赋值运算符项类型>	<运算符类型:赋值运算符类型 Id:none:=>	
第 3 项	奥运会闭幕式日期	<时间点项类型>	<变量类型:datetime:date:奥运会表 Id; 奥运会序号 Id;闭幕式日期 Id>	var1(var2)
第 4 项	前	<算术运算符项类型>	<运算符类型:算术运算符类型 Id:none:->	
第 5 项	一天	<常数项类型>	<常数类型:int:day:1>	var2

此足球编排约束的解析步骤如下:

根据图 1,此约束数据来源于约束表 Constraint,记录内容如表 3 的“约束项”列所示,可以看出约束表 Constraint 存储的是语义层面的约束。

足球编排约束例子的解析步骤如下:

先初始化两个空栈,包括一个运算对象栈和一个运算符栈。然后按照从左向右的顺序逐一分析约束中的约束项,对约束中的运算符做相应运算或操作,具体解析步骤如下:

1)分析第 1 约束项:<变量类型:datetime:date:场次表 Id;男足小项 Id,决赛场次 Id;比赛日期 Id>。处理类型是非运算符,直接入运算对象栈。

2)分析第 2 约束项:<运算符类型:赋值运算符类型 Id:none:=>。处理类型是运算符,且此时运算符栈为空,入运算符栈。

3)分析第 3 约束项:<变量类型:datetime:date:奥运会表 Id;奥运会序号 Id;闭幕式日期 Id>。处理类型是非运算符,

直接入运算对象栈。

4)分析第 4 约束项:<运算符类型:算术运算符类型 Id:none:->。处理类型是运算符,此时栈不空,与栈顶运算符即第 2 约束项进行运算符值优先级比较,“-”优先级比“=”高,第 4 约束项入运算符栈。

5)分析第 5 约束项:<常数类型:int:day:1>。处理类型是非运算符,直接入运算对象栈。

6)分析完该约束中的所有约束项后,我们开始处理运算符栈,具体步骤如下:

(A)读取运算符栈顶元素,因运算符栈不空,取出栈顶运算符即第 4 约束项,并从运算对象栈取出两个运算对象即第 3 约束项和第 5 约束项,因为第 4 运算符约束项的运算符值是“-”运算,所以用第 3 约束项的值减去第 5 约束项的值,具体操作步骤如下:

(a)根据第 3 约束项的内容<变量类型:datetime:date:奥运会表 Id;奥运会序号 Id;闭幕式日期 Id>,做以下取值步骤:

①处理类型:变量类型。说明该约束项是对象的属性值,即从对应表的行列取值。

②值类型:datetime。我们申明一个 datetime 类型的变量 var2。

③精度:date。说明该变量的数据精度为日期,取值时忽略时间部分。

④值:奥运会表 Id;奥运会序号 Id;闭幕式日期 Id。这个值给出了表、主键、列值,用该值定位表、行、列,然后取值的步骤如下:

i 定表:用 SQL 查询语句从对象表 Object 中检索出主键 Id 是奥运会表 Id 的行记录,然后从该行的 name 字段得知表名是“Olympics”,即奥运会表。

ii 定行:因为奥运会表中只存有一条记录,即当前奥运会,直接取出行记录。

iii 定列:取出当前奥运会行记录的闭幕式日期列 ClosCereDate 的值,将其放入(2)中声明的 datetime 型变量 var1 中,根据(3)中的日期精度,我们将 var1 变量中的时间部分截掉,只保留日期部分。至此,我们完成第 3 约束项的取值工作。

(b)根据第 5 约束项的内容 <常数类型:int:day:1>,直接申明一个整型变量 var2,赋值为 1。

(c)对做“-”运算,即 $var1 = var1 - var2$ 。

(d)把 var1 封装成一个常数类型的约束项,即 <变量类型:datetime:date:var1>,入运算对象栈。

(B)读取运算符栈顶元素,此时运算符栈不空,取出栈顶运算符即第 2 约束项,并从运算对象栈取出两个运算对象即第 1 约束项和封装的常数类型约束项,因为第 2 运算符约束项的运算符值是“=”运算,所以把封装的常数类型约束项的值赋值给第 1 约束项,这两个约束项的取值方法分别与第 3 约束项和第 5 约束项的取值方法相似,这里不再详述。根据第 2 约束项的运算符值“=”的赋值运算操作定义,需要改变赋值运算符“=”左边第 1 约束项所对应的关系数据库中的表的值,即把男足小项决赛场次行记录的日期列赋值为奥运会闭幕式日期前一天的日期。

(C)读取运算符栈顶元素,因运算符栈空,约束解析处理完毕。

图 4 是对上述约束解析过程的补充说明。

3.3 约束交互

用户使用约束交互功能管理数据库中的约束,进行增、删、改操作,这使得约束的处理变得简单起来,与一般的信息处理没有什么区别。

用户与系统的约束交互变成了自然约束到语义约束之间的双向交互过程,要实现约束交互功能,必须借助约束模板。约束模板中的约束项类型与该模板下约束中的约束项是一一对应的,约束项类型不仅从内容上定义了约束项的语义范围,而且为需要选择或填写的约束项规定了用户界面框架,使得只有符合该约束项类型的约束项才能被用户选择或输入。

仍以上述男子足球约束例子为例,简要描述一下增加约

束的人机交互过程,说明从自然约束到语义约束是如何交互的。

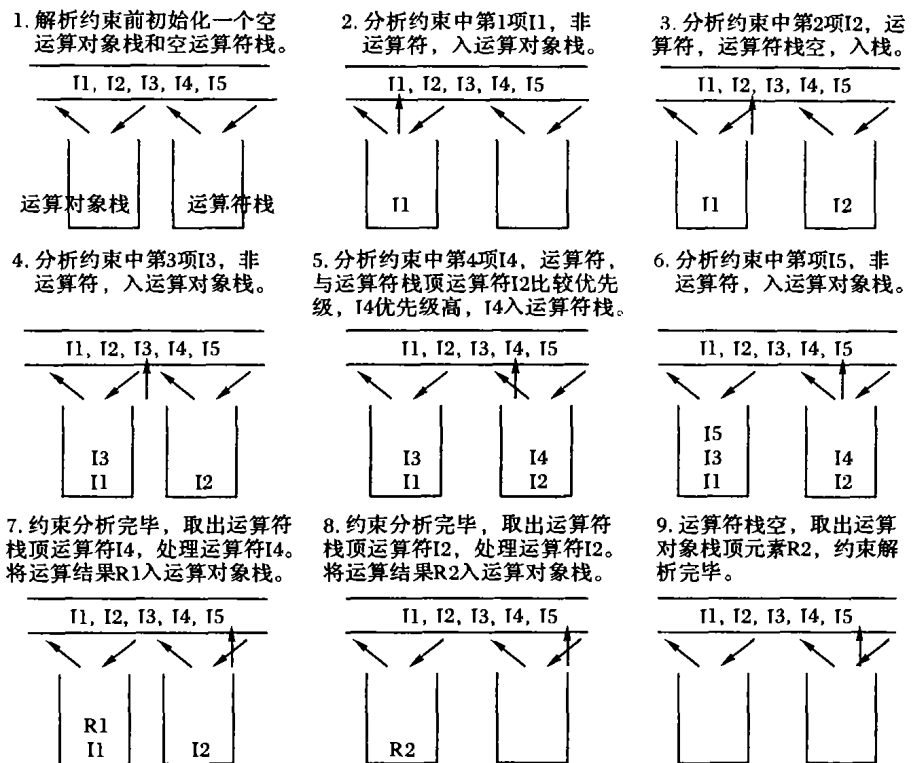


图 4 男足约束例子的解析过程

(1)进入约束设置界面,如图 5 所示。选择比赛项目约束模板类型下的比赛时间约束模板:<时间点类型><赋值运算符><时间点类型><算术运算符><常数类型>,在该模板下增加约束。

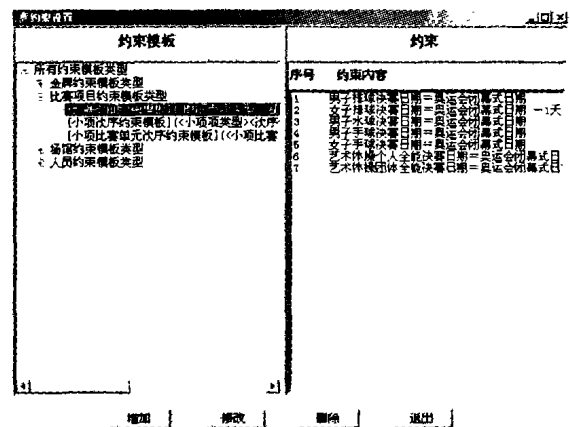


图 5 约束设置界面

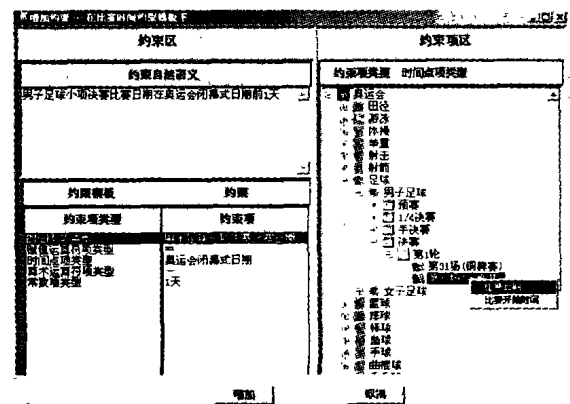


图 6 增加约束界面

计实时系统,把时间分离出来构造为一个时间方面,建立一个比较完整的时间模型来建模系统时间,促使时间方面与系统其他方面分开,能够进行单独的时间方面设计。同时又能够根据需求把设计好的时间方面织入到系统中,从而简化了实时系统建模的复杂性。UML和AOP技术的结合,充分利用AOP的思想和UML的强大的建模能力,并对UML进行扩展,是实时系统的建模的一个新途径。本文通过从静态结构模型、动态行为模型和时间方面的织入等几部分实现了电梯控制系统的整个设计过程。

目前AOP技术研究还处于初始阶段,利用UML和AOP技术结合建模实时系统还需要更深入的研究和探讨。本文只涉及到时间方面的建模和织入,但实时系统通常都很复杂,涉及到许多方面,而且方面之间也存在交互横切关系,因此这种设计方法需要进一步完善:1)进一步完善UML与AOP的结合;2)实现多维多方面的分离、建模以及织入;3)解决AOP建模实时系统的语义问题;4)进行与CASE工具结合,实现方面代码的自动产生与织入;5)扩展到实时分布式系统的开发。

参考文献:

- [1] ZAKARIA AA, HOSNY H, ZEID A. A UML Extension for Modeling Aspect-Oriented Systems[Z]. International Workshop on Aspect-Oriented Modeling with UML, 2002.
- [2] ZHANG L, GALINDO M, MARQUIE D, *et al.* Methodology of real-time system design using multiprocessors[J]. *Microprocessors and Microsystems*, 1993, 17(4): 207-214.
- [3] ZHANG LC, CHAIB-DRAA B. A Design methodology for real-time

systems to be implemented on multiprocessor machines[J]. *Journal of Systems and Software*, 1996, 33(1): 37-56.

- [4] TESANOVIC A, NYSTROM D. Aspects and Components in Real-Time System Development: Towards Reconfigurable and Reusable Software[J]. *Journal of Embedded Computing*, 2004, 1(1).
- [5] KATARA M, MIKKONEN T. Aspect-oriented specification architectures for distributed real-time systems[A]. *Proceedings of the 7th IEEE International Conference on Engineering of Complex Computer Systems*[C]. 2001.
- [6] ALDAWUD O, ELRAD T, BADER A. UML Profile for Aspect-Oriented Software Development[A]. *Proceedings of 3rd International Workshop on Aspect-Oriented Modeling*[C]. 2003.
- [7] STEIN D, HANENBERG S, UNLAND R. A UML-based Aspect-Oriented Design Notation for Aspect[J]. *Processing of 1st ACM International Conference on Aspect-Oriented Software Development*[C]. 2002.
- [8] OMG. UML Profile for Schedulability, Performance, and Time[DB/OL]. www.omg.org, 2003.
- [9] SERGIU-MIHAI DASCALU. Combining Semi-Formal and Formal Notations in Software Specification: An Approach to Modelling Time-constrained Systems[D]. Partial fulfillment of the requirements for the degree of doctor of philosophy at Dalhousie University, 2001.
- [10] MAHONEY M, BADER A, ELRAD T, *et al.* Using Aspects to Abstract and Modularize Statecharts[A]. *The 5th Aspect-Oriented Modeling Workshop in Conjunction with UML*[C]. 2004.

(上接第1873页)

(2) 增加约束界面如图6所示,我们按照比赛时间约束模板中的每一个“约束项类型”,填充对应的“约束项”内容。因为不同约束项类型对应的约束项设置方法不同,所以我们在约束项区内为每种约束项类型设置了相应的约束项设置界面。当选择某约束项类型时,就进入与该约束项类型对应的约束项设置界面,完成具体约束项内容的设置。例如图6中的约束项区就显示了在用户选择时间点项类型时与之对应的约束项设置界面。

修改约束多了一步从数据库提取约束的过程,其余操作与增加约束过程类似,这里不再详述。

3.4 约束存储

在数据库中,用约束表 Constraint 永久存储约束记录,供约束交互和约束解析使用。约束记录存储内容如前表3的“约束项”列所示。在前面的约束交互过程最后一步就已经使用了约束存储功能。

4 结语

我们在关系数据库基础上,用约束模板实现约束人机交互,用约束项实现约束解析,在语义层面上构造了奥运赛程编排约束的语义表示方法,比直接使用变量运算关系式的约束表示方法高一个层次,在表示对象间复杂关系时显得直观、灵活。约束语义表示法不仅实现了约束的表示、解析、交互和存储,还解决了约束的人机交互和动态可变性问题,还可以通过定义新运算符、新约束模板,扩展对新类型约束的表示能力。本文为奥运赛程编排约束的下一步研究工作做好了准备。

参考文献:

- [1] 佟毅. 田径竞赛编排中的两个数学模型[J]. *数理统计与管理*, 1998, 17(4): 1-3.

- [2] 庄晋林. 运动会管理信息系统的设计与实现[J]. *微型电脑应用*, 1999, 15(9): 33-36, 22.
- [3] 张建勋, 刘明华. 田径运动会竞赛日程自动编排的静态模型与算法[J]. *兰州铁道学院学报*, 1994, 13(2): 86-92.
- [4] CAMPBELL RT, CHEN DS. A Minimum Distance Basketball Scheduling Problem[A]. *Management Science in Sports*[C]. 1976. 15-25.
- [5] BEAN JC, BIRGE JR. Reducing Traveling Costs and Player Fatigue in the National Basketball Association[J]. *Interfaces*, 1980, 10(3): 98-102.
- [6] SCHREUDER JAM. Combinatorial Aspects of Construction of Competition Dutch Professional Football Leagues[J]. *Discrete Applied Mathematics*, 1992, 35: 301-312.
- [7] NEMHAUSER GL, TRICK MA. Scheduling a major college basketball conference[J]. *Operations Research*, 1998, 46(1): 1-8.
- [8] HENZ M. Scheduling a major college basketball conference-revisited[J]. *Operations Research*, 2001, 49(1): 163-168.
- [9] 刘晓平, 黄永红, 金文华, 等. 工程约束表示模型与求解算法研究[J]. *计算机学报*, 1999, 22(11): 1153-1157.
- [10] 王金敏, 王玉新, 查建中. 布局问题约束的分类及表达[J]. *计算机辅助设计与图形学学报*, 2000, 12(5): 349-354.
- [11] 陈恩红, 张奠成. 几何设计约束的表示与满足问题研究[J]. *计算机研究与发展*, 1994, 31(10): 35-39.
- [12] BOOCH G, RUMBAUGH J, JACOBSON I. UML 用户指南(中文版)[M]. 北京: 机械工业出版社, 2000.
- [13] 何彦吉, 刘兆万. 田径竞赛裁判学[M]. 沈阳: 东北大学出版社, 1998.
- [14] 编写组汇编. 体育运动竞赛组织与编排[Z]. 北京: 北京师范大学出版社, 1984.