

基于本体映射和规则推理的排课模型研究

徐 晓

(浙江广厦建设职业技术学院 信息与控制工程学院,浙江 东阳 322100)

摘 要:对排课问题进行了描述,给出了解决排课问题的多种排课方法,并且对这些排课方法进行了分析和比较。在排课模型中运用本体知识创建了 OWL 排课本体,运用本体映射方法达到数据的同步,运用 SWRL 语法对本体进行约束,运用规则推理引擎 JESS 进行推理,结合排课算法给出了整体的排课模型架构。

关键词:排课;OWL;本体映射;SWRL;规则推理

中图分类号:TP311.52 文献标识码:A 文章编号:1672-7800(2009)02-0193-02

表 1 OWL 的三个子语言

子语言	描述	例子
OWL Lite	用于提供给那些只需要一个分类层次和简单的属性约束的用户支持那些需要在推理系统上进行最大程度表达的用户,这里的推理系统能够保证计算完全性(computational completeness,即所有的结论都能够保证被计算出来)和可决定性(decidability,即所有的计算都在有限的时间内完成)。	支持基数(cardinality),只允许基数为 0 或 1
OWL DL	它包括了 OWL 语言的所有约束,但是可以被仅仅置于特定的约束下支持那些需要在没有计算保证的语法自由的 RDF 上进行最大程度表达的	当一个类可以是多个类的子类时,它被约束不能是另外一个类的实例
OWL Full	它允许在一个 Ontology 在预定义的(RDF,OWL)词汇表上增加词汇,从而任何推理软件均不能支持 OWL FULL 的所有特性	一个类可以被同时表为许多个体的一个集合以及这个集合中的一个个体

出概念(OWL Classes);②建立概念之间的关系(Object Properties);③建立概念和数据类型之间的关系(Datatype Properties);④设置限制条件(Restriction);⑤运用 SWRL(Semantic Web Rule Language)编写规则;⑥创建个体(Individual)。

1.3 语意规则语言

SWRL(Semantic Web Rule Language)是以语意的方式呈现规则的一种语言,SWRL 规则部分的概念是由 RuleML 演变而来,再结合 OWL 本体形成,目前 SWRL 已经成为 W3C 的规范之一。

在 RuleML 中以 hand 表示推论结果,body 表示推论前提的基本形态也保留在 SWRL 中,所以 SWRL 可视化为规则和本体的结合,透过两者的组合可使撰写规则时,直接使用本体中所描述的关系和字词,这些类别之间的关系本来可能还需要

1 本体描述语言及创建工具

OWL 全称 Web Ontology Language,是 W3C 推荐的语义互联网中本体描述语言的标准。它是从欧美一些研究机构的一种结合性的描述语言 DAML+OIL 发展起来的,其中 DAML 来自美国的提案 DAML-ONT,它由美国政府支持,目标是开发一种语言和一组工具,为语义互联网提供支持。OIL 是来自欧洲的一种本体描述语言,它是结合基于框架的系统、描述逻辑和互联网标准三方面工作而提出的一种通用的语义互联网标记语言。在两种本体描述语言的基础上进行了标准化,就得到了本体描述语言的标准语言 OWL。

1.1 本体描述语言归类

本体描述语言使得用户为领域模型编写清晰的、形式化的概念描述,因此它应该满足以下要求:①良好定义的语法(a well-defined syntax);②良好定义的语义(a well-defined semantics);③有效的推理支持(efficient reasoning support);④充分的表达能力(sufficient expressive power);⑤表达的方便性(convenience of expression)。

针对不同的需求 OWL 有 3 个子语言,描述如表 1 所示。

1.2 本体创建工具

Protégé 是由斯坦福大学的 Stanford Medical Informatics 开发的一个开放源码的本体编辑器,它是用 Java 编写的。Protégé 界面风格与普通 Windows 应用程序风格一致,用户比较容易学习使用。本体结构以树形的层次目录结构显示,用户可以通过点击相应的项目来增加或编辑类、子类、属性、实例等,使用户在概念层次上设计领域模型,所以本体工程师不需要了解具体的本体表示语言。Protégé 支持多重继承,并对新数据进行一致性检查,具有很强的可扩展性。

下面简要介绍 Protégé 创建本体的过程:①从问题中抽象

额外的规则描述,但 SWRL 中可以直接使用本体的描述。

2 排课本体约束与模型架构

2.1 排课本体

排课本体包含课程本体、教师本体、班级本体、时段本体、事件本体和字段本体。其中课程本体和教室本体具体描述如下:

(1)课程本体。课程分为公共基础、专业必修、专业选修和其他选修 4 种类别。考虑到公共基础和专业必修是必修课程而专业选修和其他选修是选修课程,排课时公共基础和专业必修的优先级高于专业选修和其他选修。

(2)教室本体。教室分为特殊教室和普通教室两种类别,特殊教室又分为多媒体教室、语音室、画室和实验室等类别,实验室又可分为硬件实验室和软件实验室等类别。教室是排课问题中的空间资源,考虑到空间资源的紧缺,能够在多大程度上对空间资源合理分配成为关键。

由于篇幅所限,暂列出课程本体和教室本体描述图,其他本体图类似。

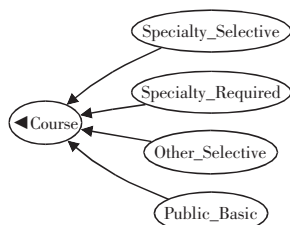


图 1 课程本体

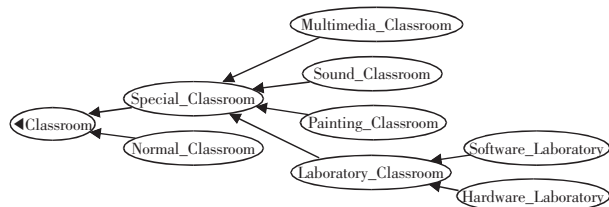


图 2 教室本体

2.2 本体约束

约束条件分为硬约束条件和软约束条件,硬约束条件必须得到满足,硬约束条件的满足可以保证非法解的淘汰;软约束条件尽量得到满足,软约束条件的满足程度反映了问题本身解决的程度。下面分别选择两种约束条件中的一种用 SWRL 进行规约,其他的硬约束条件和软约束条件可以对照以下两个例子进行设置,在此不一一给出。

硬约束条件举例:同一时段,一个教师不能上一门以上的课程。

① Teach(? teach_x) ∧ ; ② hasWeek_Slice_ID(? teach_x,? week_slice_id_a) ∧ ; ③ Week_Slice_ID(? week_slice_id_a) ∧ ; ④ hasWeek_Slice_ID(? teach_x,? week_slice_id_b) ∧ ; ⑤ Week_Slice_ID(? week_slice_id_b) ∧ ; ⑥ sameAs(? week_slice_id_a,? week_slice_id_b) ∧ ; ⑦ hasTeacher_ID(? teach_x,? teacher_id_a) ∧ ; ⑧ Teacher_ID(? teacher_id_a) ∧ ; ⑨ hasTeacher_ID(? teach_x,? teacher_id_b) ∧ ; ⑩ Teacher_ID(? teacher_id_b) ∧ ; ⑪

sameAs(? teacher_id_a,? teacher_id_b) ∧ ; ⑫ hasCourse_ID(? teach_x,? course_id_a) ∧ ; ⑬ Course_ID(? course_id_a) ∧ ; ⑭ hasCourse_ID(? teach_x,? course_id_b) ∧ ; ⑮ Course_ID(? course_id_b) ∧ ; ⑯ differentFrom(? course_id_a,? course_id_b) ∧ ; ⑰ hasRule_Punish_Parameter(? teach_x,? rule_punish_parameter) ∧ ; ⑱ Rule_Punish_Parameter(? rule_punish_parameter) ∧ ; ⑲ hasRule_Punish_ParameterData(? rule_punish_parameter,? rule_punish_value); ⑳ swrlb:add(? rule_punish_value,? rule_punish_value,RULE_PUNISH_VALUE_INCREMENT)。

说明:①-⑥表示授课事件发生在某一时段,⑦-⑪表示此授课事件中有一位教师,⑫-⑯表示此授课事件中有两门课程,⑰-⑱表示授课事件对应规则惩罚参数,⑲表示规则惩罚参数有相应的增量。整个约束条件表示由①-⑱推理出⑲,即如果同一时段,一个教师上一门以上的课程,则规则惩罚参数增加相应的量,RULE_PUNISH_VALUE_INCREMENT 表示规则惩罚参数,它的基准值可设置为 1,增量则要设置为一个很大的数,这样可以保证在计算适应值时,使适应值变得很小,从而把适应值小的淘汰掉。

软约束条件举例:多学时课程一周内的安排要尽量错开。

① Teach(? teach_x) ∧ ; ② hasCourse_ID(? teach_x,? course_id_a) ∧ ; ③ Course_ID(? course_id_a) ∧ ; ④ Teach(? teach_y) ∧ ; ⑤ hasCourse_ID(? teach_y,? course_id_b) ∧ ; ⑥ Course_ID(? course_id_b) ∧ ; ⑦ sameAs(? course_id_a,? course_id_b) ∧ ; ⑧ hasWeek_Slice_ID(? teach_x,? week_slice_id_a) ∧ ; ⑨ Week_Slice_ID(? week_slice_id_a) ∧ ; ⑩ hasWeek_Slice_IDData(? week_slice_id_a,? int_value_1) ∧ ; ⑪ hasWeek_Slice_ID(? teach_y,? week_slice_id_b) ∧ ; ⑫ Week_Slice_ID(? week_slice_id_b) ∧ ; ⑬ hasWeek_Slice_IDData(? week_slice_id_b,? int_value_2) ∧ ; ⑭ differentFrom(? week_slice_id_a,? week_slice_id_b) ∧ ; ⑮ swrlb:subtract(? temp_value_subtract,? int_value_1,? int_value_2) ∧ ; ⑯ swrlb:abs(? temp_value_abs,? temp_value_subtract) ∧ ; ⑰ swrlb:greaterThan-OrEqual(? temp_value_abs,INTERVAL) ∧ ; ⑱ swrlb:lessThan-OrEqual(? temp_value_abs,SLICE_COUNT-INTERVAL) ∧ ; ⑲ hasRule_Encourage_Parameter(? teach_x,? rule_encourage_parameter_a) ∧ ; ⑳ hasRule_Encourage_Parameter(? teach_y,? rule_encourage_parameter_b) ∧ ; ㉑ Rule_Encourage_Parameter(? rule_encourage_parameter_a) ∧ ; ㉒ Rule_Encourage_Parameter(? rule_encourage_parameter_b) ∧ ; ㉓ hasRule_Encourage_ParameterData(? rule_encourage_parameter_a,? rule_encourage_value_1) ∧ ; ㉔ hasRule_Encourage_ParameterData(? rule_encourage_parameter_b,? rule_encourage_value_2) ∧ ; ㉕ swrlb:add(? rule_encourage_value_1,? rule_encourage_value_1,RULE_ENCOURAGE_VALUE_INCREMENT) ∧ ; ㉖ swrlb:add(? rule_encourage_value_2,? rule_encourage_value_2,RULE_ENCOURAGE_VALUE_INCREMENT)。

说明:①-⑦表示两授课事件中教授一门课程,⑧-⑭表示两授课事件发生在不同时段,⑮-⑳表示两时段间有相应的时间间隔,㉑-㉒表示规则奖励参数有对应的增量。整个约束条件表示由①-⑭推理出㉑-㉒,即如果多学时课程一周内的安排有相应的错开程度,即时段有相应的间隔,则规则奖励参数增加相应的量,RULE_ENCOURAGE_VALUE_INCREMENT表示规则奖励参数,它的基准值也可设置为1,增量则要设置为适当的值,这样可以保证在计算适应值时,使得适应值满足一个软约束条件时就有相应的增量,适应值大的被选中的几率就大些,从而能够以更大的概率找到最优解。下面给出时段间隔与规则奖励参数增量之间的对应关系,在给出此关系之前需要看一看时段间隔的描述。

2.3 模型架构

为了对模型有一个整体的了解,给出排课模型的整体架构。此模型分为以下两大部分:

(1)生成排课本体。对本体进行规约并将其转换成规则推理引擎可以识别的格式;在排课模型中描述为从第①步到第⑥步。

(2)生成排课算法。进行排课算法和规则推理引擎之间的参数传递,从而生成课表;在排课模型中描述为从第①步到第⑩步。排课模型架构如图3所示。

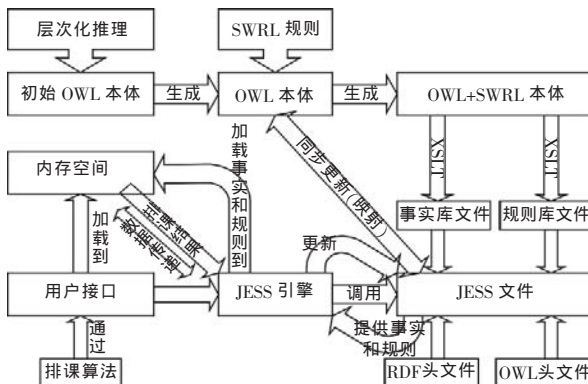


图3 排课模型架构

下面分别对这两部分进行描述。

第一部分:①根据排课问题抽象概念生成排课本体,并对排课本体进行层次化处理,生成经过层次化处理的排课本体;②运用SWRL规则对层次化后的排课本体进行约束条件的规约,生成具有约束条件限制的排课本体;③用XSLT将排课本体转换成规则推理引擎中对应的事实库部分,加入到规则推理引擎专用文件JESS文件;④用XSLT将排课本体的约束条件

转换成规则推理引擎中对应的规则库部分,加入到JESS文件;⑤加入RDF头文件到JESS文件;⑥加入OWL头文件到JESS文件。

第二部分:①生成排课算法,通过用户接口将其加载到内存空间;②使内存空间中的排课算法和JESS引擎之间进行数据传递,在此过程中JESS文件向JESS引擎提供事实库和规则库,事实库提供生成课表元素的数据,规则库提供对课表元素约束条件的规约;③在排课模型中经过多次的从③到⑦的过程,生成了最终的排课结果;④根据排课结果通过JESS引擎对JESS文件进行更新,此时JESS文件生成了根据排课结果生成的新的事实库;⑤利用影子函数对JESS文件和排课本体进行映射,在更新JESS文件的同时,排课本体由于和JESS文件的映射也达到了数据的同步更新。

3 结束语

本文首先引入了排课问题,接着给出了解决排课问题的方法,随后对这些方法进行了分析和比较,根据本体的相关知识和方法,创建排课本体,最后在排课本体的基础上运用规则进行硬约束条件和软约束条件的规约,使约束条件具有高度的可配置性。

参考文献:

- [1] Gotlieb. The Construction of Class-Teacher Time Tables [J]. Proceeding IFIP Congress, Amsterdam, 1963: 73-74.
- [2] S. Even, A. Itai, A. Shamir. On the complexity of timetable and multi-commodity flow problems [J]. SIAM Journal on Computing 5, 1976: 691-703.
- [3] Tim B. Cooper, Jeffrey H. Kingston. The Complexity of Timetable Construction Problems [J]. Proc. ICPTAT'95, 1995: 183-295.
- [4] Ferland, J. A., Roy, S. Timetabling problem for university as assignment of activities to resource [J]. Computer and Operations Research, 1985, Vol. 12, No. 2: 207-218.
- [5] 吴金荣. 求解课程表问题的分支定界算法 [J]. 运筹与管理, 2002 (1).
- [6] 赵光哲. 大学排课问题中的遗传算法设计 [J]. 延边大学学报 (自然科学版), 2006 (1).
- [7] 蒲保兴. 基于遗传算法的排课算法 [J]. 中央民族大学学报 (自然科学版), 2006 (1).
- [8] 田岭. 大学自动排课算法设计与实现 [J]. 教育信息化, 2006 (7).

(责任编辑: 杜能钢)

Research of Course Scheduling Based on Ontology Mapping and Rule Inference

Abstract: First of all, course scheduling problems are described. Next some course scheduling methods are listed. Then these methods are analyzed and compared. Course scheduling ontology is created through OWL ontology technology. Data synchronization is fulfilled through ontology mapping method. Ontology constraint is made through SWRL rules. Rule inference is practiced through JESS engine. Research of total course scheduling modeling structure is given integrating with course scheduling algorithm.

Key Words: Course Scheduling; OWL; Ontology Mapping; SWRL; Rule Inference