

中国古建的语义网络知识库及其高效实现

白卫静¹, 张松懋², 刘椿年¹

(1. 北京工业大学 计算机学院, 北京 100124; 2. 中国科学院 数学与系统科学研究院, 北京 100190)

摘要: 将全过程计算机辅助动画自动生成技术应用于中国古建筑领域, 实现了一个自动古建筑动画系统. 它根据用户对古建的描述, 自动生成三维动画来表现古建的搭建过程, 整个自动过程都是在古建筑知识库的支持下完成的, 采用语义网络技术设计并实现了知识库, 包括本体库和规则库 2 部分. 主要论述古建筑知识库的结构、本体和规则库的组成, 并以生成古建筑搭建顺序为例来说明规则推理的过程, 共实现了 2 个方案, 一个基于本体构造环境 Protégé, 另一个基于规则推理系统 Jess, 后者比前者在推理时间上节省了 99% 以上, 重点论述了高效方案的设计思想和实现技术, 并分析了其优点和不足.

关键词: 计算机辅助动画自动生成; 语义网络; 本体; 规则; Protégé; Jess

中图分类号: TP182 **文献标志码:** A **文章编号:** 1673-4785(2010)06-0510-12

A knowledge base of traditional Chinese architecture and its efficient implementation demonstrated through

BAI Wei-jing¹, ZHANG Song-mao², LIU Chun-nian¹

(1. College of Computer Science, Beijing University of Technology, Beijing 100124, China; 2. Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In applying the full life-cycle automatic animation technology, an automatic animation system was designed and implemented for traditional Chinese architecture. Based on the user's description of a traditional building, the system generated an animation demonstrating the construction of the building. The whole automation was supported by a domain knowledge base. Semantic web technology was used to implement the knowledge base, including ontological knowledge and rules. The structure and components of the ontology and rule base were presented. Moreover, the inference process through using rules to reason about the construction sequence of wooden components was illustrated. Two implementations were given: Protégé-based and Jess-based. The latter saved up to 99% more of the reasoning time than the former. The details of the Jess-based knowledge base were presented, and its merits and drawbacks were analyzed.

Keywords: computer-aided automatic animation; Semantic Web technology; ontology; rules; Protégé; Jess

起源于哲学的本体论(ontology)在信息科学领域一度引起广泛关注^[1,2], 其重要性也已在很多方面表现出来并得到广泛认同. 尤其是语义网(Semantic Web)^[3,4]的诞生, 在 W3C 组织的主导下有望解决网络信息共享的语义问题, 从而实现因特网内的知识共享和智能信息集成. 随着语义网技术的不断发展, 越来越多的本体知识库被建立起来, 用本体知识库来表示相关领域知识逐渐成为大势所趋.

本文的研究基础是陆汝钤院士提出的全过程计算机辅助动画自动生成技术^[5], 它将人工智能技术与现代多媒体技术相结合, 以底层图形学技术为支

撑, 在上层运用人工智能技术以及电影艺术实现了一个全新的动画片生产自动化的技术路线. 它的目标是只要有了一个适当的故事, 以受限自然语言的形式把它输入到计算机里, 从此时开始, 直到最终动画生成, 每一步都是在计算机的辅助下完成的. 将全过程计算机辅助动画自动生成技术应用到中国古代建筑领域, 建立了古建筑动画自动生成系统, 用户输入他想要的古建的木架结构, 系统则自动生成三维动画, 表现这个建筑一步一步的搭建过程. 为了支持这样的古建筑动画自动生成, 构建了古建筑知识库系统, 实现了对搭建顺序的自动推理、以及对每根木头的尺寸和空间位置的自动计算; 特别地, 面向知识共享和互操作, 应用语义网络技术实现了整个古建筑知识库系统, 包括建立了本体库和规则库, 并针对使用多种

收稿日期: 2010-03-05.

基金项目: 国家科技支撑计划资助项目(2006BAK31B04).

通信作者: 白卫静. E-mail: estherbaiweijing@gmail.com.

本体构建工具导致推理效率低的问题,提出了一个高效的推理实现方案.

1 研究背景

1.1 全过程古建筑动画自动生成系统

我国古代建筑文化遗产极为丰富,中国古建筑形式多种多样,归结起来有硬山、悬山、歇山、庑殿、攒尖 5 种基本形式,在这几种最基本的建筑形式中,又细分成具有不同特点的建筑,如庑殿有单檐庑殿、重檐庑殿,硬山和悬山既有 1 层,也有 2 层楼房等^[6].

尽管古代建筑形式纷繁复杂,但各个部位都有较为固定的比例关系,这些比例关系是古建筑设计 与施工共同遵循的法则.掌握这些法则,对其进行整理加工,可以用计算机来重现这些古建筑的建造过程.目前古代建筑领域的数据库或知识库,大多都是

针对某些具体的建筑或构件而建立的查询系统,并未涉及到古代建筑的搭建过程,也没有把这一搭建过程用动画形式表现出来.

古建筑动画自动生成系统填补了这个空白,它是全过程计算机辅助动画自动生成技术在古建领域的应用,以类自然语言的形式把古建结构的信息输入到计算机里,从此时开始,直到最终生成古建搭建动画,每一步都是在计算机的辅助下完成的.系统框架流程如图 1 所示,具体步骤包括对用户输入的古建结构描述的信息抽取、建筑类型和所需构件及其数目的推导、构件搭建顺序的推理、每个构件的尺寸和空间位置的计算、定性动画场景描述生成、榫卯计算和生成、定量动画场景描述生成和最后可以用于直接渲染的动画文件的生成.

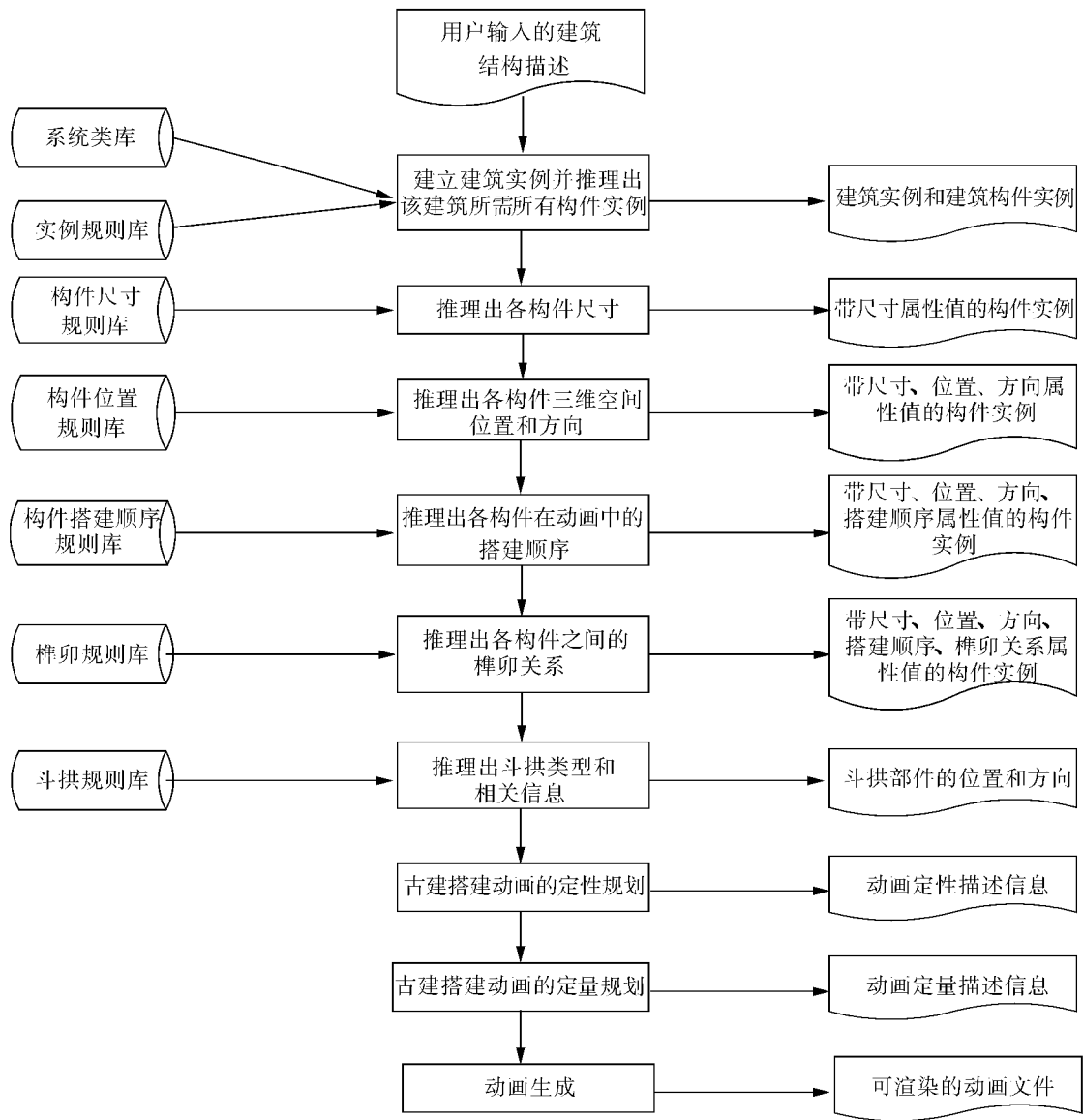


图 1 古建筑动画自动生成系统框架流程图

Fig. 1 Flow chart of the automatic animation system for Chinese traditional architecture

建筑类型和所需构件及其数目的推导、构件搭建顺序的推理、构件尺寸和空间位置的计算和榫卯生成等,都需要在**本体知识库系统的支持下实现. 目前本体的研究非常热,领域本体的覆盖范围非常广,但在中国**领域,涉及**木架结构搭建方面的知识库就很少,更是没有发现符合本项目需求的关于建筑中每根木头构件的知识库,因此需要建立一个面向**动画自动生成系统的知识库.

1.2 语义网络技术

1998年,T. B. Lee首次提出语义网络的概念,在他给出的语义网络体系结构中(如图2),构建在Unicode与URI、XML、RDF(S)等语言标准之上的本体层(ontology vocabulary)起着关键的作用,其提供的丰富的表达能力不仅能用来描述领域的概念模型,还是对知识进行推理和验证的基础^[7].

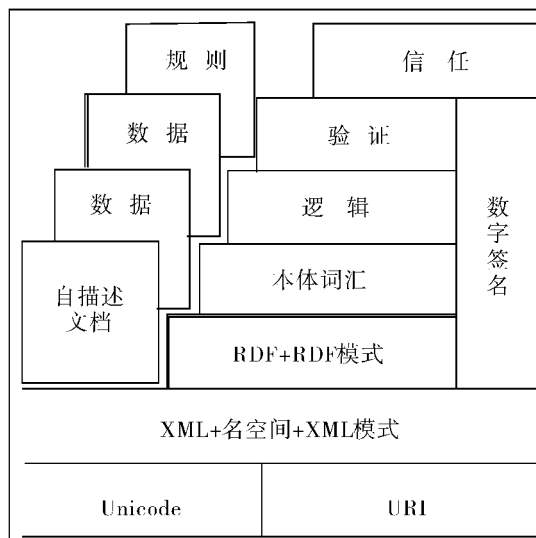


图2 语义网络体系结构

Fig. 2 Semantic Web structure layer

本体赋予网络上的信息以语义含义,因此本体作为语义载体在语义网络中起着核心的作用,是语义网络基于语义的互操作得以实现的关键因素. 本体是对领域概念的规范化表示,作为知识和概念描述的重要工具,可以清晰描述领域知识库中的概念及其关系,实现领域知识的共享和重用,也有利于领域知识库的管理和维护. 基于领域本体,可以对领域内的资源在语义层次上进行表述,使得领域内的资源从内容级别上升到语义级别,使得资源的管理和利用更加地有效和智能化. 由此,本体的构建也就成为语义网络实现的重要步骤.

正是由于其在知识共享和资源重用方面的优势,在**动画自动生成系统中采用了基于语义网络的本体这一技术来实现知识库系统的构建.

本体应用的研究领域非常广泛,包括智能信息检索、数字图书馆、网络教学、企业本体、电子邮件过滤、文档归类、文体求解、需求分析、知识管理、信息集成、机器翻译等. 另外,国家也在“973”计划、国家自然科学基金等项目中支持了基于ontology的信息集成、数字图书馆、信息检索等应用型的技术,这些项目也都以ontology技术作为核心^[8]. 本体知识库采用本体知识表示的方式,以本体作为某个领域概念、属性以及概念间内在关系的形式化表达,事实和概念的建模体现在本体知识上,进一步可以用本体概念和关系进行规则地组织,形成规则库,在**动画知识库系统中就创建了各种规则库,并利用本体推理和规则推理实现系统功能. 目前大多数的知识库只是进行本体构建和推理,并不包括规则库,未使用规则推理.

本体部分目前常用的编辑工具主要有斯坦福大学开发的Protégé系列^[9]、马德里技术大学开发的WebODE^[10]、卡尔斯鲁厄大学开发的OntoEdit^[11]和曼彻斯特大学构建的OilEd^[12]等. 这些工具各有偏重,其中斯坦福大学的Protégé系列因其界面友好、简单易学、开放源码等特性成为国内外众多本体研究机构的首选工具. 目前比较常用的本体语言的主要有XML、RDF、OIL、DAML+OIL、OWL等,其中OWL已经成为本体的标准语言,最新的版本是OWL 2. 比较常用的针对本体的规则语言包括ruleML(rule makeup languages)和SWRL(semantic web rule language),SWRL是以OWL语言为基础的规则描述语言. 目前本体推理机有Fact++、Pellet、RacerPro等,它们还支持基于本体的规则推理,另外较常用的支持规则推理的还有通用推理引擎Jess、Jena等.

2 古建筑动画自动生成系统中的知识库系统

2.1 古建筑领域知识总结

马炳坚先生在《中国古建筑木作营造技术(2版)》中曾提到:“研究古建筑,首先应当了解这些不同形式建筑的不同构造,了解每种构造都是由哪些构件组成的,它们在整体中起着什么作用. 这些构件又是怎么有机地组合起来,构成相应的建筑形式的. 只有了解了这些内容,才能指导能动地认识古建筑”.

古建搭建过程应该基于具体的建筑构件,针对古建的木架结构,把每根木头作为建筑基本构件单位,并根据它们在搭建过程中起的作用,抽象出4个

属性,形成一个描述建筑构件的四元组,包括构件名、垂直方向所跨过的层列表、面宽方向所跨过的缝列表、进深方向所跨过的缝列表.图3是古建动画系统自动生成的间数为5、檩数为7的有周围廊的庑殿建筑,其中((金柱),(1,1),(6,6),(3,3))代表图中用椭圆标识出来的一个构件,金柱表示的是建筑构件的名称,(1,1)表示这根金柱在竖直方向上是第1层,位于最底层,(6,6)表明这根金柱在面宽方向上是位于第6缝的位置,(3,3)表示这根金柱在进深方向上位于第3缝的位置;

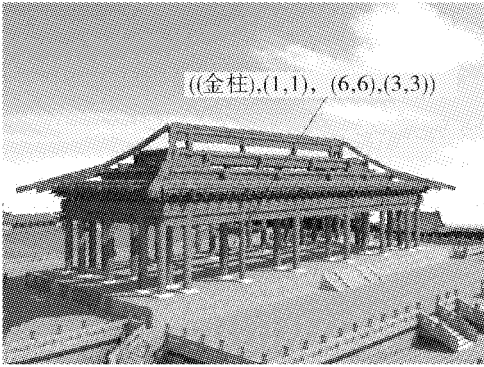


图3 5间7檩有周围廊庑殿木架结构的构件示意图
Fig.3 Wooden structure of a Wudian style building with five sections from left to right and seven purlins from front to rear

古建知识库是古建动画自动生成系统的重要组成部分,它的功能主要包括:1)把古建专家总结出来的领域知识在知识库中以本体库和规则库的形式

表示出来.2)接收用户输入的对于他想要的建筑的描述参数,如建筑制式、建筑特性、间数、檩数、柱径、柱高等,在本体库和规则库中选取适当的模块加载,由推理机推理,得到建筑的建筑构件信息,包括构件名称、数目、构件尺寸、空间位置和方向、搭建顺序等,并将这些信息以特定的形式输出给自动动画系统的下一个部分.

归纳总结出来了硬山、歇山、庑殿的领域知识,其中,庑殿建筑是中国古建筑中的最高型制.在等级森严的封建社会,这种建筑形式常用于宫殿、坛庙一类皇家建筑,是北京中轴线上主要建筑最常采用的形式,如故宫午门、太和殿、乾清宫等都是庑殿式建筑.庑殿建筑实际上已经成为皇家建筑中独有的一种建筑形式,皇家建筑之外的建筑是不允许采用这种建筑形式的,庑殿建筑的这种特殊政治地位决定了它制式复杂,用材硕大、体量雄伟、装饰华贵富丽^[6].这表明在本体知识库的实现过程中,庑殿的难度是最大的,因此在下文中,选用庑殿建筑为例来论述知识库系统的设计和实现细节.

2.2 知识库系统结构

将古建知识体现在知识库中,包括2个部分:古代建筑本体库和古代建筑规则库,推理机基于这2个部分运行,得到系统需要的信息.其中类(概念)和实例体现在本体库上,各种规则存在于规则库中,如图4所示.

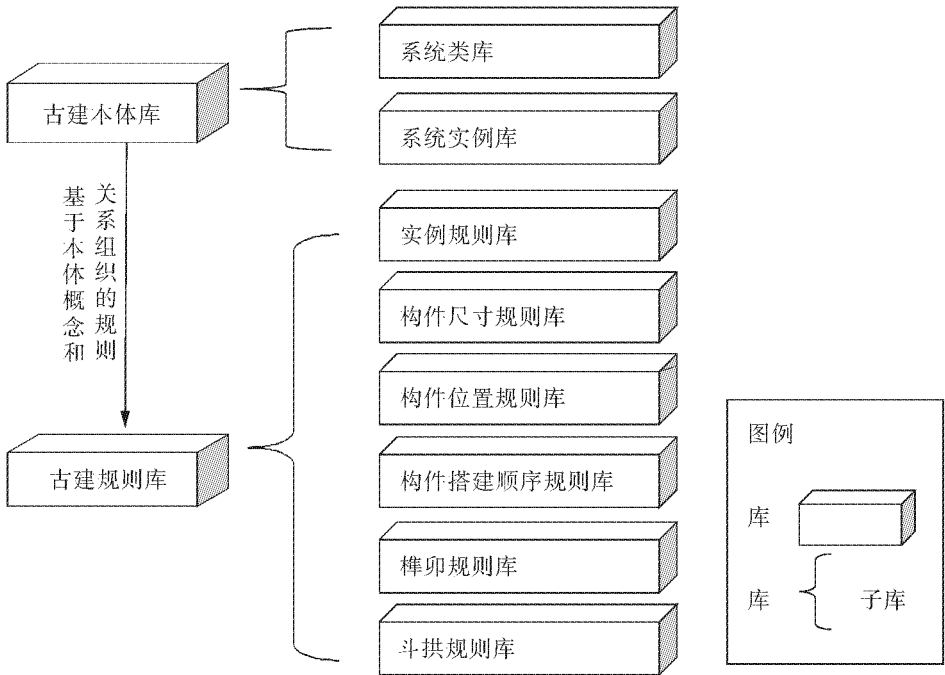


图4 古建知识库系统结构
Fig.4 Components of the knowledge base for Chinese traditional architecture

2.2.1 古建本体库

本体库包括2个子库:系统类库和系统实例库.

1) 系统类库.

系统类库存储类及类的属性,主要包括建筑制式类、建筑构件类.

建筑制式类的子类包括:硬山、庑殿、歇山等,这个类的子类可以随着领域专家的知识总结而不断扩充.其属性包括建筑制式的间数、檩数等通用属性,同时,每个子类都还包括自己特殊的一些属性,如:硬山的有无前后廊的属性,庑殿的有无周围廊的属性等.

建筑构件类的子类包括:金柱、檐柱、小额枋、桁枋、大额枋等,同样地,这个类的子类也可以随着领域知识的学习而扩充.如前文提到的,按照领域专家的总结,建筑构件可以用((构件名),(垂直方向所跨过的层列表),(面宽方向所跨过的缝列表),(进深方向所跨过的缝列表))的四元组来表示.

2) 系统实例库.

在系统未运行的时候实例库是空的,根据用户的输入,经过推理机基于本体和规则的推理,系统动态地在实例库中建立对应的建筑实例和建筑构件实例.

2.2.2 古建规则库

规则库包括6个子库:实例规则库、构件尺寸规则库、构件位置规则库、构件搭建顺序规则库、榫卯规则库和斗拱规则库.

1) 实例规则库.

基于本体库和实例库的规则,推理机推理出用户输入建筑所需要的各种建筑构件,及其具体建筑构件的数目,然后在系统实例库中建立该建筑制式的一个具体实例和其所需建筑构件的相应数目的实例.

2) 构件尺寸规则库.

基于系统实例库和尺寸规则,推理机推理出用户输入建筑中各个建筑构件的具体尺寸大小,同时将其更新写入系统实例库.

3) 构件位置规则库.

基于系统实例库、空间位置和方向规则,推理机推理出用户输入建筑中各个建筑构件的三维位置和方向,同时将其更新写入系统实例库.

4) 构件搭建顺序规则库.

基于系统实例库和搭建顺序规则,推理机推理出用户输入建筑中各个建筑构件在搭建动画中出现的先后顺序,同时更新系统实例库中各个相关构件

的顺序属性.

5) 榫卯规则库.

基于系统实例库和榫卯规则,推理机推理出用户输入建筑中建筑构件之间的榫卯关系,同时将其更新写入系统实例库.

6) 斗拱规则库.

基于系统实例库和斗拱规则,推理机推理出用户输入建筑中的斗拱类型和相关信息,同时将其更新写入系统实例库.

2.2.3 古建知识库推理机

以一个5间7檩有周围廊庑殿为例,说明知识库系统是如何基于本体库和实例规则库运行并建立相关建筑实例和构件实例的.首先用户输入的参数被传递给知识库系统,知识库系统会在系统类库中找到有周围廊庑殿的类,基于有周围廊庑殿的实例规则库,建立一个5间7檩的有周围廊庑殿实例;然后基于此实例和构件个数规则推理出这个建筑需要22根檐柱、14根金柱、22个小额枋等等,相应地系统建立22个檐柱实例、14个金柱实例、22个小额枋实例等等,并在这些构件实例和庑殿实例之间建立相应的所属关系,如图5所示.

3 基于 Jess 的知识库系统

该文的古建知识库系统全部用 Jess 实现. Jess (Java expert shell system) 是 1995 年由美国 Sandia 国家实验室分布式系统计算组成员 E. J. Friendman-Hill 用 Java 实现的一个经过扩充的 CLPIS 版本. Jess 是性能良好的开放式推理机,原则上可以处理各种领域的推理任务,只要系统能够为 Jess 提供这个领域的特定领域规则和事实信息. Jess 的规则匹配采用经典的 RETE 算法,以牺牲大量的存储空间来换取匹配的高速性. RETE 算法利用了专家系统中时间冗余性和结构相似性这 2 个特点,有效地减少了用于匹配操作的次数,因此,当系统的性能是由匹配算法的质量决定时, Jess 的优点将更加明显. 应当指出的是, RETE 算法是一个以空间换取时间的算法,所以,应用 Jess 时应当考虑内存的消耗^[13].

根据古建动画自动生成系统的需要,把所有总结得到的古建知识写入知识库的本体库和规则库中,知识库系统根据用户输入的具体参数(如 5 间 7 檩有周围廊的庑殿),进行下列推理:

1) 基于实例规则库的规则,推理出 5 间 7 檩有周

围廊庑殿建筑所需要的所有的建筑构件类别,和各构件类别的具体构件个数;然后在系统实例库中建立一个庑殿类的实例,和相应个数的构件类的实例。

2) 基于庑殿实例和构件实例,以及构件位置规则库,推理出各个构件实例的位置属性值,如所在层从第几层到第几层,面宽缝从第几缝到第几缝,进深缝从第几缝到第几缝等等,并用这些属性值更新系统实例库。

3) 基于庑殿实例和构件实例,以及构件尺寸规则库,推理出各个构件实例的尺寸等信息,同时更新

系统实例库。

4) 基于庑殿实例和构件实例,以及构件搭建顺序规则库,推理出各个构件在动画中的搭建先后顺序,同时将这些信息写入系统实例库。

5) 当动画搭建过程需要的构件信息生成完毕,将实例库中的实例及其属性值导出,以适当的形式输出给古建筑动画自动生成的下一个部分。

下面将分别介绍知识库系统的模块组成、类库的 Jess 实现、规则库的 Jess 实现、搭建顺序推理的 Jess 实现和整个知识库系统的运行流程。

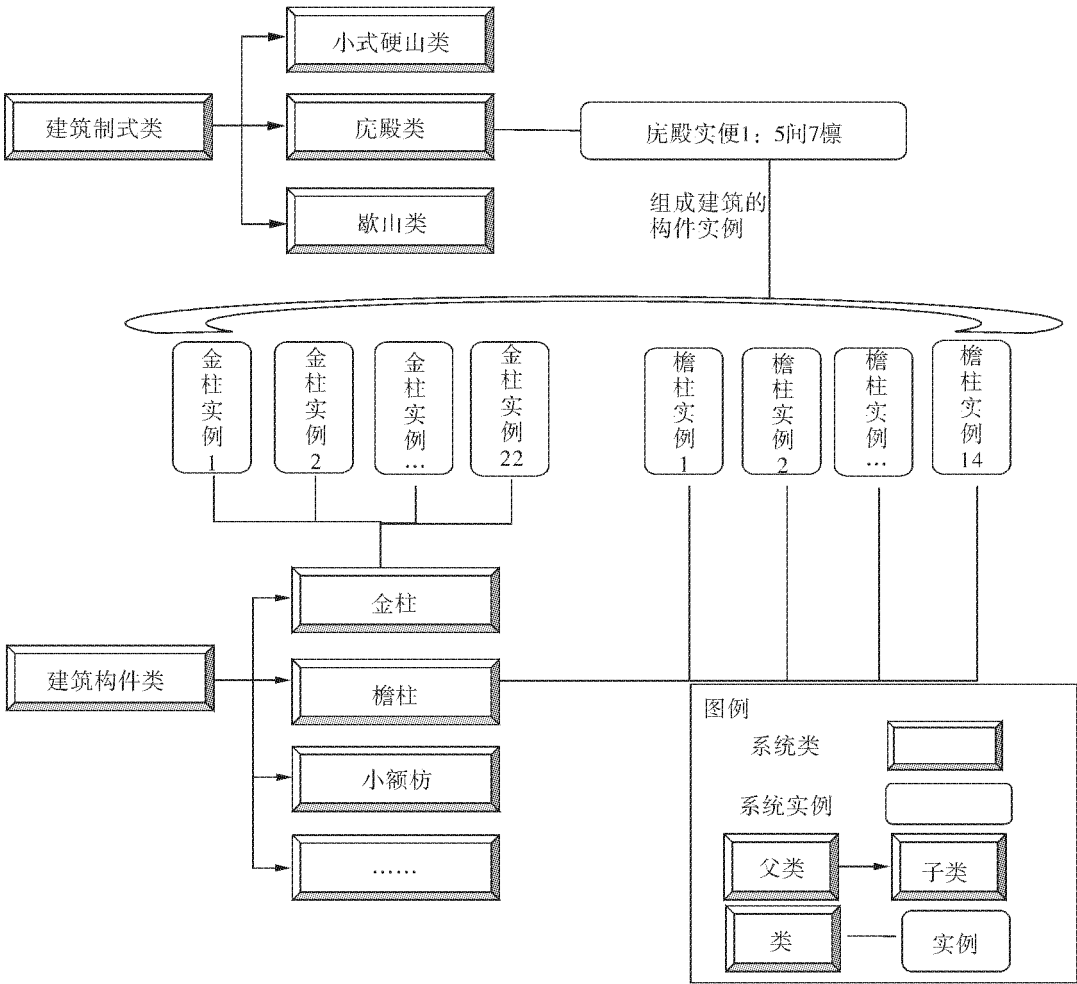


图 5 5 间 7 檩有周围廊庑殿实例规则运行结果示意图

Fig. 5 An example of generating individuals for the ontology

3.1 模块划分

基于图 4 的知识库系统的系统结构,为了方便操作和提高系统效率,把各个子库系统按照功能划

分模块,每个子库中都基于不同的建筑制式有自己不同的模块,如表 1 所示。

表1 知识库系统模块设计
Table 1 Modules in the knowledge base

子库名称	具体模块
系统类库	类的构架和定义模块
系统实例规则库	包括:硬山有廊实例规则模块、硬山无廊实例规则模块、有周围廊庑实例规则模块、重檐庑实例规则模块等
构件位置规则库	包括:硬山有廊各构件位置规则模块、硬山无廊各构件位置规则模块、有周围廊庑各构件位置规则模块、重檐庑各构件位置规则模块等
构件尺寸规则库	包括:硬山有廊各构件尺寸规则模块、硬山无廊各构件尺寸规则模块、有周围廊庑各构件尺寸规则模块、重檐庑各构件尺寸规则模块等
构件搭建顺序规则库	包括:硬山有廊顺序规则模块、硬山无廊顺序规则模块、有周围廊庑顺序规则模块、重檐庑顺序规则模块等

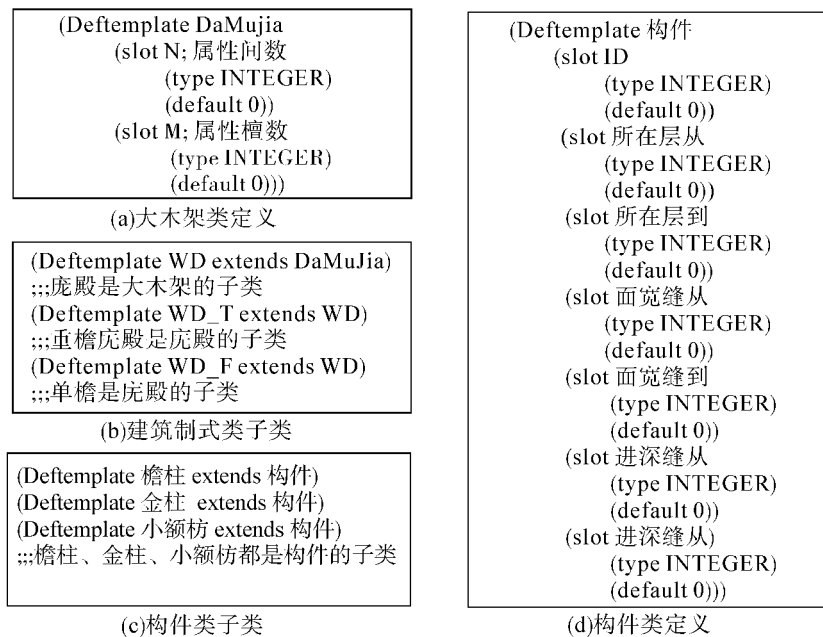


图6 类的定义

Fig. 6 Some class definitions

当用户提出一个建筑需求时,系统根据输入参数、自动选择相应的模块加载,从而节省内存空间的使用,而且效率高。例如,当用户参数表明只需要推理有周围廊庑殿的搭建顺序时,系统会自动屏蔽其他无用的模块,而只选择以下3个相关模块加载:

- 1) 系统类库(所有用户需求都需要加载的);
- 2) 有周围廊庑殿实例规则模块;
- 3) 有周围廊庑殿搭建顺序规则模块。

完成加载之后,系统开始推理,然后输出结果。

3.2 类库的实现

在 Jess 中用 deftemplate 来定义 DaMuJia(大木架)类(如图6(a)所示),其中 WD(庑殿)、XS(歇山)、XSYS(硬山)都是其子类(如图6(b)所示)。图6(d)所示是构件类的定义,属性有:ID、所在层从、所在层到、面宽缝从、面宽缝到、进深缝从、进深缝

到,这些属性的含义在前文中已经有过介绍。各种构件是构件类的子类,如金柱、檐柱、小额枋等,在 Jess 中如图6(c)表示。

3.3 实例规则库的实现

实例规则库根据领域知识在实例库中添加建筑实例和构件实例,以有周围廊庑殿的建筑型制为例,它涉及25种建筑构件,每种构件个数不相同,实例规则库根据具体的建筑实例参数来确定各个构件的个数,并为各个构件生成相应个数的实例,同时,给每个具体实例的ID属性赋值。下面给出建立构件实例的一个例子,根据学习古建领域知识的同学的总结,所有在面宽方向上第 $(J'+1)$ 缝到第 $(J-J')$ 缝的金柱的四元组表达是:

(金柱, (1,1), (j,j), (3,3)),
(金柱, (1,1), (j,j), (M,M))。

式中: $M = \text{檩数}$; $J = \text{间数} + \text{檩数} + 2$, $J' = 1 + (\text{檩数} + 1)/2$, 而 j 表示面宽方向上从第 $(J' + 1)$ 缝到第 $(J - J')$ 缝的所有值. 用如下 Jess 规则在实例库中建立这些金柱的实例, 其中 $jzid$ 是用来赋值给 ID 属性的变量, $bind$ 是 Jess 中绑定变量和变量值的语句.

```
(bind ?j (+ ?J' 1))
(bind ?j zid 1)
(while (<= ?j (- ?J ?J')) do
  (assert (JinZhu (ID ?j zid) (LayerFrom 1)
    (LayerTo 1) (MianFengFrom ?j) (MianFengTo ?j)
    (JinFengFrom 3) (JinFengTo 3)))
  (assert (JinZhu (ID (+ ?j zid 1)) (LayerFrom
    1) (LayerTo 1) (MianFengFrom ?j) (MianFengTo ?
    j) (JinFengFrom ? * M *) (JinFengTo ? * M
    *)))
  (bind ?j (+ ?j 1))
  (bind ?j zid (+ ?j zid 2))
)
```

3.4 搭建顺序推理的实现

```
(Defquery 0cengShangJia3feng
  (declare(variables ?x))
  ?f1 <-(ZhuChu(LayerFrom 0)(LayerTo 0)(MianFengFrom 3)(MianFengTo
    3)(JinFengFrom ?a&:(eq ?a(+(*?2)1)))(JinFengTo ?a))
  ?f2 <-(ZhuChu(LayerFrom 0)(LayerTo 0)(MianFengFrom 3)(MianFengTo
    3)(JinFengFrom ?*M*)(JinFengTo?*M*))
  ?f3 <-(ZhuChu(LayerFrom 0)(LayerTo 0)(MianFengFrom 3)(MianFengTo
    3)(JinFengFrom 2)(JinFengTo 2))
  ?f4 <-(ZhuChu(LayerFrom 0)(LayerTo 0)(MianFengFrom 3)(MianFengTo
    3)(JinFengFrom ?*M1*)(JinFengTo?*M1*))
)
```

图7 0层上架第3缝顺序规则

Fig.7 The first example of rules for inferring construction sequence

```
(Defquery 1cengShangJia2feng
  (declare (variables ?x))
  ?f1<-(PingBanFeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom 2)(JinFengTo 2))
  ?f2<-(PingBanFeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom ?*M1*)(JinFengTo?*M1*))
  ?f3<-(PingBanFeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom a&:(eq ?a(+(*?2)1)))(JinFengTo ?b&:(eq ?b(+?a2))))
  ?f4<-(PingBanFeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom 2)(JinFengTo 3))
  ?f5<-(PingBanFeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom ?*M*)(JinFengTo?*M1*))
  ?f6<-(ZhengXinHeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom 2)(JinFengTo 2))
  ?f7<-(ZhengXinHeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    3)(JinFengFrom ?*M1*)(JinFengTo?*M1*))
  ?f8<-(ZhengXinHeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    2)(JinFengFrom ?a)(JinFengTo ?b))
  ?f9<-(ZhengXinHeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    2)(JinFengFrom 2)(JinFengTo 3))
  ?f10<-(ZhengXinHeng(LayerFrom 1)(LayerTo 1)(MianFengFrom 2)(MianFengTo
    2)(JinFengFrom ?*M*)(JinFengTo?*M1*))
)
```

图8 1层上架第2缝顺序规则

Fig.8 The second example of rules for inferring construction sequence

关于构件尺寸、位置、搭建顺序、榫卯和斗拱,分别建立了规则库,它们虽然功能不同,但实现和运行的方式是基本一致的,下面以搭建顺序规则库中的庀殿构件顺序模块来说明规则库的实现方法.

基于已经推理出来的建筑构件实例,应用古建筑领域知识,推理出各个构件搭建的具体先后顺序,是顺序规则模块需要实现的内容. 按照建筑的搭建过程,顺序推理规则在建筑垂直方向上分为4个部分:0层、底层(1层)、顶层和中间层,其中0层、底层、顶层都是单独的一个层次,中间层根据具体建筑规模的不同有具体的数值,如1层、3层、5层等.

具体实现方法是在 Jess 的规则模块中定义规则,不同的搭建层次有不同的规则,然后在 Java 程序中调用 Jess 推理机进行规则推理,得到具体某层某缝的搭建中用到的所有构件的先后顺序. 因为0层、1层、顶层的规则相似,它们在 Java 程序中由同一个函数处理,而中间层因为处理的具体层数不是固定的,对这段规则的调用方法不同.

图7~10给出了各层具体顺序规则的示例.


```

(defquery 1cengShangJia2feng
  (declare (variables ?x)

    ?f1<-(TaiPingLiang(LayerFrom ?*M5*)(LayerTo ?*M5*)(MianFengFrom ?a&:(eq ?a ?x))(Mian
    FengTo ?a)(JinFengFrom ?*/1*)(JinFengTo ?*/3))
    ?f2<-(LeiGongZhu(LayerFrom ?*M5*)(LayerTo ?*M5*)(MianFengFrom ?a)(MianFengTo ?a)(Jin
    FengFrom ?*/2*)(JinFengTo ?*/12*))
    ?f3<-(HengFang(LayerFrom ?*M5*)(LayerTo ?*M5*)(MianFengFrom ?a)(MianFengTo ?a(Mian
    FengTo ?b&:(eq ?b(+?a1)))(JinFengFrom ?*/2*)(JinFengTo ?*/2*)
  )

```

图9 顶层下架第1缝顺序规则

Fig. 9 The third example of rules for inferring construction sequence

```

(Defquery midcengXialiajfeng2
  (declare(variables ?x?i))

  ?f1<-(GuaZhu(LayeFrom ?u&:(eq ?u?i))(LayerTo ?u)(MianFengFrom?a&:(eq ?a?x))(Mian
  FengTo ?a)(JinFeng From ?u2(+?u2)))(JinFengTo ?u2))
  ?f2<-(GuaZhu(LayeFrom ?u)(LayerTo ?u)(MianFengFrom?a)(MianFengTo ?a)(JinFengFrom
  ?u3&:(eq ?u3(-?*M1*?i)))(JinFengTo ?u3))
  ?f3<-(HengFang(LayerFrom ?u)(LayerTo ?u)(MianFengFrom ?a)(MianFengTo ?b&:(eq ?b
  (+?a1)))(JinFengFrom ?u2)(JinFengTo ?u2))
  ?f4<-(HengFeng(LayerFrom ?u)(Layer To ?u)(MianFengFrom ?a)(MianFengTo ?b)(JinFeng
  Form ?u3)(JinFengTo ?u3))
  ?f5<-(PaLiang(LayerFrom ?u)(LayerTo ?u)(MianFengFrom ?u1&:(eq ?u1(+?u1)))(MianFeng
  To ?*M2*(JinFengFrom ?u2)(JinFengTo ?u2))
  ?f6<-(PaLiang(LayerFrom ?u)(Layer To ?u)(MianFengFrom ?u1)(MianFeng To ?*M2*)(JinFeng
  ?u3)(JinFengTo ?u3))
  ?f7<-(HengFeng(LayerFrom ?u)(LayerTo ?u)(MianFengFrom ?u2)(MianFengTo ?a)(JinFengFrom
  ?u2)(JinFengTo ?u2))
  ?f8<-(HengFeng(LayerFrom ?u)(LayerTo ?u)(MianFengFrom ?u2)(MianFengTo ?a)(JinFengFrom
  ?u3)(JinFengTo ?U3))
)

```

图10 中间层下架第2缝顺序规则

Fig. 10 The fourth example of rules for inferring construction sequence

图7~10分别是0层上架第3缝顺序规则、1层上架第2缝顺序规则、顶层下架第1缝顺序规则,和中间层下架第2缝顺序规则.以0层上架第3缝顺序规则为例,其中“0cengShangJia3feng”是规则名,其中(declare (variables ?x))说明了一个名为?x的变量,在程序中调用的时候,0层上架构件第3缝中的构件个数值会通过?x传递给推理规则.

图11给出了Java程序中调用Jess推理机进行规则推理的代码,其中result = engine.runQueryStar(sRule, new ValueVector().add(num))是调用规则进行推理的语句,sRule是调用的规则段的名字,如“0cengShangJia3feng”,num即是传递给?x的具体的参数.推理结束后,Jess把结果实例存放在缓存区中,通过String name = f.getSlotValue(“Name”).stringValue(null)可以得到实例中的“Name”(即名称属性存放在String型变量name中),存放在不同的变量中的其他属性也可以类似得到.

3.5 知识库系统的运行流程

以有周围廊庑的搭建顺序推理过程为例,说明整个知识库系统的推理实现过程.根据前面的论

述,对于有周围廊的庑殿,系统会自动屏蔽其他无用的模块,而只选择以下3个相关模块加载运行:

- 1) 系统类库(所有用户需求都需要加载的);
- 2) 有周围廊庑实例规则模块;
- 3) 有周围廊庑顺序规则模块.

2) 模块的规则运行时,根据具体的建筑实例参数来计算出各个构件的个数,并为各个构件构造相应个数的实例.一个5间7檩的有周围廊的庑殿共有560个具体的构件实例,其中14个金柱、22个檐柱、22个小额枋等等;而11间13檩的有周围廊庑殿有1076个建筑构件实例,每个实例都是构件类的实例,属性包括:ID、所在层从、所在层到、面宽缝从、面宽缝到、进深缝从、进深缝到.

3) 模块的规则运行时,Java程序调用知识库中Jess不同的规则模块,基于2)部分的实例结果,利用搭建顺序规则进行推理,得到结果以规定的形式写到XML文件中,作为系统下个部分的输入.根据专门学习古建筑知识的学生归纳总结出来的领域知识,庑殿的搭建是分层进行的,其中每个层又分为上架、下架、由戗、角梁等,具体到每一架的搭建又按照

不同的缝来进行,同一架的不同缝之间的搭建是相互独立、互不影响的.据此,把庀殿构件顺序规则模

块分解成互不影响互相独立的26个子模块,具体如表2所示.

```
try{
    result=engine.runquerystar(srule,new valuevector().add(num));

    if(result.next()){
        int i=1
        {
            while(i<nurmofofinstance)
            {
                string fi= "f" +i;
                system.out.println(fi);
                integer id=result.get(fi).intvalue(null);
                fact f=engine.findfactbyid(id);

                string name=f.getslotvalue( "name" ).stringvalue(null);
                string lf=f.getslotvalue( "layerfrom" ).stringvalue(null);
                string lt=f.getslotvalue( "layerto" ).stringvalue(null);
                string mf=f.getslotvalue( "mianfengfrom" ).stringvalue(null);
                string mt=f.getslotvalue( "mianfengto" ).stringvalue(null);
                string jf=f.getslotvalue( "jinfengfrom" ).stringvalue(null);
                string jt=f.getslotvalue( "jinfengto" ).stringvalue(null);

                i++;
            }
        }
    }
}
```

图 11 调用规则推理的 Java 代码

Fig. 11 Java codes for starting the rule inference

表 2 庀殿顺序部分推理规则模块划分

Table 2 Modules of rules for inferring construction sequence

层次	架	推理规则模块名称
第 0 层	下架	0 层下架第 1 缝
		0 层上架第 2 缝
		0 层上架第 3 缝
	上架	0 层上架第 j 缝 1
		0 层上架第 j 缝 2
第 1 层	下架	1 层下架第 2 缝
		1 层下架第 3 缝
		1 层下架第 j 缝 1
		1 层下架第 j 缝 2
	上架	1 层上架第 2 缝
		1 层上架第 3 缝
		1 层上架第 j 缝 1
		1 层上架第 j 缝 2
中间层	角梁	1 层角梁第 2 缝
	下架	中间层下架第 j 缝 1
		中间层下架第 j 缝 2
		中间层下架第 j 缝 3
	上架	中间层上架第 j 缝 1
		中间层上架第 j 缝 2
	由戗	中间层上架第 j 缝 3
		中间层由戗第 j 缝

层次	架	推理规则模块名称
顶层	下架	顶层下架第 j 缝 1
		顶层下架第 j 缝 2
		顶层上架第 j 缝 1
	上架	顶层上架第 j 缝 2
		顶层由戗第 j 缝
	由戗	顶层由戗第 j 缝

4 实验结果

在整个古建本体知识库的实现过程中,使用 Jess 和 Java 完成了所有的功能,而没有选用在本体研究领域应用广泛的 Protégé 系列工具.

在古建知识库的第 1 版^[14]中,使用了 Protégé 系列工具,具体选用 Protégé 3. 4 作为本体编辑工具,使用 OWL DL 作为本体描述语言,使用 SWRL 语言来描写规则,选用 Jess 作为推理机,驱使规则和 OWL 知识本体结合. 不同的部分使用了不同的专业工具,使得整个知识库系统搭建过程是递增的. 然而,当系统运行时出现了内存溢出的问题. 为了解决内存溢出的问题,使系统可以顺利运行,把系统中原本应该一次推理的内容分成几次进行,这种采用时间换取空间的方法,使推理时间变得非常长. 表 3 的左边部分列出了带周围廊庀殿的不同制式的搭建顺序推理的时间,运行环境是 CPU2. 8GHZ、内存 1.5 GB 的微机,并为知识库分配了 1 GB 内存.

表3 庑殿建筑搭建顺序推理的时间
Table 3 Times for inferring construction sequence

庑殿制式	基于 Protégé 版本的 顺序推理时间/ms	基于 Jess 版本的	
		顺序推理时间/ms	提高比例/%
3 间 5 檩	124 203	1 281	98.97
5 间 7 檩	151 140	1 235	99.18
7 间 9 檩	198 875	1 250	99.37
9 间 11 檩	293 156	1 402	99.52
11 间 13 檩	443 750	1 484	99.67

搭建顺序推理作为整个古建动画自动生成的一部分,要运行 7~8 min,这是不能接受的,不能达到快速实时地生成符合用户描述的古建动画的需求。

导致效率低下的主要原因是使用的工具过多,包括用 Protégé 来建立本体,用 SWRL 来建立规则,用 Jess + Jena 来做推理机。在知识库开始推理之前,库中的本体和规则要在这几个不同的工具之间不断地载入、载出,并要进行相应的格式转换,在推理运行过程中,某一实例的更改也要在不同工具间进行格式转换、载入、载出,系统在这个过程中耗费了很多时间,而规则推理本身的时间其实只占了极小的一部分。

针对这个效率问题,使用 Jess 和 Java 实现了古建本体知识库的全部内容,推理时间见表3的右部分,可见效率有了极其显著的提高,对于不同制式的建筑规格,新版运行时间都比旧版节省了 99% 以上。

5 结 论

随着语义网络技术逐步走向成熟,基于本体的知识库系统越来越多,涉及各个领域;目前大部分的知识库都是构建领域本体,推理也大都基于本体推理,在本体之上进行规则推理的系统相对少。

本系统是古建动画自动生成系统的一个重要组成部分,为古建动画自动生成系统提供了知识库支持,并推理生成系统所需要的信息。本系统是语义网络技术在中国古代木架结构建筑领域的一个应用,首次以建筑构件的粒度来模拟实现了古建木架结构的整个搭建过程。

用 Jess 表达所有的领域知识,用 Jess + Java 实现了包括本体、实例和规则一起的整个知识库系统的构建和执行,而且,针对旧版本中出现的内存溢出的空间效率问题,采取了以下策略解决:

1) 简化系统流程。旧版的知识库系统用 Protégé 来建立本体,用 SWRL 来建立规则,用 Jess + Jena 来做推理机,导致知识库系统在推理时要在不同的工具之间进行内容的载入、载出、格式转换等,系统效率非常低;新版本采用 Jess 统一表示知识,减少了知识在各个工具之间载入、载出和格式转换,节省了

时间和空间。

2) 实例生成部分减少了规则之间以及实例之间的交互。在旧版本中,一个实例的生成往往以其他的实例为基础,或者本身是其他实例生成的基础,这样就加大了实例建立部分的内存和时间开销;而在新版本中,重新整理了规则,每个实例的生成都是单独的规则,和其他规则和实例间没有依赖关系,节省了时空开销。

3) 顺序规则执行过程去掉了对内存的修改。在旧版本中,顺序规则的执行是以每条规则不断修改实例中的属性来实现的,根据 Jess 的特性,每对内存进行一次修改,推理机都会对修改了的内存重新进行一次扫描;而新版本中,顺序规则的实现是通过查询语句实现的,只是将实例按照规则指定的顺序从知识库中查询出来,并不涉及到内存中实例的修改,所以比旧版节省了时间开销。

4) 不同方式的分模块执行。在旧版本中,顺序规则的分模块执行是指每次一个模块推理成功都要做一次 3) 中提到的往内存中写的操作,而本次模块推理结束之后、下次模块推理之前,又要清空之前对内存的修改,目的是不影响下次推理的结果;在新版本中,虽然顺序推理也是基于不同的规则模块,但是各个模块中都没有涉及到对内存的修改,相互之间也没有交互,每个模块都是单纯的查询语句,所以大大节省了时空开销。

除了效率上的优势之外,还采用了分模块加载运行策略,知识库系统会根据古建动画自动生成系统的需求,针对用户输入选取适当模块自动加载运行,因此,随着领域学习的加深,更多的建筑制式的知识都可以按照已经完成的运行模式被添加进来,不需要修改其他模块,也不需要改变系统的控制流程,整个古建本体知识库系统具有良好的可扩充性。

本知识库系统仍在完善中,存在的问题包括:

1) 旧版本用 Protégé 等工具实现,本体与规则部分分开构建,构建过程标准,在知识库扩充和修改时会更简洁,更易懂,符合语义网络的规范;而相比于 Protégé 等工具的简便易用,以及友好清晰的界面,新版系统在界面友好、易用性等方面还有许多需要

改进的地方。

2) 目前构件实例规则模块和搭建顺序模块已经加入到系统,其他的模块还需要一些工作,才能加入系统中来。

3) 本古建知识库是根据全过程古建动画自动生成系统的需求来构造的,领域知识覆盖面是非常有限的;中国古代建筑博大精深、浩如烟海,建立一个相对全面的、能支持各种应用需求的、高效的古代建筑本体知识库,是一个长期的目标。

致谢

北京工业大学硕士宫丽环基于 Protégé 设计实现了古建知识库系统的第一版,北京工业大学硕士孙凯、王巍峰、未宫瑾和尹梅芳学习总结了古建领域知识,设计并实现了古建动画自动生成系统的顺序推理、尺寸计算和空间位置计算,他们的工作为本文的研究打下了良好的基础,在此表示感谢。

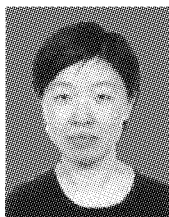
参考文献:

- [1] GUARINO N. Formal ontology and information systems [C]//Proc of the 1st Int'l Conf on Formal Ontology in Information Systems. Trento, Italy: IOS Press, 1998: 3-15.
- [2] USCHOLD M, GRUNINGER M. Ontologies: principles, methods, and applications[J]. Knowledge Engineering Review, 1996, 11(2): 93-155.
- [3] LEE B T, HENDLER J, LASSILA O. The semantic Web [J]. Scientific American, 2001, 284(5): 34-43.
- [4] LEE B T. Semantic Web road map [EB/OL]. [2010-01-01]. <http://www.org/DesignIssues/Semantic.html>.
- [5] 陆汝钐,张松懋. 从故事到动画片——全过程计算机辅助动画自动生成[J]. 自动化学报, 2002, 28(3): 322-334.
LU Ruqian, ZHANG Songmao. From story to animation: full life-cycle automatic animation generation[J]. Acta Automatica, 2002, 28(3): 322-334.
- [6] 马炳坚. 中国古建筑木作营造技术[M]. 2版. 北京: 科学出版社, 2003: 1-2.
- [7] 李善平. 本体论研究综述[J]. 计算机研究与发展, 2004, 41(7): 1041-1052.
LI Shanping. Survey of ontology research[J]. Journal of Computer Research and Development, 2004, 41(7): 1041-1052.
- [8] 姚绍文. 语义化 web 的关键技术及其应用研究[D]. 成都: 电子科技大学, 2002.
YAO Shaowen. Key technologies of Semantic Web and their application research[D]. Chengdu: University of Electronic Science and Technology of China, 2002.
- [9] The Protégé project [EB/OL]. [2010-01-01]. <http://Protege.stanford.edu>.
- [10] ARPIREZ J C, CORCHO O, FEMANDEZ-LOPEZ M, GOMEZ-PEREZ A. WebODE: a scalable ontological engineering workbench [C]//Proc of the K-CAP 2001. New York: ACM Press, 2001: 6-13.
- [11] SURE Y, ERDMANN M, ANGELE J, STAAB S, STUDER R, WENKE D. OntoEdit: collaborative ontology engineering for the Semantic Web [C]//International Semantic Web Conference. Sardinia, Italy, 2002: 245-249.
- [12] BECHHOFFER S, HORROCKS I, GOBLE C, STEVENS R. OILED: a reasonable ontology editor for the Semantic Web [C]//KI2001, Joint German/Austrian Conference on Artificial Intelligence. [S.l.], 2001: 396-408.
- [13] 吴沁奕, 陈英, 吴鹤龄. 新一代专家系统工具[J]. 计算机世界报, 2001, 44: B10-B11.
WU Qinyi, CHEN Ying, WU Heling. The new generation expert system tools [J]. China Computer World, 2001, 44: B10-B11.
- [14] 宫丽环. 动画自动生成中复杂情节展开的研究及本体知识库构建[D]. 北京: 北京工业大学, 2008.
GONG Lihuan. Complex plot design and the ontological knowledge base for automatic animation [D]. Beijing: Beijing University of Technology, 2008.

作者简介:



白卫静,女,1984年生,硕士研究生,主要研究方向为人工智能中的知识库技术。



张松懋,女,1967年生,研究员,博士生导师,博士。曾在美国、澳大利亚、德国和法国的研究机构作访问学者。主要研究方向为计算机科学中的人工智能理论和技术,曾担任国家自然科学基金项目和国家“863”计划项目的负责人。发表学术论文40余篇。



刘椿年,男,1944年生,曾主持过多项国家自然科学基金面上项目和早期“863”计划项目。曾获北京市科技进步奖2项,在科学出版社有著译多种出版,在国内外学术刊物和重要国际学术会议上发表论文100余篇。