

# 面向存储与传输的手绘草图信息表示模型<sup>\*</sup>)

孙怀志 孙正兴

(南京大学计算机软件新技术国家重点实验室 计算机科学与技术系 南京 210093)

**摘要** 草图信息的存储与传输是草图系统在信息共享乃至网络协同方面需要解决的问题,其核心在于草图信息的一致表示。本文从信息表示和信息传输两个维度,建立了草图信息表示模型框架。该框架采用域模型思想,包括域属性、逻辑模型、转换模型、对象模型和控制模型,它们实现了草图信息的一致表示、统一管理和信息存储、传输。最后,从静态和动态两方面,对草图信息的建模过程进行了详细描述,给出了草图信息表示的 XML 实现策略,并对信息传输技术进行了分析。

**关键词** 草图信息,草图信息表示,草图信息表示模型

## A Representation Model for Sketchy Data Storage and Transmission of Sketchy Data

SUN Huai-Zhi SUN Zheng-Xing

(State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093)

**Abstract** The storage and transmission of sketchy data is a key problem in data communication and cooperative work. The main difficulty is how to represent sketchy shapes in a consistent way. A framework of consistent representation of sketchy shapes based on domain model idea is proposed. The framework contains domain property, logic model, transformation model, object model and control model in order to represent, store and transmit sketchy data. Detailed description of modeling sketchy data is given in both static and dynamic ways in succession. An XML based strategy of sketchy data representation and an analysis of sketchy data transmission technology are also finally proposed.

**Keywords** Sketchy data, Sketchy data representation, Sketchy data representation model

## 1 引言

手绘草图(Sketching)是人类一种自然而直接的思路外化和交流方式,其人机交互在许多领域得到广泛的研究和应用<sup>[1]</sup>。近年来,基于分布式计算和网络协同技术的手绘草图人机交互日益引起关注。在这种计算模式下,可以把草图系统分为计算模型和表示模型。计算模型针对草图识别与检索等关键技术,注重目标实现。表示模型侧重于草图信息表示和信息存储、传输,分别表明信息存在的方式和应用。当前研究主要集中在草图计算模型方面,针对草图信息表示模型的研究还很少。根据草图信息的组成关系,可以将草图信息划分为构成信息和特征信息两部分,其中构成信息指手绘草图的原始信息,由它可以复原手绘的图形。特征信息从构成信息中抽取特征得到。本文所指的草图信息是草图的构成信息。在草图构成信息表示方面,现有草图系统存在三个主要问题:首先,缺乏草图信息描述标准,每个系统都有自己的信息表示格式,难以实现草图信息的共享。其次,草图信息存储方式单一,基本采用独立的文本文件,限制系统的可扩展性和灵活性。最后,从草图信息表示到信息管理之间缺乏统一描述。基于以上存在的问题,本文从草图信息管理的角度出发,提出了面向存储与传输的手绘草图信息表示模型,以实现草图信息的统一格式表示、信息存储和信息传输。

一些系统与工作已经对手绘草图信息模型进行了研究。JOT<sup>[2]</sup>针对笔迹存储给出了一个 ink 描述的规范,并针对 ink 操作与平台交互提出了通用标准。Li<sup>[3,4]</sup>提出了一种基于域

模型参与管理 ink 数据信息和应用语义处理的方法,并描述了工具系统 Penbuilder 域模型思想,域的内容视为轻量级对象,以 ink 作为主要数据类型描述笔迹信息。W3C 推出了 InkML<sup>[5]</sup>标准草案,将“数字墨水”作为一种新型多媒体数据类型,对手绘草图等笔迹信息提供了统一的描述格式。然而这其中大部分都面向特定的应用需求,或者仅仅着眼于笔迹信息的管理,无法针对草图的信息结构建立一个完整的模型。徐<sup>[6]</sup>提出面向语义的草图描述模型,仅用来表示草图的形状信息。当前并不存在一个描述草图信息模型的完备框架。

手绘草图信息表示模型借助 Penbuilder 域模型的思想进行组织,把草图信息视为具有层次的对象,并划分由低到高的三个层次实现草图信息的表示、信息的存储和传输,这三个层次分别为:点序列、笔划(包括笔划和图元)和图形层。这样,一方面使计算在同一层次进行,计算粒度相同,管理粒度一致,便于信息维护。另一方面便于建立草图信息到草图表示信息的映射关系,可以在任何一个层次输出草图的表示信息,减少了计算和传输的信息量,这在网络环境中尤为重要。本文从信息表示和信息传输两个维度阐述草图信息表示模型,并对草图信息的建模、信息的 XML 表示方法和信息传输机制进行分析,最后用实验对系统的整体性能作出评价,并验证不同层次、不同规模的草图信息对系统性能的影响。

## 2 系统框架

根据域模型的思想,域中的对象分为两种:一个是域,一个是域中的内容。域定义用户界面的特性和行为,每个域对

<sup>\*</sup> 基金项目:国家自然科学基金(69903006、60373065)、教育部“新世纪优秀人才支持计划”。孙怀志 硕士研究生,主要研究方向:多媒体计算、图形识别;孙正兴 博士,教授,博士生导师,主要研究方向:智能人机交互、多媒体计算和计算机视觉。

自己的域空间进行管理。域中的内容是草图信息。由于应用领域多样性,域按照问题域的不同,可以划分若干子域,并允许嵌套或递归。这样就形成了一个具有层次性、结构化的体系结构,实现基于树状结构组织,有利于同领域的信息共享。

域可以看作一个抽象的类,由它派生出了草图信息表示模型。模型包含五个部分,分别为:域属性(Property)、对象模型(Object Model)、逻辑模型(Logic Model)、转换模型(Transform Model)和控制模型(Control Model),它们之间的关系构成了表示模型的系统框架,见图1。由此,域可以用一个五元组来描述:

$$D = \{Property, Object, Logic, Transform, Control\}, \text{记为 } D_{POLTC}$$

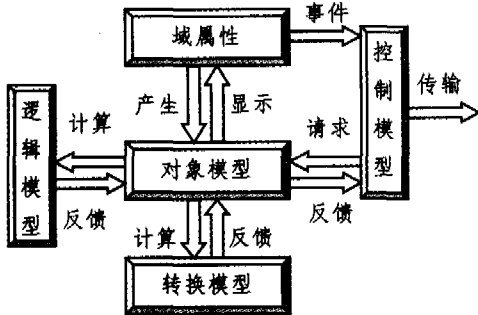


图1 系统框架图

域属性(Property)可以理解为系统界面定义的元素,如界面风格、输入工具、按钮、菜单等,它是草图信息的来源。

对象模型(Object)对域内的信息表示过程进行全局控制。对象模型包括两个重要内容:对象和对象管理。域中的对象包括元对象和派生对象。元对象由域属性获取,主要是草图信息对象,作为对象模型的基类,提供对外接口和自管理功能;派生对象继承自元对象,它主要指用XML表示的草图信息。

逻辑模型(Logic)的任务是根据草图信息逻辑结构,生成关于表示信息的元素以及元素间组成关系的描述,这种描述采用索引表表示,为对象模型和转换模型服务。

转换模型(Transform)负责建立草图信息与逻辑结构的映射关系,完成由草图信息到XML格式表示的转换。

控制模型(Control)实现草图表示信息的存储和传输。控制模型提供了对网络的支持,从而使域具备了网络环境下的信息交换能力。

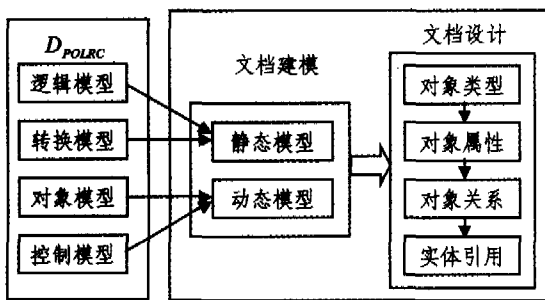


图2 草图信息表示的实现过程

草图信息到表示信息的转换和传输过程是由草图信息模型描述的。为了详细说明信息模型的建模过程,将信息模型分为文档建模和文档设计两个步骤,其中文档模型又可分为

静态模型和动态模型。静态模型注重信息的表示,动态模型侧重信息的存储和传输。文档设计主要是将文档模型转换为规则集合,以确定信息表示的逻辑结构,这种结构用文档类型定义-DTD表示。从域中各个模型的功能可以看出,逻辑模型和转换模型是静态模型的具体实现,对象模型和控制模型为动态模型提供支持。它们之间的对应关系和实现过程见图2。

### 3 草图信息表示

草图信息表示是将草图信息转化为XML表示的过程,它由草图信息模型,即由文档建模和文档设计描述。

#### 3.1 文档建模

文档建模的目的是理解信息表示中的元素结构和含义,它们由静态模型描述。静态模型按照草图信息的划分层次进行组织,并把层次表示转化为模型中的对象类型。这样,静态模型就包含三类对象,分别为点序列、笔划(包括笔划和图元)和图形。现对它们的对象特性描述如下。

点序列(Point):采样点的集合,其BNF(Backus-Naur Form)表示形式为:

〈点序列〉 ::= 〈〈采样点〉〉<sub>1</sub><sup>n</sup>,

〈采样点〉 ::= 〈(横坐标,纵坐标,时间)〉。

笔划(Stroke):用户在一次的落笔和提笔间的笔迹被称为笔划,它们的特性主要表现为与笔迹相关的采样点序列。草图系统往往通过笔划分割,而后将分割后的笔划拟合成与之最相近的直线(Line)、弧(Arc)和椭圆(Ellipse)等基本图元。这时,笔划的BNF表示形式为:

〈笔划〉 ::= 〈〈直线〉〈弧〉〈椭圆〉〉<sub>1</sub><sup>n</sup>。直线(Line)、弧(Arc)和椭圆(Ellipse)分别用它们的数学特征表示。

图形(Sketch):模板库中匹配成功的图形集合,目前的草图系统通常用它们的ID号标识。

为确保草图信息表示的完整性,高层次的对象除了自身的特性外,还包含低层对象的特性。点序列作为笔划最基本的信息,使它包含在每个笔划中,它们之间构成一对一关系,不再建立点序列与基本图元的对应关系,这样,点序列和基本图元定义在一个层次,维护了基本信息整体性。为便于管理,把整个草图视为一个抽象对象 Sketches,它位于草图信息的最高层,这样便形成了图3所示的组织结构。结构图中体现了表示事物,将事物组织为类层次对象、对象间的约束等建模过程,构成了静态模型的骨架,而后通过添加特性来填充对象,完成整个建模过程,整个过程是按照面向对象的思想进行的。

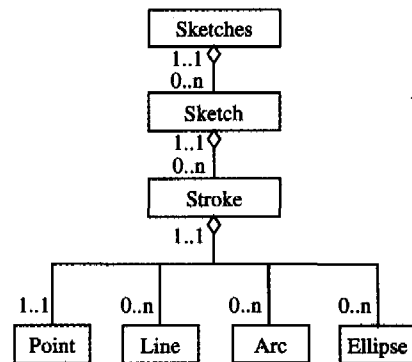


图3 草图对象UML结构

#### 3.2 文档设计

草图表示信息的最终形态为 XML 文档,文档设计负责生成信息表示的逻辑结构 DTD。整个文档设计分四个步骤进行,包括对象类型表示、属性表示、关系表示和实体引用。

进行文档设计时,首先将静态模型中的对象类型转换为 DTD 的元素类型,包括元素名称和对象分层两方面。对象类型是类层次的一部分,通过类层次决定从哪一层开始建立 XML 元素。随后根据对象的特性进行属性表示,属性表示包括属性命名与属性值编码。表示关系要结合 XML 的具体实现技术,元素间的关系大体可分为嵌套、并列、链接三种,分别对应于父子关系、兄弟关系和其它离散关系。

DTD 中的实体 Entity 是文档设计的重要环节,它不仅仅是物理文档的一部分,逻辑上也有一定用途,尤其对松散信息的总体结构有很好的控制作用。将文档中的部分内容放入外部实体,使得它们能够独立于文档的其余内容独立更新,这种方式非常有价值。

### 3.3 逻辑模型

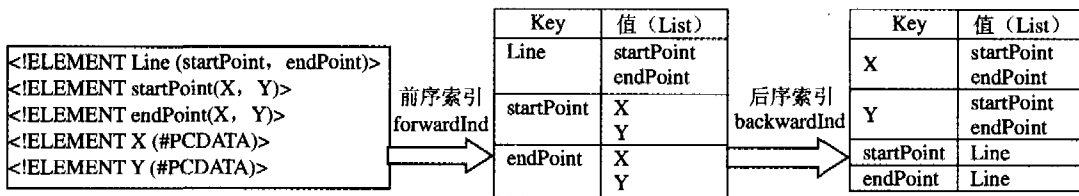


图 4 LineDTD 的转换过程

### 3.4 转换模型

转换模型负责建立草图信息与索引表的映射关系,并由映射关系生成草图信息的 XML 表示。整个过程由 DOM API 以树型结构维护,最终的计算结果对应着某个层次的一棵子树。以 Line 对象为例,子树生成过程见图 5,现作简要说明:

①通过 forwardInd 建立树的根节点 Line 和第一棵子树节点 startPoint,同时建立路径索引 pathInd,把处理的节点按照顺序加入到路径索引中,指针指向当前处理的节点。子树构造完毕,由 backwardInd 控制 pathInd 指针回溯,到达当前子树父节点停止。

②按照①构造其它子树;

③将新生成的子树加到 pathInd 指针指向的节点,也就是树的 Line 节点。

以上算法生成的子树是属于某个层次的,它需要对象模型建立与父节点的连接,从而形成完整的草图信息 XML 表示。

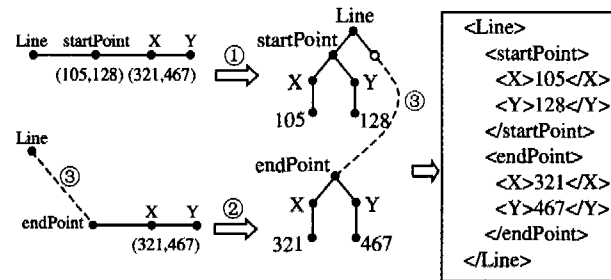


图 5 Line 子树生成过程与 XML 文档

## 4 草图信息传输

动态模型描述了信息的传输机制,采用数据流模型实现。

逻辑模型实行分层定义,DTD 的层次与草图信息的层次一一对应,分别为:PointDTD、StrokeDTD 和 SketchDTD,这三个层次间相互独立。它们描述了表示信息在每个层次的逻辑结构,确保草图信息表示在层次上的一致性。之所以没有定义全局的描述,是因为草图信息转换往往在中间层次上进行,而没有到达最高层,也就无需关注高层的信息表示,由对象模型完成信息表示的动态组织。这种分层表示、集中管理的模式会更加灵活和合理。

DTD 仅是对表示信息逻辑结构的描述。在信息表示过程中,为了控制表示信息的结构和内容,需由逻辑模型将 DTD 的结构转换为索引表实现。生成的索引表包括前序索引和后续索引,它们作为草图逻辑结构的元信息描述,形成了对 DTD 内容的完备表示,索引表由相应层次的对象进行管理。图 4 描述了笔划层的 Line 对象从 DTD 到索引表的转换过程。

数据流模型描述了数据存储、处理器和数据流。其中数据存储规定了信息的保留方式,处理器针对数据的管理,数据流负责数据的传输。处理器由对象模型实现,控制模型实现了数据存储和数据流描述的功能。

### 4.1 对象模型

模型中的对象抽象为 POINT、STROKE、SKETCH,它们均为域中的轻量级对象。POINT 是 Object 的元对象,其它类为派生对象。每一层次的派生对象除了拥有自身的对象信息外,还维护前一层级的对象信息,从而构成具有嵌套结构的上下文解释。对象管理采用自我管理控制派生对象的状态,这种自我管理功能的一部分是元对象具有的,派生对象通过继承元对象获得。而另外一些功能是派生对象特有的,如处理控制模型的请求,不同层次的派生对象功能有所区别。对象管理采用外部事件驱动模式控制对象的状态转换,状态转换见图 6。对象管理过程中,状态层次由低级迈向高级,如果把每个状态用对应层次的子计算实现,子计算的结果为轻量级对象,低层子计算结果作为下一层子计算的输入,这样就形成了一个完整的对象计算模型(图 7)。计算模型中每个层次的子计算负责生成本层的轻量级对象,其中包含了表示信息。表示信息生成过程如下:

//假设当前子计算层次为第三层 SKETCH,那么第一、第二层已经计算完毕

①创建轻量级对象 SKETCH

//由 Logic 第三层的 Logic3 生成与逻辑结构 SketchDTD 对应的索引表 SketchInd

②SketchInd ← Logic3(SketchDTD)

//Transform 分析 SketchInd,由 DOM API 建立该层次的对象结构并加载草图数据 dataList,生成表示信息对象 ObjectDOM

③ObjectDOM ← Transform(SketchInd,dataList)

//Object 的对象管理将前一层子计算生成的表示信息对象 StrokeDOM 作为子节点添加到 ObjectDOM

④ SketchDOM ← ObjectDOM. AppendChild (StrokeDOM)

⑤ SKETCH ← SketchDOM



图6 状态转换图

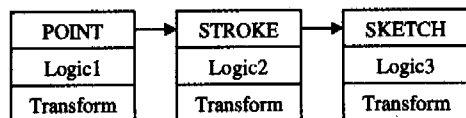


图7 对象计算模型

从控制事件可获得子计算终止的层次。当子计算结束，不再跃向更高层时，就把本层的轻量级对象由对象管理转化为结果对象，通过访问结果对象可以获得草图的表示信息。

#### 4.2 控制模型

控制模型通过分析域属性的事件控制草图信息的 XML 表示和信息传输，可以定义一个三元组表示：

Control={Analyzer, CalAct, TransAct}，框架见图8。

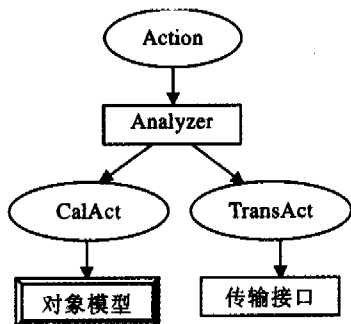


图8 控制模型框架

Analyzer 是事件分析器，它是控制模型的核心。Analyzer 负责对输入事件进行判别，确定发生何种具体事件，根据分析结果调用处理对象 CalAct 和 TransAct。

CalAct 针对草图信息表示过程实施控制。因为该过程是分层次进行的，CalAct 的主要功能是建立事件状态与层次的映射关系，以确定在哪个层次上完成信息表示，然后将层次标识和相关信息封装成消息，发送给对象模型处理。

域属性定义的事件和域状态是密切相关的，而且它每个阶段的处理对象，如点序列、笔划、图形，都具有由低到高的层次性。通过记录每个阶段的状态信息，就可确定当前对象的层次，再映射到草图信息表示的层次上。

TransAct 完成两个功能：信息的存储和传输。在此之前，草图的信息是以 DOM 树的形态驻留在内存中，因此上述功能实际上是对 DOM 树的操作。TransAct 通过序列化对象输出 XML 文档，实现信息的本地存储。信息传输是通过传输接口实现的，传输接口是域与外部通信的唯一接口，它使用 SOAP(简单对象访问协议)来传递数据或整个 XML 文档。

#### 4.3 传输机制

控制模型刻画了对草图表示信息的动态管理功能，主要实现信息的存储和传输。动态管理的核心是 XML 的存取机制和消息机制。通过存取机制实现了草图表示信息的存储，对于信息传输，可由 XML 的消息机制 SOAP 实现。

数据存取机制由 DOM 实现，它是 XML 分析器的一个主要标准 API，通常是把对 XML 文档的操作转化为对 DOM 中对象的操作，直接通过 DOM 接口访问数据。这样，使整个数据存取过程变得更加平滑。最后再序列化，输出 XML 文档，或者由 TrAX 通过 XSLT 转换模板输出其它格式的文件，如文本文件、Html、PDF 格式等。草图信息用 XML 表示，信息存取由 DOM 接口实现，二者的结合使得草图信息的表示和传输能够获得良好的统一。需要说明的是，可以针对任何层次输出草图的 XML 表示信息，而且草图中的细粒度对象，如 Line、Arc 等都是参数化表示，需要维护的信息量不大，用 DOM 接口实现数据存取机制有着天然的优势。

SOAP 可以在 HTTP 上传递 XML 文档来支持远程过程调用，并向服务器发送任意的 XML 元素，这使得可以由 SOAP 实现草图 XML 表示信息的全部或部分传输，增加了传输的灵活性。

### 5 实验组织与分析

#### 5.1 试验组织

根据草图信息表示模型设计了基于 C/S 模式的应用系统 Sketch2XML。模型中的域属性由小工具 MyGUI 实现，它位于客户端，用来生成不同层次的草图信息。其它模块在服务器端，发布服务器为 Tomcat4。Sketch2XML 分别控制在 Sketch 和 Sketches 两个层次上输出草图的 XML 表示信息，并记录系统消耗的时间，用来验证系统的整体效率以及不同层次、不同规模的草图信息对系统性能的影响。

草图信息规模由生成的 XML 文档字节数量衡量，本次试验选取了 10k、50k、100k、150k、200k、300k 六种。系统在 CPU 为 Celeron 1.3G、内存 128M 的 PC 机上运行，试验结果见图9。

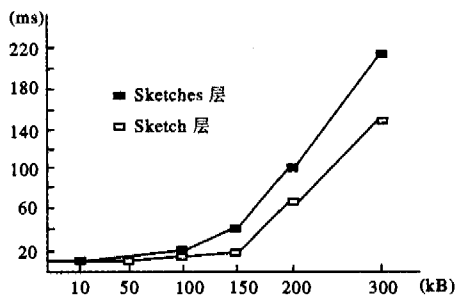


图9 试验结果

#### 5.2 结果分析

计算时间的消耗主要体现在由 DOM 实现的草图信息到 XML 表示信息转换过程和 DOM 序列化输出 XML 文档。可以看出，在 100k 以内，耗用的时间与 XML 嵌套的层次关系不大，基本在 20ms 左右。随着信息规模增加，嵌套层次越多，需要的时间也越多。也就是说，针对大规模的草图信息，尽可能地减少信息表示层次，有利于提高系统效率。通常来说，100k 的信息足够描述非常复杂的草图，即使信息规模较大，系统也有较快的响应时间。

**结论与展望** 草图信息表示与传输的统一是整个草图系

统需要解决的问题,是实现信息共享和网络协同的前提。基于这样的目的,本文从模型实现的功能和计算效率出发,对草图信息进行分层管理。在此次基础上,从信息表示和信息传输两个维度,建立了草图信息表示模型。文章还从静态和动态两方面,分别对草图信息的建模过程、草图信息到 XML DTD 的转换规则以及信息表示的实现策略、信息传输技术进行了详细叙述。通过实验证明,草图信息表示模型能够满足草图信息的一致表示、统一管理和信息传输要求。由于模型基于域管理的思想,可以通过域派生机制实现功能扩充,从而完成更复杂的草图信息管理。

今后工作中,将对草图系统在网络环境下共享与协同方面进行研究,同时完善和扩充草图信息表示模型,最终使草图系统的计算模型和表示模型形成完整的统一,以进一步扩展手绘草图的功能和应用领域。

## 参考文献

- 1 周若鸿,孙正兴,张莉莎,等. 草图理解研究进展[J]. 计算机科学, 2004,31(4):140~146

- 2 Slate Corporation, Scottsdale A Z. JOT - A Specification for an Ink Storage and Interchange Format [J/OL]. <http://hwr.nici.kun.nl/unipen/jot.html>
- 3 栗阳,关志伟,戴国忠. 笔式用户界面开发工具研究[J]. 软件学报, 2003,14(03):392~400
- 4 Li Y, Guan ZW, Chen YD, Dai Gz. Penbuilder: platform for the development of PUI (pen-based user interface). In: Tan T, Shi Y, Gao W, eds. Proceedings of the 3rd International Conference on Multimodal User Interfaces (ICMI 2000) [J]. Springer-Verlag, 2000. 534~541
- 5 Chee Yi-Min, Magaña J, et al. Ink Markup Language -W3C Working Draft 28 September 2004 [S]. <http://www.w3.org/TR/InkML>
- 6 徐晓刚. 草图理解系统及其关键技术研究: [硕士学位论文]. 南京: 南京大学计算机科学与技术系, 2003
- 7 W3C. Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004 [S]. <http://www.w3.org/TR/2004/REC-xml-20040204>

(上接第 173 页)

sPattern="司马懿是 \$-1- \$ 的父亲"

测试结果:

锁定事实 F1: 司马懿是司马昭的父亲。

变量 \$-1- \$ 代换: "司马昭"。返回 true 值。

测试用例 3:

sPattern="司马昭是司马炎的父亲"

测试结果:

锁定事实 F2: 司马昭是司马炎的父亲

变量 \$-1- \$ 代换: "父亲"。返回 true 值。

测试结果分析:

测试结果表明,本算法可以有效地解决目标模式与事实库的匹配问题。并且对于目标模式中的变量,可以得到对应的代换它的常量字符串。

4.4.2 测试二: 目标模式与规则库的匹配

规则库 RuleBase:

R1: \$-1- \$ 是 \$-2- \$ 的父亲, \$-2- \$ 是 \$-3- \$ 的父亲 → \$-1- \$ 是 \$-3- \$ 的祖父

R2: \$-1- \$ 是 \$-2- \$ 的母亲, \$-2- \$ 是 \$-3- \$ 的父亲 → \$-1- \$ 是 \$-3- \$ 的祖母

测试用例 1:

sPattern=" \$-1- \$ 是司马炎的祖父"

测试结果:

锁定规则 R1: " \$-1- \$ 是 \$-2- \$ 的父亲, \$-2- \$ 是 \$-3- \$ 的父亲 → \$-1- \$ 是 \$-3- \$ 的祖父"

变量 \$-1- \$ 代换: " \$-1- \$", 返回 true 值。

测试用例 2:

sPattern="司马懿是 \$-1- \$ 的祖父"

测试结果:

锁定规则 R1: " \$-1- \$ 是 \$-2- \$ 的父亲, \$-2- \$ 是 \$-3- \$ 的父亲 → \$-1- \$ 是 \$-3- \$ 的祖父"

变量 \$-1- \$ 代换: " \$-1- \$"。返回 true 值。

测试用例 3:

sPattern="司马懿是司马炎的父亲"

测试结果:

锁定规则 R1: " \$-1- \$ 是 \$-2- \$ 的父亲, \$-2- \$ 是 \$-3- \$ 的父亲 → \$-1- \$ 是 \$-3- \$ 的祖父"

锁定规则 R2: " \$-1- \$ 是 \$-2- \$ 的母亲, \$-2- \$ 是

\$-3- \$ 的父亲 → \$-1- \$ 是 \$-3- \$ 的祖母"。

变量 \$-1- \$ 代换: "祖父"、"祖母"。返回 true 值。

测试结果分析:

实验结果表明,本算法可以有效地解决目标模式与规则库的匹配问题,可以有效地锁定匹配成功的规则。

## 5 “图检索”算法在问答系统中的应用

上述算法可以递归进行。在问答系统中的具体应用如下:

①如果目标模式匹配事实库、规则库均失败,那么本次模式推理失败,返回 false;

②如果匹配事实库成功,那么本次模式推理成功,返回 true;

③如果匹配事实库失败,但是匹配规则库成功,那么通过代换规则库中的变量,得到新的目标模式,继续进行模式推理。

将用户查询进行处理,作为目标模式。通过上述算法,可以得到用户查询的答案。

总之,本算法可以应用于问答系统,是提高问答系统准确率的一条有效途径。所以,对于问答系统来说,多模式合一有很高的应用价值。

**结论和下一步的工作** 本文首先介绍了多模式合一的相关定义、研究现状,然后给出了多模式合一的模型和机制。在此基础上,重点给出了多模式合一的算法——“图检索”算法。实验结果表明,本算法可以有效地解决多模式合一问题。并且,本算法可以直接应用到问答系统中,较大地提高问答系统的准确率 (precision)。下一步工作,将致力于减小算法的时间复杂性,进一步提高算法的效率。

## 参考文献

- 1 Turing A M. Computing Machinery and Intelligence. Mind, 1950,59(236):433~460
- 2 Searle J R. Minds, brains, and programs. Behavioral and Brain Sciences, 1980,3:417~424
- 3 Lin Dekang, Pantel P. Discovery of Inference Rules for Question Answering. Natural Language Engineering, 2001, 7(4): 343~360
- 4 Simmons R F. Natural Language Question-Answering Systems. Communications of the ACM, 1969,13(1): 15~30
- 5 白硕. 语言计算与基于内容的文本处理. 见: 大规模内容计算. 北京: 清华大学出版社, 2003. 13~25