

文章编号:1007-130X(2012)02-0111-05

# 虚拟计算环境中的上下文建模与推理方法研究<sup>\*</sup>

## Research on Context Modeling and Reasoning in Virtual Computing Environments

曹晓舂,王珊珊,吴刚

CAO Xiao-chong, WANG Shan-shan, WU Gang

(上海交通大学软件学院,上海 200240)

(School of Software, Shanghai Jiaotong University, Shanghai 200240, China)

**摘要:**上下文建模与推理是支持虚拟计算环境自适应调整的技术基础,基于本体的建模是其中一种重要的方法,但缺乏相关的定量分析。本文分析了不同的上下文建模方法,重点以基于 OWL 和 XML 的建模方法为代表,通过实验从推理效率、建模能力和易操作性三方面对比了基于本体的语义化方法和基于标记语言的非语义方法。

**Abstract:** Context modeling and reasoning is the basis of virtual computing environments for dynamic adapting. The ontology-based modeling is one of the important approaches, with the lack of correlative quantitative analysis. This paper analyzes several existing context modeling methods, and focuses on the comparison between the ontology-based modeling and the markup modeling in the three aspects like reasoning efficiency, modeling capability and programming convenience. OWL is chosen to describe the ontology and XML is chosen as the markup language respectively.

**关键词:**上下文推理;效率;本体;虚拟计算环境

**Key words:** context reasoning; efficiency; ontology; virtual computing environment

**doi:**10.3969/j.issn.1007-130X.2012.02.021

**中图分类号:**TP311

**文献标识码:**A

## 1 引言

随着网络技术的快速发展和广泛应用,互联网已经不只是一个信息共享的平台,而且也是一个计算平台;人们不断探索在 Internet 平台上如何构造虚拟计算环境,以支持大规模的计算或数据处理。谷歌、亚马逊等公司推出了商业云计算平台,加州大学伯克利分校提出了用于网格计算的中间件系统(BOINC)<sup>[1]</sup>;国防科技大学提出了一个基于互联网的虚拟计算环境 iVCE 概念模型<sup>[2]</sup>,并开发了相应的语言设施和支撑环境;北京大学、南京大学等单位提出了 Internetware<sup>[3]</sup>的概念,以支持互联

网上新的计算模式。

在虚拟计算环境中,一个关键的问题是如何感知环境的变化并对计算结构与行为做出动态调整。为了解决这个问题,需要获取环境上下文,并进行相应的推理工作,以支持正确的决策。目前已经有许多不同的建模方法及推理技术<sup>[4]</sup>,基于本体的上下文建模与推理是其中一种被广为关注的方法。

本体(Ontology)是万维网发明者 Tim Berners-Lee 提出的语义网体系架构中一个重要的层次。本体是关于领域知识的概念化、形式化的明确规范,是对领域知识共同的理解与描述。它作为一种描述概念及其关系的模型,可以支持对虚拟计算环境中上下文语义的描述,并且在没有额外规则信

<sup>\*</sup> 收稿日期:2010-12-01;修订日期:2011-03-02

基金项目:国家 863 计划资助项目(2009AA01Z123)

通讯地址:200240 上海市闵行区东川路 800 号上海交通大学 788 信箱

Address: Mailbox 788, Shanghai Jiaotong University, 800 Dongchuan Rd, Minhang District, Shanghai 200240, P. R. China

息的情况下,仍提供相应的推理能力。但是,本体推理的效率,尤其是在大规模虚拟计算环境中的推理效率始终是人们关注的问题。

本文分析对比了基于本体的上下文建模与推理方法和传统的非语义化方法,并通过实验模拟了虚拟计算环境,从建模能力、易操作性和推理效率三个方面对这两类方法进行了比较。

## 2 上下文建模和推理

### 2.1 上下文建模方法概述

上下文指环境中任何实体或事件的信息,其中实体可以是一个人或一个地方,也可以是物理或计算对象。一些研究也给出了关于上下文的定义<sup>[5~7]</sup>,在面向服务的虚拟计算环境中,上下文可以包括对象、事件、结构。对象是指服务的属性和状态,而事件描述在一段时间内服务的属性或状态的转变,结构显示了服务之间的关系。

文献[4]总结了一些上下文建模的方法,并将这些方法根据建模的数据结构进行分类,包括键值对模型、标记方法模型、图形模型、面向对象模型、基于逻辑的模型和基于本体的模型。文献还对这些方法在分布式组成、可验证性、信息质量、不完整性和二义性、形式化程度等方面进行了性质上的评估。键值对方法本身简单的特性使得它易于管理和避免错误,但是当需要考虑到二义性问题的時候,其缺点就表现出来了。标记方法模型支持类型检测等验证,但是相应的不完整性和二义性必须在应用层处理。而图形的方法多被用做手工的模型构建。基于逻辑的方法有很高的形式化程度,但是其他方面表现得却不是很好。面向对象的方法和基于本体的方法在各个维度都表现优异,尤其是基于本体的方法。

### 2.2 本体建模方法与标记语言建模方法的比较

在上述建模方法中,标记方法模型尤其是基于XML的标记方法模型是当前一种主流的建模方法,而随着近年来语义网的发展,以OWL本体建模语言为代表的基于本体的建模方法也逐渐被关注。因此,本文选取基于OWL和基于XML的方法,来比较基于本体的上下文建模方法与基于标记语言的非语义建模方法。

扩展性标记语言(XML)广泛用于数据传输,使用XML进行上下文建模属于标记方法建模的范畴。XML是一种自描述语言,并且易于解析,但

它不能表达语义信息。

为了使数据获得可被应用程序理解的语义性,出现了资源描述框架(RDF)。它提供了由一系列“对象-属性-值”三元组为陈述所组成的数据模型和基于XML的序列化语法,但是其表达能力仍然有限。而构建在RDF之上的OWL是W3C组织推荐的本体描述语言的标准。OWL通过一些带有形式化语义的词汇,支持比XML、RDF更好的机器可理解性<sup>[8]</sup>。使用OWL进行上下文建模是一种基于本体的建模方法。

为了支持上下文推理,每一种建模方式的实现都要包括上下文表达、规则表达和规则执行三部分。图1显示了上下文建模和推理的关系。由于XML不能表达语义信息,也就是说,它只有语法的信息,因此传统的非语义上下文推理没有在上下文描述中嵌入规则。其规则定义与执行需要完全依靠用户自己实现。而对于基于本体的上下文推理,推理规则可以分为两部分,一部分包含在上下文表达中,另一部分是用户定义的规则。

从图1可以看出,基于本体的推理方法不仅是依靠上下文和用户自定义规则,它还依赖于本体的内嵌规则。对于标记方法推理,它仅仅依赖于上下文和用户自定义规则。

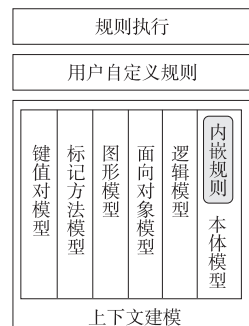


图1 上下文建模和推理

## 3 实验设计和结果

### 3.1 实验设计

在虚拟计算环境中,单独的服务往往需要通过某种机制集成其他服务来最终形成一个分布式的计算环境。在服务集成之前,它们需要知道其它服务的状态。例如哪一个服务是可用的,或者服务质量如何。本文按照文献[9]的方法构建了上下文本体,并使用本体推理中的实例归类推理来获得这样的高层语义信息。

本文设计了三组对比实验用于比较基于本体

的上下文建模方法和传统的非语义方法在推理效率上的差异。第一组实验比较它们在实例归类方法上的效率,第二组实验研究随着实例规模增大之后,它们在推理时间变化上趋势的不同,第三组实验研究随着推理规则数量增大之后,它们在推理时间变化上趋势的不同。

### 3.2 上下文建模

基于 XML 方法的建模过程中,通过在 XML 文件中定义标签来描述网络服务的各项参数。其中一个 XML 片段如图 2 所示,用户自定义的推理规则模板如图 3 所示。图 3 中“XPath”用来确定 XML 中具体元素或属性的具体位置,“Value”表示该元素或属性的要求值,“Relationship”用来表示一种关系,具体有大于、小于、等于和包含等,“Action”表示某方法。整个规则表示的含义是:当“XPath”位置上的值对于“Value”具有“Relationship”的关系,则执行“Action”的方法。和通常的表达式一样,“+”表示一条规则可以由一个或多个三元组组成。通过以上的建模及规则定义的方法,避免了在代码中定义规则,提高了推理的可扩展性。为了表述方便,下文实验中所指的传统非语义建模方法即指基于 XML 的建模方法。

```
<Service name="service1">
  <state>on</state>
  <Availability>80</Availability>
  <Cost>100</Cost>
  <MTBF>100</MTBF>
  <ResponseTime>100</ResponseTime>
</Service>
```

图 2 传统非语义上下文建模的 XML 片段

```
(XPath, Relationship, Value)+ → Action
```

图 3 传统非语义上下文推理的规则格式

图 4 给出了本文实验场景中的本体定义片段,它描述了服务是否可用的性质。本文选择 Jena2<sup>[10]</sup>作为实验中的上下文推理工具,Jena2 是一种基于 Java 的用于创建语义网应用系统的框架,它为 OWL 提供了一个程序开发环境,包括创建、解析、推理等环节。在本体领域中,Jena2 得到了较广泛的应用。Jena2 提供的 OWL 推理功能中包括了传递性、子类、子属性等内嵌规则,另外结合用户自定义规则推理,可以满足基于本体的上下文推理需求。

### 3.3 实验结果

进行实验的计算机配置为:Windows7 操作系统,2G 内存,2.4GHz Intel Duo Core CPU。

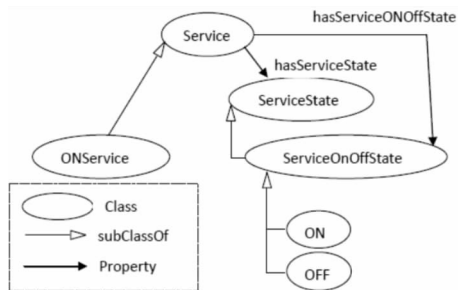


图 4 基于本体的上下文推理中的本体定义片段  
(1)实例归类。

该实验的实验内容为检查一个网络服务是否处于“ONService”状态,即是否是一个运行中的服务。基于本体的方法中,使用实例归类的方法,判断某个服务实例是否可以归类为“ONService”这个类。

为了在基于本体的方法中实现上面所设计的实验场景,除了图 4 中表示的本体定义片段外,需要进一步完整该本体的定义。增加的定义内容如图 5 所示,其中 hasServiceOnOffState 是 hasServiceState 的子属性;存在一个类型为 Service 的 service1 实例,存在一个类型为 ON 的 onstate 实例,service1 的 hasServiceOnOffState 属性的值为 onstate。这些定义需要按照 OWL 的语法加入到所构建的 OWL 文件中。

```
Schema: (hasServiceState subPropertyOf hasServiceOnOffState)
Data: (service1 rdf:type Service)
      (onstate rdf:type ON)
      (service1 hasServiceONOffState onstate)
Rule: (?s rdf:type Service) (?z rdf:type ON) (?s hasServiceState ?z) → (?s rdf:type ONService)
```

图 5 本体定义中的增加内容

图 6 显示了推理的部分过程:Fact(1)和 Rule(2)推理出 Fact(3),Fact(3)(4)(5)和 Rule(6)推理出 Fact(7),从而将实例 service1 归类为 ONService。这里需要强调的是,使用 Jena 的推理功能,需要指出所要推理的内容,此处选择分别推理 service1 所有可能存在的所有实例归类,推理可归类到 OnService 的所有实例以及推理 OnService 和 service1 的关系。这三个推理的前提都是将实例 service1 归类为 ONService,也就是说,三个推理的推理结果中都能得出 service1 是否为 ONService 类型的结论。

在传统的推理方法中,以图 2 中的 XML 片段所描述的上下文作为一个服务实例,制定了[(‘\Services\state’ equal ‘on’) -> ‘MarkAsONService’]这样一条规则,其中,作为 Action 的

‘MarkAsONService’指代码中的某方法,该方法用来描述该 Service 的实例是处在运行的状态中。

Fact:(service1 hasServiceONOffState onstate)	(1)
Rule:?a hasServiceOnOffState ?b→?a hasServiceState ?b	(2)
Fact:(service1 hasServiceState onstate)	(3)
Fact:(service1 rdf:type Service)	(4)
Fact:(onstate rdf:type ON)	(5)
Rule:(?s rdf:type Service)/(Z rdf:type ON)(?s hasServiceState ?z)→(?s rdf:type ONService)	(6)
Fact:(service1 rdf:type ONService)	(7)

图 6 本体推理部分过程

表 1 显示了实验的结果,其中的平均推理时间为五次实验的平均值,可以看出传统非语义方法在推理的时间效率上有很大的优势,而基于本体方法的三种不同推理内容在效率上也有差异。另一方面,基于本体方法的代码实现更为方便,代码量比传统方法少很多,这是由于部分工作由 Jena 实现。该实验设计较为简单,但从实验结果中可以得到两种方法在效率以及代码量上的大致区别。

表 1 实例归类实验结果

建模方法		平均推理时间(ms)	代码量(行)
传统非语义方法		1	473
基于本体方法	推理可归类为 ONService 的所有实例	1 431	55
	推理 service1 可能存在的所有归类	2 865	
	推理 service1 和 ONService 的关系	1 357	

(2)实例数量规模对推理效率的影响。

本实验研究当实例数量规模增大时推理时间的变化趋势。按照上一个实验确定的推理规则,逐步从 1 到 2 000 增大服务实例的数量规模,推理内容是推理可归类到 OnService 的所有实例。其中的服务实例均状态相同。

图 7 显示了实验结果。当实例数量增加时,两种方法的总体推理时间都是增长的趋势。在非语义方法的推理以及当实例数量在 500 以内的本体推理中,对于每个实例的平均推理时间是减小的趋势,也就是说推理时间的增加速度要小于实例数量的增加速度。而当本体推理中实例数量超过 500 时,本体推理的推理时间会产生快速的的增长,已不具备对比价值。

(3)自定义规则数量规模对推理效率的影响。

本实验研究当用户自定义规则数量规模增大时推理时间的变化趋势。在前面的实验中,使用的自定义规则可描述为当实例满足某条件,则可将该

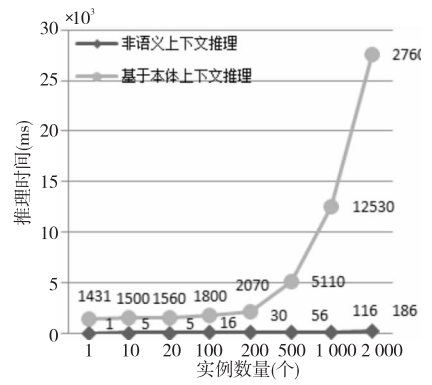


图 7 实例数量变化时推理时间变化趋势

实例归至某个特定类,参照这个推理规则,可定义其他的同类规则,即当实例满足不同的条件时,可被归至不同的类,其中每条自定义规则中所需要满足的条件及可能的归类都是各不相同的。实验中,逐步从 1 到 2 000 增大自定义规则的数量规模,推理内容是推理 service 可能存在的所有归类。

图 8 显示了实验结果。同样,当规则数量增加时,两种方法的总体推理时间都是增长的趋势。基于本体的推理方面,当规则数量为 200 时,推理时间对比规则数量为 100 时的推理时间出现大幅增长,已经超过 1 分钟。这个推理时间在虚拟计算的应用场景中已经不可以接受,所以更大的规则数量下的本体推理在本实验中并未涉及。另外需要指出的是,进行规则数量为 100 的本体推理实验时,默认启动选项下的程序已经出现堆栈溢出的错误,调整了分配堆栈的大小后后续规则数量更大的实验才可以进行。从实验的过程与结果来看,本体推理中自定义规则数量的增大会对推理时间及计算机资源的消耗产生很大影响。对于传统非语义推理来说,规则数量增大对推理效率的影响与实例数量增大产生的影响类似,推理效率方面对比本体推理具有很大的优势。

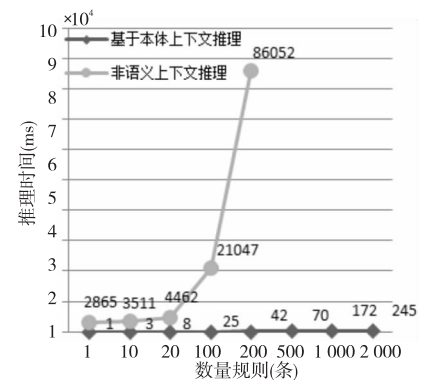


图 8 自定义规则数量变化时推理时间变化趋势

(4)其它方面的比较。

从实验的过程来看,使用 XML 方法进行建模

比较简单,但它不具有语义性,推理能力完全依赖于用户自定义的规则。而基于本体的建模方法,其语义性使得上下文模型具有了实例归类、不一致性检测、完整性检测等能力,对于模型的维护也更为方便。因此,在建模能力上基于本体的方法大大强于传统方法。

在系统实现的易操作性和易维护性方面,传统方法的推理完全需要用户编码实现。虽然实验中已经采取了一些机制,如图3中定义的推理规则模板,来降低代码量以及提高可扩展性,但相对于本体方法,其推理的部分更难实现。当模型和规则发生变化时,代码实现的工作量较大。

## 4 结束语

本文对一种基于本体的上下文建模与推理方法和一种传统的非语义的方法,在建模能力、易操作性和推理效率方面进行了比较。在实际的虚拟计算应用场景中,当构建一个上下文感知的计算平台时,上述实验与分析结果可以用来帮助选择正确的建模和推理方法。实验的结果是基于Jena2默认配置,在将来的工作中,将进一步研究如何结合具体的应用场景在本体建模以及推理机的使用上更为合理,以提高推理的效率。

### 参考文献:

- [1] Anderson D P. BOINC: A System for Public-Resource Computing and Storage[C]// Proc of the 5th IEEE/ACM Int'l Workshop on Grid Computing, 2004:365-372.
- [2] 卢锡城,王怀民,王戟. 虚拟计算环境 iVCE:概念与体系结构[J]. 中国科学 E 辑, 2006,36(10):1081-1099.
- [3] YANG F Q, Lü J, MEI H. Technical Framework for Internetwork: An Architecture Centric Approach[J]. Science in China, Series F, 2008,51(6):610-622.
- [4] Strang T, Linnhoff P C. A Context Modeling Survey[C]// Proc of the 1st Int'l Workshop on Advanced Context Modeling, Reasoning and Management, in coordination with the 6th International Conference on Ubiquitous Computing (Ubi-

Comp 2004), Springer, 2004.

- [5] Schilit B N, Adams N L, Want R. Context-aware Computing Applications[C]// Proc of IEEE Workshop on Mobile Computing Systems and Applications, 1994:85-90.
- [6] Pascoe J. Adding Generic Contextual Capabilities to Wearable Computers[C]// Proc of the 2nd Int'l Symp on Wearable Computers (ISWC 1998), 1998:92-99.
- [7] Schmidt A, Beigl M, Gellersen H W. There Is More to Context Than Location[J]. Computers and Graphics, 1999,23(6):893-901.
- [8] <http://www.w3.org/TR/owl-features/>.
- [9] 黄骏,吴刚,马天笑. 一种情境驱动的网构软件体系架构与支撑环境[J]. 计算机工程与科学, 2010,32(7):130-133.
- [10] <http://jena.sourceforge.net/>.



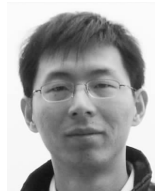
曹晓翀(1986-),男,江苏宜兴人,硕士生,研究方向为情景感知计算。**E-mail:** quriyu@gmail.com

**CAO Xiao-chong**, born in 1986, MS candidate, his research interest includes context aware-computing.



王珊珊(1987-),女,山东临沂人,硕士生,研究方向为情景感知计算和虚拟计算环境。**E-mail:** shanshan.z.wang@gmail.com

**WANG Shan-shan**, born in 1987, MS candidate, her research interests include context-aware computing, and virtual computing environment.



吴刚(1973-),男,江苏南通人,博士,副教授,CCF 会员(E200010717S),研究方向为情景感知计算、分布计算和普适计算。**E-mail:** wugang@cs.sjtu.edu.cn

**WU Gang**, born in 1973, PhD, associated professor, CCF member (E200010717S), his research interests include context-aware computing, distributed computing, and pervasive computing.