

基于本体的上下文感知中间件框架

刘 威¹,王汝传¹,叶 宁^{1,2},马守明¹

(1. 南京邮电大学 计算机学院,江苏 南京 210003;

2. 南京人口管理干部学院 信息科学系,江苏 南京 210042)

摘 要:上下文感知是普适计算的核心技术之一,而描述和理解上下文信息是上下文感知的前提。由于上下文信息种类繁多、感知方式迥异,目前开发面向特定应用的上下文感知系统缺乏统一的机制和通用的架构,增加了系统开发的成本。引入语义 Web 技术,利用本体对上下文信息进行建模,采用本体描述语言描述上下文模型,提供了一个公共的上下文本体以实现多个独立开发的上下文感知系统对知识的共享和推理,构建了通用的上下文感知中间件框架,从而实现对域内上下文知识的共同理解。

关键词:普适计算;本体;上下文感知;中间件

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2010)05-0051-05

Context-aware Middleware Framework Based on Ontology

LIU Wei¹, WANG Ru-chuan¹, YE Ning^{1,2}, MA Shou-ming¹

(1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. Department of Information Science, Nanjing College for Population Programme Management, Nanjing 210042, China)

Abstract: Context-aware is one of core technologies of pervasive computing, and the first thing of context-aware is to describe and understand context information. Because of diverse types of context information and different way of perception, to develop application-specific context-aware systems lack a unified mechanism and a common architecture, and increase system development costs. Introduces semantic Web technology to construct common context-aware middleware framework with use of ontology for context information and ontology Web language for context model. A public ontology is provided to achieve a number of independently developed context-aware systems for knowledge sharing and reasoning. As a result, a common understanding with context knowledge is achieved in one domain.

Key words: pervasive computing; ontology; context-aware; middleware

0 引 言

随着计算机及其相关技术的发展,通信能力和计算能力的价格正变得越来越便宜,所占用的体积也越

来越小,各种新形态的传感器、联网设备快速发展,计算开始分布于每个人的周围。普适计算(Pervasive Computing)成为继主机计算、桌面计算之后的一种新的计算模式,最早起源于1988年Xerox PARC实验室计划,由Mark Weiser首先提出普适计算^[1]的概念。而最早由Schilit在1994年提出来的上下文感知(Context-aware)是普适计算的核心技术之一^[2]。

上下文感知是提高系统智能性的重要途径,其实现的前提是正确表达、理解和利用上下文信息。上下文信息分为3类:

1) 计算上下文,如网络的可用性、网络带宽、通信开销、显示器等资源。

2) 用户上下文,包括用户的个性、位置、社会关系等。

3) 物理上下文,包括光线的明暗、噪声大小、气候、温度等。

收稿日期:2009-08-08;修回日期:2009-11-09

基金项目:国家自然科学基金(60973139, 60773041);江苏省自然科学基金(BK2008451);国家高科技863项目(2007AA01Z404, 2007AA01Z478);现代通信国家重点实验室基金(9140C1105040805);国家和江苏省博士后基金(0801019C, 20090451240, 20090451241);江苏高校科技创新计划项目(CX08B-086Z);江苏省高校自然科学基金(09KJB510020);江苏省六大高峰人才项目(2008118);江苏省青蓝工程项目

作者简介:刘 威(1987-),男,硕士研究生,研究方向为普适计算、无线传感器网络;王汝传,博士生导师,教授,研究方向为网络与信息安全、可信计算技术、无线传感器网络、普适计算、移动代理技术;叶 宁,副教授,博士研究生,研究方向为计算机软件在通信中的应用、天线传感器网络。

Dey A 给出了上下文的定义^[3]：“上下文是环境本身以及环境中各实体所明示或隐含的可用于描述其状态(含历史状态)的任何信息。其中,实体可以是人、地点等物理实体,也可以是诸如软件、程序、网络等虚拟实体。”

从 20 世纪 90 年代初,业界开发了许多上下文感知系统,如 Xerox Palo Alto 实验室开发了 ParcTab 系统,乔治亚理工发布了提供导游服务的 Cyberguide 项目,以及基于用户位置实现电话自动转接的 Active Badge 系统等。不过这些系统都是面向特定应用的,没有一个统一的模型,与底层硬件密切相关,很难作为通用的方法推广到其他项目中。随后 Stanford 大学提出了 ROME 模型^[4],虽然具有一定的普适性,但存在不足。中科院对 ROME 模型进行了改进,在此基础上提出一个基于上下文触发的事务模型 CTMPC(Context-based Triggered Task Model in Pervasive Computing)^[5],该模型适用于在动态的移动环境中开发上下文感知系统。Context Toolkit^[6]是基于组件的分层模型,用功能组件集合屏蔽了底层平台细节,使用 context widget、context server、context interpreter 对上下文信息进行抽象。Context widget 处理底层感知细节,使上下文获取和使用分离,隐藏了传感器的复杂性,容易实现功能扩展和多应用融合。但是该模型缺乏知识共享机制,在大规模系统中不能保持知识的一致性,同时不能提供一个公共的上下文本体来实现系统内多个服务对上下文知识的共享和推理。

文中引入语义 Web 技术,使用本体进行上下文建模,用 OWL(Ontology Web Language,本体描述语言)描述上下文模型,构建通用的上下文感知中间件框架,实现域内上下文知识的共同理解,为独立开发各种面向不同应用的上下文感知系统提供基础支持。

1 基本概念

1.1 上下文感知

上下文感知是指计算系统自动的对上下文、上下文变化以及上下文历史进行感知和应用,并据此做出决策和自动提供相应的响应或者服务^[7],其特征有:将信息和服务呈现给用户;自动执行一个服务;将上下文信息保存下来用于以后的检索。上下文感知涉及上下文的建模与表示、上下文的获取与解释、上下文的存储查询和管理以及上下文的有效利用等。由于上下文种类的繁多和感知方式的迥异,开发面向特定应用的上下文感知系统缺乏统一的机制和通用的架构,增加了系统开发的成本。将底层信息采集处理与高层应用分开,通过引入中间件这样的软件基础设施来处理原始

上下文信息的采集、融合、推理、传播等以降低应用开发的复杂度成为目前比较通用的办法。上下文感知中间件框架主要负责对设备、上下文、物理环境等构成的计算环境进行管理和协调,建立实体对象间互操作的基础,同时屏蔽计算环境的复杂和动态性,为应用开发提供统一的框架和应用编程接口。系统中的上下文来源分为两类:一是从传感器或其他感知设备直接得到的低层上下文,比如位置、温度、时间等;二是利用低层上下文,结合人工智能技术进行推理得到的高层上下文。高层上下文揭示了系统内深层次的关系,是上下文感知系统进行决策的重要依据。

1.2 本体

语义 Web 对当今的 Web 进行了扩充,给其中的信息添加了丰富的语义。在语义 Web 中信息不但能够被人理解和使用,也能被计算机共享和处理。而本体(Ontology)作为语义载体在语义 Web 中起着核心作用,是基于语义的互操作得以实现的关键因素。

本体的概念源于哲学范畴,指的是对客观存在的系统的解释和说明。在不同的领域,本体有不同的定义,关注的焦点也不同。在知识工程领域,Studer 等人给出本体的定义:“本体是共享概念模型的明确的形式化规格说明”^[8]。其中,概念模型是指通过抽象出客观世界中一些现象的相关概念而得到的模型;明确是指所使用的概念以及这些概念的约束都有明确的定义;形式化指本体是计算机可处理的;共享指本体所体现的是共同认可的知识,是由一个团体承认的。本体用于组织较高层次的知识抽象,描述领域知识,捕获相关的领域知识,确定领域内共同认可的概念和概念间的关系,从而形成对该领域的共同理解。

本体的表示方式多种多样,可以用自然语言、框架和逻辑语言等来描述本体。在语义 Web 中两种主要的描述语言是 RDF(Resource Discription Framework)和 OWL。

1) RDF/RDFS。

RDF 是一个建立在 XML 技术之上的资源描述框架,其基本数据类型包括:资源(Resource)、陈述(Statement)、属性(Properties)。任何一个实体都被认为是资源,通过 URI 来标识,属性由属性类型标识,属性有属性值。资源、属性和属性值通过三元组陈述来描述,利用这些陈述的组合建立起对一个客观世界的描述。RDFS 是对 RDF 资源描述文件中出现词汇的定义。

2) OWL。

OWL 是对 RDF 语义描述能力的扩充,能够清晰地表达词汇表中词条的含义以及这些词条之间的关系,同时从 RDF 继承了一些模式特性,基本数据类型

也包括资源、描述和属性。OWL 中的属性分两种:数据类型属性和对象属性,前者表示类的实例与 OWL 固有数据类型的关系,后者表示两个类实例之间的关系。OWL 分为三个子语言,分别是 OWL Lite、OWL DL 和 OWL Full。文中使用 OWL Lite 描述上下文本体。

2 基于本体的上下文感知中间件框架

考虑到本体有利于知识共享和推理等优点,文中提出基于本体的上下文感知中间件框架。支持上下文感知的中间件系统目的在于简化上下文感知应用系统的开发,降低开发成本,一般需实现以下几个目标:一是能够从异构、复杂多样的信息源中获取上下文信息;二是能够保持上下文信息的一致性,解决冲突和矛盾;三是高层服务能够共享上下文知识,使得服务智能化;四是支持用户隐私保护策略。图 1 是基于本体的上下文感知中间件框架。

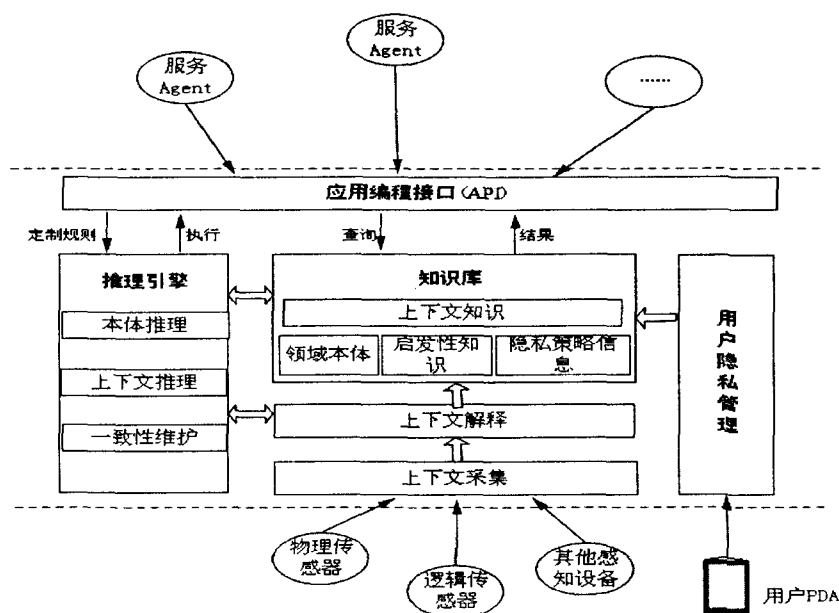


图 1 基于本体的上下文感知中间件框架

在该系统中,高层的上下文应用通过中间件提供的应用编程接口查询、管理和利用上下文知识,而原始上下文的获取由上下文采集模块完成,原始上下文经上下文解释模块解释后和高层上下文一并存入知识库,这样就将上下文的采集和利用分开,屏蔽了底层硬件操作的细节。同时考虑到传感器、手持设备等移动感知设备资源有限,不适合进行上下文解释、推理等计算量大的工作,因此将系统部署在计算资源丰富的服务器上,服务器可以是一台,随着系统规模的扩大,也可部署在多台服务器上,多台服务器之间采用特定的协议通信。下面对各个模块进行详细说明。

2.1 上下文采集

上下文采集的数据来源可以是物理传感器、逻辑传感器或者是其他提供上下文信息的感知设备,如智能卡。逻辑传感器是指能提供上下文信息的软件或网络,如 Web services, Internet。上下文采集模块通过有线或无线的通信方式与各种异构的上下文信息源交互,将获得的原始信息交给上下文解释模块处理,向上层屏蔽了底层的硬件细节。

2.2 上下文解释

由于信息源的异构性,上下文采集的时间、方式各异,因而上下文信息的表示方式不一致,并且具有单一、低层、不精确、不稳定等特点。上下文解释模块将原始的上下文抽取、转换成用 OWL 描述的统一格式,然后将这些上下文知识存储在知识库中。为了消除原始上下文的不精确性,甚至冲突,上下文解释模块与推理引擎频繁的交互,利用推理引擎的一致性维护模块解决冲突。同时为了获得更有价值的高层上下文,解

释模块借助于本体推理,通过对低层上下文的推导形成高层上下文,并存入知识库。冲突解决和形成高层上下文的方法将在推理引擎模块中详细说明。

2.3 知识库

知识库存储着大量的领域本体和上下文等信息,是系统的重要组成部分。它包括:上下文知识、启发性知识、隐私策略信息和领域本体。

1) 上下文知识,是经上下文解释模块处理后的上下文信息,也即领域本体的实例,用 OWL 统一描述。可以通过 API 被上层实时查询,也是上下文推理的基础。

2) 启发性知识,被推理引擎的一致性维护模块使用,用来检查知识库中的上下文信息是否存在不一致性。如“同一个人不可能同时在两个不同的地点出现”、“人的体温不可能是 100 度”等都是启发性知识。

3) 隐私策略信息,是对用户的上下文信息采取的访问控制机制,由用户通过 PDA 或者其他手段提交给系统的用户隐私管理模块,该模块向知识库中写入隐私策略信息。隐私策略信息规定了用户哪些上下文信息可以被服务 Agent 使用,哪些信息不可以被使用,从而有效保障了在动态开放的环境下用户的安全。

4) 领域本体,由领域专家和知识工程师手动或机器自动构建,并预先存储在知识库中,领域本体构建的

好坏在很大程度上影响着上下文感知系统的性能。如果要开发多个领域的上下文感知应用系统,就必须对知识库的领域本体进行扩充,以满足多领域的要求。下面用 OWL 语言对智能会议室 SmartMeetingRoom 的部分本体进行描述,定义了 Place, Room, Building, People 四个类以及数据类型属性 name, role 和对象类的属性 isNowIn, isPartOf, 同时给出预先定义和系统实时采集的该本体的实例。

```
<owl:Ontology rdf:about="http://www.njupt.edu.cn/SmartMeetingRoom">
  </owl:Ontology>
  <owl:Class rdf:ID="Place" />
  <owl:Class rdf:ID="Room" />
  <owl:Class rdf:ID="Building" />
  <owl:Class rdf:ID="People" />
  <owl:DatatypeProperty rdf:ID="name">
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="role">
    <rdfs:domain rdf:resource="#People" />
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="isPartOf">
    <rdfs:domain rdf:resource="#Room" />
    <rdfs:range rdf:resource="#Building" />
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isNowIn">
    <rdfs:domain rdf:resource="#People" />
    <rdfs:range rdf:resource="#Place" />
  </owl:ObjectProperty>
  <owl:Class rdf:ID="Room">
    <rdfs:subClassOf rdf:resource="#Place">
  </owl:Class>
  <owl:Class rdf:ID="Building">
    <rdfs:subClassOf rdf:resource="#Place">
  </owl:Class>
  <Building rdf:ID="TechBuilding">
    <name>科研楼</name>
  </Building>
  <Room rdf:ID="MeetingRoom">
    <name>会议室</name>
```

```
<isPartOf> TechBuilding</isPartOf>
</Room>
<Room rdf:ID="RetiringRoom">
  <name>休息室</name>
</Room>
<People rdf:ID="P1">
  <name>Alice</name>
  <role>speaker</role>
</People>
```

2.4 推理引擎

推理引擎是上下文感知中间件框架的核心,包括三个子模块:本体推理、一致性维护和上下文推理。

1) 本体推理,结合领域本体,将低层上下文信息转换为高层上下文并存入知识库,由上下文解释模块调用。下面根据智能会议室的一个场景具体说明本体推理的过程:会议成员 Alice 进入会议室,部署在会议室的位置传感器感知到 Alice 携带的 PDA 而得知 Alice 进入了会议室,于是向系统中发送了这样一条上下文:

```
<People rdf:ID="P1">
  <name>Alice</name>
  <isNowIn> MeetingRoom</isNowIn>
</People>
此时,知识库中已有上下文:
<Room rdf:ID="MeetingRoom">
  <name>会议室</name>
  <isPartOf> TechBuilding</isPartOf>
</Room>
```

通过本体推理推导出 Alice 现在在科研楼。

2) 一致性维护,维护知识库中知识的一致性。知识的不一致包括两个方面:一是从底层设备带来的上下文不稳定、不精确性;二是知识库中知识没有及时反应出环境的动态变化。一致性维护模块借助于启发性知识解决知识的不一致问题。有下面的场景:Alice 把 PDA 落在会议桌上,去休息室休息,休息室内的传感器感知到 Alice 穿戴的智能卡,识别出 Alice 进入休息室,于是向系统通报一条上下文信息:

```
<People rdf:ID="P1">
  <name>Alice</name>
  <isNowIn> RetiringRoom</isNowIn>
</People>
但是系统知识库中已有一条上下文信息:
<People rdf:ID="P1">
  <name>Alice</name>
  <isNowIn> MeetingRoom</isNowIn>
</People>
```

根据启发性知识“同一个人不可能同时在两个不同的地点出现”,一致性维护模块发现知识库中存在不一致的知识,利用“身份识别智能卡优于PDA”的策略,该模块删除后一条上下文,只保留前一条。

3) 上下文推理,最能体现上下文感知系统的智能化服务。该模块根据用户向系统定制的规则,读取知识库中上下文知识与规则进行匹配,如果满足规则的前提就执行规则指定的动作。规则可以采用三元组的形式表示,由于上下文知识采用 OWL 语法描述,在与规则匹配之前需要转化为三元组形式,具体转化方法可参考文献[9],上下文推理模块可以用 JESS(Java Expert System Shell,用 Java 实现的基于规则的专家系统)构建^[10]。假如系统中存在如下规则:

IF (isNowIn, Alice, MeetingRoom) ^ (role, Alice, speaker) ^ nowTime(AM9:00) ^ (beginAt, Conference, 9:00)

THEN prepare for Alice's presentation

这条规则表示如果报告人 Alice 在会议室,并且此刻是会议开始时间上午 9 点,那么智能会议室开始为 Alice 的报告作准备,如启动投影机,准备幻灯片等。

2.5 用户隐私管理

用户首次进入系统后,通过手持 PDA,向用户隐私管理模块提交自定义的隐私策略文件,文件规定用户上下文信息的访问策略。该模块将隐私策略文件或文件的引用存入知识库中,当上层服务访问用户上下文信息时,服务 Agent 通过 API 读取用户隐私策略信息,并按照该策略访问授权的上下文信息,未被授权的信息禁止访问,从而有效地保护了用户的隐私。

2.6 应用编程接口(API)

应用编程接口(API)为上层应用提供开发接口,以便开发者充分利用中间件系统的功能快速构建感知应用,它包括一组接口和函数,使用这些接口和函数可以直接对知识库进行操作。知识库中存放的上下文知识以及领域本体等都是用 OWL 语言描述的,因而对知识库的查询和管理需要借助 RDQL(RDF Data Query Language)实现,RDQL 查询语言类似于 SQL。HP 的 Jena 支持 RDQL,包含对上下文信息进行操作的接口,所以应用编程接口(API)必须封装这些接口实现对知识库中上下文信息的查询、添加和修改。

3 结束语

文中引入语义 Web 技术,使用本体建模上下文信息,用 OWL 语言描述本体模型,构建了通用的上下文感知中间件框架。该系统相比其他上下文感知系统框架优点在于:上下文知识使用统一的描述,实现了不同

服务之间的知识共享,形成了对域内知识的共同理解。并且本体本身具有语义信息,使用本体推理增强了推理引擎的功能。最终实现了上下文感知系统分布式开发和软件复用的目的。

然而本体论本身还不完善,特别是本体构建方法、方式都不成熟,因而本体的应用还处于初级阶段,基于本体构建上下文感知系统也因此面临许多挑战。领域本体的构建、基于 JESS 的推理引擎的设计实现以及结合人工智能技术对推理机制进一步的研究将是下一阶段的研究重点。

参考文献:

- [1] Weiser M. The computer for the 21st century[J]. Scientific American, 1991, 265(3): 66-75.
- [2] Schilit B N, Adams N, Want R. Context-aware Computing Applications[C]//IEEE Workshop on Mobile Computing Systems and Applications. Washington, D. C., USA: IEEE Computer Society, 1994.
- [3] Dey A, Abowd G. Towards a better understanding of context and context-awareness[C]//Workshop on the what, who, where, when and how to context-awareness at CHI 2000. New York, USA: ACM Transactions on Computer Systems, 2000.
- [4] Andrew C H, Benjamin C L, Shankar P. Pervasive computing: What is Good for[C]//The International Workshop on Data Engineering of Wireless and Mobile Access(MobiDE) in conjunction with ACM MobiCom 99. New York, USA: ACM Transactions on Computer Systems, 1999: 84-91.
- [5] 罗俊伟,秦晓,陈思功.普适计算中基于上下文触发的事务模型[J].小型微型计算机系统,2004,25(8):1542-1545.
- [6] Daniel S, Anind K D, Gregory D A. The context toolkit: aiding the development of context-enabled application[C]//Proceedings of the CHI 99 Conference on Human Factors in Computing Systems. New York, USA: ACM Transactions on Computer Systems, 1999: 434-441.
- [7] Chen G, Kotz D. A survey of context-aware mobile computing research[R]. Hanover, N. H., USA: Dartmouth College, 2000.
- [8] Studer R, Benjamins V R, Fensel D. Knowledge Engineering: Principles and Methods[J]. IEEE Transactions on Data and Knowledge Engineering, 1998, 25: 161-197.
- [9] Kopena J, Regli W C. DAMLJessKB: A tool for reasoning with Semantic Web[J]. IEEE intelligent systems & their applications, 2003, 18(3): 74-77.
- [10] Ernest J, Friedman-Hill Jess. The Expert System Shell for the Java Platform[M]. Livermore, California, USA: Sandia National Laboratories, 2003.