

文章编号:1001-9081(2005)12-2957-04

## 有上下文觉察能力的智能会议室系统模型的研究

满君丰<sup>1</sup>,金可音<sup>1,2</sup>,胡永祥<sup>1</sup>

(1. 株洲工学院 计算机系, 湖南 株洲 412008; 2. 中南大学 信息科学与工程学院, 湖南 长沙 410083)  
(mjfok@tom.com)

**摘 要:**通过利用多 Agent、语义 Web、上下文觉察和逻辑推理等技术,参考早期的普适计算系统,提出了一个新的智能会议室系统模型。该模型使用本体 Web 语言——OWL 实现知识共享和上下文推理;使用逻辑推理检测和解决上下文知识的不一致问题,并向使用者提供一种策略语言来控制他们的私人信息。

**关键词:**语义 Web;本体;普适计算;智能会议室;上下文;推理

**中图分类号:** TP312 **文献标识码:** A

## Research on the model of smart meeting room with context-aware capacity

MAN Jun-feng<sup>1</sup>, JIN Ke-yin<sup>1,2</sup>, HU Yong-xiang<sup>1</sup>

(1. Computer Science Department, Zhuzhou Institute of Technology, Zhuzhou Hunan 4120081, China;  
2. College of Information Science and Technology, Central South University, Changsha Hunan 4100831, China)

**Abstract:** Taking advantages of the technologies of multi-agent system, Semantic Web, context-aware and logical inference, consulting earlier pervasive computing system, a new model of Smart Meeting Room (SMR) was presented. SMR used Ontology Web Language-OWL to support knowledge share and context reasoning, used logical inferences to detect and resolve inconsistent context knowledge, and provided users with a policy language to control their private information.

**Key words:** semantic Web; Ontology; pervasive computing; smart meeting room; context; reasoning

## 0 引言

在普适计算过程中,计算机系统将无缝地融入每一个用户的日常生活,以随时随地的方式为他们提供信息和服务。在这种公开的、动态的环境下,智能信息处理实体必须要实现知识共享、对周边环境的推理和服务互操作等功能。

目前,众多研究结构纷纷开发出自己的原型系统和体系结构,例如,用有上下文觉察能力的手持设备为博物馆来访者创建个性化指导<sup>[1]</sup>;当用户进入办公室时,可以将资源有限且处理能力低下的移动设备(如手机或 PDA)的应用程序动态迁移到附近的资源丰富且处理能力强大的台式机上<sup>[2]</sup>。然而,这种信息的迁移既费时又费力,因为没有为这些设备提供全局性的控制和协调机制,各设备基本上是处于“信息孤岛”状态,所以对知识共享和推理只能提供微弱的支持,这难以满足智能空间环境下以智能设备为载体自如地交互信息和共享知识的要求。为了在普适计算过程中真正实现智能化,应该提高以往系统的数据处理效率和推理能力。本文通过一个新的智能会议室系统(Smart Meeting Room, SMR)模型,对智能空间环境下如何实现知识共享、上下文推理、检测和解决上下文不一致问题进行探讨。

SMR 是由智能代理中心、服务、设备和传感器等部分组成的分布式系统,这些部分构成一个团体,它们相互协调,为参加会议者(如发言者、与会者和组织者)提供基于上下文环境的相关信息和服务。

SMR 系统实现的关键在于具备独特的上下文觉察功能<sup>[3]</sup>。上下文可以用来表征个人或计算机实体状态的任何

信息。文献[1,2]把位置信息看作上下文的重要组成部分。我们认为:除了位置信息,对上下文的理解还应该包括一些其他的信息,这些信息可以用来描述系统的能力,提供用于查找操作的服务,用户和计算机实体参与的活动和任务,以及他们的角色、信任度、要求和意图等。

SMR 的重要特性包括使用语义 Web 语言表示知识;支持上下文推理;使用策略语言<sup>[4]</sup>控制用户上下文信息共享;使用逻辑推理检测和解决上下文知识的不一致问题。

## 1 SMR 的结构

### 1.1 智能会议室的应用情形

R101 是一间智能会议室,其入口处装备有 RFID 传感器,可以对参加会议人员进行考勤。当 Tom 走进会议室时,这些传感器通过获取其手机上的信息来通知 R101 的上下文代理中心(Context Broker Center, CBC) Tom 进入会议室了,然后 CBC 将该事实加入到知识库中。

当 Tom 坐下,他手机中的 Agent 会通过蓝牙设备发现 R101 的 CBC,然后将手机主人的个人信息控制策略告知 CBC,这样手机和代理就可以使用握手协议(例如验证代理身份后建立信任)建立通信联系。下面这个控制策略描述了 Tom 要求代理中心该做什么:(1)通知 CBC 是否允许其他设备存取他的上下文信息(如允许存取他的位置和角色,不允许存取他拨打的电话号码);(2)当他的上下文信息发生变化时,CBC 应通知其他 Agent(如把他的位置信息通知给其在家的个人 Agent);(3)允许其他代理获取他的上下文信息(如当他在会议室时,会议室里的所有其他 Agent 都可以获取他的

收稿日期:2005-06-21;修订日期:2005-08-26

基金项目:国家自然科学基金资助项目(60373062);湖南省杰出中青年基金资助项目(02JJYB01)

作者简介:满君丰(1976-),男(满族),黑龙江海伦人,讲师,硕士,主要研究方向:语义 Web、普适计算;金可音(1954-),男,湖南湘潭人,教授,主要研究方向:语义 Web、网络信息安全;胡永祥(1973-),男,湖南邵阳人,讲师,硕士,主要研究方向:图形处理、数字水印。

上下文信息)。

在接到 Tom 的个人信息控制策略后, CBC 为他创建一个简要的描述信息(Profile), 当 CBC 处理和他有关的上下文知识信息时, 将遵守该描述信息的规则和约束。例如, 在上述控制策略条件下, Tom 的描述信息会指导代理: (1) 获取和推理 Tom 的位置和行动的上下文信息; (2) 当 Tom 的上下文信息变化时, 通知他在家的个人 Agent; (3) 会议室里的 Agent 共享他的上下文信息。

知道 Tom 的手机现在在 R101, 并且没有与之相反的事实, 代理推断出 Tom 也在那里。另外, 因为 R101 是计算机楼的一部分, 而计算机楼又是学校的一部分, 于是代理推断出 Tom 在计算机楼, 也在学校。这些结论被加入到代理的知识库中。

按照其描述信息, CBC 与 Tom 的个人 Agent 联系。因为 Tom 已经把今天的工作日程委托给其个人 Agent 管理, 则其个人 Agent 知道在这个时间段内 Tom 将在学校做会议报告, 个人 Agent 就会将该事实告知 R101 的 CBC。

R101 的 CBC 收到有关 Tom 意图的信息后, 会把这个消息告知 R101 的放映机 Agent 和灯光控制 Agent。几分钟后, 放映机 Agent 从 Tom 的个人主页处下载他要用的幻灯片, 灯光控制代理调暗灯光, Tom 就可以开始会议报告了。

## 1.2 SMR 的结构

由于需要支持资源相对有限的小型设备进行上下文信息的获取和推理, 所以 SMR 的体系结构采用集中式模式。SMR 的核心是 CBC。在智能空间中, CBC 具有如下责任: (1) 提供一个上下文集中式模型, 它可以被空间中的所有设备、服务和代理共享; (2) 从那些设备难以到达的信息源处获取信息; (3) 通过推理获取不能直接从传感器获得的信息(如意图、角色、暂时的空间关系等); (4) 检测和解决在上下文共享模型中的不一致的知识; (5) 通过使用用户定义的控制策略来控制共享和上下文信息的使用, 以便保护他们的隐私。

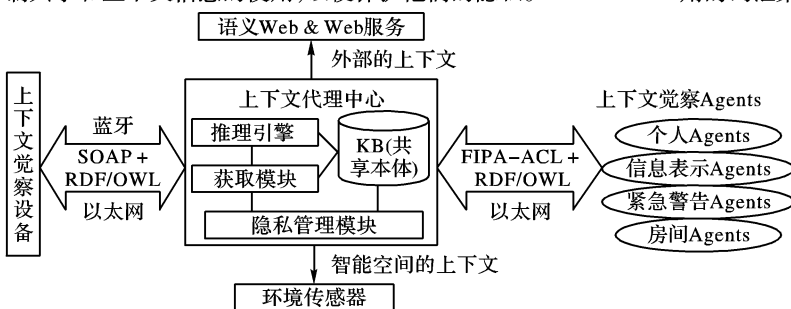


图1 SMR 的体系结构

图1 是 SMR 的体系结构。其 CBC 有下面四个功能部件:

(1) 上下文知识库: 存储上下文知识。它为代理的其他组件提供了一套用于存取知识的 API。它还包括特定的智能空间本体和一些与时间、空间有关的启发性知识(如 13:00 到 15:00 开会; 在同一个时间段, 一个人不可能同时出现在两个不同的地方等)。

(2) 上下文推理引擎: 一个利用储存的上下文知识进行推理的交互性推理引擎。在该引擎中可以进行两种推理: ①使用本体推导出上下文知识; ②利用启发式知识检测和解决不一致的知识。

(3) 上下文获取模块: 一个为上下文获取而形成的中间件抽象程序库, 它为高层应用程序屏蔽了低层的传感设备。

(4) 策略管理模块: 一套为实施用户策略而设置的推理规则。一些推理规则用来确定不同信息处理实体共享一个上下文信息的正确权限, 一些推理规则负责当上下文信息变化

时通知与其相关的 Agent。

为了实现上述 SMR 结构, 我们应该考虑到以下问题: (1) 怎样开发一套支持知识共享和上下文推理的共享本体; (2) 如何用逻辑推理来检测和解决上下文知识的不一致; (3) 如何为用户提供策略语言来控制他们的私人信息。由于文章篇幅限制, 我们主要阐明前两个问题。

## 2 面向普适计算的共享本体

OWL 语言是一种被计算机应用程序使用的语义 Web 语言, 该语言作为语义 Web 的一部分是由 W3C 开发的。OWL 是一种用于定义和实例化本体的表示语言, 它不仅为用户显示信息, 而且有助于计算机理解 and 处理信息<sup>[5]</sup>。

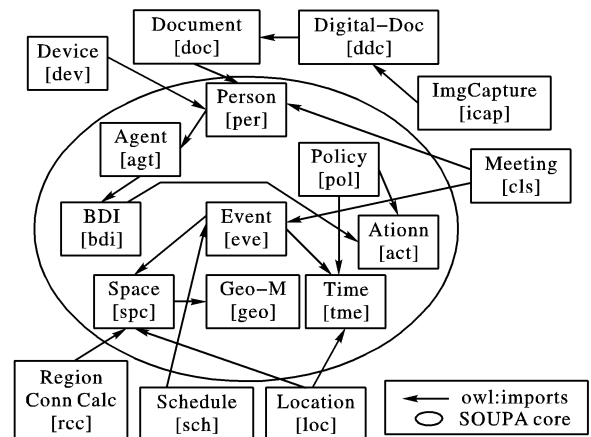


图2 SO4PC 的结构关系

为了实现知识共享和上下文推理, 我们设计了一套用于普适计算的共享本体(Shared Ontologies for Pervasive Computing, SO4PC)。SO4PC 的优点在于为普适计算的开发者提供一个共享本体, 该本体将来自不同共识本体的许多有用的词汇集中在一起<sup>[6]</sup>。我们相信 SO4PC 不仅能胜任 SMR, 还能帮助那些不熟悉知识表示的开发者快速建立本体应用程序, 而不需要重新定义本体, 使他们可以集中更多的精力用于系统的实现。

SO4PC 参考的本体包括 Friend-Of-A-Friend 本体(FOAF)<sup>[7]</sup>、DAML-Time、OpenCyc<sup>[8]</sup>的空间本体、区域联系演算(Regional Connection Calculus, RCC)<sup>[9]</sup>、COBRA-ONT、MoGATU BDI<sup>[10]</sup>本体和 Rei 策略本体<sup>[4]</sup>。SO4PC 由两部分组成: SO4PC 核心和 SO4PC 扩展。

SO4PC 核心本体集合定义了通用词汇, 它们适用于不同的普适计算的应用程序。由核心本体扩充而来的 SO4PC 扩展本体集合定义额外的词汇用于支持特殊类型的应用程序, 并且为未来的本体拓展提供实例。本体集合由表示概念的词汇组成, 这些概念和人、Agent、信念-期望-意图(Belief-Desire-Intention, BDI)、行动、策略、时间、空间和事件有关。SO4PC 包含 9 个不同的本体文件。图 2 表明了本体文件和它们的相互关系。

## 3 上下文推理

SMR 支持两种类型的上下文推理: (1) 建立上下文共享模型的推理; (2) 维护上下文一致性模型的推理。第一种类型的推理包含使用推理来解释觉察信息, 第二种类型的推理包含使用推理来检测和解决不一致信息。

### 3.1 上下文推理的方法

上下文推理对每一个上下文觉察系统来说都是至关重要

的。上下文觉察系统的功能取决于对上下文推理的支持。在早期的系统中<sup>[1,2]</sup>,典型的上下文推理是通过使用 Java 和 C++ 这样的语言编写过程程序实现的。这些面向对象语言在 API 上支持代码重用,然而他们对于知识表示缺乏表达力。当使用这些语言进行上下文推理时,逻辑推理的非陈述性表示往往会使代码修改和错误检查很难进行。为了解决这些问题,我们设计一套基于规则的逻辑推理方法。这个方法的优点如下:

(1) 一个上下文推理规则的显式表示有助于把高层推理逻辑和低层的功能实现划分开来。通过这种划分,开发者可以修改或替换上下文推理的某些组件,而不需要重新编写大量程序;

(2) 基于规则的方法允许很多已定义的一般概念的逻辑模型被直接映射到上下文推理实现上,如时间和空间。这种和一般概念有关的逻辑推理经常用于上下文推理;

(3) 当明确表示出上下文解释规则时,元推理技术可以用来检测和解决上下文解释的不一致。

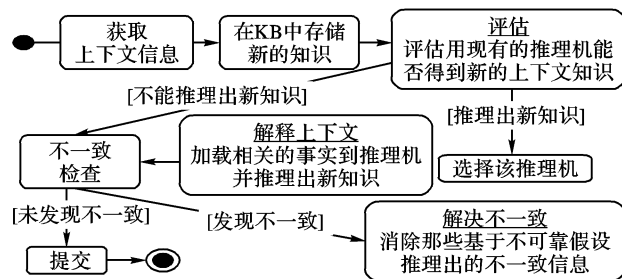


图3 用UML描述上下文推理过程

### 3.2 时间推理

在描述事件时,时间是一个中心概念。在 SMR 中可以包括统计会议出席情况、行为的实现、代理之间发送和接收信息、服务和设备内部状态的变化等事件。描述这些事件都用到了时间,所以 CBC 可以推理出他们相互之间的时间顺序关系。这些针对时间推理的逻辑结论是以基于 SO4PC 的时间本体为基础的。

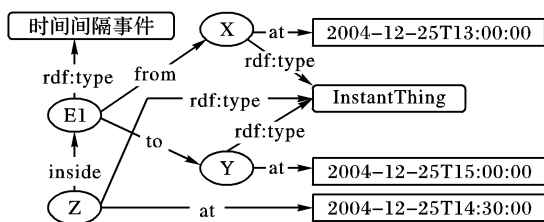


图4 用本体表示时间关系

时间的基本表示有由 `tme: Time-Instant` 和 `tme: TimeInterval` 两个类组成。`tme: Time-Instant` 是一种瞬间时间类型, `tme: TimeInterval` 是一种时间间隔类型, 这两个类结合起来构成了 `tme: TemporalThing` 类。为了把与时间有关的事件与日期/时间值联系起来, 用 XML 格式的 `xsd: dateTime` 数据类型值(如 2004-12-25T14:30:00)把 `tme: at` 属性定义为与 `tme: Instant-Thing` 的实例相联系; 用两个不同的 `tme: Time-Instant` 时刻把 `tme: from` 和 `tme: to` 属性定义为与 `tme: IntervalThing` 的实例相联系。

例1:推断出手机的主人在开会。

A1: `TimeInterval(13:00am, 15:00am)`, 在 R101 举行会议。

A2: 现在是 14:30am (即 `Time-Instant(14:30am)`)。

A3: CBC 已经检测出 `DeviceAt("Tom", "Mobile", "R101")`。

A4: 没有证据可以证明 Tom 在其他地方。

A5  $\leq A1 + A2$ ; A2 是 A1 的一时间实例。

A6  $\leq A3 + A4 + A5$ ; 14:30am 时刻, Tom 在 R101 开会。

### 3.3 空间推理

空间信息是上下文信息的重要部分。对某个对象的位置信息的逻辑推断被称为空间推理。在 SMR 中, 空间推理不仅仅是知道物理对象的位置, 还包括使用地理空间数据和本体来获得那些不能直接用传感器得到的新的空间知识。

在 SMR 中, SO4PC 空间本体是空间推理实现的核心部分。这些推理假定所有的空间信息都可以用于 SO4PC 空间本体的表示。基于本体语义, 支持以下逻辑推断: (1) 给定地理位置 X 和 Y (它们是 `spc: GeographicalSpace` 类的实例), 推断出 Y 是否包含 X, 或 X 是否包含 Y; (2) 给定两个地理位置 X 和 Y, 推断出它们是不是相联系的; (3) 如果 Y 在空间上包含 X, 根据已定义经度和纬度的地理位置 X, 推断出 Y 的地理位置。

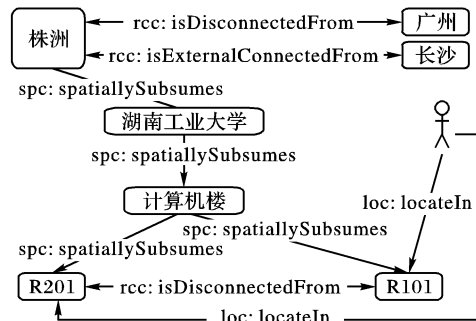


图5 用空间本体推理

CBC 可以在很多不同方面利用空间推理: (1) 利用空间推理推断不能直接从传感器获得的一个位置的空间属性。例如, 从 GPS 设备上读到的个人 PDA 的信息, CBC 可以推理获得有关这个人位置的一些基本信息(如经纬度); (2) 空间推理可用于检测有关个人位置的不一致的信息。例如, 根据已定义的本体, 房间 A 和房间 B 在空间上不连接, 但 CBC 获得的两个不同的信息表明同一个人在同一个时间间隔内既在房间 A 中, 又在房间 B 中。利用空间推理可以推断出一个人不可能同时在两个不相连接的房间中出现。因此, 得出结论: 位置信息存在不一致。第三, 通过协调用户位置信息的粒度, CBC 可以利用空间推理来保护他们的个人信息。例如, 基于一套已定义好的用户位置信息的粒度规则, CBC 可以泛化一个人的位置信息, 该信息可以被该人还未信任的 Agent 共享, 这样既共享了必要的信息, 又在适当的粒度上保护了该人的个人隐私。

例2:确定 Tom 是否在计算机楼。

B1: `Person("Tom")` 拥有属性 `isCurrentlyIn("R101")`。

B2: 对于任何一个有属性 `isCurrentlyIn()` 且其属性范围 `rdfs: range` 限制为 `isPartOfBuilding` 的人, 那么该人一定是 `PersonInBuilding` 的一个实例(即那个人在这所建筑内)。

B3  $\leq B1 + B2$ : `Person("Tom")` 是 `PersonInBuilding` 类的一个实例(即 Tom 现在在这所建筑内)。再者, 因为 `Room("R101") isPartOf Building("计算机楼")`, 代理可以推断出 Tom 现在在计算机楼。

例3:确定 Tom 今天是否在学校。

C1: `Person("Tom")` 现在在 `Building("计算机楼")` 中。

C2: `Building("计算机楼") isPartOf Campus("湖南工业大学")`。

C3  $\leq C1 + C2$ : `Person("Tom")` 今天在学校。

### 3.4 检测不一致

使用 OWL 语言可以检测不一致的信息, 即将上下文进行本体模型化, 并利用本体推理来检测不一致的信息。OWL 语

言是表示性语言,它可以表示上下文的不同方面。这包括上下文中的对象,这些对象之间的关系,和对这些关系的约束。一旦本体被定义,就可以将观测和获取的上下文信息作为本体的数据实例表示出来。给定一个本体和一些实例数据,本体推理可用来推断实例数据描述的上下文是不是和本体定义的模型存在不一致。

假设如下情形:Tom 进入会议室(R101)。这间会议室检测到他的手机中的蓝牙设备。由于没有其他与之相反的证据,代理推断出 Tom 也在会议室,并通知 CBC。片刻后, Tom 匆忙离开会议室,但把手机忘在了那里。从会议室离开后, Tom 回了办公室(R201)。当他进入办公室时,其屋内安装的语音识别设备识别出他的声音,并通知 CBC: Tom 在他的办公室。根据空间本体知识(如图 5), R201 在空间上和 R101 是相互独立的两个房间。根据这些信息,并且没有证据证明 Tom 已经从任一房间离开, CBC 得出的结论是关于 Tom 的位置信息存在不一致。

例 4: 为了检测 Tom 的不一致的位置信息。

D1: Person (“Tom”) 有属性 isCurrentlyIn (“R101”)。

D2: Person (“Tom”) 有属性 isCurrentlyIn (“R201”)。

D3: 根据已有知识, R101 和 R201 在空间上是相互独立的房间。

D4  $\Leftarrow$  D1 + D2 + D3: CBC 得出的结论是关于 Tom 的位置信息存在不一致。

### 3.5 使用逻辑推理解决不一致

信息不一致是由提供信息的源 Agent 引起的。因为这些 Agent 可能已经使用了不准确的传感器数据或有缺陷的领域推断来解释上下文,所以他们获得的信息可能存在不一致。第 3.4 节描述的情形是一个典型的例子,它表明了由于源 Agent 检测的局限性和推理不准确从而导致了信息不一致情况的出现。

我们开发了一个基于假设的逻辑推理方法来解决信息的不一致。在这个方法中,上下文 CBC 通过衡量各 Agent 的用于得出他们结论的假定的可信度来得出最可靠的假定。

我们使用逻辑推理系统来实现关于假设的建模和推理。在实现中, CBC 有一个关于假设的明确模型,其他 Agent 用该模型来解释上下文。当 CBC 检测不一致的信息时,通过和其他 Agent 通信,它动态获得假设,然后使用基于假设的逻辑推理来解决信息的不一致问题。

在实现中,推理系统设计采用如下形式:

(1) 假定所有的 CBC 所使用的本体都是一致的。CBC 不需要检测和解决由于不准确的本体规范而导致的的不一致信息的问题;

(2) 所有的 Agent 都和 CBC 共享一个通用的本体。这个通用的本体包括词汇,以及用于描述上下文和模型化推理假设的相关语义;

(3) 当 CBC 要求 Agent 证明他们对上下文的信任时,所有的 Agent 都不应有意提供错误信息来欺骗上下文 CBC。换句话说就是,没有说谎的 Agent,且所有的 Agent 都是相互合作的;

(4) CBC 只负责解决各种不一致的信息,而且它有足够的相关知识去解决这些不一致情况。CBC 的内在逻辑推断先定义这些知识。

现在用第 3.4 节的情形阐述该推理方法。当检测到 Tom

的不一致的位置信息时, CBC 查询对应的传感器来证明它的假定:(1) 查询监测器 S1 来获知 Tom 的手机 cellphone(moto\_v80)的状态;(2) 查询传感器 S2 来获得 R201 的噪音情况。通过 S1 的查询结果, CBC 知道 Tom 的手机已经有 8 个来电没有接听,而且最近时间内没有拨出电话。通过 S2 的查询结果, CBC 知道 R201 的噪音水平是 50db,属于正常范围。这样 CBC 证实了对 S2 的假设更可靠,不一致问题得到解决。

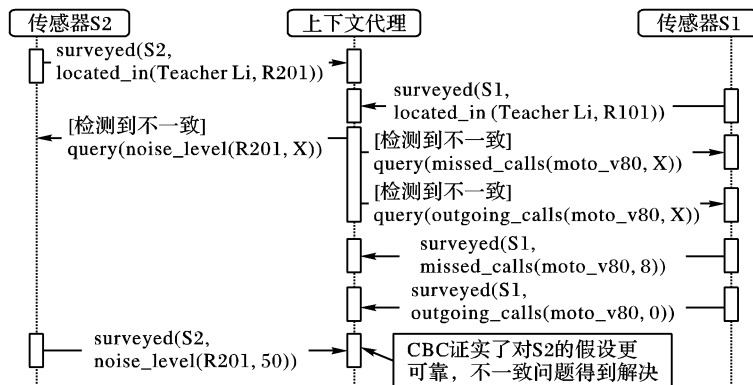


图6 用基于假定的推理解决不一致

## 4 结语

该 SMR 模型论证了如下可行性:(1) 使用 OWL 本体可以使分散的 Agent 共享知识;(2) 对上下文信息进行逻辑推理;(3) 用策略语言来控制他们的个人隐私。但要使 SMR 模型真正实施,还有很多问题要解决,包括在分布的、动态环境下知识共享的可测量性;在大量检测数据出现时,上下文推理的性能和时间复杂度,以及与编辑和维护用户隐私的控制策略有关的用户接口问题。将进一步扩大上下文觉察支持,包括追踪缺席者的位置,追踪便携式投影设备的可用性。

### 参考文献:

- [1] KINDBERG T, BARTON J. A Web-based nomadic computing system[J]. Computer Networks, 2001, 35(4): 443-456.
- [2] BENNETT F, RICHARDSON T, HARTER A. Teleporting - making applications mobile[A]. Proceedings of 1994 Workshop on Mobile Computing[C]. 1994.
- [3] CHEN H, FININ T, JOSHI A. Semantic Web in the context broker architecture[A]. Proceedings of PerCom 2004[C]. 2004.
- [4] KAGAL L, FININ T, JOSHI A. A Policy Based Approach to Security for the Semantic Web[A]. 2nd International Semantic Web Conference (ISWC2003)[C]. 2003.
- [5] 满君丰, 邱银安, 陈青. 语义 Web 服务框架模型研究[J]. 计算机集成制造系统, 2005, 24(10).
- [6] MAN JF, CHEN Q, DENG XH. The Design and Implementation of Shared Ontology for Smart Space Application[A]. The proceeding of the fourth International Conference on Machine Learning and Cybernetics(ICMLC 2005)[C]. Guangzhou, 2005.
- [7] MCGUINNESS DL, VAN HARMELEN F. OWL Web Ontology Language Overview[A]. In Proposed Recommendation (PR) for OWL [C]. Web Ontology Working Group, W3C, 2003.
- [8] BRICKLEY D, MILLER L. FOAF vocabulary specification [Z]. In RDF Web Namespace Document, RDFWeb, xmlns.com, 2003.
- [9] PAN F, HOBBS JR. Time in OWL-S[A]. Proceedings of AAAI-04 Spring Symposium on Semantic Web Services[C]. Stanford University, California, 2004.
- [10] PERICH F. MoGATU BDI Ontology[EB/OL]. http://www. MoBDI.com/bdi.owl, 2004.