

# 一种情境感知服务系统框架

莫 同 李伟平 吴中海 褚伟杰

(北京大学软件与微电子学院 北京 100871)

**摘 要** 根据感知的情境信息将业务服务组合成情境感知服务是提高服务水平的有效途径. 现有研究多关注于情境信息及服务的表示, 而对情境感知服务的构造特征和业务运行机制缺乏充分考虑. 一方面, 各情境感知服务的情境信息和业务服务重复性较强, 通过预定义流程的紧耦合构造方式难以支持重用; 另一方面, 情境信息取值变化具有随机性和动态性, 基于推理的触发式调用执行代价过高. 基于情境感知服务特征分析提出了一种情境感知服务系统框架 CBS<sup>3</sup>, 给出了情境信息与业务服务的配置模型和构造机制. 通过社区老年情境感知服务的案例说明使用 CBS<sup>3</sup> 建立情境感知服务的过程. 对比分析的结果表明, 与传统方法相比, CBS<sup>3</sup> 能够提高情境感知服务重用度和执行效率.

**关键词** 情境信息; 情境感知; 情境感知服务; 配置模型

中图法分类号 TP311 DOI号: 10.3724/SP.J.1016.2010.02084

## Framework of Context-Aware Based Service System

MO Tong LI Wei-Ping WU Zhong-Hai CHU Wei-Jie

(School of Software and Microelectronics, Peking University, Beijing 100871)

**Abstract** Using context information to composite business services into context-aware based service is an efficient path to improve service level. Existing researches pay more attention to the description of context information and context-aware based services, and lack full consideration of structural features and operational mechanisms. On one hand, context information and business services are repeated in a variety of context-aware based service, a pre-defined tight coupling process can't reuse them. On the other hand, context information has the randomness and dynamic values, and triggered mechanism based on context reasoning is a little more costly. Based on the analysis of structural features, a new system framework Context-aware Based Smart Service System (CBS<sup>3</sup>) is proposed. The configuration model and system operational mechanisms are discussed. A case is prepared to explain how to build a context-aware based service based on CBS<sup>3</sup>. Compared with traditional methods, CBS<sup>3</sup> can improve the reusability and efficiency of context-aware based service.

**Keywords** context information; context aware; context-aware based service; configuration model

## 1 引 言

随着信息通信技术与传感器技术的迅猛发展和

广泛深入的应用, 地球上的万事万物逐渐变得可感知和互联互通<sup>[1]</sup>. 互联网和物联网将现实世界和信息世界充分覆盖与融合, 为信息采集、传递和服务决策提供了强有力的技术支撑. 情境感知服务是在这

收稿日期: 2010-06-08; 最终修改稿收到日期: 2010-08-27. 本课题得到丹麦战略研究委员会合作项目 GENIES(2106-08-0046)、国家“八六三”高技术研究发展计划项目基金(2009AA04Z120)和国家自然科学基金(61033005)资助. 莫 同, 男, 1981 年生, 博士, 主要研究方向为服务工程、情境感知、情境感知服务. E-mail: motong@ss.pku.edu.cn. 李伟平, 男, 1973 年生, 博士, 副教授, 主要研究方向为软件工程、服务工程、服务计算等. 吴中海, 男, 1968 年生, 博士, 教授, 博士生导师, 主要研究领域为软件工程、数字媒体. 褚伟杰, 男, 1980 年生, 讲师, 主要研究方向为移动计算、RFID 技术.

些技术背景下产生的一种能够极大改善人们生活的新服务方式<sup>[2]</sup>。

### 1.1 情境感知服务

情境感知服务是指通过传感器采集/感知被服务对象的情境信息,根据情境信息分析判断被服务对象当前的状况,然后选择并提供适当的业务服务<sup>[3]</sup>。其目标是根据被服务对象的情境信息,自动判断当前所需的服务并予以提供。情境感知服务具有以下 4 个主要特点:

(1) 自动化。情境感知服务由传感器自动采集的数据触发而并非人,因此除了能够节省人们的时间精力外,更适用于夜间或危险环境等人们难以控制,或者睡眠、遗忘与丧失意识等无法进行自主行为以及公共区域等缺乏明确责任的情形。

(2) 全天候/方位。情境感知服务不受人工工作的作息时间限制,通过传感器可以实现对被服务对象进行 7×24h 全天候监控。随着传感器功能的进一步完善,可以实现对生产生活中的方方面面进行数据采集。

(3) 即时/前瞻性。情境感知服务的延迟仅受数据采集频率、传输速度和服务响应时间影响,在正常情况下,这种延迟几乎可以忽略不计而被视为即时服务。此外,情境感知服务也可对顾客的历史数据进行分析处理,及时发现数据趋势,通过前瞻性服务做到未雨绸缪或防患于未然。

(4) 个性定制。由于情境感知服务是基于服务对象的情境数据,因此这种服务是真正的个性化服务。此外,数据的采集配置、服务策略的制订也可根据顾客的需求进行定制。

已有文献主要通过两种方式构造情境感知服务。一种是自上而下,将情境信息与业务服务通过业务流程紧耦合绑定,然后部署执行<sup>[4]</sup>。这种方式由于业务服务的调用流程已预定义,调用控制代价较小。但是,顾客的情境信息在不同情境感知服务中无法共享。此外,结构中调用过程描述会随着情境信息种类和取值的增加变得十分复杂,这种方式仅适用于情境信息种类较少,取值相对简单的情况。

另一种是自下而上,将情境信息与业务服务分别定义,通过规则或配置进行关联,根据情境信息取值触发调用相应的业务服务<sup>[5]</sup>。这种方式的关联定义较为灵活,且支持情境信息重用,但是在控制实现上较前者更为复杂。目前主流的基于本体推理的控制机制代价较大,难以支撑多服务共同运行。

目前有大量的业务服务和情境信息的提供商,所欠缺的是一种简便快捷的方式能够将二者结合形

成情境感知服务向顾客提供。一种可行的思路是将上述两种方式相结合,将情境信息与业务服务独立定义在平台上,通过建立松耦合关联,并将调用控制预定义在配置中,然后根据监控的情境信息取值选择相应的配置调用业务服务。这种方式既能保留关联定义的灵活性,保证情境信息的重用,也能够通过预定义的配置为降低调用控制代价提供可能。

### 1.2 情境感知系统

在国际上,情境感知被称为 Context-aware, context 在国内通常被翻译为“上下文”,上下文感知计算是对设备、上下文、物理环境等构成的计算环境进行管理、协调和调度,建立实体对象间互操作的基础,为应用开发提供统一的框架和应用程序接口<sup>[6]</sup>。上下文更多用于描述对象的计算环境,而本文所关注的 context 是描述对象的自然环境以及其它表征对象的信息,如时间、地点等,进而被用于判断对象所需的服务。这类 context 我们称之为情境<sup>[3]</sup>。

目前情境感知系统框架的研究主要面向普适计算领域。浙江大学的吴朝晖教授对普适计算的介绍中,提出基于情境感知系统框架主要的研究问题包括情境信息的获取、情境信息的建模、情境信息的管理和上下文信息的推理<sup>[7]</sup>。南京大学的吕建教授提出可以通过一个分布式的、面向服务的软件基础架构来支持情境感知系统的开发和部署<sup>[8]</sup>。常见的情境感知架构有 Gaia、ACAI、CAPpella、CAPNET、SOCAM、CASM、CASA、ACF 和 Jena 等<sup>[6]</sup>。这些情境感知框架注重对情境信息的描述和处理,主要关注情境模型的建立和推理。

情境模型用于定义和表示情境,其发展过程分为面向理解、面向交互,面向推理和面向本体 4 个阶段。面向理解的模型是通过符号、图形以及数值等方式直观地描述情境内容,如 ContextToolkit 中的键值对模型<sup>[9]</sup>,基于图的情境建模语言 CML<sup>[10]</sup>等。这种方式人为因素很强,缺乏统一的格式支持情境交互,因此后来出现了利用标记语言描述的交互模型,如 ConteXtML<sup>[11]</sup>和 ContextUML<sup>[12]</sup>。简单、直观的情境信息可以通过终端设备或传感器直接获取,而一些基于这些可检测情境的、抽象的复杂情境则需要综合已知的情境进行推理判断。为此,McCarthy 等人将情境定义成一系列的事件、公式和规则<sup>[13]</sup>,通过规则逻辑支持对情境的推理,此外,也有学者从有限状态机<sup>[14]</sup>或 Petri 网<sup>[15]</sup>等角度定义情境。本体具有表达能力强,支持逻辑推理,易于理解以及便于知识重用和共享等优点,适合描述和定义情境以及情境之间的关系,也便于计算机进行推理<sup>[16]</sup>。综合

来看,目前基于本体来定义和建立服务情境模型是目前情境模型发展的主要方向和趋势。

在获得了第一手的情境数据后,通常需要对情境数据进行进一步的处理,然后才能得出顾客当前所处的完整情境,这一过程通过情境推理来实现。目前,根据情境模型的不同,主要的推理方法分为基于规则的方法和基于本体的方法两类。

基于规则的推理方法的实现有两种方式,一种是将逻辑规则写入程序代码中,通过程序自动实现推理;另一种是通过逻辑规则和推理引擎共同实现推理。由于前一种方法简单易于实现,但仅适用于规则较少的情况,在代码中固化逻辑规则难以维护和变更。规则与推理的分离降低了推理实现过程的耦合度,通过规则的定义和修改可以方便地扩展规则来应对不同的情况,因此现在基于规则的推理普遍采用这种方式<sup>[17]</sup>。

基于本体的推理方法通过本体描述语言将顾客的情境、情境之间的关系和推理规则定义成本体、本体的属性和约束关联关系,通过本体的推理来实现对情境的推理<sup>[16]</sup>。由于基于本体的情境模型在情境表述上的优势和本体推理工具的日益完善,利用本体推理来实现情境推理是一条简介的途径,而在本体推理中也可以融入基于规则的推理思想,即抽象出若干可修改和扩展的本体推理规则,建立一种混合的推理机制。

现有的情境感知框架注重对情境信息本身进行管理,如情境的表示、过滤、推理、存储和查询等。其目的是建立一个统一的、开放的环境模型以及环境信息的共享、使用和管理机制。在建立基于情境感知的情境感知服务时,除了要采集服务对象的情境信息之外,还需要建立不同情境信息与业务服务的关联关系,说明在何种情况下,用户需要哪种服务,然后根据感知的情境信息值判断当前服务对象的情境状态,提供相应的服务。

基于此,本文提出一种情境感知服务集成框架,该框架通过配置模型定义建立情境信息与业务服务的松耦合关联关系,利用值到值的映射和配置模型查询进行业务服务调用控制,进而实现情境感知服务。

本文工作的特点在于:(1)总结情境感知服务的松耦合和触发式调用特征,分析了现有方式的优点和不足;(2)提出通过配置模型方式直观表示情境信息与业务服务的松耦合关系,易于不同情境感知服务之间的情境信息与业务服务的重用;(3)建立轮询-调用的触发式调用控制机制,将业务服务调用控制简化为情境信息值到值的映射与配置模型查

询,降低情境感知服务执行代价。

## 2 情境感知服务特征

为提高情境感知服务的重用性,降低执行代价,需从结构关系和运行机制两方面分析情境感知服务特征,根据这些特征建立适应的情境感知服务集成框架。

### 2.1 松耦合

情境感知服务中的情境信息与业务服务的提供通常由不同的提供商提供,顾客在购买情境感知服务时需要分别购买。虽然通常顾客可以从情境感知服务提供商处直接购买,但情境感知服务提供商仅起到代理销售的作用,二者之间不存在直接关联关系。这体现了情境信息与业务服务之间的松耦合关系。

但在文献[4]中的情境感知服务需要将二者绑定销售,而这种绑定提高了顾客的服务成本。当顾客购买两个使用相同情境信息或业务服务的情境感知服务时,通常需要重复购买。造成这种情况的原因是情境感知服务提供商将二者原本松耦合的关系变成紧耦合绑定销售,使情境感知服务之间无法重用已有的情境信息或业务服务。

文献[18]给出了情境感知服务的结构,这也是当前情境感知服务所采用的通用结构。该结构中,每个情境感知服务包含一个情境信息集合。情境信息及关联关系通过本体进行定义描述。情境信息的取值可来自于传感器或根据其它情境信息取值与关联关系通过推理获得。每个情境感知服务还包含业务服务集合。业务服务的调用与表示服务对象状态的情境信息相关联,这类信息通常通过推理得到。

这种结构虽然能够体现情境信息与业务服务的松耦合关系,但由于是推理之后再调用,可重用的部分需要通过分析推理规则得到。情境信息与业务服务的关系缺乏直观表示。

在文献[18]的结构基础上,本文通过情境信息与业务服务的配置模型表示二者的关联关系。使用相同情境信息或业务服务的情境感知服务可以通过复制和简单修改配置模型的方式进行定义,进而体现松耦合特征在情境信息和业务服务重用中的作用。

### 2.2 触发式服务调用

情境感知服务运行时,传感器实时采集情境信息,控制单元根据采集值调用相应的业务服务向顾客提供。由于情境信息是实时采集,因此很可能多次的采集具有相同取值。业务服务通常只在情境信息首次满足条件时提供,即情境信息值变化时触发调用。

由于顾客的行为是动态随机的,情境信息的变化以及变化的时间难以预计.一种执行机制是将情境感知服务固化成一个循环流程,通过分支穷举各种情境信息情况,通过循环不断判断所处分支并调用相应业务服务<sup>[4]</sup>.这种方式虽然一定程度上弱化了情境信息变化的随机性,利于控制,但其本质仍是触发式调用.

文献[18]通过轮询-调用的触发式调用机制进行业务服务调用.其调用控制逻辑通过基于本体的情境信息推理实现,受推理引擎的效率限制,该结构难以支持大量情境信息和业务服务的情况.

本文沿用文献[18]中的轮询-调用方式,将部分本体推理转变为配置模型查询,以期降低调用控制代价,进而支持多情境感知服务.

### 3 情境感知服务系统框架

根据情境感知服务的松耦合和触发式调用特征,提出一种情境感知服务框架 CBS<sup>3</sup> (Context-aware Based Smart Service System).

#### 3.1 系统框架

CBS<sup>3</sup>系统框架如图1所示.

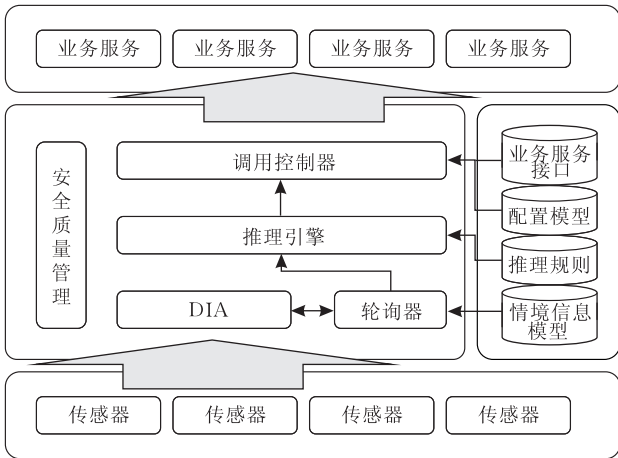


图1 CBS<sup>3</sup>框架

CBS<sup>3</sup>框架包括传感器层、情境感知引擎、存储库和业务服务层4个部分.

(1) 传感器层. 包含各种直接提供情境信息的单元,情境信息的来源可能是硬件设备,软件服务,也可能是人.

(2) 情境感知引擎. 情境感知引擎是CBS<sup>3</sup>框架的核心,包括DIA、轮询器、推理引擎、调用控制器和安全质量管理.

(3) 存储库. 存储库保存情境感知引擎所使用的情境信息模型、推理规则、配置模型和业务服务接口.

(4) 业务服务层. 包含向顾客提供的各种业务服务,业务服务可以是Web service,也可以是现实生活中的实体服务,如报警、医疗服务等.

#### 3.2 情境信息采集

由于传感器层的情境信息来源差异很大,为便于情境感知引擎使用,CBS<sup>3</sup>框架通过设备信息访问接口DIA(Devices Information Access)屏蔽下层访问环境.DIA向上可以被看作是一个统一的情境数据访问接口,向下需要采用不同的方式适配各种情境信息来源.

轮询器根据情境信息的采集频率调用DIA接口获取情境信息的取值,然后将值传递给情境推理引擎.DIA和轮询器的作用是为情境感知引擎获取情境信息取值,具体实现详见文献[3].

#### 3.3 情境推理

有的情境信息无法通过传感器层直接获取值,而是需要根据与其它情境信息的关联关系进行值判定.这种判定在CBS<sup>3</sup>框架中通过推理引擎实现.

**定义1.** 情境推理是情境信息  $cxt$  的值  $v_{cxt}$  和与之关联的情境信息集合  $CXT = \{cxt_1, cxt_2, \dots, cxt_n\}$  的值集合  $V_{CXT} = \{v_{cxt_1}, v_{cxt_2}, \dots, v_{cxt_n}\}$  的映射  $\varphi$ ,  $v_{cxt} = \varphi(V_{CXT})$ .

映射  $\varphi$  通过推理规则表示,CBS<sup>3</sup>框架中的推理规则由本体表示的条件组成,包括取值条件  $cv$  和关系条件  $cr$ .

**定义2.** 取值条件  $cv = \langle obj, cxt, v_{cxt} \rangle$ , 关系条件  $cr = \langle obj, relation, obj \rangle$ , 其中

- (1)  $obj$  是对象;
- (2)  $cxt$  是情境信息;
- (3)  $v_{cxt}$  是情境信息的值;
- (4)  $relation$  是对象之间的关系.

$obj, cxt$  和  $relation$  均为本体或本体属性;  $v_{cxt}$  可能为本体,也可能为字符串、整数等数值类型.

**定义3.** 映射  $\varphi$  的推理规则  $l\varphi = \langle A, S \rangle$ , 其中

- (1)  $A$  是前置条件集;
- (2)  $S$  是后置条件集;
- (3)  $A$  与  $S$  满足如下约束:

- ① 对任意  $A$ , 如果有  $A \neq \emptyset$ , 则必存在  $cv \in A$ ;
- ② 对任意  $S$ , 有  $S \neq \emptyset$ , 且必不存在  $cr \in S$ .

前置条件集  $A$  可能由取值条件和关系条件共同组成,但必包含取值条件;后置条件集  $S$  仅由取值条件组成.

以文献[5]中的推理为例,如果人在卧室,卧室房间的灯是关的,时间为夜晚,那么推导出人在睡觉.该例包含人、房间、卧室、灯、时间和位置6个本

体. 其中人、房间和灯是 *obj*, 卧室是房间的子概念; 时间、位置和睡眠状态是人的 *cxt*, 开关状态是灯的 *cxt*; 灯和卧室之间具有 *relation* 属性 *locate-in*. 上述内容可用如下推理规则描述:

$$l\varphi = \langle A, S \rangle$$

$$A = \{ \langle \text{人}, \text{位置}, \text{卧室} \rangle, \langle \text{灯}, \text{locate-in}, \text{卧室} \rangle, \langle \text{灯}, \text{开关状态}, \text{关} \rangle, \langle \text{人}, \text{时间}, \text{夜晚} \rangle \}$$

$$S = \{ \langle \text{人}, \text{睡眠状态}, \text{睡觉} \rangle \}$$

在情境推理中, 有一类特殊的映射  $d \in \emptyset$ , 映射  $d$  两端的 *cxt* 相同, 这样的映射称之为情境划分. 情境划分的意义在于将连续型情境信息离散化. 例如将房间温度离散化成热 ( $>26^\circ$ ), 适中 ( $18^\circ \leq \text{且} \leq 26^\circ$ ), 冷 ( $<18^\circ$ ). 在睡眠状态推理中, 情境信息时间也采用了情境划分值(夜晚).

### 3.4 业务服务调用

情境信息经过采集和推理后被送到调用控制器进行分析判断是否需要调用以及具体调用哪个业务服务向哪位顾客提供.

调用控制器以配置模型、采集信息和缓存为输入, 输出为业务服务.

**定义 4.** 配置模型是一个五元组  $conf = \langle confID, Cus, Cxts, CXT-V, Bs \rangle$ , 其中

- (1) *confID* 是 *conf* 的标识信息;
- (2) *Cus* 是顾客;
- (3) *Cxts* 是情境感知服务;
- (4)  $CXT-V = \{ \langle cxt_1, v_{1x} \rangle, \langle cxt_2, v_{2y} \rangle, \dots, \langle cxt_n, v_{nz} \rangle \}$  是情境信息-值对的集合, 其中  $cxt_1, cxt_2, \dots, cxt_n$  是 *Cxts* 的情境信息,  $v_{ij}$  是  $cxt_j$  的值;
- (5) *Bs* 是业务服务.

**定义 5.** 采集信息是一个五元组  $seni = \langle seniID, sen, Cus, Cxt, val \rangle$ , 其中

- (1) *seniID* 是 *seni* 的标识信息;
- (2) *sen* 是传感器, 推理得到的情境信息此处为推理引擎;
- (3) *Cus* 是顾客;
- (4) *Cxt* 说明 *sen* 采集的是哪种情境信息;
- (5) *val* 是采集值.

缓存信息是上一采集周期的采集信息.

调用控制算法见算法 1.

#### 算法 1. 调用控制算法.

输入: 配置模型集合 *CONF*, 采集信息集合 *SENI*, 缓存信息 *SENI\_T*

输出: 业务服务集合 *BS*

主要步骤:

1.  $SENI' = \emptyset; CHANGE = \emptyset; BS = \emptyset$  // 初始化
2. for (all  $seni \in SENI$ ) {
  - 2.1. select  $seni\_t$  from *SENI\_T* where  $seni\_t.sen = seni.sen$  and  $seni\_t.cus = seni.cus$  and  $seni\_t.cxt = seni.cxt$ ;
  - 2.2. if  $seni\_t = \emptyset$  then insert *seni* into *SENI\_T*;
  - else if  $seni\_t.val \neq seni.val$  then  $seni\_t.val = seni.val$ ;
  - insert *seni* into *SENI'* ;
 // 查找所有发生变化的采集值并更新缓存
3. for (all  $seni' \in SENI'$ ) { // 有变化时触发控制调用
  - 3.1. select  $CHANGE' = (cus, cxts)$  from *CONF* where  $conf.cus = seni'.cus$  and  $(seni'.cxt, seni'.val) \in conf.CXT-V$ ;
  - 3.2.  $CHANGE = CHANGE \cup CHANGE'$  ;
 // 查找受采集值变化影响的顾客及其情境感知服务
4. for (all  $change \in CHANGE$ ) {
  - 4.1. select *CXT* from *CONF*  $change.cus = conf.cus$  and  $change.cxts = conf.cxts$ ;
  - // 查找该顾客的情境感知服务的所有情境信息
  - 4.2. select  $CXT-V' = (cxt, val)$  from *SENI\_T* where  $seni\_t.cus = change.cus$  and  $CXT-V' = CXT$  and  $seni\_t.cxt = cxt$ ;
  - // 查找这些情境信息的最新采集值
  - 4.3. select *bs* from *CONF* where  $change.cus = conf.cus$  and  $change.cxts = conf.cxts$  and  $conf.CXT-V = CXT-V'$ ;
  - 4.4. insert *bs* into *BS* ;
  - // 查找所需调用的业务服务
5. return (*BS*).

### 3.5 安全质量管理

情境信息涉及个人隐私, 因此情境感知服务必须对情境信息的使用进行安全控制. 原则上, 每个顾客能且只能查询到自己的情境信息. 此外, 也需控制各提供商们对情境信息、业务服务和情境感知服务的查询与管理权限. CBS<sup>3</sup> 框架的安全控制关系如图 2 所示.

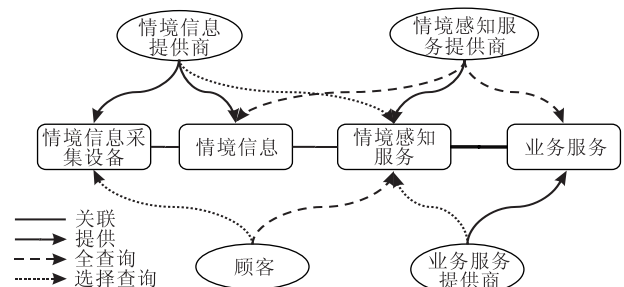


图 2 CBS<sup>3</sup> 框架安全控制关系

如图 2 所示, 在 CBS<sup>3</sup> 框架中, 主要被控制的对象包括情境感知服务、业务服务、情境信息和情境信

息采集设备. 其中情境感知服务需要情境信息和业务服务, 而情境信息一方面是建立情境感知服务的基础, 一方面依赖情境信息采集设备提供数据.

CBS<sup>3</sup>框架的安全控制关系分为提供、全查询和选择查询 3 种. 提供关系说明对象由相应的角色提供, 该角色对其所提供的对象具有增删改查等一切权利. 全查询关系说明该角色可以查询对象中的所有个体. 选择查询说明该角色只能够查询对象中与其相关的个体.

情境感知服务质量  $Q_s$  由情境感知质量  $Q_{CXT}$ 、业务服务质量  $Q_{BS}$  和平台系统质量  $Q_{pf}$  共同组成.

$Q_{CXT}$  由情境信息提供商所提供的情境感知数据获取方法的及时性、准确性、获取方式的配置程度以及续航能力所决定.  $Q_{CXT}$  主要针对硬件采集型情境感知质量, 读取型和推理型情境由软件系统自动感知, 通常质量要远高于硬件采集型情境感知, 质量影响可以忽略不计. 硬件采集型情境的感知受硬件设备的信号传输和处理速度影响, 情境信息的获取无法作到完全的实时采集, 采集频率越快则情境数据越接近真实值. 硬件设备的工作环境为自然环境, 信息采集和传输会受到噪声影响, 因此需要一定的过滤降噪处理以减少误报次数, 提高数据准确性. 高级的感知硬件通常会提供多种可配置的数据采集方式以满足不同情况下的情境信息获取需要, 可配置程度越高则说明硬件的适应能力越强. 通常, 情境信息采集设备采用电池供电, 受体积、重量等限制, 携带的电池容量极为有限, 低能耗高续航的设备更加环保、便捷和经济.

业务服务质量评价与业务服务所属行业领域相关. 各个行业领域均有各自的质量评价方法.

情境感知质量和业务服务质量决定了情境感知服务本身的质量, 而平台系统质量则影响情境感知服务的建立和使用速度, 主要包括信息处理速度、存储容量、网络带宽等平台系统硬件环境指标.

目前, 业务服务质量和硬件环境质量的相关研究已经比较成熟, 二者的评价可以参照相关方式进行, 本文不再进行叙述.

## 4 应用案例

本节以一个情境感知服务 iCARE 为例, 说明如何利用 CBS<sup>3</sup> 框架将情境信息与业务服务相结合建立情境感知服务.

### 4.1 业务背景

许多老人患有老年痴呆或其它疾病造成自主意识丧失, 导致频繁出现走失等事件. 其子女由于工作等原因无法一直陪护在老人身边, 但需要解决方案帮助其避免老人走失.

随着各种定位技术日趋成熟, A 公司的手机蓝牙定位能够十分经济有效地对手机携带者进行室内定位. B 公司希望借手机定位建立老人看护服务 iCARE.

iCARE 的主要业务是通过在顾客的居住社区内布置的蓝牙节点采集数据, 判断老人是否走出预订区域. 顾客将手机放在老人身上作为定位传感器, 并设置老人的正常活动区域, 例如家. 老人走出正常区域时会被非正常区域的节点探测到, 比如楼过道或者小区出口, iCARE 向子女或社区保安报警, 以避免老人丢失. 当老人正常外出时, 子女可暂停该服务防止误报警. iCARE 服务的业务过程如图 3 所示.

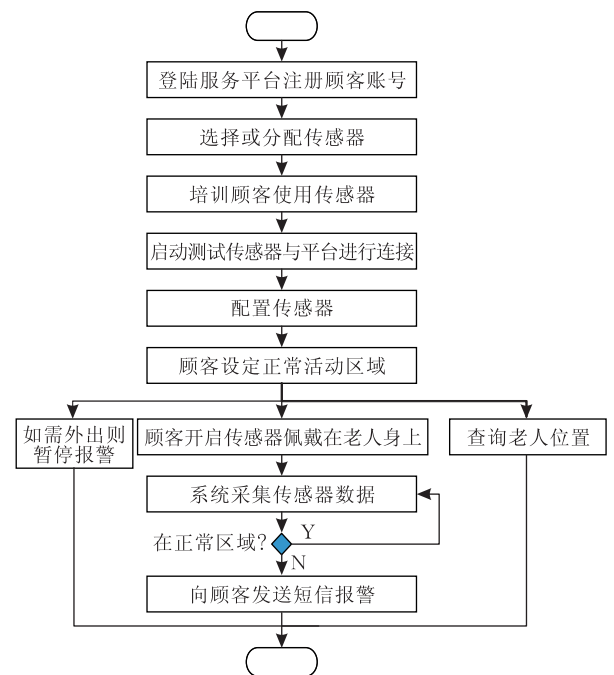


图 3 iCARE 服务业务过程

### 4.2 iCARE 建立

A 公司作为情境信息提供商注册登录 CBS<sup>3</sup>, 在情境分类“定位-蓝牙定位”下增加情境信息“A-手机蓝牙定位”. C 公司是一家短信服务提供商, 在 CBS<sup>3</sup> 上作为业务服务提供商注册并声明一个通过 webservice 发送短信报警的服务“f-SM”.

B 公司在 CBS<sup>3</sup> 上作为情境感知服务提供商注册并建立一个情境感知服务“iCARE”, 并为 iCARE 选择情境信息 A-手机蓝牙定位. 根据情境感知服

务逻辑的需要, B 公司又为 iCARE 增加两个情境信息: O-state, P-state, 其中 O-state 表示老人丢失状态, P-state 是服务暂停状态, O-state 的值根据 A-手机蓝牙定位的值推理得到, 推理规则定义如下:

老人, 位置是 *obj*; 位于、状态是老人的 *cxt*.

$A = \{\langle \text{老人}, \text{位于}, \text{位置} \rangle\}$ ,

$S = \{\langle \text{老人}, \text{状态}, \text{丢失/正常} \rangle\}$ .

#### 4.3 iCARE 使用

D 社区业主 E、F 购买 iCARE 服务帮助照看家中的老年痴呆患者. B 公司在 E、F 家中和 D 社区布置 8 个 A 公司的蓝牙感知节点, 其中节点 1 和节点 2 分别在住户 E、F 家中, 其它 6 个节点分布在社区其它位置.

顾客 E、F 在 A 公司处通过提供验证密码和相关凭证的方式, 证明他们分别是 phone1 和 phone2 的合法用户, A 公司在 A-手机蓝牙定位下为其增加两个传感器 ID, 并提供给他们相应的密码.

顾客 E、F 分别注册登陆 CBS<sup>3</sup>, 选择 iCARE 服务, 并通过验证密码的方式声明他们是情境信息 A-手机蓝牙定位传感器 phone1 和 phone2 的合法用户. 在定义说明正常位置以及报警短信发送手机号后, E、F 将 phone1 和 phone2 分别佩戴在老人身上, 开始由 iCARE 服务协助看护老人. CBS<sup>3</sup> 相关配置模型见表 1.

表 1 配置模型

| <i>confID</i> | <i>Cus</i> | <i>Cxts</i> | <i>CXT-V</i>   | <i>Bs</i> |
|---------------|------------|-------------|--|-----------|
| 00001         | E          | iCARE       | $\langle \text{O-state}, \text{丢失} \rangle$<br>$\langle \text{P-state}, \text{正常} \rangle$ | f-SM      |
| 00001         | F          | iCARE       | $\langle \text{O-state}, \text{丢失} \rangle$<br>$\langle \text{P-state}, \text{正常} \rangle$ | f-SM      |

B 公司为推理本体和规则添加实例, 主要本体和规则实例见表 2.

表 2 主要推理本体和规则的实例

| 本体/规则 | 实例   |
|-------|--|
| 老人    | phone1, phone2   |
| 位置    | loc1, loc2, ..., loc8  |
| 规则    | $(A = \{\langle \text{phone1}, \text{位于}, \text{loc1} \rangle\}, S = \{\langle \text{老人}, \text{状态}, \text{正常} \rangle\})$ ,<br>$(A = \{\langle \text{phone2}, \text{位于}, \text{loc2} \rangle\}, S = \{\langle \text{老人}, \text{状态}, \text{正常} \rangle\})$ ,<br>$(A = \{\langle \text{phone1}, \text{位于}, \text{loc3} \rangle\}, S = \{\langle \text{老人}, \text{状态}, \text{丢失} \rangle\})$ ,<br>$(A = \{\langle \text{phone2}, \text{位于}, \text{loc3} \rangle\}, S = \{\langle \text{老人}, \text{状态}, \text{丢失} \rangle\})$ ... |

B 公司在 CBS<sup>3</sup> 上启动 iCARE 服务后, CBS<sup>3</sup> 根据配置的采集周期采集 phone1 和 phone2 的数据.

当顾客 E 正常启用 iCARE 服务时, P-state 值为正常, 如果老人 (phone1) 进入楼道, 楼道对应节点为 loc3, 经过推理 O-state 值为丢失. 调用控制器从推理引擎获得的相关采集信息见表 3.

表 3 采集信息

| <i>seniID</i> | <i>sen</i> | <i>Cus</i> | <i>Cxt</i> | <i>val</i> |
|---------------|------------|------------|------------|------------|
| 00001         | phone1     | E          | A-手机蓝牙定位   | loc3       |
| 00002         | 推理引擎       | E          | O-state    | 丢失         |
| 00003         | Db1        | E          | P-state    | 正常         |

根据表 1 的配置模型, CBS<sup>3</sup> 调用业务服务 f-SM 向顾客 E 发送短信报警.

当老人正常外出时, 例如在家人的陪同下外出散步, 顾客 E 可以通过用户接口将 P-state 设置为暂停. 这样, 无论老人走到任何位置, 均不会出现误报警情况.

## 5 对比分析

本文选取文献[4]中基于 BPEL 的预定义控制执行和文献[18]中基于本体推理的框架两种较具代表性的情境感知服务实现方式与 CBS<sup>3</sup> 框架进行重用度和调用控制效率比较.

### 5.1 重用度

由于目前情境感知服务尚未普及, 缺乏同时能够通过上述 3 种方式实现, 又具有较多交叉重复内容的批量对比实验案例, 因此难以通过如被重用情境信息数与案例数比值这类可直接量化计算的重用度进行比较. 本文通过分析实现机制, 从情境信息在情境感知服务中的使用方式和情境信息与业务服务的耦合关系出发, 定性地比较 3 种方式在情境信息和服务建立两方面的重用度.

如表 4 所示, 文献[4]通过 BPEL 将情境信息和业务服务紧耦合成情境感知服务系统. 二者通过 BPEL 流程进行关联, 相似的服务可通过修改 BPEL 流程方式进行重用, 而不同系统间的情境信息难以重用. 文献[18]将情境信息融入本体推理规则之中, 通过规则与业务服务关联, 二者之间是一种不直观的准耦合关系. 不同情境感知服务中的规则可以使用同一

表 4 重用度对比

| 方面   | 重用度     |        |                     |
|------|---------|--------|---------------------|
|      | 文献[4]   | 文献[18] | CBS <sup>3</sup> 框架 |
| 使用方式 | BPEL 流程 | 本体推理规则 | 配置模型                |
| 耦合关系 | 紧耦合     | 准松耦合   | 松耦合                 |
| 情境信息 | 较难重用    | 较易重用   | 较易重用                |
| 服务建立 | 中       | 较难重用   | 中                   |

情境信息. 因规则受本体定义影响, 不同本体体会导致相同关联关系的规则不同, 故较难重用. CBS<sup>3</sup> 框架通过配置模型直观地表示情境信息与业务服务的松耦合关系. 情境信息可以在不同配置模型中使用, 相似的服务可通过修改配置模型方式进行重用.

## 5.2 调用控制效率

为便于实验比较, 本文以 iCARE 服务为例, 给出业务环境及各方法实现结果.

### (1) 业务规模模拟.

通过重复多个 iCARE 服务和添加多个顾客的方式模拟不同业务规模的运行环境. 设每个服务的顾客数相同, 业务规模 = 服务数 × 顾客数. 各服务节点个数为 8, 顾客的允许范围随机产生.

### (2) CBS<sup>3</sup> 框架实现.

推理规则和配置模型通过数据库保存, 配置模型通过 sql 语句进行操作, 数据库为 Mysql, 推理引擎为 Jena.

### (3) 文献[4]实现方法模拟.

以 Java 程序和本机服务代替 BPEL 流程和 webservice, 执行环境为 Eclipse.

### (4) 文献[18]实现方法模拟.

在表 2 所示的推理规则基础上, 增加表示情境信息值与业务服务关联的推理规则. 推理部分实现同 CBS<sup>3</sup> 框架.

3 种方法的运行环境为 CPU 双核 2.5GHz, 内存 3GB 的 PC 机, 操作系统为 WindowsXP. 不同业务规模下 3 种方法的执行代价如图 4 所示.

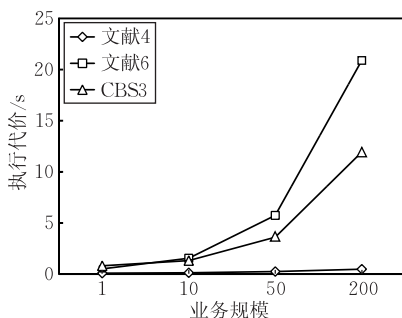


图 4 执行代价对比

图 4 表明, 3 种方法的执行代价都随业务规模的增加而增加. 文献[4]方法执行代价受业务规模增加的影响最小, 且远小于其它两种方法的执行代价; 其原因是执行代价为 Java 程序运行代价与服务调用代价之和. 文献[18]方法执行代价随业务规模的增加显著增长; 这是因为推理引擎的执行效率远低于普通 Java 程序执行效率, 业务规模的增加使得推理引擎所需处理的推理规则增加. CBS<sup>3</sup> 框架执行代

价随业务规模的增加速度位于文献[4]和文献[18]方法之间, 其原因是虽然 CBS<sup>3</sup> 框架也使用了推理引擎, 但将部分推理内容改由效率更高的数据库查询实现; 在业务规模很小的时候, 这种方式要比文献[18]额外多付出一些代价, 但在可接受范围之内; 随着业务规模的增加, 数据库查询与推理引擎的效率差异所带来的效率提高逐渐明显.

## 5.3 小结

综合上述两方面的对比分析结果, 文献[4]方法的执行效率最高, 随情境感知服务和顾客数量的增多影响最低, 但其紧耦合的实现方式导致最难进行情境信息和服务建立的重用. 文献[18]方法的执行效率最低, 基于本体推理规则的准松耦合实现方式易于情境信息的重用但不利于服务建立的重用. 本文提出的 CBS<sup>3</sup> 框架执行效率位于上述两种方法之间, 能够显式地表示情境信息与业务服务的松耦合关系, 利于情境信息和服务建立的重用.

## 6 结束语

情境感知服务以其自动化、全天候/方位、即时/前瞻性和个性定制等特点能够极大地改善人们的生活, 必将成为人们未来生活必不可少的部分. 为了快速、有效地支持情境感知服务的建立, 本文提出了一套情境感知服务系统框架 CBS<sup>3</sup>. CBS<sup>3</sup> 框架通过配置模型将情境信息与业务服务相结合, 支持情境感知服务的敏捷构建. CBS<sup>3</sup> 框架采用基于 Web 的松散耦合方式, 一方面有利于情境信息、业务服务提供商, 使他们的情境信息和业务服务能够被多个情境感知服务使用, 提高使用量和重用性; 另一方面便于情境感知服务提供商根据实际需要灵活的选择情境采集方式和业务服务, 通过简便的配置策略定义与查询实现情境感知服务的低代价实现.

下一步工作包括: 完善 CBS<sup>3</sup> 框架开发; 将情境感知服务建立过程中的相关问题进一步抽象和形式化, 如配置策略管理的数学模型表示及优化, 调用控制算法的优化; 将 iCARE 服务进行实际应用.

**致谢** 在此, 作者向哈尔滨工业大学的徐晓飞教授和中国科学院计算技术研究所的韩燕波教授对本文工作给予的启发、建议和指导表示感谢!

## 参 考 文 献

- [1] Thompson C W. Smart devices and soft controllers. Internet Computing, 2005, 9(1): 82-85

- [2] Kortuem G, Kawsar F, Fitton D, Sundramoorthy V. Smart objects as building blocks for the Internet of things. *Internet Computing*, 2010, 14(1): 44-51
- [3] Staunstrup J, Tong F, Yu L, Li W, Lin H, Chu W, Hansen J P, Glenstrup A, Hildebrandt T. Services in context. *Computer Systems Application*, 2009, 18(6): 161-167
- [4] Yamato Y, Nakano Y, Sunaga H. Study and evaluation of context-aware service composition and change-over using BPEL engine and semantic Web techniques//*Proceedings of the Consumer Communications and Networking Conference*. Las Vegas, Nevada, 2008: 863-867
- [5] Zualkernan I A, Al-Ali A R, Jabbar M A, Zabalawi I, Wasfy A. InfoPods; Zigbee-based remote information monitoring devices for smart-homes. *IEEE Transactions on Consumer Electronics*, 2009, 55(3): 1221-1226
- [6] Li R, Li R. A survey of context-aware computing and its system infrastructure. *Journal of Computer Research and Development*, 2007, 44(2): 269-276
- [7] Wu Z, Pan G. Ubiquitous computing//*China Computer Federation Proceedings (CCFP-0002)*. 2005: 175-187
- [8] Bu Y, Li J, Chen S, Han H, Tao X, Lv J. Research and development of context-aware computing//*Proceedings of the 1st Joint National Conference on Harmonic Human Machine Environment (HHME'05)*. Kunming, China, 2005: 556-563
- [9] Dey A K. Providing architectural support for building context-aware applications[Ph. D. dissertation]. Georgia Institute of Technology, Atlanta, GA, 2000: 1-119
- [10] Henricksen K. Developing context-aware pervasive computing applications: Models and approach. *Journal of Pervasive and Mobile Computing*, 2006, 2(1): 37-64
- [11] Ryan N. ConteXtML: Exchanging contextual information between a mobile client and the FieldNote server. <http://www.cs.kent.ac.uk/projects/mobicomp/fnc/ConteXtML.html>, 2000
- [12] Sheng Q Z, Benatallah B. ContextUML: A UML based modeling language for model driven development of context-aware Web services//*Proceedings of the International Conference on Mobile Business*, Sydney, Australia, 2005: 206-212
- [13] McCarthy J. Notes on formalizing contexts//*Proceedings of the 13th International Joint Conference on Artificial Intelligence*. Chambéry, France, 1993: 555-560
- [14] Wang Y. An FSM model for situation-aware mobile application software systems//*Proceedings of the ACM 42nd Southeast Conference (ACMSE'04)*. Huntsville, Alabama, 2004: 52-57
- [15] Kwon O B. Modeling and generating context-aware agent based applications with amended colored Petri nets. *Expert Systems with Applications*, 2004, 27(4): 609-621
- [16] Wang X, Zhang D, Gu T, Pung H. Ontology based context modeling and reasoning using OWL//*Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOM'04)*. Orlando, Florida, 2004: 18-22
- [17] Chen H, Finin T, Joshi A. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 2003, 18(3): 197-207
- [18] Gu T, Pung H K, Zhang D Q. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 2005, 28(1): 1-18



**MO Tong**, born in 1981, Ph. D. . His research interests include service engineering, service modeling, context-aware service, smart service.

**LI Wei-Ping**, born in 1973, Ph. D. , associate professor. His current research interests include software engineering, service engineering and service computing.

**WU Zhong-Hai**, born in 1968, Ph. D. , professor, Ph. D. supervisor. His research interests include software engineering and digital media.

**CHU Wei-Jie**, born in 1980, lecturer. His research interests include mobile computing and RFID.

## Background

Research work is supported by the Danish Strategic Research Council (grant No. 2106-08-0046) and the National High Technology Research and Development Program (863 Program) of China under grant No. 2009AA04Z120.

Using context-awareness to obtain information and giving proper services automatically is the future way of convenient life. Though there are some successful applications of context-aware based service such as Smart Home, it still lacks a loose coupling way to combine context-awareness and business service to build context-aware based service agilely.

Attention to the context limits the overall vision of context-aware based service system. The structural features and

operational mechanisms of are abstracted from the whole process of providing context-aware based service. Based on them, a framework of context-aware based service system CABS<sup>3</sup> is naturally presented. CABS<sup>3</sup> gives us a clear and concise engineering approach to establish a context-aware based service system. A sample application is also prepared to a perceptual cognition of how to use CABS<sup>3</sup> to build context-aware based service. And a comparative analysis between CABS<sup>3</sup> and other framework is discussed to show the CBS<sup>3</sup> can improve the efficiency and reusability of context-aware based service. This paper aims to provide an empirical framework for building context-aware based service systems.