

# 普适计算环境下基于中间件的上下文质量管理框架研究

郑 笛<sup>1</sup> 王 俊<sup>2</sup> 贲可荣<sup>1</sup>

(海军工程大学计算机工程系 武汉 430033)<sup>1</sup> (空军雷达学院重点实验室 武汉 430019)<sup>2</sup>

**摘 要** 随着信息技术的快速发展,分布式计算技术逐渐向普适计算技术演化,从而达到信息空间和物理空间融合的最终目标,为用户提供普适的智能化服务。为了达到这个目标,一个主要的困难就是如何有效地连续监测、捕获与解释环境相关的上下文信息来确保精确的上下文感知性。很多研究者已先后投身于上下文感知的普适应用的研究工作中,但大多数往往直接针对原始上下文进行处理,没有考虑上下文质量(QoC)的影响。因此,提出了一种基于中间件的上下文质量管理框架,即通过上下文的质量门阀管理、重复与不一致的上下文丢弃等不同层次的控制机制,为上下文感知服务和应用用户提供有效而可靠的上下文服务。

**关键词** 普适计算,中间件,上下文质量,上下文感知

**中图法分类号** TP311.5 **文献标识码** A

## Middleware-based Framework for the Quality Management of Context-aware Pervasive Applications

ZHENG Di<sup>1</sup> WANG Jun<sup>2</sup> BEN Ke-rong<sup>1</sup>

(Department of Computer Science, Naval University of Engineering, Wuhan 430033, China)<sup>1</sup>

(Key Research Lab, Wuhan Radar Institute, Wuhan 430019, China)<sup>2</sup>

**Abstract** With the rapid development of the information technology, it is inevitable that the distributed mobile computing will evolve to the pervasive computing gradually whose final goal is fusing the information space composed of computers with the physical space in which the people are working and living in. To achieve this goal, one of the problems is how to continuously monitor/capture and interpret the environment related information efficiently to assure high context awareness. Many attentions have been paid to the research of the context-aware pervasive applications. However, most of them just use the raw context directly or take just some aspects of the Quality of Context(QoC) into account. Therefore, we proposed a middleware based context-aware framework that supports QoC management in various layers. By this framework we can refine raw context, discard duplicate and inconsistent context so as to protect and provide QoS-enriched context information of users to context-aware applications and services.

**Keywords** Pervasive computing, Middleware, Quality of context, Context-aware

## 1 前言

随着无线网络、移动计算和传感器网络技术的发展,普适计算的设想正逐渐变成现实。普适计算力图使人们能够以一种方便的方式实现在任意地点与任意对方的联系,从而给传统的应用带来了新的挑战。“普适计算”将我们从“以设备为中心”的计算模式带入到“以人为中心”的计算模式,已成为当前计算模式发展的潮流,成为学术界和工业界研究的热点和焦点<sup>[1,2]</sup>。

在普适计算环境中,由于所面对的移动设备往往只具有有限的资源(如内存、电池和CPU),同时应用的执行上下文(如用户的位置、手持设备的可视面积)<sup>[3]</sup>也不断发生变化,因此要求对应的应用必须是上下文相关的,从而使应用能够根据上下文信息适配自身的行为。因此上下文相关技术成为普适计算环境中应用服务不可或缺的核心技术之一<sup>[4]</sup>。不同的

上下文来源所投递的数据通常是异构的,并且更新频率经常发生变化,同时要考虑准确性和异构的表达格式。这些统称为上下文的质量 QoC(Quality of Context),它是指用于描述信息质量的任何信息,而不是提供信息的过程和相应的硬件设备。如果不能反映上下文信息要建模的世界的真实状态,信息将不正确;如果包括矛盾信息,则将导致不一致性的发生;如果上下文的某些方面不可知,则导致信息的不完整。学术界已展开大量的关于上下文质量的研究工作,但往往关注的只是上下文质量的部分方面,没有从各个层面实现对上下文质量的综合管理。

同时,中间件已经被广泛地应用于分布式计算中。对于普适应用来说,通过中间件平台能够自动监测用户、应用和外部环境间所发生的交互以及这些交互所导致的上下文变化,并做出相应的反应,而不需要用户的干涉<sup>[5]</sup>。为了提供上下文相关的服务,支持普适计算的中间件平台必须能够识别上

到稿日期:2010-12-01 返修日期:2011-04-15 本文受海军工程大学自然科学基金项目资助。

郑 笛(1977-),女,博士,讲师,主要研究方向为普适计算、中间件,E-mail:dizheng1216@163.com;王 俊(1978-),男,博士,讲师,主要研究方向为无线计算、云计算。

下文的变化,同时通过聚合来提高上下文的质量或通过推理以产生更高级别的上下文,从而使应用能够采用适当的动作来适配这些变化。在已有的研究中,很多上下文感知的系统已经先后被建立起来以支持普适应用,并开始关注上下文的质量管理问题<sup>[5-11]</sup>。但这些系统只关注了上下文质量管理的一部分方面。因此,我们基于上下文感知中间件提出了一种全面的针对上下文的质量管理框架,通过该框架能够对上下文的质量进行有效的管理,从而能够更好地支持普适应用的需求。

## 2 基于中间件的上下文质量管理框架

### 2.1 上下文感知中间件体系结构

图1所示为我们前期工作中建立的基于上下文感知的中间件的体系结构<sup>[12-14]</sup>。核心构件管理模块为我们在前续工作所开发的遵循CORBA构件模型CCM的构件化中间件,为基于构件的应用,构件实例提供平台无关的服务,为执行平台资源提供了统一的平台无关访问。构件在容器中执行,为分布式应用的事务、安全、一致性和资源管理提供隐式的支持,从而使应用开发者能够更着重于应用本身的开发。同时可以提供有效的商业逻辑的重用,容器能够提供包括生命周期管理和构件发现在内的非功能性特性。

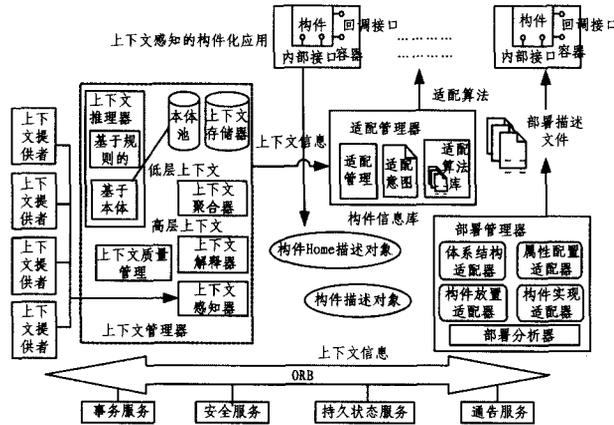


图1 用于构件化应用的上下文感知中间件结构

同时,中间件提供下面3个核心服务模块:

- 上下文管理器模块,对上下文进行监测和管理,以应对相关的变化。
- 适配管理器模块,对上下文的变化进行推理,并基于构件属性的体系结构描述以决定相应的构件适配行为。
- 配置器模块,对应用变量进行重配置,以实现上下文相关的构件适配。
- 自主管理器模块,通过一定的自主策略,实现对构件和服务的动态绑定与替换。

上下文管理器负责传感和捕捉上下文信息以及它们的变化,提供对上下文信息的访问,并将上下文信息的变化通知适配管理器模块。上下文管理器模块同时负责存储用户对应用服务的需求和偏好,它必须提供灵活的上下文监测。我们将上下文管理器开发为一个构件化的框架,从而可以插入新的上下文传感器模块。上下文管理器模块可以为上下文提供高级的推理操作。例如,它可以从基本的上下文元素中聚合出复杂的上下文元素或从上下文中派生出用户需求。上下文管理器模块同时保存上下文变化的历史信息。

适配管理器模块通过一定的适配规则负责对上下文的变化对应用所产生的冲击进行推理,并为当前的上下文计划和选择最适合的应用变量。作为推理的一部分,适配管理器模块需要在当前的上下文中访问这些变量的应用。适配管理器模块动态产生一个最适合上下文的应用变量的配置模板,通过这样的配置模板来表示一个所有应用变量均被解决的模型。

配置器模块负责协调应用的初始实例,并完成应用或设备重配置。配置器模块通过应用配置模板执行适配管理器决定的适配。适配管理器和配置器模块基于公共的信息元素——配置模板进行操作,因而是紧耦合的关系。

### 2.2 基于本体的上下文质量管理模型

基于本体的建模方式通过定义一个公共的词汇表以在一个具体的领域中进行上下文信息的共享,并且包括对领域中基本概念的可解释的定义和这些概念之间的关系。因此,借助于本体对上下文信息进行建模能够通过用户在用户、设备和服务之间共享上下文信息的结构,以实现语义的互操作性。

我们将上下文本体划分为上层本体和领域相关本体。上层本体是一个高层本体,捕获一般的上下文知识。每一个领域中的上下文共享使用一种一般化的上下文模型建模的公共的概念,和细节特性有巨大的差别。因此,应用领域的分离鼓励了一般概念的重用,并为定义应用相关的知识提供了一个灵活的接口,能够减少上下文处理的负担,并使得在移动客户端上解释上下文信息变得可能。领域相关的本体是一个定义了一般的概念细节和每个子领域的属性的低层本体的集合。当环境发生变化时,每一个子领域中的低层本体可以从高层本体被动态地插入和拔出。

在文献[15]中,我们详细分析了上下文处理的一般流程和上下文管理器的组成模块。要进行上下文质量管理,首先要从上下文建模的方式上进行增强。

如图2所示,在已有的基于本体的上下文的模型基础上,我们为每个上下文对象添加了6个参数,通过这些参数中的一个或多个构成上下文质量管理元组。

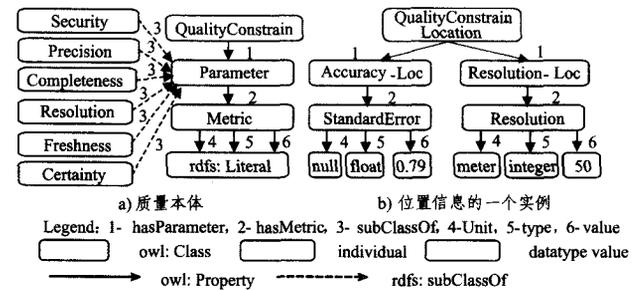


图2 基于本体的上下文质量管理模型

上下文的安全度(Security):上下文的安全性用来表示上下文安全地投递给消费者的概率。这个参数能够很好地用来反映上下文在传递过程中安全性保持的程度。

上下文的精度(Precision):定义为上下文信息描述一个现实世界形态的粒度。出于精数量化的目的,上下文信息值可以分为4种类型:布尔型、数值型、递增集、加权集。

上下文的完整性(Completeness):定义为相对收集到的所有上下文的可用上下文的比率。这个参数用来衡量用户定义的不同上下文之间的关系。

上下文的新鲜度(Freshness):是指上下文信息确定到被投递给请求者期间所消耗的时间。因为过时的上下文对请求

者来说是没有用的,同样上下文的新鲜度和信息的私有敏感度息息相关。上下文感知服务将评估应用的需求以及用户的个人爱好,以决定所采取的缓存策略。

上下文的空间分辨率(Resolution):定义为上下文信息的实例可用的物理区域的精度。空间分辨率是一个用于保护用户私人性的指标,因为更本地化的信息代表了更高的私人敏感度。例如,用户可能允许一个大楼安全系统访问大楼中的人数,但不允许访问这些人分别在什么房间中,从而避免系统推断用户是否在开会。

上下文的确定性(Certainty):定义为上下文信息的确定性。

### 2.3 基于质量管理的上下文处理流程

图3所示为考虑质量管理的上下文管理器的结构。通过阈值、重复上下文检测与消除、不一致上下文检测与消除3个层次来完成对上下文的质量管理。其中阈值管理通过对上下文质量管理参数设定一定的阈值来对获取的原始上下文进行初步的筛选,接下来在上下文的解释和聚合过程对重复上下文和不一致上下文进行连贯的处理。

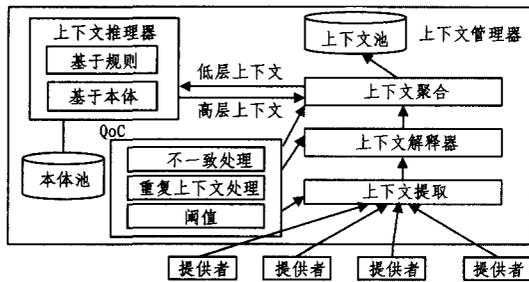


图3 用于质量管理的上下文管理器结构

图4给出了考虑质量管理的上下文处理过程。第一步为收集原始上下文,即在较短的固定时间内从不同的传感器来源中收集原始的上下文信息。第二步是解析不一致性,我们之所以选择在本步中解析不同原始上下文中存在的不一致性,是因为原始上下文中存在的不一致性将最终导致高层上下文中的不一致性,而这种高层的不一致性将是非常难以解决的。采用批处理的方式来处理原始的上下文,每一批原始上下文中的不一致性均在上下文推理之前清除完,从而使高层上下文的不一致性能够被限制在一个特定的程度。第三步是重构原始上下文,将使用原始上下文更新上下文池并对相关性进行检查。过时或不正确的高层上下文将在本步中被删除。如果它们没被删除,将在推理后导致严重的上下文不一致性。然后,我们运用基于规则的推理以产生高层的上下文,这些推理器均配置为可追踪,从而能够在上下文池中对相关的图进行更新。

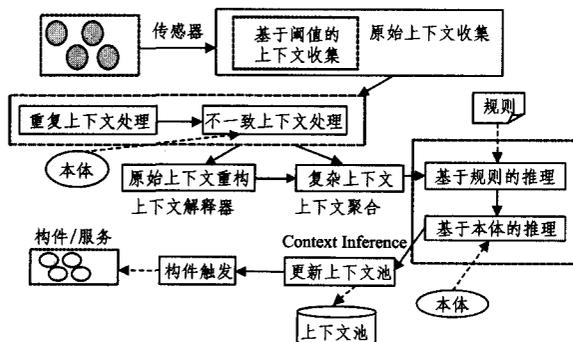


图4 基于质量管理的上下文的处理流程

### 2.4 重复上下文的检测与丢弃

当系统完成基于阈值的原始上下文筛选后,接下来需要考虑的就是重复上下文的清除。在上下文模型中,为每个上下文设置了上下文ID。因此所谓重复上下文,就是具有相同的上下文ID,或者具有相同的上下文名/值对。

算法1给出了重复上下文的检测方法。首先判断新到达的上下文的ID号和名/值对,是否与现有的上下文重复。如果有上下文具有相同的ID号和名/值对,则检查这些上下文的来源。如果它们具有相同的来源,则它们分别来自不同的时段,根据相应的质量参数进行丢弃即可。如果它们来源不同,则上下文的收集过程存在错误,需要采取一定的检查措施。

#### 算法1 重复上下文检测算法

INPUT:新到达的上下文

1. 获得上下文ID
2. 如果存在具有相同ID的上下文
3.     如果相应的上下文来源相同
4.     如果具有相同的时间戳
5.         找到重复的上下文,执行丢弃
6.     否则 检查上下文的收集
7. 如果找到具有相同名/值对的上下文
8.     按质量元组进行丢弃

### 2.5 不一致上下文检测与丢弃

对于所检测到的重复上下文,我们采取了前期工作中提出的如下的几种不一致丢弃算法<sup>[16]</sup>:

(1)全丢弃不一致性消除:对于检测出的冲突上下文,要通过删除相应的上下文来消除不一致性。所谓全丢弃不一致性消除是最为直观的不一致性消除方法,即将产生冲突的上下文实例不加区分地完全删除。

(2)最新上下文丢弃不一致性消除:相对于最新上下文丢弃不一致性消除算法来说,最新上下文丢弃不一致性消除并不丢弃所有发生冲突的上下文,因为很多冲突的发生是由于新进入的上下文实例和已有的上下文实例之间的冲突,甚至这部分已有的上下文实例可能即将加入上下文池进行存储。因此通过比较上下文实例的 *starttime* 域和 *updatetime* 域,能够判定上下文实例的新旧程度。

(3)基于确定性上下文丢弃:基于确定性上下文丢弃不一致性消除算法相对最新上下文丢弃算法具有更高的准确性,同时相对全丢弃不一致性消除算法能够为应用保留更多的上下文信息。但基于确定性的上下文丢弃算法需要对每个上下文实例进行确定性的判定和比较,从而会加大算法的额外开销,降低算法的效率。

(4)基于相关性上下文丢弃:对于不同的上下文来说,更频繁的原始上下文比不频繁的那些上下文具有更高的优先级,也可能是正确的上下文。然而,要为不同种类的上下文比较频率是非常困难的。基于这个原因,考虑这些上下文彼此之间的相关性,也就是它们被使用的频繁程度为每一个原始上下文进行度量,具有更大的相对频率值的上下文近来更为频繁,因此它们更可能是正确的上下文。当冲突发生时,丢弃具有较小的相对频率值的上下文。下式为计算原始上下文的相对频率的方法。

$$context_i.rf = \begin{cases} context_i.period \cdot context_i.freshness \\ currenttime - context_i.starttime \end{cases} \text{ (动态上下文)} \\ \infty \text{ (持久上下文)}$$

### 3 性能测试

如图 5 所示,我们比较和分析了全丢弃、最新上下文丢弃、基于确定性上下文丢弃和基于相关性上下文丢弃几种不一致性消除算法的作用。

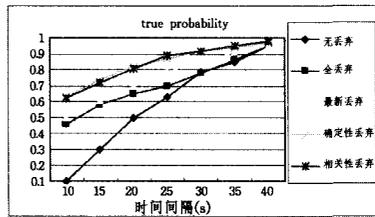


图 5 非一致性解析的正确性概率分析

实验中共使用 3 台计算机,一个 4G RAM,2 Xeon CPUs 的工作站和两个通过局域网互联的 PC 客户端。一个客户端负责充当原始上下文的提供者,而另一个则充当上下文的消费者。在实验中,为了简化对上下文信息正确性的分析,只使用了两种类型的传感器。要指出的是,在这个试验中尽管只使用了两种类型的传感器,我们的体系结构和算法也能够不做任何修改地适用于更多的传感器类型,因为它们针对语义上下文设计,而非物理传感器。当引入新的传感器类型时,唯一需要做的就是增加特定的原始上下文提供者。

从图 5 中可以看出,基于确定性和基于相关性的不一致性消除算法具有更好的性能。同时在工作站上对上述算法的性能进行了比较和分析,发现随着内存中总的上下文的增加,算法的效率将会相应地下降。

图 6 以基于相关性的丢弃算法为例,当缓冲区上下文实例为 1000~2000 时,使用的时间为 1.5s~2.0s。而在上下文实例为 3000~4000 时,需要 6s。同时,全丢弃算法和最新丢弃算法相对来说具有较低的不一致性处理时间,而基于确定性的丢弃算法由于需要对上下文实例的确定性进行判定,因此具有相对最高的处理时间。

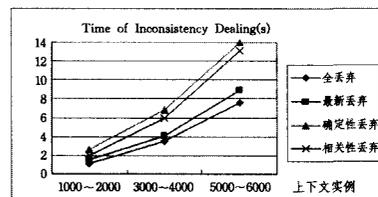


图 6 非一致性解析的性能分析

同时我们对几种不同算法的错误率进行了比较。为了进行比较,我们将 3 个应用分别运行了 600 次(总共 30 个大循环,每个循环 20 次),并记录了每一格循环的错误率。在试验中,所有被记录下来的错误是应用的不协调的行为,而系统的错误,如内存溢出等则不加以考虑。从试验结果中同样可以看出,随着正确性概率的提高,各上下文不一致性管理机制能够在不同程度上提高上下文感知应用的健壮性,大量由于上下文冲突所产生的不协调的行为被减少,从而提高了上下文感知应用的性能。

**结束语** 上下文感知系统通常利用来自于不同的物理传感器的感知上下文信息。而由于传感器的易错性,所产生的上下文信息的质量无法得到有效的保证。现有的上下文质量

方面的研究往往只关注上下文质量的某一方面。因此,在前期工作的基础上,本文提出了基于中间件的上下文质量管理框架,实现了对上下文质量的多层次的管理。实验证明,该方法能够有效提高上下文质量的管理效果。

### 参考文献

- [1] 徐光祐,史元春,谢伟凯. 普适计算[J]. 计算机学报,2003,26(9):1042-1050
- [2] 郑增威,吴朝晖. 普适计算综述[J]. 计算机科学,2003,30(4):18-29
- [3] 徐文拴,辛运祎,卢桂章. 普适计算中间件的研究与进展[J]. 计算机科学,2007,34(11):1-5
- [4] Dey A K. Understanding and using context [J]. Personal and Ubiquitous Computing,2001,5(1):4-7
- [5] Chen G,Kotz D. A survey of context-aware mobile computing research[R]. Hanover,NH,USA,2000
- [6] Buchholz T,Küpper A,Schiffers M. Quality of context:What it is and why we need it[C]//HPOVUA 2003. Geneva,2003
- [7] Kim Y,Lee K. A quality measurement method of context information in ubiquitous environments[C]//ICHIT'06. Washington,DC,USA;IEEE Computer Society,2006:576-581
- [8] Razzaque P N M, Dobson S. Categorization and modelling of quality in context information[C]//Proceedings of the IJCAI 2005. 2005
- [9] Preuveneers D,Berbers Y. Quality Extensions and Uncertainty Handling for Context Ontologies[C]//Proceedings of (C&O 2006). Shvaiko P,Euzenat J,L'eger A, et al. eds. Riva del Garda,Italy,August 2006:62-64
- [10] Sheikh K,Wegdam M,van Sinderen M. Quality-of-context and its use for protecting privacy in context aware systems[J]. JSW,2008,3(3):83-93
- [11] Manzoor A, Truong H L, Dustdar S. On the evaluation of quality of context[C]//Proceedings of Third European Conference(EuroSSC 2008). Zurich,Switzerland,October 2008:140-153
- [12] Zheng Di, Wang Jun, Jia Yan, et al. Deployment of Context-aware Component-based Applications based on Middleware[C]//Proceedings of The 4th International Conference on Ubiquitous Intelligence and Computing(UIC 2007). Hong Kong,China,July 2007:908-918
- [13] Zheng Di, Wang Jun, Jia Yan, et al. Middleware based Autonomic Management of the Context-aware Component-based Applications[C]//Proceedings of the Autonomic and Trusted Computing(ATC 2007). Hong Kong,China,July 2007:161-171
- [14] Zheng Di, Jia Yan, Zhou Peng, et al. Context-aware Middleware Support for Component based Applications in Pervasive Computing[C]//Proceedings of The 7th International Conference on Advanced Parallel Processing Technologies. Guangzhou, China, November 2007:161-171
- [15] 郑笛,贾焰,韩伟红,等. 普适计算环境下基于构件化中间件的上下文管理技术研究[J]. 计算机工程与科学,2008,30(5):106-112
- [16] 郑笛,朱珊. 普适计算环境下上下文不一致性的消除算法研究[J]. 计算机应用研究,2009,26(1)