

# 面向复杂物联网应用的上下文质量感知技术研究

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

(海军工程大学电子工程学院计算机工程系 武汉 430033)<sup>1</sup>

(空军预警学院兵器运用工程军队重点实验室 武汉 430019)<sup>2</sup>

**摘 要** 随着物联网应用的快速发展,传感信息日益多元化,传感器网络规模广域化,底层传感器网络构成异构化,传感信息数量大数据化,相应地,这也使得底层传感信息中所蕴含的不一致性、不完整性、不准确性等影响信息质量的因素大大增加。而传统的上下文感知技术没有充分考虑上下文质量对感知过程的影响,因此,在现有的上下文感知系统框架的基础上,充分研究不一致性、不完整性、不准确性等低质量传感器上下文的消除问题,通过上下文质量因子分类配置、不准确与不一致上下文丢弃、不完整上下文填充等方法实现了不同层次的控制机制,降低了信息的不确定性,从而有效提高了物联网应用的上下文处理质量。

**关键词** 物联网,上下文感知,上下文质量,传感器网络

**中图分类号** TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.033

## Research of Quality-aware Contexts Management for Complex IOT Applications

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

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

(Key Research Lab, Air force Early Warning Academy, Wuhan 430019, Chian)<sup>2</sup>

**Abstract** With the rapid development of IOT, it will lead to more multiple-dimensional and more numerous information with wide-area and heterogeneous sensor networks. Accordingly, the information from these kinds of networks may be more uncertain, incomplete, inconsistent and inaccurate. Therefore, we proposed a context management framework based on research of eliminating incomplete, inconsistent and inaccurate contexts. In this framework, we used quality factors configuration for contexts, regarding methods for inconsistent and inaccurate contexts as well as filling methods for incomplete contexts. Results show that the method can reduce the uncertainty of the contexts and improve the quality of the IOT applications efficiently.

**Keywords** IOT, Context-aware, Quality of context, Sensor networks

## 1 引言

随着无线网络、移动计算和传感器网络技术的发展,物联网技术获得了飞速的发展,物联网的应用涵盖智能家居、交通物流、环境保护、公共安全、智能消防、工业监测、个人健康等多种领域。但对于这些日益复杂的物联网应用来说,应用所处的环境更为复杂,传感器自身受到损害的可能性也更大,因此这些传感器在信息处理的过程将中不可避免地受到传感器信息质量的影响。

如果不能反映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,则将导致不一致性发生;如果上下文的某些方面不可知,将导致信息的不完整。因此,上下文的模型需要指定上下文信息的一系列特性,包括时效特性(刷新率和历史)、准确性、解析度(粒度)、上下文信息的准确度中的信任度。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,同时准确性和异构的表达格式也不同,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。本文着重关注上下文信息的不完备性、不精确性和不一致性的问题。

1)上下文信息中的不一致性

上下文不一致性的发生有许多原因,可能是因为不准确的位置检测或不正确的活动推理。而在现实生活中,因为推理资源的缺乏,上下文不一致性经常发生。

2)上下文信息中的不完备性

上下文的不完备性是指上下文感知系统中的某类上下文的样本数据向量中缺失样本的比例。我们将缺失样本定义为按照传感器的采集频率上下文的样本数据向量的对应位置缺失的样本。

3)上下文信息中的不精确性

上下文的不精确性是指上下文感知系统的某类上下文的样本数据向量中不精确样本的比例。不精确样本是指上下文的格式或上下文数据某方面的精确度不符合上下文感知系统要求的样本。

到稿日期:2014-07-31 返修日期:2014-09-03 本文受国家自然科学基金项目(61100041),中国博士后基金项目(2012M512131)资助。

郑 笛(1977—),女,博士,讲师,主要研究方向为普适计算,E-mail:dizheng1216@163.com;贲可荣(1963—),男,博士,教授,博士生导师,主要研究方向为软件工程;王 俊(1978—),男,博士,讲师,主要研究方向为分布式计算。

## 2 上下文质量管理方法

要解决不一致性、不完备性、不精确性这些上下文质量问题,首先要建立合适的上下文质量模型。

### 2.1 基于本体的上下文质量管理模型

如图 1 所示,采用层次化的方式实现上下文感知管理,将整个感知过程依次划分为传感层、获取层、处理层、分发层、应用层。与现有处理方式不同,将整个质量管理机制贯穿到各个层次中,在上下文信息获取层,通过质量因子度量器和阈值控制机制实现自定义的上下文信息丢弃,降低系统需要处理的上下文数量,提高系统处理的原始上下文质量。同时根据不同的度量机制生成不同的质量参数,为后续的处理算法提供输入。在上下文信息处理层,扩展现有的不完备、不精确、不一致上下文以及上下文的推理机制,针对各种质量因子实现定制,提供更加高效的上下文处理机制。在上下文的分发层,实现基于质量信息的上下文适配、上下文选择等多种机制。同时,考虑在应用层针对用户的特有需求和私密性进行进一步扩展。

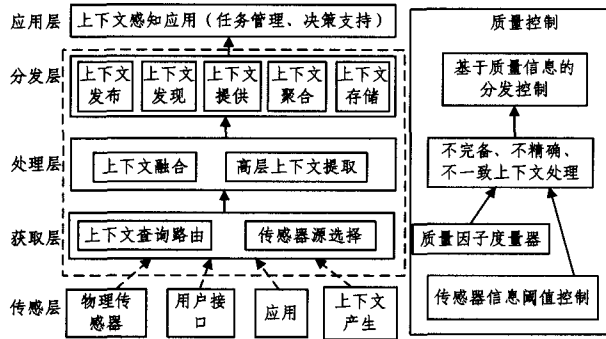


图 1 一般化的上下文质量管理框架

### 2.2 不完整上下文处理

对于不完整上下文,可以通过完整性因子进行管理,当完整性因子的计算值低于设定阈值时,可考虑丢弃该来源的上下文数据或发出报警。当上下文数据的完整性在阈值范围内,且不完备性呈现离散性特性时,可以考虑使用历史数据进行数据填充。上下文感知系统的每轮运行都会搜集到大量的上下文向量,随着运行次数的增多,积累的上下文向量样本会详细记录各类上下文的历史数据。当后续的某次运行出现缺失样本时,可以充分利用该上下文的历史数据,从中发掘出该上下文的样本期望值,最后用该期望值填充缺失的样本。

本文基于经典的期望-极大化(Expectation Maximization)算法<sup>[9]</sup>生成缺失的样本,其步骤如下所示。EM 是一个在已知部分相关变量的情况下估计未知变量的迭代算法。该算法的每次迭代是由期望(Expectation)和极大(Maximization)两步操作构成。第一步是计算期望(E),也就是将缺失变量好像能够观测到一样包含在内,计算最大似然的期望值;第二步是极大化(M),也就是极大化在 E 步骤上找到的最大似然的期望值,从而计算参数的最大似然估计。M 步骤上找到的参数将被用于下一次迭代的 E 步骤,这个过程不断交替进行。通过 EM 算法,首先在给出缺失上下文数据(样本)初值的条件下估计出参数值,然后根据参数值估计出缺失上下文数据的值;再根据估计出的缺失数据值对参数值进行更新,如此反复迭代直至收敛。将收敛时刻的缺失样本值作为缺失样本的最终取值,从而解决上下文的缺失问题。

### 算法 1 基于 EM 方法的上下文不完整消除算法

INPUT:新到达的上下文

1. 检测上下文的完整性
2. 如果存在属性缺失或其它不完整
3. 查找不完整上下文的历史数据
4. 计算不完整上下文样本的期望
5. 通过期望填充上下文的不完整属性
6. 运用极大似然估计方法极大化找到的最大似然的期望值,从而计算参数的最大似然估计
7. 如果收敛,则上下文填充完成,导出完整的上下文
8. 否则重复步骤 5,继续进行期望填充
9. 不完整上下文处理完成

本算法的第 4 步首先利用缺失样本所在上下文的样本历史数据计算得到缺失样本的期望值,为了得到相关性更好的期望值,可以使用与缺失样本具有相同其它上下文的历史样本计算该期望值(同一个情景的某些底层上下文之间具有相关性,系统根据情景的迁移决定进一步的行为在不知不觉中利用了这种相关性)。但这两种获取缺失样本初值的方式对最终的估计结果影响不大,这是因为 EM 方法是一个迭代过程,初值的较小差异只会影响收敛时的迭代次数,即计算时间有所差异。第 6 步的本质是执行 EM 算法的 E 步骤和 M 步骤,若 EM 算法收敛,即估计值趋于稳定值时,执行第 7 步,将最近这次迭代的对缺失样本的估计值作为缺失样本的最终值;若 EM 算法不收敛,则执行第 8 步,使用第 5 步估计出的参数值重新评估缺失样本的期望值,并调回第 5 步,开始新一轮迭代。在估计出缺失样本后,上下文感知系统可以像没有出现样本丢失时的情况一样,继续进一步的行为,即上下文不完备性问题得以解决。

### 2.3 不精确上下文处理

事实上,上下文的不精确性往往是瞬发的,这些数据并不一定会造成严重的影响。对不精确性的直观判断包括:该来源的上下文在一致性处理中被标记为引起不一致性的源头;在其它来源没有特别行为变化时,该数据源的上下文出现一定的跳变。

因此,对于不精确上下文,当出现异常畸变时,则可以直接丢弃;当未出现异常畸变,但精确性因子的计算值低于设定阈值或者被标记为引起不一致性的源头时,则可以考虑使用类似不完备性处理的方式用历史数据进行数据填充。

事实上,本文处理上下文的不完备性问题和不精确性问题的差异主要体现在如何确定缺失样本或不精确样本的初值上。针对不精确的上下文样本,可以使用以下两种方式为 EM 算法确定该样本的初值:1)直接将不精确的样本数据作为初值;2)将该不精确的样本丢弃,并将之看成不完备的样本,再按照解决不完备性问题时使用的方法来确定初值。在确定了不精确样本的初值后,不精确性问题就转化成了不完备性问题,算法如下所示。

### 算法 2 基于 EM 方法的上下文不精确性消除算法

INPUT:新到达的上下文

1. 检测上下文的精确性
2. 如果上下文数值出现过大的畸变,则丢弃
3. 如果该来源被标记为精确性存疑,则实施 EM 方法进行检查
4. 查找不精确上下文的历史数据
5. 计算不精确上下文样本的期望
6. 通过期望填充上下文的不精确属性
7. 运用极大似然估计方法极大化找到的最大似然的期望值,从而计算参数的最大似然估计

8. 如果收敛,则上下文填充完成,导出精确的上下文
9. 否则重复步骤 5,继续进行期望填充
10. 否则直接提交后续推理处理
11. 不精确上下文处理完成

## 2.4 上下文不一致处理

同样地,当上下文出现多来源信息不一致性时,则通过自定义的各种可信度量度方式进行一致性处理,丢弃可信度低的冲突上下文。

### 算法 3 上下文不一致性检测

1. (1)T1:上下文预处理线程
2. 等待任何输入的上下文实例 ins
3. 为 ins 匹配的每一个模式 pat
4. 将 ins 添加到 pat 的匹配队列 pat\_que 中
5. If ins 是 pat\_que 中的第一个元素
6. 基于模式的刷新需求和 ins 的创建时间为模式创建一个定时器 t
7. (2)T2:不一致性触发线程
8. 等待模式匹配队列 pat\_que 中任何新的上下文实例 ins
9. 对所有其它的模式 pat<sub>1</sub>, pat<sub>2</sub>, ..., pat<sub>n</sub>
10. 在 pat 的触发器 tgr 中
11. if exists ins<sub>1</sub> in pat\_que<sub>1</sub>, ins<sub>2</sub> in pat\_que<sub>2</sub>, ..., and ins<sub>n</sub> in pat\_que<sub>n</sub>  
并且 tgr 对 ins<sub>1</sub>, ins<sub>2</sub>, ..., ins<sub>n</sub> 的约束都满足
12. then if tgr 的约束也满足
13. then 检测到不一致性
14. delete 所有冲突的上下文实例
15. 通过定时控制线程的定时器将未冲突的上下文实例加入上下文池中
16. (3)T3:定时控制线程
17. 循环等待任何过时的定时器 t
18. 从 t 相关联的模式 pat 的匹配队列 pat\_que 中移去第一个元素
19. if pat\_que<sub>1</sub> = ∅
20. then cancel t
21. else 基于 pat 的更新需求更新 t 和新的第一个元素的产生时间

算法包括 3 个线程: T1、T2 和 T3。T1 为每一个进入的上下文实例执行匹配,并为每一个匹配模式的匹配队列附着一个 copy。T2 监控所有的匹配队列以检测是否有一组上下文实例,能够使用所有满足的条件和约束激活一个触发器,并在发生不一致冲突时丢弃可信度较低的上下文。T1 和 T2 作为上下文实例的生产者和消费者对,用于非一致性触发过程。T3 管理所有的运行定时器,并当必要时从它们的本地匹配队列中移出过期的上下文实例。

可信度的计算按照不同的质量因子的选择或组合来进行,例如:

- 1) 选择时间因子来判断不一致上下文处理,只丢弃相对较老的上下文信息;
- 2) 选择完整性因子来判断不一致上下文处理;
- 3) 选择有效性因子来判断不一致上下文处理,通过有效性因子来判断信息融合的正确性概率;
- 4) 通过确定性因子来判断不一致上下文处理,通过确定性来判断信息融合的正确概率;
- 5) 综合考虑上述几种因子的组合。

可信度的计算对于系统的性能有着重要的影响,对于节点数目较多、数据量较大且数据变化相对较为平滑的情况,可以采用较为简单的处理机制,只根据时间的先后来进行处理;对于数据波动较大的部分节点,则可以采用精度更高的一致

性处理方法,提高数据的精确性。同时,系统充分利用并发机制,将质量因子的计算、历史数据分析等相关处理采用异步并发的方式在后台进行,以提高处理效率。

## 2.5 关注质量的上下文感知计算流程

图 2 所示为关注质量的上下文感知计算流程,第一步为原始上下文的收集,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。第二步是不完整性、不精确性、不一致性上下文的解析,选择在这一步中解析不同原始上下文中存在的不完整性、不精确性、不一致性,是因为原始上下文中存在的不完整性、不精确性、不一致性将最终导致高层上下文融合性能,而这种高层上下文的不确定性将是非常难以解决的。

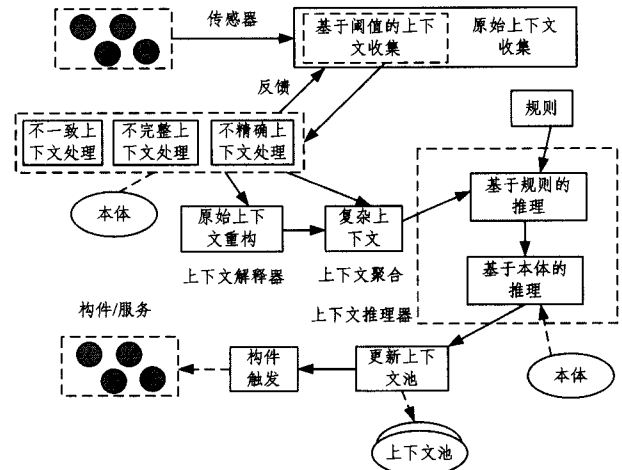


图 2 基于质量管理的上下文处理流程

采用批处理的方式来对原始的上下文进行处理,每一批原始上下文中的不完整性、不精确性、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确定性能够被限制在一个特定的程度。第三步是原始上下文的重构,在本步中我们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确定性。然后,运用基于规则的推理来产生高层的上下文,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。

## 3 性能评价

本文以某车联网系统作为实验平台,实验中共使用 3 台计算机,一个运行 Windows 2008 R2 的服务器和两个 PC 客户端,服务器运行上下文解析处理服务和车载综合环境监测系统。PC 客户端模拟多个终端传感器实际数据的上报,每类上下文上报的时间间隔设为 30s。因为传感器数量和范围的限制,本实验没有开启阈值机制,所有上下文信息不经过采集时丢弃的环节,直接转发给处理端。实验分为不完整上下文处理、不精确上下文处理和一致上下文处理 3 个部分。

### 3.1 不完整上下文处理性能

不完整上下文主要考察系统内的定位数据上下文和 OBD 数据上下文,这两类上下文都属于连续变化上下文,没有突变性,历史数据具有一定的可重复性。在数据中,以不同的比例植入不完整信息,如定位信息中缺少经度或纬度, OBD 信息中缺少里程、转速等信息,相应的信息融合的正确性概率如图 3 所示。

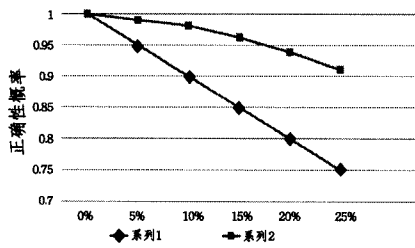


图3 不完整上下文处理性能

如图3所示,当使用不完整上下文处理时,能够有效提高上下文处理的正确性概率。图中带“◆”曲线代表未使用完整性处理的情况,逐次提高系统中不完整上下文的比例,可以看到正确性概率随之下降。事实上,由于模拟中使用的上下文种类较少,同时定位信息与OBD信息并不构成矛盾,因此相互间没有互相影响,系统上下文的正确性概率能够基本保持与注入不完整上下文相同的水平。但对于更为复杂的物联网系统,当传感器上下文种类众多且存在相互影响时,系统的正确性概率将超过注入不完整上下文的比例。同时,对于增加了不完整上下文处理的情况,随着不完整上下文比例的增加,完整性处理的有效性也会缓慢受到影响,因为这种增加会使得系统处理的局部性减弱。当然,当不完整性超过阈值时,在实际应用系统中会舍弃该来源或进行报警。

相应地,完整性处理由于迭代运算的存在会在一定程度上影响系统的性能,对于有大量上下文来源的系统,则需要二者之间达成一个平衡。

### 3.2 不精确上下文处理性能

不精确上下文主要考察系统内的定位数据上下文和OBD数据上下文,在数据中,以不同的比例修改正确的信息,如定位信息中经度或纬度发生偏移、OBD信息中某些参数发生偏移等,相应的信息融合的正确性概率如图4所示。

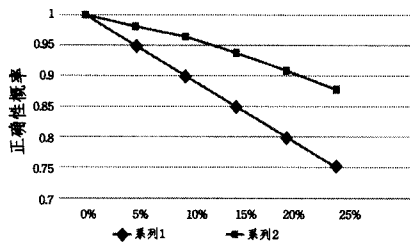


图4 不精确上下文处理性能

如图4所示,当使用不精确上下文处理时,能够有效提高上下文处理的正确性概率。图中带“◆”的曲线代表未使用完整性处理的情况,逐次提高系统中不完整上下文的比例,可以看到正确性概率随之下降。对于增加了不精确上下文处理的情况,随着不精确上下文比例的增加,精确性处理的有效性也会缓慢受到影响,因为这种增加会使得系统处理的局部性减弱。同时,由于精确性的衡量标准不如完整性那样精确,数值的偏差依赖于设定的规则和取值的范围,如果连续几个值发生了不同的偏差,则往往会导致相对不精确的结果出现。因此,相比完整性上下文处理,精确性处理的正确性概率的结果要差一些。

相应地,精确性处理由于迭代运算的存在会在一定程度上影响系统的性能,对于有大量上下文来源的系统,则需要二者之间达成一个平衡。

### 3.3 不一致上下文处理性能

在完整性和精确性处理的基础上,综合考虑不一致性的

处理。人为增强上下文的不确定性,在不完整性、不连续性、不准确性的基础上,增加OBD传感器和尾气传感器信息的不一致性,增加临近位置车辆数据的信息不一致性。每个节点的定位数据、OBD数据、尾气数据的故障率设定为一个相对均匀的比例,同时不同节点的不一致设定也基本保持均匀。

本文对融合后信息的正确性概率进行分析统计。针对上下文的设置,分别采用下列几种质量计算机制:

- 1) 系列1,不进行质量管理,直接将原始信息进行融合;
- 2) 系列2,只通过完整性因子来判断不一致上下文处理,对应采集数据的不完整;
- 3) 系列3,只通过有效性因子来判断不一致上下文处理,通过有效性因子来判断信息融合正确性概率,对应OBD传感器数据的跳变;
- 4) 系列4,只通过确定性因子来判断不一致上下文处理,通过确定性来判断信息融合正确概率,对应尾气传感器数据的高故障性和环境数据的不一致性;
- 5) 系列5,综合考虑上述几种因子的组合,为便于讨论,这里预处理的过程对因子的权重生成相同的权值。

几类不确定性上下文的比例如图5所示。

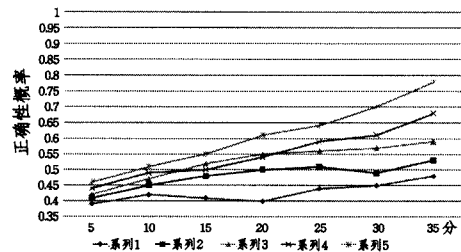


图5 信息处理正确性概率

相应地,增加每台终端计算机上对应的实例数目,对应的信息融合的正确性概率分别如图6所示。

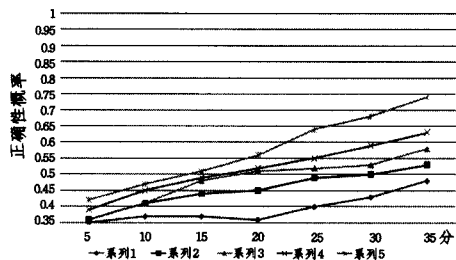


图6 增加节点数后的信息处理正确性概率

其它的调整过程和性能开销的列表不一一列举,可以看出:

- 1) 综合考虑几种因子的可信度计算方法在本例中具有最好的信息融合效果,获得的信息融合可信度最高。
- 2) 不同的可信度计算方法会得到不同的可信概率,它取决于具体的实际应用,如本实例中每类不确定问题相对比较独立,所以各自的变化曲线相对比较均匀,没有出现剧烈变化的情况。
- 3) 随着时间的推移,不论采用哪种可信度计算方法获得的正确性概率都会逐渐增加,并向1靠拢,但综合的可信度计算方式会带来更快的收敛速度。
- 4) 节点数的数目对信息融合的处理过程会造成一定的影响,但这种影响主要体现在融合过程的开销和复杂度上,只要不是上下文耦合度非常高的应用,不会出现剧烈的变化曲线。

要说明的是,因子组合的方法会带来相对更大的开销,而

不确定性出现的原因也多种多样,并非实验中设定的那样均匀。因此在系统实际应用过程中,往往采用变化的策略,如果有多个不一致的情况出现,则选用复合型的可信度计算方法;如果相对较少且比较平稳,则可以选用快速的信息融合方法。

**结束语** 物联网在广泛应用的同时不可避免地带来对传感器感知的上下文质量的需求。本文在传统的上下文感知技术的基础上,充分研究了不一致性、不完整性、不准确性等低质量传感器上下文的消除问题,通过上下文质量因子分类配置、不准确与不一致上下文丢弃、不完整上下文填充等方法实现不同层次的控制机制,降低信息的不确定性,从而有效提高物联网应用的上下文处理质量。

## 参考文献

- [1] Vermesan. Internet of Things: Strategic Research Roadmap [OL]. [http://www.internet-of-things-research.eu/pdf/IoT\\_Cluster\\_Strategic\\_Research\\_Agenda\\_2011.pdf](http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf)
- [2] Wang J, et al. A study on wireless sensor network based indoor positioning systems for context-aware applications[J]. *Wireless Communications and Mobile Computing*, 2012, 12(1): 53-70
- [3] Claudio, et al. A survey of context modeling and reasoning techniques[J]. *Pervasive and Mobile Computing*, 2010, 6(2): 161-180
- [4] Gaia Group[OL]. <http://gaia.cs.uiuc.edu>
- [5] Gu T, Pung H K, Zhang D Q. A Service-Oriented Middleware for Building Context-Aware Services [J]. *Elsevier Journal of Network and Computer Applications*, 2004, 28(1): 1-18
- [6] Chen H. An Intelligent Broker Architecture for Pervasive Context-Aware Systems[D]. University of Maryland, 2004
- [7] Chen H, Finin T, Joshi A. An ontology for context-aware pervasive computing environments[J]. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 2003, 18(3)
- [8] Hong J L. An Architecture for Privacy-Sensitive Ubiquitous Computing[D]. University of California, Berkeley, 2005
- [9] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. *Journal of the Royal Statistical Society(Series B)*, 1977, 39(1): 1-38
- [10] Roy N, et al. Resource-Optimized Quality-Assured Ambiguous Context Mediation Framework in Pervasive Environments[J]. *IEEE Transactions on Mobile Computing*, 2012, 11(2): 218-229
- [11] Sheikh K, Wegdam M, van Sinderen M. Quality-of-context and its use for protecting privacy in context aware systems[J]. *JSW*, 2008, 11(3): 83-93
- [12] Manzoor A, Truong H L, Dustdar S. On the evaluation of quality of context[C]// *Smart Sensing and Context, Third European Conference(EuroSSC)*. 2008: 140-153
- [13] Filho. Modeling and Measuring Quality of Context Information in Pervasive Environments[C]// *Proc of 24th IEEE International Conference on Advanced Information Networking and Applications*. 2010: 690-697
- [14] Sheikh K, Wegdam M, van Sinderen M. Middleware support for quality of context in pervasive context-aware systems [C] // *PERCOMW'07*. Washington DC, USA: IEEE Computer Society. 2007: 461-466
- [15] 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
- [16] Allen R, Garlan D. A formal basis for architectural connection [J]. *ACM Trans. on Software Engineering and Methodology*, 1997, 6(3): 213-249
- [17] Hoare C A R. Communicating sequential processes[J]. *Communications of the ACM*, 1978, 21(8): 666-677
- [18] Ryan P, Schneider S, Goldsmith M, et al. The Modelling and Analysis of Security Protocols: The CSP Approach[M]. Boston: Addison-Wesley, 2000
- [19] Roman G C. Specifying software/hardware interactions in distributed systems[C]// *Proc. of the 9th Int'l Conf. on Software Engineering*. Los Alamitos: IEEE Computer Society, 1987: 126-139
- [20] Hoare C A R, He J. Unifying Theories of Programming[M]. New York: Prentice-Hall, 1998
- [21] Harrison M, Barnard P. On defining requirements for interaction [C]// *Proc. of the IEEE Int'l Symp. on Requirements Engineering(RE'93)*. Washington: IEEE Computer Society, 1993: 50-54
- [22] FDR. ProBE[OL]. <http://www.fsel.com>
- [23] Hall A, Chapman R. Correctness by construction: Developing a commercial secure system[J]. *IEEE Software*, 2002, 19(1): 18-25
- [24] Peleska J. Applied formal methods-From CSP to executable hybrid specifications[C]// *Abdallah A E, Jones C B, Sanders J W, eds. Proc. of the 2004 Int'l Conf. on Communicating Sequential Processes; The 1st 25 Years*. Heidelberg: Springer-Verlag, 2004: 293-320
- [25] Creese S. Industrial strength CSP: Opportunities and challenges in model-checking[C]// *Abdallah A E, Jones C B, Sanders J W, eds. Proc. of the 2004 Int'l Conf. on Communicating Sequential Processes; The 1st 25 Years*. Heidelberg: Springer-Verlag, 2004: 292
- [26] Harel D. Statecharts: A visual formalism for complex systems [J]. *Science of Computer Programming*, 1987, 8(3): 231-274
- [27] Li Z. Progressing problems from requirements to specifications in problem frames[C]// *Proc. of the 3rd Int'l Workshop on Applications and Advances of Problem Frames (IWAAPF 2008)*. New York: ACM Press, 2008: 53-59
- [28] Li Z, Hall J G, Rapanotti L. On the systematic transformation of requirements to specifications[J]. *Requirements Engineering*, 2014, 19(4): 397-419
- [29] 李智, 庞柳, 刘国源, 等. 一种模型驱动的软件需求分析方法及技术支持[J]. *广西师范大学学报(自然科学版)*, 2013, 31(2): 19-26
- [30] Li Z, Pang L, Liu G-Y, et al. A Model-Driven Software Requirements Analysis Method and Its Technical Support[J]. *Journal of Guangxi Normal University(Natural Science Edition)*, 2013, 31(2): 19-26
- [31] <http://problemoriented.wikispaces.com/References+and+Links>
- [32] [http://en.wikipedia.org/wiki/Problem-oriented\\_development](http://en.wikipedia.org/wiki/Problem-oriented_development)
- [33] Seater R, Jackson D, Gheyi R. Requirements progression in problem frames[J]. *Deriving Specifications From Requirements*, 2007, 12(2): 77-102
- [34] 朱利鲁. 关于问题框架理论中领域间因果行为形式化验证的研究[D]. 桂林: 广西师范大学, 2015
- [35] Zhu L-L. The Study About Formal Validation of Causal Behaviors of Domains In Problem Frames[D]. Guilin: Guangxi Normal University, 2015