

# ScudCORE: 一个情境驱动的推理引擎

潘 纲, 李 彤, 任豪毅, 李石坚\*, 姚 敏

(浙江大学计算机学院, 浙江杭州 310027)

**摘 要:** 普适计算对推理引擎提出了有别于传统方法的诸多新挑战. 面向普适计算的推理引擎需要基于情境驱动, 可灵活应对情境时变频繁、实时性、开放性等问题. 本文在分析了普适计算中情境感知特性的基础上, 提出了一个以一阶谓词逻辑为基础的情境驱动的推理引擎 ScudCORE (Scud Context-driven Reasoning Engine). ScudCORE 采用情境过滤机制弱化了由情境时变频繁带来的推理过度频繁、效率低下的问题; 采用本体建模技术使推理引擎内外之间的情境信息能相互共享、重用、理解, 从而实现推理引擎的主动触发; 通过规则管理器增强规则的动态管理功能, 提高推理引擎的灵活性.

**关键词:** 推理引擎; 情境驱动; 普适计算

**中图分类号:** TP182 **文献标识码:** A **文章编号:** 0372-2112 (2009) 4A-070-05

## ScudCORE: A Context-Driven Reasoning Engine

PAN Gang, LI Tong, REN Hao-yi, LI Shi-jian, YAO Min

(Department of Computer Science, Zhejiang University, Hangzhou, Zhejiang 310027, China)

**Abstract:** Pervasive computing puts forward a number of challenges on reasoning engine, such as: to be context-driven, to deal with frequently change of contexts, and to work in a real-time manner. Traditional reasoning engines can hardly meet these new requirements. In this paper, we present a context-driven reasoning engine ScudCORE (Scud Context-driven Reasoning Engine), which takes the first order predicate logic as basis. When contexts vary frequently, ScudCORE reduces the reasoning problems of excessive frequency and inefficiency by the context filtering mechanism. We employ ontology to make the reasoning engine convenient to share, reuse and understand context information both inside and outside it. Also we increase its flexibility with a dynamic rule management module.

**Key words:** reasoning engine; context-driven; pervasive computing

### 1 引言

近年来, 国内外普适计算各项关键技术的研究发展迅速, 情境感知能力是普适计算的核心特征之一. 情境指整个物理环境、计算环境、以及用户等方面的静态与动态的状态信息<sup>[1]</sup>. 研究者希望通过情境感知使部署在日常生活中的各种计算设备或系统能透明地提供主动服务, 和谐地辅助人的工作与生活. 而情境感知技术能否根据具体情境在合适的时间、合适的地点、以合适的方式自发透明地为人们提供主动服务, 其关键在于是否具有一个有效的情境驱动的推理机制<sup>[2]</sup>.

普适计算环境下的情境驱动推理与传统推理相比具有如下特点和需求:

(1) 情境驱动: 情境感知推理过程为主动触发式, 在情境信息符合服务需求时自动触发推理, 然后驱动计算

设备为用户提供服务; 已有推理引擎都是目标驱动或问题驱动的;

(2) 情境时变频繁: 在普适计算环境中有着丰富的情境信息, 这些情境信息随着时空的变化会频繁更新, 现有推理引擎对此处理效率较低;

(3) 实时性: 由于要为用户提供透明的、主动的服务, 和谐地辅助人的工作与生活, 这就需要情境驱动推理具有实时性, 能及时为用户提供服务;

(4) 开放性: 建立的推理模块并不是一成不变的, 随着使用的应用场景的改变, 需要及时更改规则, 以适用于新的环境.

当前普适计算领域科研人员提出了许多知名的情境感知原型及系统, 如 Context Toolkit、Gaia、CORTEX、SO-CAM 等, 多着重于支持情境感知的体系结构的研究, 其中通常直接使用现有的传统推理引擎, 而未专门针对情

收稿日期: 2008-11-22; 修回日期: 2008-12-30

基金项目: 国家科技支撑计划 (No. 2006BAH02A01); 国家 863 计划 (No. 2006AA01Z198, No. 2008AA01Z132); 国家自然科学基金 (No. 60525202, No. 60533040)

\* 本文通信作者: 李石坚.

境驱动的推理引擎展开研究. 人工智能领域虽对推理技术已有了很多研究, 并有许多实用的推理引擎, 如 Prolog、Visual LISP、CLIPS、JESS、JENA、JRules、FaCT、RACER 等, 但并不能很好适用基于情境驱动的推理特点. 传统推理引擎与基于情境驱动的推理引擎的比较如表 1 所示.

表 1 传统推理引擎与情境驱动的推理引擎比较

	传统推理引擎	情境驱动的推理引擎
驱动机制	目标驱动	情境驱动
实时性要求	中低	高
对时空频变支持的需求	低	高

因此, 亟待研究一种能更好支持普适计算应用环境的情境驱动的推理引擎. 本文以一阶谓词逻辑<sup>[3]</sup>为基础, 为解决情境时变频繁带来的推理计算过度、效率低下等问题, 提出了一个情境驱动的推理引擎体系结构, 引入独特的情境过滤方法, 建立了有效的情境驱动机制; 并改造开源的专家系统开发工具 CLIPS<sup>[4]</sup>, 构建了一个支持情境感知系统开发的情境驱动的推理引擎 ScudCORE (Scud Context-driven Reasoning Engine).

## 2 ScudCORE 的体系结构

在总结传统推理引擎的功能及研究情境感知推理特点的基础上, 我们提出了能更高效地支持情境感知计算的推理引擎 ScudCORE, 它采用情境过滤机制弱化了情境时变频繁的问题, 利用本体建模技术<sup>[5]</sup>实现了情境信息的共享、重用、理解功能, 通过规则管理器模块增强了推理引擎的灵活性. 本节将介绍我们 ScudCORE 体系结构中各个模块的功能.

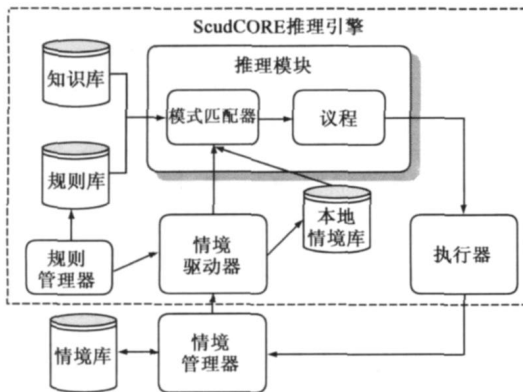


图1 推理引擎ScudCORE的体系结构

ScudCORE 推理引擎的体系结构如图 1 所示, 它主要包括推理模块、知识库、规则库、规则管理器、情境驱动器、本地情境库和执行器等部分. 其中推理模块、知识库规则库和执行器与经典规则推理引擎类似, 而规则管理器、情境驱动器和本地情境库是我们为了实现情境驱动机制而添加的部分. 各部分的功能如下:

(1) 知识库和规则库: 知识库存放一些领域的专家知识事实和通用知识事实, 这些知识一般都是静态的; 规则库则存储用于推导用户需求的规则集.

(2) 推理模块: 推理模块包括模式匹配器和议程, 其中模式匹配器的输入包括知识库、规则库及本地情境库, 主要决定有哪些规则被事实所满足; 议程则给这些规则授予优先级.

(3) 执行器: 执行器选择并执行具有最高优先级的规则, 并把执行的结果传给外部的情境管理器.

(4) 规则管理器: 为动态添加、删除、修改规则提供接口, 检查规则库改变之后有可能产生的规则冲突, 并将更新规则消息发送到情境驱动器.

(5) 情境驱动器: 是 ScudCORE 引擎具有情境驱动机制的核心模块, 主要为情境信息提供各种特殊管理机制, 解决了情境时变频繁等问题, 提高引擎的推理速度及灵活性.

(6) 本地情境库: 为了方便我们管理情境信息这些动态的“事实”, 在 ScudCORE 中, 我们并不直接把情境信息存储在知识库, 而是为推理引擎专门建立了一个独立的数据库来存放推理需要的情境信息.

(7) 情境管理器和情境库: 这两个不是 ScudCORE 的内部功能模块, 而是整个普适计算情境感知系统中与 ScudCORE 直接相关的部分. 情境管理器主要为 ScudCORE 提供两个功能: (a) 为情境驱动器提供所需要的情境信息; (b) 根据推理结果更新情境库或触发事件, 驱动计算设备为用户提供服务. 情境库存储所有的情境信息.

## 3 ScudCORE 的情境驱动机制

为建立有效的情境驱动机制, 使 ScudCORE 满足情境感知推理的特殊需求, 本文采取的主要策略包括:

(1) 基于本体情境表达: 实现了推理引擎与情境管理器对情境的一致性理解与共用, 在语义层次上打通两者之间的情境数据通道, 是情境驱动机制的基础;

(2) 情境过滤机制: 由于普适计算环境中含有丰富的情境信息, 并且这些情境信息时变频繁, 为了加快推理速度, 使推理系统具有实时性, 我们在 ScudCORE 中增加了情境驱动器, 达到过滤大量情境信息的目的;

(3) 规则动态管理: 根据当前环境推理需求, 可动态向推理引擎增加、删除规则, 使推理引擎更具灵活性, 大大增强了情境驱动能力.

在具体实现上, 我们以开源的专家系统开发工具 CLIPS<sup>[4]</sup>为基础, 在新的体系结构下对其进行改造, 建立了新型的情境驱动推理引擎 ScudCORE.

### 3.1 基于本体情境表达

为实现推理引擎主动驱动, 须使从外部环境采

集到的情境信息为推理引擎所理解,如此,当外部环境中情境信息发生改变时,即可驱动推理引擎.为使采集到的情境信息能转变为推理引擎可以直接理解的事实,在推理引擎与外部交互方面,我们采用了本体技术<sup>[5]</sup>.本体是一个概念体系的显示的形式化规范.利用本体,不仅使推理引擎内部与外部的语境信息能共享同一概念,也使我们的推理引擎与系统内别的模块能在语义层面相互通信,同时便于我们在开发普适计算情境感知系统时形成统一规范.

在 ScudCORE 中,我们采用了基于 SOUPA<sup>[6]</sup>的情境本体模型,SOUPA 是一个可扩展的标准本体模型.事实表示为一阶谓词逻辑,可用 RDF 三元组格式来表示.图 3 是一个将基于 RDF 表示的情境信息进入推理引擎内部的例子.一个名为 GPan 的人现在位于车内的这个语境信息用一阶谓词逻辑表示为 locationOf(GPan, Car),用 RDF 三元组格式表示,如图 2(a) 所示.进入推理引擎之后,转换为其内部能理解的事实,如图 2(b) 框中所示.



图2 一个RDF表示的情境信息进入推理引擎内部的例子

### 3.2 情境过滤机制

#### 3.2.1 基于规则关联的情境过滤

情境信息采集为我们提供大量的情境信息,但不是所有的情境信息都是推理系统所需要的,我们只需关注与规则有关的那些情境信息,这样可以大大减轻推理引擎负担,提高推理速度.

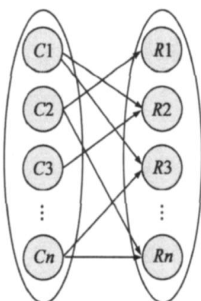


图3 情境规则关联表示意图

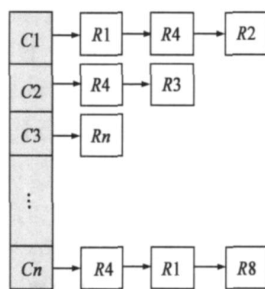


图4 情境规则关联表数据结构

为此,我们维护了一张情境规则关联表,如图 3 所示.其中 C 代表情境信息, R 表示规则,数字下标区分不同情境信息和规则.情境与规则为多对多关系,即,一个情境信息可被多条规则所用,同时一条规则也会用到多个不同情境信息.通过该表我们可动态获得所有规则可能会用到的情境信息集合.

我们用数组存储情境信息名,如图 4 所示,根据建立的情境本体,每个情境都有一个唯一的 Key,采用哈希映射能快速访问我们所需查找的情境信息.用链表存储与该情境信息有关的规则集,当添加一条新的规则时,根据规则用到的情境信息,在每个情境信息的链表中添加该规则的节点,当规则所用到的情境信息不存在时,添加该情境信息并增加规则节点.同时在删除规则时,如果一个情境信息后面的链表为空就删除数组中该情境信息.图 5 显示了添加一条新规则时,如何对情境规则关联表进行维护的伪码.

```

Context-memory :array of Context
Structure Rule :
RuleName :symbol
ContextChildren :list of contextname
End
Structure Context :
ContextName :symbol
RuleChildren :list of rulename
End

procedure add-rule (R :Rule)
for each Cname in R.ContextChildren
do context-memory-activation (Cname ,R.RuleName)
end

procedure context-memory-activation
(CName :symbol ,Rname :symbol)
c-men ← lookup-in-hash-table (CName)
if c-men ≠ Nil
then
insert Rname at the head of
Context-memory[c-men].RuleChildren else
{c-men = hash (CName)
Context-memory[c-men].ContextName = CName
insert Rname at the head of
Context-memory[c-men].RuleChildren}
end
  
```

图5 情境规则关联表的更新算法伪码:增加一条规则

#### 3.2.2 基于订阅的情境过滤

有了情境规则关联表,我们就可以有选择地向情境管理器订阅所需要的情境信息.由于在普适计算环境下,服务的触发是由当前情境发生改变引起的,因此只有当推理引擎关注的情境信息发生变化时,情境管理器才向情境驱动器发送情境信息.这样有效降低了推理引擎对大量情境信息的处理开销.

但有些情境信息处于连续变化状态,例如温度、用户位置、光强、速度等等,面对这些特殊的情境信息,采

用变化机制向情境管理器订阅,推理引擎需要处理的情境信息量还是很大.我们这里采用增量  $T$  机制向情境管理器订阅情境信息,当新采集的情境信息  $T_1$  与本地情境库中的当前情境信息  $T_0$  的增量  $T$  大于某个值的时候,情境管理器才将该情境信息发送给情境驱动器,同时情境驱动器用  $T_1$  更新本地情境库中的  $T_0$ .这种情境过滤机制需要具有该增量订阅模式的情境管理器支持.

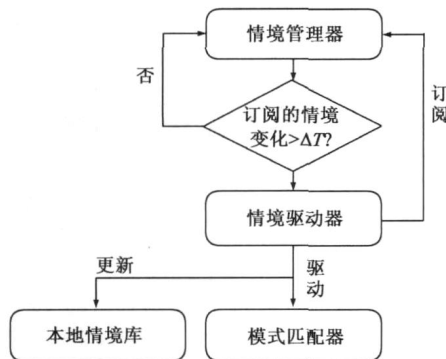


图6 情境增量订阅及驱动流程图

### 3.3 规则动态管理

情境感知系统需每时每刻为用户提供主动服务,这就需要推理引擎持续处于运行状态,而我们的普适计算应用场景不可能一成不变的,经常会添加或删除一个计算设备,甚至于整个应用场景会发生改变.这时现有的推理规则就有可能不符合当前需求,需要更改规则来满足新的应用情况.若关闭推理引擎,然后更改规则再重新启动推理引擎,必将使整个系统在这段时间内处于瘫痪状态,代价昂贵.为此,我们在 ScudCORE 中增加规则管理器模块,提供动态添加、删除、修改规则的功能,使推理引擎更具灵活性.

在 ScudCORE 推理引擎中,我们采用 Rete 算法<sup>[7]</sup>作为 ScudCORE 的模式匹配算法,Rete 算法做为一个经典模式匹配算法,不仅解决事实与规则进行快速匹配的问题,而且能为我们实现推理运行时动态添加与删除规则而不影响整个推理引擎的运行提供了很好的支持.

当有规则更新时,我们首先要检测更新的规则是否与规则库中的原有规则是否有冲突,若没有冲突,则更新规则库,同时规则管理器通知情境驱动器规则更新,此时需要更新情境规则关联表,维护关联表状态与当前规则库之间的一致性,从而实现正确的情境信息过滤.针对规则冲突问题,目前我们的规则管理器只是简单的采用了专家系统规则冲突消除策略来解决这个问题<sup>[8]</sup>.以下是我们提出的规则更新算法框架,包括规则的添加、删除与修改.

规则更新算法框架:

1. 规则冲突检测
2. IF 无规则冲突 THEN
3. 更新规则库;
4. 更新 Rete 匹配网;
5. 更新情境规则关联表;
6. 修改订阅信息;
7. ELSE
8. 规则冲突消除策略;
9. IF 需要更新规则 THEN
10. CALL 规则更新.

## 4 在智能汽车空间的应用

本课题已开发了一个智能汽车空间<sup>[9]</sup>.我们把 ScudCORE 整合到 ScudSCAPE 软件平台中作为智能汽车空间<sup>[9]</sup>中的推理引擎(如图 7 所示),其中 ScudSCAPE<sup>[10]</sup>是一个专门面向智能汽车空间的情境感知的主动信息服务软件平台.整个系统利用部署在智能汽车空间中的各种情境采集设备,获取情境信息,经过 ScudCORE 推理,最后通过 ScudSCAPE 软件平台为用户适时地提供各种主动服务.

情境采集通过情境采集设备如传感器、车内摄像头、GPS 接收器、麦克风、手机、CAN 总线等获取,包括动态情境信息与静态情境信息,然后传送给情境管理器.情境管理器将之存入情境库中.当有满足 ScudCORE 订阅的情境信息事件发生时,情境管理器将借助智能汽车本体,将该情境信息转变为 ScudCORE 能理解的事实,然后发送给 ScudCORE,从而驱动 ScudCORE 进行推理.ScudCORE 会将推理结果返回给情境管理器:(1)推理结果为高级情境信息时,情境管理器将更新情境库,并根据订阅机制是否再次触发 ScudCORE;(2)若推理结果为触发行时,情境管理器将通知 ScudSCAPE 执行该行为.

一个典型的智能汽车空间应用场景如下:当司机

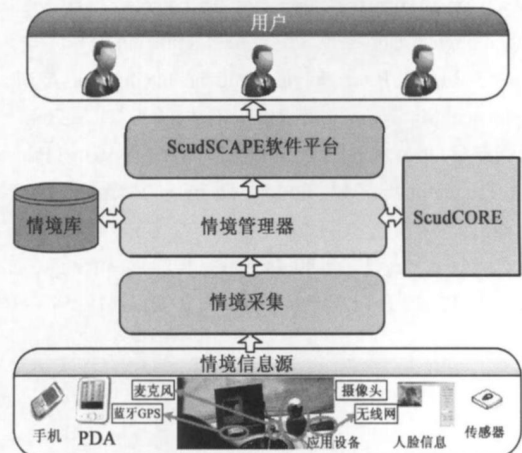


图7 在智能汽车空间中的应用

准备进入车内时,情境采集设备感知确认司机身份;自动为司机开启车门并激活车内软件平台功能;自动设置其个人偏好信息;为其自动调整好座位位置、后视镜角度及方向盘高低位置等.在司机驾驶过程中,如果通过 ScudSCAPE 软件平台接收到用户新邮件、手机短信、即时通信软件(QQ、MSN 等),会自动分析情境而选择语音阅读或屏幕显示的输出方式.若车内正在播放音乐,系统会更新情境,自动调节音乐音量,例如,当车内有人说话时,系统会自动调低音量;当系统通过空气质量传感器发现车内空气浑浊的时候,系统会选择合适的时间开窗换气.如果系统通过瞳孔分析发现司机正在疲劳驾驶,则会加大音乐播放的音量,并建议司机到休息区或下高速公路休息.

## 5 总结

为更高效地支持情境感知系统的开发,本文对普适计算环境下的推理引擎进行了研究,重点探讨了其中的情境驱动机制.提出了一个情境驱动的推理引擎体系结构 ScudCORE,它采用情境过滤机制弱化了情境时变快带来的推理过度频繁、效率低下问题;同时,利用本体建模技术实现了情境信息的共享、重用、理解功能;而对规则的动态管理机制,则使推理引擎更具灵活性,大大增强了情境驱动能力.

我们下一步的工作重点主要为:1)更好的规则冲突解决机制;2)推理过程的高效的模式匹配算法;3)支持情境时效性的推理机制等.

## 参考文献:

- [1] Dey A. K. Understanding and using context[J]. *Personal and Ubiquitous Computing*, 2001, 5(1): 4 - 7.
- [2] 吴朝晖,潘纲.普适计算,中国计算机科学技术发展报告 2005[M].北京:清华大学出版社,2006.  
Wu Zhaohui, Pan Gang. *Pervasive Computing. Annual Progress Report on Computer Science and Technology 2005* [M]. Beijing: Tsinghua University Press, 2006. (in Chinese)
- [3] Stuart J Russell, Peter Norvig. *Artificial Intelligence: A Modern Approach* [M]. New York: Prentice Hall, 2006. 240 - 268.
- [4] Joseph C Giarratano, Gary D Riley. *Expert Systems: Principles and Programming* [M]. Boston: Thomson Learning, 2005. 235 - 466.
- [5] A Gómez-Pérez, O Corcho. *Ontology languages for the semantic web* [J]. *IEEE Intelligent Systems*, 2002, 17(1): 54 - 60.

- [6] H Chen, F Perich, T Finin, A Joshi. *SOUPA: standard ontology for ubiquitous and pervasive applications* [A]. *Mobile and Ubiquitous Systems: Networking and Services* [C]. Boston: IEEE Press, 2004. 258 - 267.
- [7] Charles L Forgy. *Rete: a fast algorithm for the many pattern/multiple object pattern match problem* [J]. *Artificial Intelligence*, 1982, 19(1): 17 - 37.
- [8] Michael Negnevitsky. *Artificial Intelligence: a guide to intelligent systems* [M]. Boston: Addison Wesley, 2004. 32 - 37.
- [9] Gang Pan, Zhaohui Wu, Jie Sun. *Towards a smart space inside car* [A]. *The 9th International Conference on Ubiquitous Computing* [C]. Innsbruck: Springer, 2007. 16 - 19.
- [10] 罗威. *智能汽车空间情境感知的主动信息服务* [D]. 杭州: 浙江大学计算机学院, 2008.  
Wei Luo. *Context-aware information service in smart car space* [D]. Hangzhou: Zhejiang University, 2008. (in Chinese).

## 作者简介:



潘纲 男, 1976 年 10 月出生于浙江永嘉. 现为浙江大学计算机学院副教授, 研究方向为普适计算、计算机视觉与模式识别.  
E-mail: gpan@zju.edu.cn



李彤 女, 1986 年 10 月出生于浙江台州. 现为浙江大学计算机学院研究生, 研究方向为普适计算.  
E-mail: tongtonglee@zju.edu.cn



李石坚 男, 1979 年 12 月出生于湖南益阳. 现为浙江大学计算机学院讲师, 研究方向为普适计算、传感器网络.  
E-mail: shijianli@zju.edu.cn