

基于本体的智能空间情境信息模 型研究

Ontology-based Context Model for Smart Space

(申请清华大学工学硕士学位论文)

培 养 单 位 : 计算机科学与技术系
学 科 : 计算机科学与技术
研 究 生 : 秦 伟 俊
指 导 教 师 : 徐 光 佑 教 授
 史 元 春 教 授

二〇〇五年十二月

基于本体的智能空间情境信息模型研究

秦伟俊

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后遵守此规定）

作者签名： _____

导师签名： _____

日 期： _____

日 期： _____

摘 要

随着移动设备的广泛使用，用户可以随时随地进入或者离开智能空间，从而决定了智能空间环境是一个高度动态的、开放的、智能的分布式系统。为了让智能空间中用户透明地使用智能空间提供的各种计算服务，智能空间应该具备获取和处理环境和用户活动相关的情境信息的能力。研究表明只有对智能空间中不同实体进行形式化的描述，才能便于计算机系统的处理，因此建立情境信息模型，成为近年来智能空间研究的重要课题。本文是作者围绕智能空间应用使用本体描述的方法进行情境信息建模和情境推理的研究总结。

本文结合智能空间应用过程，分析面向智能空间应用的情境信息，并对情境信息进行分类。为了形式化描述智能空间中的不同情境信息实体，我们采用本体的方法对情境信息进行形式化建模和描述，以便于智能空间中知识的共享和复用。智能空间中存在多种多样的情境信息，根据情境信息类型的不同，我们提出基于本体的情境信息分层模型，将智能空间的实体本体分为核心分体和扩展本体。核心本体是由用户、服务、环境、活动和平台等基本实体构成，并阐明了各实体间的相互关系；扩展本体根据不同的应用领域，继承和扩展核心概念，设计领域特定的本体概念。

情境推理是指智能空间系统利用当前用户和环境资源的情境来获取智能空间中存在的高层语义。本文提出基于规则的情境推理机制，将情境推理分为本体推理和自定义规则推理。使用本体推理来解决本体使用过程中本体的冲突检测、表达优化、本体的融合和验证知识的正确性等问题。使用自定义规则推理则根据不同应用领域中用户自定义的情景规则对底层的情境信息进行高层抽象和映射。

本文设计的情境感知原型系统，利用本体描述的方法针对智能教室应用进行情境建模，采用规则描述语言来自定义智能导播应用中的推理规则，验证基于本体的智能空间情境感知技术的方法。

关键词：智能空间，情境感知，本体，情境建模，情境推理

Abstract

With the broad utilization of mobile devices, the users can enter or leave Smart Space freely at will. It's acknowledged that Smart Space environment is a highly dynamic, open, intelligent distributed system. In order to enable the users in the smart space to use the services provided by the Smart Space system, the system should have the capabilities to acquire and process the contextual information related to environment and user activities. Many researchers have indicated that various entities in Smart Space can be processed by computers only if the entities are represented in levels of formality. Therefore, it's one of significant research topics to build formal context model to describe various entities in Smart Space in recent years. In this paper, we are investigating an ontology-based approach to modeling and reasoning the contextual information in Smart Space.

With the experience of Smart Space application, we analyze and categorize various types of contextual information at first. In order to describe the entities in Smart Space, we investigate an ontology method to modeling and representing contexts for the purpose of knowledge sharing and reuse. We propose ontology-based context layered model, and divide context ontology into two sets: core ontology and extended ontologies. Core ontology consists of concepts and relations of fundamental contexts: User, Service, Environment, Activity and Platform. Extended ontologies inherit core ontology, and define vocabularies for domain-specific ontology concepts.

Context reasoning refers to the process that Smart Space system makes use of the contextual information about users and environment in order to acquire and extract high-level semantics. In this paper, we propose a rule-based context reasoning mechanism which includes ontology reasoning and user-defined rule reasoning. Ontology reasoning aims to solve ontology conflict detection, expression optimization, ontology composition and

knowledge validation. User-defined rule reasoning aims to facilitate high-level abstraction and mapping from low-level contextual information adapted to different domains.

With the help of context-aware prototype system, we apply ontology-based approach to build formal context model specific to Smart Classroom, and investigate to utilize rule description language to build user-defined inference rules for Smart Cameraman application.

Keywords: Smart Space, Context-awareness, Ontology, Context Modeling, Context Reasoning

目 录

第 1 章 引言	1
1.1 课题研究背景	1
1.2 问题的提出	6
1.2.1 情境与情境感知	7
1.2.2 情境感知的典型应用	8
1.2.3 情境感知研究面临的问题	9
1.2.4 情境感知研究的基本内容	9
1.2.5 相关研究分析	11
1.3 本文的研究范畴	14
1.3.1 情境信息表示与建模	14
1.3.2 情境信息解释与推理	14
1.4 本文的工作	15
1.4.1 基于本体论的情境信息模型	15
1.4.2 基于规则的情境推理机制	15
1.4.3 支持情境感知的智能空间支撑平台设计方案	16
1.5 论文的组织	16
第 2 章 基于本体的情境信息模型	17
2.1 智能空间中的情境信息分类	17
2.1.1 按照信息类型划分	17
2.1.2 按照信息产生方式划分	18
2.2 情境信息建模概述	19
2.2.1 情境信息建模的限制条件	19
2.2.2 情境信息模型的分类和比较分析	20
2.3 本体论及描述逻辑简介	23
2.3.1 本体论	23
2.3.2 描述逻辑	24
2.3.3 OWL 语言	26

2.4	基于本体的情境信息模型	27
2.4.1	COSS 模型核心本体	28
2.4.2	扩展本体	37
2.4.3	实体间的关系	38
2.5	本章小结	38
第 3 章	基于规则的情境推理机制	40
3.1	情境推理的定义	40
3.2	基于规则的情境推理的优点	42
3.3	基于规则的情境推理的任务	42
3.4	本体推理的分析和应用	43
3.4.1	本体语言的分析	43
3.4.2	本体推理的应用	43
3.5	自定义规则推理的分析	47
3.5.1	基于一阶谓词逻辑的情境表示方法	47
3.5.2	自定义规则推理方法	48
3.6	情境推理的设计与实现	50
3.6.1	情境推理的总体流程	50
3.6.2	情境推理的实现工具	51
3.7	本章小结	53
第 4 章	面向智能空间应用的情境感知系统	54
4.1	情境感知原型系统	54
4.1.1	应用需求	54
4.1.2	设计原则	55
4.1.3	原型系统框架	56
4.2	情境感知在智能教室中的应用示例	57
4.2.1	场景描述	57
4.2.2	实现步骤	59
4.3	本章小结	65
第 5 章	总结和展望	66
5.1	论文工作的总结	66

目 录

5.1.1	基于本体的情境信息模型	66
5.1.2	基于规则的情境推理机制	67
5.2	论文工作的贡献	68
5.3	需进一步开展的工作	69
参考文献		71
致 谢		75
个人简历、在学期间发表的学术论文与研究成果		77

第1章 引言

本文主要研究面向智能空间应用的基于本体论的情境信息模型，和基于规则的情境推理机制。本章首先简要介绍论文研究的背景——普适计算、智能空间和情境感知计算；然后介绍本文的研究对象——情境信息建模和情境信息推理，并介绍它们在情境感知计算研究中的地位以及研究现状；最后，本文概括工作的成果并给出正文的组织结构。

1.1 课题研究背景

随着计算机和通信技术的发展，计算和通信能力的性价比不断提高，移动设备、嵌入式设备等新型计算设备广泛应用，各种新形态的传感器设备，计算/联网设备的蓬勃发展；与此同时，人们希望计算机不只扮演科学计算工具的角色，而是能帮助提高人们日常工作生活质量的先进的电子日用品。因而，如何让人们能够随时随地、透明地享用丰富的计算资源和服务，是普适计算(Pervasive Computing 或者 Ubiquitous Computing) 研究的目标。

普适计算的概念是由已故的 Xerox 公司 Palo Alto 研究中心的首席技术管 Mark Weiser 于二十世纪九十年代初首先提出的，强调把计算机嵌入人们的工作生活空间中，将计算机技术融入人们的日常生活中，让人们的注意力回归到要完成的任务本身[1]。Weiser 在《The Coming Age of Calm Computing》[2]一文中指出计算科学发展 50 年来历经三代计算模式：主机计算模式 (Mainframe Computing Paradigm)、桌面计算模式 (Desktop Computing Paradigm) 和普适计算 (Pervasive/Ubiquitous Computing Paradigm)。从人与计算机的数量关系的角度来看，在开始的主机时代，人与计算机的关系是“多对一”的关系，多人共享一台计算机；在桌面计算时代，人与计算机的关系则是“一对一”的关系，个人计算机广泛使用；而在普适计算时代中，人与计算机的关系发生革命性的改变，变成“一对多”的关系，甚至是“多对多”的关系。计算机的使用者不再只是经过专业培训的人员，而是遍及普通大众；计算机不再仅限于桌面，也不仅是科学计算工具，它将以多种新形态 (例如手持设备、可穿戴设备等) 融入我们日常工作生活空间中。普适计算的思想吸引众多研究者投身于研究如何将

计算技术无缝地融合到日常的生活空间中。清华大学计算机系普适计算组从智能教室 (Smart Classroom) [11]研究入手研究普适计算的关键技术, 经过几年时间的研究, 我们对普适计算的理解如下: “普适计算是通过先进的计算、通信和数字媒体技术把相关的信息空间与物理空间融合成一个和谐丰富的人机交互环境或空间, 在此环境或空间中人们可以随时随地、透明地获取和使用所需要的信息和服务” [8]。

作为新的计算模式, 普适计算具有两个典型的本质特征: 物理集成 (Physical Integration) 与自发互操作 (Spontaneous Interoperation) [3]。物理集成的含义是信息空间的计算节点同物理空间的环境与设备的融合, 物理空间中的环境与设备是信息空间中计算节点自然的载体, 计算节点的计算和通讯能力透明地隐藏在环境与设备之中。例如在一个智能会议室中, 人们依旧采用原来的方式开会, 由于智能会议室中嵌入了计算、通信以及传感设备, 就能够觉察当前会议室中开会人员的状态和活动, 并能够通过会议室中的设备为用户提供服务, 例如使用手持个人数字助理 (PDA, Personal Digital Assistant) 控制投影仪的投放、白板记录用户板书内容等服务。自发互操作的含义是普适环境中存在的计算实体 (例如计算设备、服务、程序和 Agent 等) 需要具备在高度动态的环境中自发地完成与其他计算实体和环境之间的交互, 尽可能少的甚至不需要用户的干预。还是以上述智能会议室为例, 一位讲演者手持 PDA 进入智能会议室中, 无需任何手动配置, 利用智能会议室中的投影仪投放 PPT, 其中设备 (PDA、投影仪) 和环境 (智能会议室) 之间的交互是自发进行的, 无需用户的手动配置操作。

智能空间 (Smart Space) 系统是普适计算的典型应用实例, 按照美国国家标准和技术学会 (NIST) 给出的定义, 智能空间是 “一个嵌入了计算、信息设备和传感器的工作空间, 其目的是使用户能够非常方便地在其中访问信息和获取计算机的服务来高效地进行单独工作与其他人的协作” [4]。智能空间技术在普适计算研究中占有重要的地位, 被普遍认为是研究普适计算理想的实验床, 能够集中体现普适计算本质特征及其关键技术。上述例子中的智能会议室就可以认为是典型的智能空间实例。

目前已经有很多学术界和工业界的研究者开展了关于智能空间的研究计划并各有特色, 比较著名的研究计划有: MIT 的 CSAIL 实验室提出的 AIRE (Agent-based Interactive Reactive Environment) 计划[5]提出基于 Agent 的软件平台设计方式, 设计的原型系统称为 Intelligent Room; Stanford 大学的 Interactive

Workspace 计划[10]致力于研究用户如何同存在丰富计算资源的工作空间进行交互,解决不同尺寸的屏幕之间切换的问题,设计的原型系统称为 iRoom; Microsoft 公司的 Easyliving 计划[9]着眼于研究在充满大量交互设备的智能环境中用户体验的问题。

清华大学普适计算组则以智能教室[11]作为典型的智能空间实例,重点研究智能空间中软件基础平台 Smart Platform[6][7]的研究,并以此从软件平台设计的角度上总结智能空间发展的三个阶段[7],如图 1.1 所示。

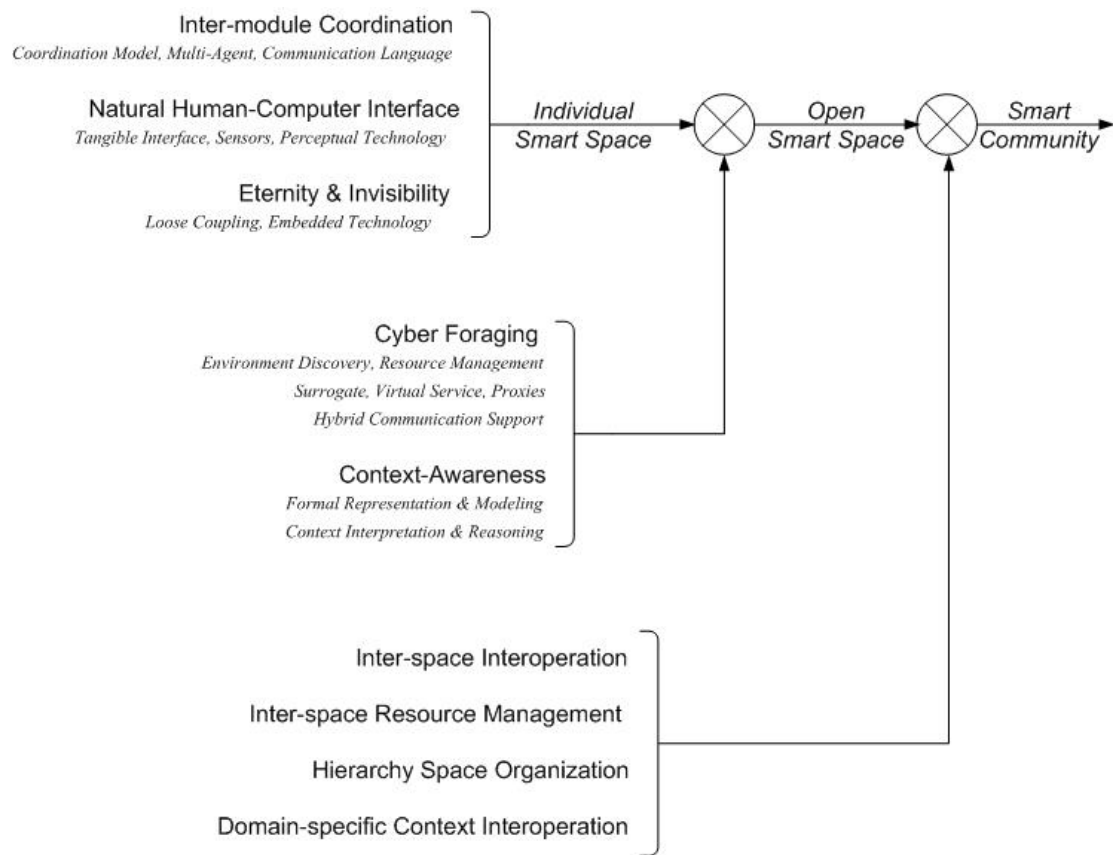


图 1.1 智能空间发展的三个阶段

智能空间研究的第一个阶段是单个智能空间 (Individual Smart Space)。单个智能空间阶段的研究主要集中在空间内模块的通讯与协调形式,自然的人机交互接口,同时要求系统具有不间断性 (Eternity) 和不可见性 (Invisibility) 等特性。为了支持模块间的协调,研究主要集中在智能空间软件支撑平台的协调模型、多 Agent 系统以及通讯机制等;为了支持自然的人机交互方式,研究主要集中在通过嵌入大量可触摸接口 (Tangible Interface)、传感器以及基于识别技术的

感知模块等多模态接口来获取友好的人机交互能力；松散耦合（Loose Coupling）的应用使系统的鲁棒性和不间断性，不会因为系统的升级、调试等原因停止；嵌入式技术的应用使得具有计算能力的设备（例如智能灰尘[12]）能够嵌入到物理空间中，对使用者来说是不可见的。

智能空间研究的第二个阶段是开放的智能空间（Open Smart Space）。开放的智能空间具备两个特点：一个特点是智能空间的开放性或者动态性，主要体现在随着移动设备的广泛使用，用户可以随时随地进入或者离开智能空间，用户所持的移动设备随着用户在空间中漫游，能够自动发现环境的存在，并自发地与环境进行交互，如图 1.2 所示。需要解决的研究问题可以用“游牧服务（Cyber Foraging）”来总结，主要包括环境的发现、异构通讯信道的互通、资源管理、代理（Surrogate）[13]以及自适应等技术的应用来，支持移动用户随时随地访问智能空间，透明地使用智能空间里的计算资源和服务。另一个特点是感知能力（Awareness），为了让用户透明地使用智能空间中的服务，智能空间应该具备主动地获取和处理环境和用户活动相关情境信息（Context）的能力。该方面的研究主要包括情境信息的表示与建模、情境信息的解释与推理等研究内容。

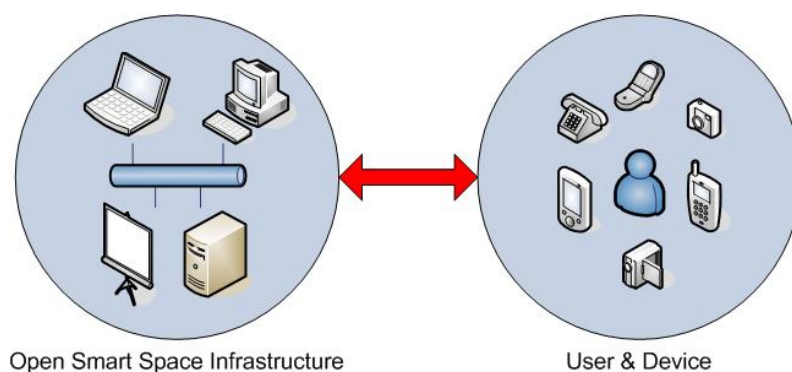


图 1.2 开放的智能空间

智能空间研究的第三个阶段是智能社区（Smart Community）。有学者指出，将来不太可能出现全球统一的智能空间，而是存在许多因为管理区划、地域区划和文化区划而有明确的边界划分的智能空间系统[3]。我们认为将来的智能空间系统具有一定层次拓扑结构。空间与空间之间互连，若干个开放的智能空间能够联合起来构成更大的智能空间（例如 FIT 大楼 3 区实验室构成整个智能媒体所），这样的空间能够继续构成更为复杂的空间（例如所有 FIT 大楼里各个单位构成 FIT 信息智能空间），最终智能空间将以智能社区的形式组织在一起。如

图 1.3 所示是智能社区的结构示意图，智能空间是嵌入计算、信息设备和多模态传感器的工作空间，只是逻辑意义上的概念，并不特指某种实物。通过网络和通讯技术，智能空间之间实现通讯互连，而且用户和设备在不同的智能空间中能够自由进出。我们认为智能社区阶段的研究将着重在如何进行跨空间的交互，如何进行跨空间的资源访问，如何共享智能空间中的情境信息和情境知识的映射。

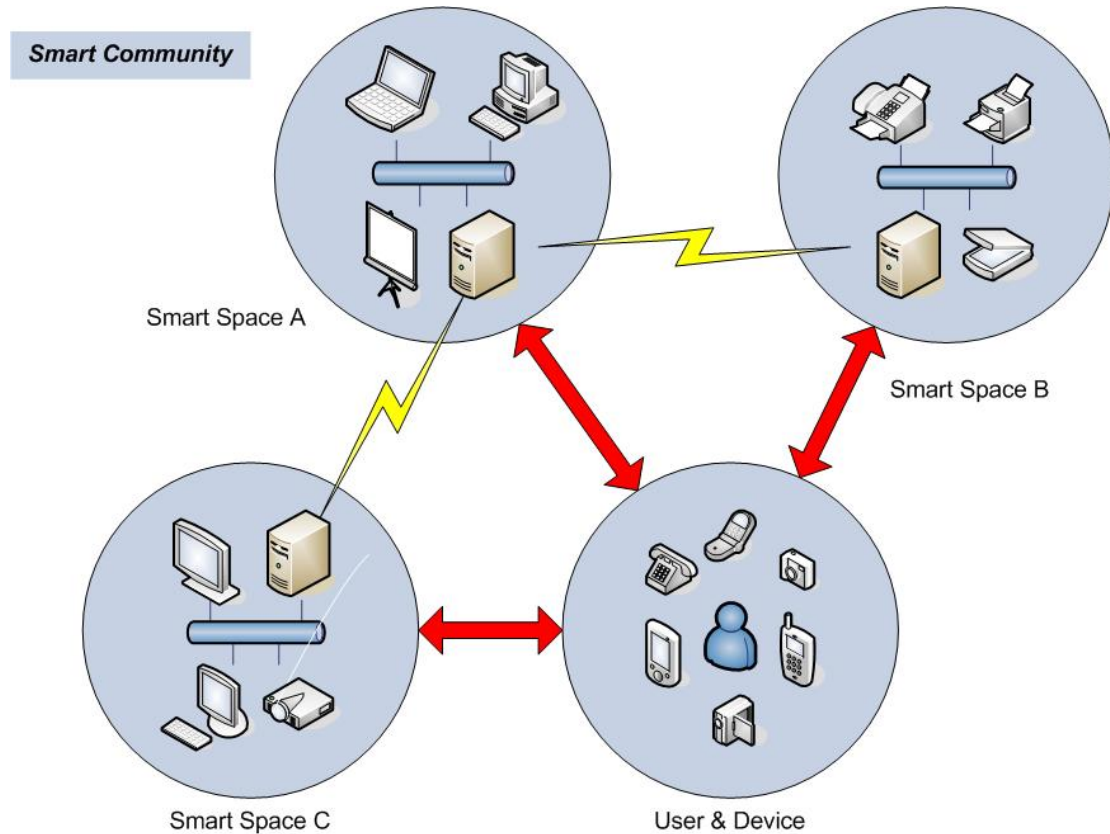


图 1.3 智能社区的结构示意图

智能空间发展的三个阶段的顺序是逻辑关系而非时间关系，随着智能空间研究的发展，从左至右新的研究问题不断引入，每个阶段的研究重点各有不同，但并不意味前一阶段的研究问题在下一阶段中不重要，我们只是假设前一阶段的问题已经有很好的解决方案的基础上开展下一阶段的研究。就目前的研究而言，多数项目已经对第一阶段的研究问题有很深入的研究，研究重点放在第二阶段的研究上。

1.2 问题的提出

我们认为普适计算的目标是在计算和通信无所不在的条件下建立“以人为中心”的环境[8]。实现普适计算“以人为中心”的目标需要计算机科学研究多方面的支持，本文主要针对智能空间的第二个研究阶段的支撑技术——情境感知技术。

随着计算机应用不断发展，虽然绝大数的计算机不再需要我们精心地维护，但是在传统的计算机系统中计算机还是需要专业人员的耐心伺候：一方面，从人机交互方式的角度来看，人与计算机交互的方式还是采用传统的键盘、鼠标方式，需要经过专业化的培训，使用者才能使用计算机；另一方面，从使用者的角度来看，使用者在环境中面对多种相对独立的设备，花费大量的精力管理和控制计算机。在普适计算时代，智能空间的目标是建立嵌入大量的计算、信息设备和传感器的智能环境，通过觉察用户的意图来主动地向用户提供随时随地透明的伺候式服务。我们认为智能性是智能空间的本质特征，因为在智能空间中，用户所处的环境中存在大量的计算设备和服务；为了让用户在智能空间中专注于本身的工作任务，智能空间应该尽量减少用户参与系统配置、控制管理计算设备和服务的精力和时间，这就要求智能空间系统具备一定的智能化行为和决策能力。

为什么在传统的计算机系统中人机交互存在巨大的障碍呢？原因肯定有诸多方面的，我们拿人机交互方式与人和人沟通方式进行比较。众所周知，人与人的交流和沟通是很和谐便利的，有诸多因素辅助人与人之间的交流，例如人与人交流时所采用的语言具有很丰富的表达能力，人与人交流时有相同的背景信息的理解，也就是说人与人的交流过程中对情境信息（Context）有共同的理解。举例来说，两个人从同一扇门进入会议室，前进门的人对后者说“请关门”。由于两者都具有共同的背景信息的理解能力，前者则不必指明自己想要关闭的大门来阐明自己的意图。由此可见，人机之间的交互存在很大的局限性，正是由于一方面缺乏具有丰富表达能力的语言进行沟通，另一方面人与计算机之间缺乏共同的背景信息的理解。

情境感知（Context-awareness）是智能系统智能性的外在表现，情境感知的研究实际上是对人类智能的探索和研究[14]。计算机科学中许多学科开展对情境感知的研究，例如普适计算（Ubiquitous Computing）[20][22]、人机交互（Human-Computer Interaction）[15]、人工智能（Artificial Intelligence）[16]、信

息检索 (Information Retrieval) [17]、游牧计算 (Nomadic Computing) [18]和传感器网络 (Sensor Networks) [19]等等。不同的学科研究情境感知的出发点和角度各有不同,有些学科研究情境感的理论基础,有些学科则研究情境感的工程性应用,但是普遍的共识是认为情境感知是智能信息系统的重要特征。本文是从智能空间角度入手,研究面向智能空间应用的情境感知技术。

1.2.1 情境与情境感知

研究智能空间中的情境感知,建立情境感知系统,首先需要理解情境 (Context) 的定义,因为只有充分地理解自己研究范畴中情境的含义,才能使设计人员决定选择系统中的哪类数据或者信息作为有效的情境信息,采用何种抽象模型来设计情境感知系统。

从词义上说,按照美国《大学词典》(Merriam Webber's College Dictionary 1989)的解释,Context (本文译为情境,也有学者译为上下文)是指“1)上下文文章或叙述中围绕着某个特定词或段落并且决定其含义的部分;2)某物存在或者某事发生的相关背景和条件”。

在普适计算研究领域,研究者认为上述定义不足以描述计算机系统中情境的含义[21],分别提出对情境 (Context) 的理解。Schilit 等人认为情境是情境感知系统中与用户描述相关的信息,给出情境的定义是“在移动分布式计算系统中,情境是指用户的位置 (location)、身份 (Identity)、用户身边的物理对象 (Physical Object) 以及同用户交互的设备的状态 (State)” [23]。Schilit 针对情境的定义在 ParcTab 项目[23]中得到使用,从 Schilit 的定义中可以把情境信息分成下列环境相关的信息[20][22]:

- 1) 计算环境 (Computing Environment), 包括计算机处理器的运算能力、用户输入和显示的设备、网络设施等;
- 2) 用户环境 (User Environment), 包括用户位置、用户周围的人、所属的社会关系等;
- 3) 物理环境 (Physical Environment), 包括环境的光照情况、噪声水平等。

Dey 和 Abowd 等人认为情境的定义不仅仅指用于描述用户或者系统的信息,排除其他类型的情境信息,例如用户的意图 (User Intent)、用户信息 (User Profile) 等,从而提出被广泛认可的更全面概括的情境的定义“情境是任何可以用来表现实体状态的信息。实体是与用户和应用之间交互相关的人、地点或者对象,

包括用户与应用本身” [20][22]。同时, Dey 和 Abowd 等人提出对情境感知的理解, 认为情境感知是指“系统具备依据用户任务的需求, 使用情境向用户提供相关信息和服务的能力” [20][22]。

1.2.2 情境感知的典型应用

我们认为情境感知是智能空间的典型特征和基础服务, 为了直观地说明情境感知, 下面列举几个情境感知在智能教室和智能会议室使用的典型应用:

■ 智能导播

智能教室是清华大学普适计算组设计和开发的应用于远程教育的智能空间系统[11]。在智能教室中, 需要解决的重要研究问题是增强远程学生参与的现场感, 智能教室采用远程音视频的方式将现场教学活动通过 Internet 传送到远程学生, 那么现场教学活动焦点的选取直接影响到远程学生学习的质量和效果。智能教室系统中的智能导播模块使用智能教室中情境(例如教师板书、现场学生发言等), 通过选取教室中预先布置的多个摄像头的视角(例如教师近景视角、教室远景视角等), 向远程学生反映现场教学活动的焦点, 增强远程学生的现场感。

■ 智能会议助手

智能会议室是智能空间应用的典型范例, 智能会议助手作为智能会议室中情境感知模块, 为与会的讲演者提供全过程、透明的服务:

- 当讲演者进入会议室时, 智能会议助手根据会议室的日程安排推理出目前的活动是该讲演者准备做报告。智能会议助手根据讲演者携带的个人信息 (User Profile) 中获得讲演者准备的讲演稿 (例如讲演者的个人主页), 并下载到本地。
- 当讲演者使用投影仪播放讲演稿时, 智能会议助手根据会议室的光照条件发现投影的亮度太高, 就自动地调节会议室的照明状况。
- 智能会议助手根据本次讲演的信息, 发现某些对本次报告感兴趣的人员因故缺席。智能会议助手就会根据讲演稿的涉密级别, 分别通过缺席人员的 Email 联系方式发送讲演稿的拷贝。
- 讲演结束后, 智能会议助手根据讲演的时间和预定时间来安排提问时间。同时, 智能会议助手根据讲演者的个人日程安排信息, 安排会后的活动, 例如讲演者会后需要赶往机场, 智能会议助手就预定出租车。

1.2.3 情境感知研究面临的问题

情境感知的研究自从 1992 年 Xerox Parc 项目的研究工作开始已有十余年的时间，目前成为普适计算研究的热点和重点，但是情境感知技术尚未达到非常成熟的阶段，研究仍然面临着许多的困难：

1) 情境信息源的异构性

情境信息的获取一般来自于普适环境中的硬件传感器和软件服务等，情境信息的来源存在异构性的特点，例如在智能教室环境中用户身份（User Identity）可能是从定位系统等硬件设备读取的，也可能是从人脸识别软件模块识别出来的结果。即便是定位系统来说既可能是从 Active Badge[24]得到，也有可能是从 RFID 传感器得到。因而，情境信息源的异构性使得情境感知系统很难具有通用性。

2) 缺乏软件支撑平台的支持

从系统模块的重用性（Reusability）和互操作性（Interoperability）的角度来说，情境感知服务需要智能空间系统软件平台的支持，以往的情境感知系统都缺乏通用的软件支撑平台来提供高层的抽象模型和重用模块。设计通用的情境感知软件支撑平台的重要意义是便于开发人员快速开发可重用的情境感知系统模块或原型系统。

3) 缺乏通用的情境模型

目前情境信息模型一般是根据特定应用设计的，缺乏通用的情境模型，不便于异构模块之间情境知识共享（Knowledge Sharing）、知识的重用（Knowledge Reuse）和情境知识逻辑推理（Logic Inference）等[25]。

4) 缺乏安全和隐私机制

用户的个人隐私和系统的安全性都是系统设计中必须考虑的因素，由于情境信息的高度动态性使得情境感知系统中安全机制和隐私维护的解决方案更加趋于复杂。

1.2.4 情境感知研究的基本内容

情境感知研究的内容涉及计算机科学的诸多方向，为了便于将情境感知研究进行系统性的分类，总结情境感知研究的基本内容和研究方向：

1) 情境信息获取（Context Acquisition）

情境信息获取是情境感知计算研究中最底层的研究点。情境信息获取是从多

样的情境信息源中获取情境数据的重要机制，一般而言，情境信息的获取和硬件传感器是紧密耦合的。同时，将情境信息获取过程独立出来也便于情境信息的双重用。

2) 情境模型与表示 (Context Model and Representation)

情境信息的模型是情境信息表示和高层情境推理的基础，目前很多情境感知系统都提出自己的情境信息模型，情景信息模型的类型选取主要是根据应用需求的特点和选取的情境数据的类型和表示形式决定的。经过几年的研究过程，研究者在设计情境模型上遇到一些问题，比如如何在情境感知系统中实现知识的共享、语义的互操作和模块的重用等问题，因而建立通用的情境信息模型成为研究的热点[20][22][26]。

3) 情境信息聚合 (Context Aggregation)

情境信息是从多种信息源获取的，情景信息聚合的研究集中在如何将多种情境信息源里的数据收集和存储起来。其中的一种实现方案就是设计集中式的情境信息聚合机制来提供持久性的存储、情境信息的集成和管理以及情境信息的共享和读取。基于统一的情境数据模型，情境信息聚合还能为上层的情境信息解释提供数据关联和查询服务。

4) 情境信息查询 (Context Query)

作为情境感知服务的重要组成部分，情境信息查询提供访问情境数据的接口，由于情境数据的异构性和存储的分布性，情境信息查询的研究主要集中在情境查询语言、事件通知和查询优化等方面。

5) 情境信息解释 (Context Interpretation)

底层的情境信息(例如 RFID 传感器的数据)很难被上层的应用程序所理解，情境信息解释的作用就是将底层的情境信息解释为高层的情境信息，其中研究的方法主要是利用机器学习和知识推理等人工智能方法，目前的研究主要集中在 Description Logic[27]、First-Order Logic[28]、Bayesian Networks 等方法。

6) 情境信息发现 (Context Discovery)

在情境感知系统中，情境感知将作为系统中的服务给用户或者软件模块使用，因而情境感知系统中必须具备情境信息发现机制，其研究主要集中在情境服务的描述、发布和事件订阅机制等[29]。

1.2.5 相关研究分析

目前国内外开展情境感知的研究项目很多，其中比较有代表性的研究项目有美国 GaTech 的 Context Toolkit 项目[20][22]、美国 UC Berkeley 大学的 Context Fabric 项目[30]、美国 UIUC 大学的 Gaia 项目[31]和新加坡国立大学的 Semantic Space 项目[26]。

■ Context Toolkit

GaTech 的 Context Toolkit[20][22]开发了一套针对传感器数据处理的情境抽象模型，以便辅助情境感知应用的设计和重用。Context Toolkit 的设计将底层的情境数据的获取和高层的情境应用，设计 Context Toolkit 中间件，其功能是收集传感器的原始数据信息，将原始数据转换成应用程序能够理解和处理的数据格式，并分发给应用程序使用。在 Context Toolkit 中，Context Widget 模块提供情境信息的访问接口，隐藏应用程序访问异构的传感器数据带来的计算复杂度上的问题，同时也使得传感器数据的重用变得更加便捷。在 Context Toolkit 中，为了能够处理情境信息查询和事件通知等服务，每一个 Context Widget 都状态描述，该描述保存 Context Widget 的各种属性和事件触发调用的函数接口。Context Widget 从传感器获取原始数据，并将数据传给 Context Interpreter 模块或者 Context Aggregator 模块进行数据抽象或者融合，其中分布式通讯采用 HTTP 协议进行通信，采用 XML 语言对消息进行格式编码。

■ Context Fabric

Context Fabric[30]是 ParcTab 项目的后续项目，Context Fabric 项目中设计的情境感知软件基础平台提供两个基本内置服务：事件服务和查询服务，提供情境数据的获取和检索。Context Fabric 采用实体-关系型 (Entity-relation) 逻辑情境数据模型来表示四种基本概率：实体 (Entity)、属性 (Attribute)、关系 (Relationship) 和集合体 (Aggregate)。在 Context Fabric 中采用 XML 语言对情境数据进行编码存储在本地文件系统中，使用 XPATH 来作为 XML 树型结构的查询语言。

■ Gaia

Gaia 项目中情境感知研究[28]借鉴 Context Toolkit 的设计思想，情境模型是基于—阶逻辑 (First-order Logic) 和布尔代数 (Boolean Algebra) 的概率预测模型来对情境信息 (Context Information) 进行推理和决策，并且利用 Ontology 来表示和定义模型中情境谓词 (Context Predicate) 的结构和语义，其中基本元素

是事实 (Fact) 和规则 (Rule)。Gaia 也使用概率逻辑 (Probabilistic Logic)、模糊逻辑 (Fuzzy Logic) 或贝叶斯网 (Bayesian Networks) 等模型来进行情境信息不确定性推理[32]。Gaia 中情境服务的软件基础平台包括情境供应模块 (Context Provider)、情境合成模块 (Context Synthesizer)、情境使用模块 (Context Consumer)、情境供应查找服务 (Context Provider Lookup Service) 和 Ontology 服务器 (Ontology Server) 等模块。Gaia 中将应用程序 (Application)、设备 (Device) 和服务 (Service) 都认为是实体 (Entity)。每个实体都关联一个知识库 (Knowledge Base) 来断言和认证实体相关的事实和规则。知识库以谓词多元组的方式来存储事实, 也包括其他服务的信息。当情境供应模块发现当前的情境发生变化时, 该模块就通过该模块关联的事件通道向知识库发送事件更新信息, 从而更新知识库中的内容。当实体需要使用情境信息时, 实体通过固定的 API 向推理引擎 (Reasoning Engine) 询问它所需要的情境信息或者推理结果。推理引擎中采用插件方式来集成推理模型, 例如概率逻辑模型、贝叶斯网模型等。

■ Semantic Space

Semantic Space[26]是新加坡 NUS 结合 Semantic Web 技术来解决智能空间中情景信息的语义表示、查询和推理等问题。在 Semantic Space 项目中, 与 Context Toolkit 很类似, Semantic Space 的情境软件基础平台包括 Context Wrappers、Context Aggregator、Context Knowledge Base、Context Query Engine 和 Context Reasoner 等模块构成。Context Wrapper 获取硬件传感器或者软件服务中的数据, 并对数据进行情境标注 (Context Markup); Context Aggregator 发现和收集情境标注, 实现上采用 UPnP Control Point 来解决 Context Wrapper 的发现和情境事件的订阅问题; Context Knowledge Base 用于存储情境标注数据和扩展的特定应用的情境 Ontology 数据, 并为 Context Query Engine 提供访问接口; Context Query Engine 选用 RDF Data Query Language 作为情境查询语言, RDQL 支持基于三元组 (<subject, predicate, object>) 语义模型上的查询方式; Context Reasoner 利用启发式规则对情境信息进行抽象和推理, 其中需要解决的一个问题是避免情境信息发生冲突的问题, 实现上采用 Jena 2[47]的通用规则引擎来对 CKB 中情境信息的推理。

针对情境感知研究的基本内容, 表 1.1 对列出的研究项目进行分类。

表 1.1 情境感知系统的比较

	Context Toolkit	Context Fabric	Gaia	Semantic Space
Context Acquisition	Components decoupling low-level sensing	Similar to Context Toolkit	Similar to Context Toolkit	Similar to Context Toolkit
Context Model	Name-Value pairs in XML	Entity-relation model in XML	Ontology in DAML+OIL	Ontology in OWL
Context Aggregation	Relational Database	Local storage of XML files	N/A	N/A
Context Interpretation	Ad hoc	Ad hoc	Rule based reasoning using XSB	Description Logic using Jena 2
Context Query	Simple matching	XPATH based XML matching	N/A	RDF Data Query Language
Context Discovery	Simple private protocol	N/A	CORBA naming/trading service	UPnP
Context Delivery	HTTP based private protocol	Private protocol	CORBA IDL	Jena 2
Developing Platform	N/A	N/A	CORBA	Jena 2

1.3 本文的研究范畴

情境感知研究内容涉及面广泛，例如情境信息获取、情境建模与表示、情境信息聚合、情境信息查询、情境信息推理与解释和情境服务发现等方面。本文主要针对情境感知研究的重点：情境信息的表示与建模（Context Modeling and Representation）、情境信息的解释与推理（Context Interpretation and Reasoning）。

1.3.1 情境信息表示与建模

智能空间中的情境信息多种多样，例如时空关系、位置、用户活动、计算实体的状态等。情境感知重要研究课题是如何有效地使用情境信息，首先要解决的问题是在计算机中如何合适地表示情境信息，适合于机器的处理。在情境感知研究初期的系统中，例如 Context Toolkit，情境信息一般使用面向对象语言（例如 Java 或 C++）进行对象的封装，这种描述方法的关键问题是情境信息虽然具有很好的封装性，但缺乏扩展性。后续的研究项目，例如 Context Fabric 项目，则是采用元语言（meta-language），例如 XML，来结构化地描述情境信息，这类描述方法虽然具有一定的扩展性，但是由于限于描述语言的能力的原因，这类描述方法在表示和互操作性上只具有语法层的描述能力。目前研究的情境感知系统中，例如 Semantic Space，则是采用 OWL 语言（Web Ontology Language）进行描述情境信息，采用这类描述方法的重要目的在于能从语义层对情境信息进行表示。

1.3.2 情境信息解释与推理

情境信息解释与推理是情境感知研究富有挑战性的研究内容。从情境感知系统设计的角度来说，情境信息解释与推理的作用体现在两个方面[33]：在数据获取（Data Sensing）的过程中，情境信息解释与推理的作用是将底层的传感器数据信息解释或者推理得到高层的情境信息，例如将定位系统得到的底层位置坐标数据（<26, 87, 29>）解释为高层的情境信息（<AT FIT 3-526>）。在情境知识共享（Knowledge Sharing）的过程中，情境信息解释与推理的作用是融合多样的情境信息，检查情境信息的一致性，从而保证情境信息的互操作性。

在目前已有的情境感知系统中，情境信息的解释与推理有两种研究思路：一种思路是在情境信息的解释与推理过程中采用本体论和逻辑推理的方法来实现情境信息的映射和共享，另一种思路是采用概率统计的方法进行推理。前者研

究利用基于规则的逻辑推理方法来推理利用本体论语言（例如 OWL 语言）描述的情境信息；后者研究各种不同的贝叶斯网络来应用于情境信息概率模型[6]。

1.4 本文的工作

针对上面介绍的情境感知研究的重要内容，本文展开了相应的研究，并有如下的研究结果。

1.4.1 基于本体论的情境信息模型

我们借鉴已有的情境信息模型，结合智能空间应用的特点以及普适计算组已有的研究结果，提出基于本体论的情境信息模型。

- 1) 分析了智能空间中情境信息的分类，情境信息建模的限制条件以及情境信息模型的分类，指出基于本体论的情境信息模型在知识表示、逻辑推理、知识共享等方面体现出的特点和优势。
- 2) 设计基于本体论的情境信息模型。我们利用本体论的方法来针对智能空间中的不同实体概念（包括用户、设备、环境以及服务等）进行形式化建模，使得能够形式化地描述和表示智能空间存在的情境信息。同时，本体的使用使得不同实体对特殊领域（Domain）中的各种概念有共同的本体认识，有利于情境信息的互操作和一致性，从而智能空间中不同实体之间的交互更加便利。
- 3) 利用本体描述语言（例如 OWL 语言）来描述和实例化智能空间中不同实体概念，便于情境感知系统处理和使用语义化描述的情境。

1.4.2 基于规则的情境推理机制

我们基于利用本体描述的情境信息表示模型，结合近年来人工智能中逻辑推理和本体论方面的研究工作，研究基于规则的情境推理机制。

- 1) 分析了情境信息推理在情境感知系统中所起到的作用和情境信息推理中基于规则的逻辑推理方法。
- 2) 结合描述逻辑（Description Logics）的研究工作，研究了面向智能空间应用的基于描述逻辑的情境推理的机制。

基于规则的逻辑推理的应用，使得我们从逻辑推理的角度形式化地研究情境推理的机制，有利于情境信息的知识共享和互操作。

1.4.3 支持情境感知的智能空间支撑平台设计方案

我们结合普适计算组已有的智能空间支撑平台 Smart Platform 研究的基础上，提出支持情境感知的智能空间支撑平台设计体系结构。在智能空间支撑平台的基础上，借鉴 Semantic Space 项目中 Context Stack 分层模型的做法，设计智能空间支撑平台中情境感知系统框架。

1.5 论文的组织

本文后续章节按照下列方式展开。

第二章首先介绍了情境信息模型的限制条件和分类，然后介绍本体论与 Semantic Web 技术的基础知识，通过本体论的方法来说明用以描述智能空间中不同实体概念和关系的情境信息模型。

第三章首先介绍情境信息推理的基础知识，然后说明基于规则的情境信息推理的方法和作用。

第四章介绍了结合已有的智能空间支撑平台 Smart Platform 的研究工作，设计面向智能空间应用的基于本体论的情境感知的研究方案。

最后一章总结本文的研究结果以及将来需要进一步展开的工作。

第2章 基于本体的情境信息模型

在前一章我们已经讨论过，随着移动设备的广泛使用，用户可以随时随地进入或者离开智能空间，从而决定了智能空间环境是一个高度动态的、开放的、智能的分布式系统。为了让智能空间中用户透明地使用智能空间中的服务，智能空间应该具备获取和处理环境和用户活动相关的情境信息的能力。同时只有对智能空间中不同实体进行形式化的描述，才能便于计算机系统的处理，因而形式化描述智能空间中的实体，建立情境信息模型，成为近年来智能空间研究的重要课题。“本体是共享的概念模型的形式化的规范说明” [34]是在语义层上描述智能空间中不同实体概念和关系的有力的工具。

本章先分析智能空间中的情境信息的类别，从而引入智能空间需要描述的实体类型，然后介绍和分析现有的不同类型的情境信息模型，并总结智能空间中情境信息的类别，随后介绍必要的预备知识——本体论和描述逻辑的基础理论知识，最后采用本体论的方法对智能空间中的不同实体进行形式化的描述，并给出实体之间的逻辑关系。

2.1 智能空间中的情境信息分类

智能空间环境是一个高度动态、开放的、智能的普适计算环境，情境感知是智能空间系统的典型特点，在前一章中我们已经给出研究者普遍接受的情境的定义“情境是任何可以用来表现实体状态的信息。实体是与用户和应用之间交互相关的人、地点或者对象，包括用户与应用本身” [20][22]。但是这个定义在实际应用过程中太抽象笼统，为了便于情境感知的研究，许多研究者都对情境信息针对不同的应用领域（例如人机交互领域、移动计算领域等）背景进行分类[20][21][22]，我们从智能空间应用出发通过不同的角度来阐明情境感知应用中所使用的不同情境，从而说明智能空间中存在的不同实体类型。

2.1.1 按照信息类型划分

针对智能空间中存在的情境信息的类型（从某种程度上说是按照情境信息的来源和类别）进行情境信息划分：

- 1) 用户的情境信息(Person): 用户本身和与用户相关的信息, 包括用户的 Profile 信息、用户偏好 (Preference)、用户所属的社会关系 (Social situation)、用户任务 (Task) 以及与用户交互的人等信息, 其中用户的 Profile 信息包含用户的身份信息和联系方式等, 例如用户姓名、性别、年龄、生日, 还有用户的 Email、手机号码、通讯地址等。用户偏好因人而异, 例如智能家庭中家居光照条件的喜好, 有人喜欢明亮的室内光照条件, 有人则喜欢柔和暗淡的光照条件。用户所属的社会关系用于描述不同情况下用户不同的身份以及人与人之间的关系, 例如在智能家庭中, 空间中的人是房间的主人、朋友还是陌生人是相关的情境, 在智能会议室中, 出席的人是会议室的工作人员还是列席嘉宾是相关的情境。
- 2) 时间的情境信息 (Time): 是上午还是下午, 是工作日还是休息日, 是春季还是秋季。时间的情境信息所描述的维度有所不同, 是指时间点 (例如 2005 年 10 月 16 日 16 时 54 分 10 秒), 还是指时间段 (例如下午 2 点至 4 点)。
- 3) 空间的情境信息 (Space): 比如人和物体的绝对位置、相对位置 (是否在一起、是否在可见范围内、是否在同一房间内)、人的朝向等等。空间的情境信息所描述的层次也不同, 是指物体的空间坐标 (例如三维坐标 $\langle 15m, 29m, 0.5m \rangle$), 还是地理位置 (例如 FIT 大楼 3 区 527 室)。
- 4) 设备的情境信息 (Device): 设备的性能参数以及计算和通讯能力, 比如显示设备的显示屏幕尺寸和分辨率, 计算设备的 CPU 和内存的性能, 网络设备的网络带宽。
- 5) 物理环境的情境信息 (Physical Environment): 智能空间的物理条件, 例如光照亮度、噪声级别、天气、温度、湿度等。
- 6) 活动的情境信息 (Activity): 房间是会议室还是教室、房间中是否进行会议讨论等。

2.1.2 按照信息产生方式划分

从情境信息获取的角度来看, 根据情境信息产生的方式有所不同进行划分:

- 1) 预设的情境信息 (Profiled Context): 这类情境信息相对稳定, 可以通过系统开发者或使用者进行预先配置和输入设定, 或者系统能够预先自动获取, 例如预先输入的用户 Profile 信息, 房间的空间关系等, 又如系统可以通过下载会议室的时间表, 来预先获知房间正在进行的活动内容 (例如做学术报告)。

- 2) 实时的情境信息 (Real-time Sensed Context): 这类情境信息变化比较频繁, 通常是从传感器或者传感器网络获得的实时信息, 例如当前用户的位置和朝向, 当前的天气状况, 室内房间的温度和湿度等。
- 3) 派生的情境信息 (Derived Context): 这类情境信息是根据将多种信息综合派生起来, 比如通过查询用户使用的 Microsoft Outlook 中保存的工作时间表, 得到用户当前工作安排是项目例会, 另外从 GPS 定位系统得到用户当前的位置在会议室, 那么判断出用户当前的活动是开会中, 从而派生出新的情境信息。

2.2 情境信息建模概述

2.2.1 情境信息建模的限制条件

情境感知研究结果表明情境信息模型直接影响情境感知系统的设计和运行效率。在讨论不同的情境模型之前, 我们首先研究情境感知研究中情境信息建模过程中需要满足的六点限制条件[35]。

R.1 分布式合成 (Distributed Composition)

由于普适计算系统 (包括智能空间系统) 在软件体系结构上一般采用分布式系统的设计方案, 同时在分布式系统结构中, 系统缺乏中心控制逻辑单元来负责生成、发布、合成、维护和管理情境信息数据和服务, 因此情境信息模型的设计应便于分布式的情境信息的合成。

R.2 部分验证 (Partial Validation)

在情境应用过程中存在需要将情境知识在组织结构或实例层次上同情境信息模型进行部分验证的需求。

R.3 信息质量 (Quality of Information)

在情境信息获取的过程中, 传感器中得到的数据总是随着时间而不断变化的, 不同的传感器获得数据的信息质量也可能不同, 因而情境模型应该能够表征出有效信息量的差别。

R.4 不完备性与模糊性 (Incompleteness and Ambiguity)

在情境信息获取的过程中, 从传感器得到的数据往往是不完备的或者是模糊的, 特别是数据从传感器网络 (Sensor Networks) 中得到的, 在模型设计过程中需要考虑情境数据不完备或不确定的特性。

R.5 形式化程度 (Level of Formality)

情境信息模型的形式化是指用正则精确的方式来描述情境信息中的事实与关系。例如，任务要求是“在距离我最近的打印机上打印文档”，如何表述“距离最近”的概念，对情境模型的表示提出很高的要求。同时，形式化程度直接决定了情境信息理解共享的程度。

R.6 可应用性 (Applicability of Existing Environments)

从系统实现的角度来看，情境信息模型对现有普适计算环境的软件平台（例如基于 Web Services 的服务框架等）的可应用性的支持是设计情境信息模型时需要考虑的问题。

2.2.2 情境信息模型分类和比较分析

随着情境感知系统研究的不断深入，情境感知研究项目中采用各种不同类型的情境信息模型，Strang 等人依据用于情境信息交换的数据类型将情境信息模型分为键值模型、标记模式模型、图模型、面向对象模型、基于逻辑的模型以及基于本体的模型等六类，本节中将从情境信息建模的限制条件出发分别简要说明不同类别的情境信息模型的特点以及典型应用示例。

1) 键值模型 (Key-Value Model)

键值模型是情境信息模型中最为简单的类型，在该模型中，情境数据采用键值对 (key-value pair) 的数据结构进行描述。键值模型的优点是简单，便于数据管理；缺点是情境数据的描述不够精细，缺乏有效的情境信息检索算法。键值模型在早期的情境感知系统得到采用，典型应用有 ParcTab 项目中 Schilit 等人采用键值模型用于描述位置信息[23]。键值模型在限制条件 R.1 到 R.5 方面的能力都偏弱，由于键值模型的简易性，键值模型在一定程度上满足限制条件 R.6 的要求。

2) 标记模式模型 (Markup Scheme Model)

标记模式模型是采用带有属性 (Attribute) 和内容 (Content) 的标记符 (markup tag) 组成的分层数据结构来描述情境信息的方法，描述语言一般都是从 XML (eXtensible Markup Language) 演化出来的。典型应用有 UAProf (User Agent Profile)、CC/PP (Composite Capabilities / Preferences Profile)。标记模式模型强调部分验证 (R.2) 和形式化 (R.5) 的需求，因为标记模式模型的描述语言中存在高度形式化的 Scheme 定义，而且目前已有一些验证工具来进行类型验证。

3) 图模型 (Graphical Model)

图模型是指采用面向图的建模工具 (例如 UML) 进行描述情境信息的方法, 一般是用于描述情境信息的结构。图模型的典型示例是 Henricksen 等人提出的面向情境感知应用的扩展 ORM (Object-Role Modeling) 模型[36]。Henricksen 等人扩展 ORM 模型的应用对 ORM 模型中基本概念——事实 (fact) 进行分类, 根据持久性与数据源类别将事实分为静态事实类型 (static fact type) 和动态事实类型 (dynamic fact type), 又根据情境数据的来源及依赖关系将动态事实分为预先描述事实类型 (profiled fact type)、感知事实类型 (sensed fact type) 和派生事实类型 (derived fact type)。

4) 面向对象模型 (Object-Oriented Model)

面向对象模型是利用面向对象设计思想对情境信息进行对象封装和重用的方法。典型应用有 TEA 项目[37]中利用 cue 的概念来对物理的或者逻辑的传感器进行抽象和封装。面向对象模型主要是强调分布式合成 (R.1) 的需求, 因为采用面向对象模型, 能够方便对分布的情境信息类型 (type) 及其实例 (instance) 进行类 (class) 定义和对象 (object) 封装。另外, 面向对象模型可以通过编译器和运行环境在实例层上进行部分验证 (R.2); 通过加入类属性的方法进行信息质量 (R.3) 的描述 (例如 TEA 项目) 以及不完备性 (R.4) 的描述。

5) 基于逻辑的模型 (Logic-based Model)

基于逻辑的情境模型是借用逻辑正则系统来描述情境数据和推理演绎过程。在基于逻辑的情境模型中, 情境一般定义为事实 (Fact), 表达式 (Expression) 和规则 (Rule), 逻辑主要用于定义从事实或者表达式集合演绎推理出派生的事实或者表达式集合的约束条件。典型应用有 Gray 等人提出的 Sensed Context Model, Sensed Context Model 使用一阶预测逻辑来作为情境信息陈述和关系的正则表示方法。基于逻辑的模型的优点在于有很强的形式化表示能力 (R.5), 但是在部分验证 (R.2) 和可应用性 (R.6) 等方面有局限性。

6) 基于本体的模型 (Ontology-based Model)

基于本体的情境模型是利用本体论的方法来描述情境信息的概念 (Concept) 和相互关系 (Interrelation)。基于本体的情境模型的典型应用主要有 Chen 等提出的 SOUPA 本体[38]、Wang 和 Gu 等提出的 CONON 本体[26]。

在 SOUPA 本体中, Chen 等根据智能空间中的不同实体将 SOUPA 本体分为核心本体和针对特殊应用的扩展本体, 在核心本体中主要描述智能空间中的人

(Person)、时间 (Time)、空间 (Space)、事件 (Event)、计算实体 Agent 以及策略 (Policy) 等概念, 在扩展本体中针对智能空间中典型应用 (例如智能会议室系统) 设计相关的本体, 比如会议时间表、数字文档等。

在 CONON 本体中, Wang 和 Gu 等将普适计算环境中的人 (Person)、位置 (Location)、活动 (Activity) 和计算实体 (Computing Entity) 被认为上层本体 (Upper Ontology), 是普适计算环境中的核心概念。然后根据不同的领域 (例如智能会议室、智能家居等) 定制领域例化的本体 (Domain-specific Ontology)。

基于本体的情境模型的优点是具有很强的形式化表示和语义表达能力 (R.5); 与面向对象模型相似, 利用概念和事实的方法对实体进行形式化描述有利于分布式合成 (R.1) 和部分验证 (R.2)。

综上所述, 我们在表 2.1 中比较上述六类情境信息模型在满足建模限制条件需求上的差别, 表中符号“+”表示能力强, 符号“-”表示能力弱, 这里需要指出的是不同的情境信息模型之间没有绝对的优劣差别。因而, 针对不同的应用领域, 采用合适的情境信息模型才是合理的使用方法。

表 2.1 不同情境模型比较

	键值模型	标记模式模型	图模型	面向对象模型	基于逻辑的模型	基于本体的模型
分布式合成 (R.1)	-	+	-	++	++	++
部分验证 (R.2)	-	++	-	+	-	++
信息质量 (R.3)	--	-	+	+	-	+
不完备性与模糊性	--	-	-	+	-	+

(R.4)						
形式化程度 (R.5)	--	+	+	+	++	++
可应用性 (R.6)	+	++	+	+	-	+

2.3 本体论及描述逻辑简介

2.3.1 本体论

本体 (Ontology) 最早是一个哲学上的概念, 从哲学的范畴来看, 本体是客观存在的一个系统的解释或说明, 关心的是客观现实的抽象本质。北京大学的邓志鸿[39]、浙江大学的李善平[40]等研究者对本体给出了很好的综述。在人工智能界, 最早给出本体定义的是 Neches 等人, 将本体定义为“本体定义了组成主题领域的词汇表的基本术语及其关系, 以及结合这些术语和关系来定义词汇表外延的规则。(An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary)”。Gruber 等人给出了本体普遍接受的定义, 即“本体是概念模型的明确的规范说明 (An ontology is an explicit specification of a conceptualization)” [41]。Borst 等人在此基础上给出了本体的另一个定义“本体是共享概念模型的形式化规范说明 (An ontology is a formal specification of a shared conceptualization)” [34]。Studer 等对上述两个定义进行了深入的研究, 认为本体是共享概念模型的明确的形式化规范说明, 包含四个层次的含义:

- 1) 概念模型 (Conceptualization): 通过抽象出客观世界中一些现象 (Phenomenon) 的相关概念而得到的模型。
- 2) 明确 (Explicit): 所使用的概念及使用这些概念的约束都有明确的含义。
- 3) 形式化 (Formal): 本体是计算机可读的、可处理的。
- 4) 共享 (Share): 本体中体现的是共同认可的知识, 反映的是相关领域中公认的概念集。

2.3.2 描述逻辑

描述逻辑 (Description Logics) [27] 是一种基于对象的知识表示的形式化, 它吸取了 KL-ONE 的主要思想, 也叫概念表示语言 (Concept Language) 或术语逻辑 (Terminological Logics)。它建立在概念 (Concept) 和关系 (Role) 之上, 其中概念解释为对象的集合, 关系解释为对象之间的二元关系。描述逻辑是一阶逻辑的一个可判定的子集。描述逻辑是语义网络 (Semantic Networks) 派生来的, 它和框架理论 (Frame Theory) 有着很多的相似性。描述逻辑被认为是以对象为中心的表示语言的最为重要的归一形式。

描述逻辑的重要特征是很强的表达能力和可判定性, 它能保证推理算法总能停止, 并返回正确的结果。在众多知识表示的形式化方法中, 描述逻辑在十多年来受到人们的特别关注, 主要原因在于: 它们有清晰的模型-理论机制; 很适合于通过概念分类学来表示应用领域; 并提供了很好的推理服务。

最基本的描述逻辑语言是 AL (Attributive Language), 它的核心算法 Tableau 能在多项式时间内判断描述逻辑 AL 概念的可满足性问题, 其它描述逻辑语言都是对 AL 的扩展。目前, Tableau 算法已用于各种描述逻辑中, 现在主要研究各种描述逻辑中 Tableau 算法的扩展、复杂性及优化策略等。

描述逻辑在许多领域中被作为知识表示的工具。英国曼彻斯特大学的 Iran Horrocks 教授对表达能力较强的描述逻辑进行了深入地研究, 建立了一些逻辑框架, 如 SHF (它对基本描述逻辑 ALC 进行了扩充, 增加了传递关系、函数关系和层次关系), SHIQ(D) (它对 SHF 进行了扩充, 增加了逆关系和数量约束), 并开发了一个描述逻辑推理系统 FaCT (它实现了 SHF 和 SHIQ 的推理), SHOQ(D) (它扩充了 SHIQ, 它表达的描述逻辑能够处理命名个体和具体的数据类型, 它强大的表达能力几乎与最新的 DAMIL+OIL 一样) 等等。

Iran Horrocks 教授和 Dieter Fensel 等人将描述逻辑、语义网和 DAML(DARPA Agent Markup Language) 结合起来, 提出了 DAML+OIL, 进一步发展成 OWL(Web Ontology Language), 其中以描述逻辑作为核心的表示和推理基础, 并在 XML 及其 RDF 上面进行了扩展, 用描述逻辑来研究语义网络和本体论。

2.3.2.1 描述逻辑系统的构成

一个描述逻辑系统包含四个基本组成部分：

- 1). 表示概念和关系(Role)的构造集。
- 2). Tbox——(terminological knowledge, 简称为 Tbox) 关于概念术语的断言, 它是描述领域结构的公理的集合。
- 3). Abox——(assertional knowledge, 简称为 Abox) 关于个体的断言, 描述具体情形的公理的集合。其中概念断言表示一个对象是否属于某个概念、关系断言表示两个对象是否满足一定的关系。
- 4). Tbox 和 Abox 上的推理机制。

通常我们把 Tbox 和 Abox 称为某描述逻辑系统的知识库, 即知识库:

$$K = \langle Tbox, Abox \rangle$$

下面介绍最基本的描述逻辑描述语言 AL (Attributive Language)。

2.3.2.2 最基本描述逻辑语言

为了形式化地定义 AL 的语法, 下文使用 A 和 B 表示原子概念、 R 表示原子关系, C 和 D 表示概念。为了定义 AL 概念的形式化语义, 使用符号 I 来表示解释, 则非空集合 Δ' 是所有解释的定义域。最基本逻辑描述语言 AL 的语法和语义如表 2.2 所示:

表 2.2 AL 的语法和语义

构造算子	语法	语义
原子概念	A	$A^I \subseteq \Delta'$
原子关系	R	$R^I \subseteq \Delta' \times \Delta'$
顶部	T	$T^I = \Delta'$
底部	\perp	$\perp^I = \emptyset$
非	$\neg A$	$(\neg A)^I \subseteq \Delta' \setminus A^I$
交	$C \cap D$	$(C \cap D)^I = C^I \cap D^I$
全称量词	$\forall R.C$	$(\forall R.C)^I = \{a \in \Delta' \mid \forall b, (a, b) \in R^I \rightarrow b \in C^I\}$

限制存在量词	$\exists R.T$	$(\exists R.T)^I = \{a \in \Delta^I \mid \exists b, (a,b) \in R^I\}$
--------	---------------	--

一个解释 I 是包含断言 $C \subseteq D$ 的模型，当且仅当 $C^I \subseteq D^I$ ；解释 I 是实例断言 $C(a)$ 的模型，当且仅当 $a \in C^I$ ；解释 I 是 $R(a,b)$ 的模型，当且仅当 $(a,b) \in R^I$ 。

2.3.3 OWL 语言

描述逻辑的特性很大程度上满足和代表了新一代本体描述语言发展的要求，最近几个主要的 Web 本体语言 DAML，OIL，DAML+OIL 以及已成为 W3C 国际标准 OWL (Ontology Web Language) 就是建立在描述逻辑的基础上。OWL 建立在 XML/RDF 等已有标准基础上，通过添加大量的基于描述逻辑的语义原语来描述和构建各种本体。OWL 根据表示和推理能力分为 3 类[48]:

- OWL Lite: 局限于对概念(类)的层次分类和简单的约束等进行描述;
- OWL DL: 以描述逻辑为基础, 在不失掉计算完全性和可判定性条件下, 支持最大的表示能力;
- OWL Full: 与 RDF 保持最大程度的兼容, 具有最大的表示能力, 但不能保证计算性能。

表 2.3 给出了 OWL 的主要类构造符 (Constructors) 与描述逻辑语法的对应关系及其应用示例 (C 表示概念, R 表示关系, x 表示实例)。

表 2.3 OWL 类构造符与描述逻辑语法的对应

OWL 类构造符	描述逻辑语法	应用示例
intersectionOf	$C_1 \cap \dots \cap C_n$	Human \cap Male
unionOf	$C_1 \cup \dots \cup C_n$	Doctor \cup Lawyer
complementOf	$\neg C$	\neg Male
one of	$\{x_1, \dots, x_n\}$	{Chen, Xu}
toClass	$\forall R.C$	\forall hasChild.Doctor
hasClass	$\exists R.C$	\exists hasChild.Lawyer

hasValue	$\exists R.\{x\}$	$\exists \text{ citizenOf.}\{CHN\}$
minCardinalityQ	$\geq nR.C$	$\geq 2\text{hasChild.Laywer}$
maxCardinalityQ	$\leq nR.C$	$\leq 1\text{hasChild.Male}$
cardinalityQ	$= R.C$	$=1\text{hasParent.Female}$

表 2.4 给出了 OWL 的主要公理构造符(Axioms)与描述逻辑语法的对应关系及其应用示例(C 、 D 表示概念, R 、 S 表示关系, x 、 y 表示实例)。

表 2.4 OWL 公理构造符与描述逻辑语法的对应

OWL 公理构造符	描述逻辑语法	应用示例
subClassOf	$C \subseteq D$	$\text{Human} \subseteq \text{Animal} \cap \text{Biped}$
sameClassAs	$C \doteq D$	$\text{Man} \doteq \text{Huamn} \cap \text{Male}$
subPropertyOf	$R \subseteq S$	$\text{hasDaughter} \subseteq \text{hasChild}$
samePropertyOf	$R \doteq S$	$\text{cost} \doteq \text{price}$
sameIndividualAs	$x \doteq y$	$\text{Chairman_Hu} \doteq \text{Hu_jingtao}$
disjointWith	$C \subseteq \neg D$	$\text{Male} \subseteq \neg \text{Female}$
differentIndividualFrom	$\{x\} \subseteq \neg\{y\}$	$\{\text{chen}\} \subseteq \neg\{\text{xu}\}$
inverseOf	$R \doteq S^{-}$	$\text{hasChild} \doteq \text{hasParent}^{-}$
transitiveProperty	$R^{+} \subseteq R$	$\text{ancestor}^{+} \subseteq \text{ancestor}$
uniqueProperty	$\text{Thing} \subseteq \leq 1R$	$\text{Thing} \subseteq \leq 1\text{hasMother}$
unambiguousProperty	$\text{Thing} \subseteq \leq 1R^{-}$	$\text{Thing} \subseteq \leq 1\text{isMotherOf}$

几乎所有的公理都可以简化为用 subClassOf 和 subPropertyOf 来表示。

2.4 基于本体的情境信息模型

在情境感知计算的研究领域中, 研究者在构建情境感知系统的过程都认为情境信息模型是情境感知计算研究中极为重要的组成部分, 因为情境信息模型设计所采用的方法直接决定了情境感知系统中情境信息组织和访问的方式[35]。在本节中, 我们研究采用本体的方法来建立面向智能空间的情境信息模型 COSS (Context Ontology for Smart Space), 用以描述智能环境中涉及的各种实体, 并利用 W3C 国际标准 OWL 语言进行形式化的表示。

2.4.1 COSS 模型核心本体

COSS 模型将情境本体分为两个部分：核心本体（Core Ontology）和扩展本体（Extended Ontology）。核心本体用于描述智能空间中通用的概念实体，COSS 模型中选取用户（User）、环境（Environment）、服务（Service）、活动（Activity）以及平台（Platform）作为核心本体中的组成元素，认为上述基本元素构成了智能空间的基本框架[19][20][22][23]：

$$C = \{P, A, U, S, E\}$$

根据第 2.3 节中本体理论的简介，我们将核心概念命名为 owl:CoreOnt，智能空间中的实体与 CoreOnt 均是子类的关系 rdfs:subClassOf。在下文中将对核心本体的概念依次说明，部分描述用 List 2.1 表示。

【List 2.1】

```
<owl:Class rdf:ID="CoreOnt" />
<owl:Class rdf:ID="User">
  <rdfs:subClassOf rdf:resource="#CoreOnt" />
</owl:Class>
<owl:Class rdf:ID="Activity">
  <rdfs:subClassOf rdf:resource="#CoreOnt" />
</owl:Class>
<owl:Class rdf:ID="Service">
  <rdfs:subClassOf rdf:resource="#CoreOnt" />
</owl:Class>
<owl:Class rdf:ID="Environment">
  <rdfs:subClassOf rdf:resource="#CoreOnt" />
</owl:Class>
<owl:Class rdf:ID="Platform">
  <rdfs:subClassOf rdf:resource="#CoreOnt" />
</owl:Class>
```

2.4.1.1 用户实体（User）

从 Dey 对情境给出的定义，我们可以看出所谓的情境信息是和用户的任务或者活动相关的，因而用户（User）是智能空间中重要的情境信息。收集用户自身相关的情境信息将有助于应用程序以及服务来提高自身的可用性。以根据不同用户的偏好来定制应用程序的用户友好界面为例，在设计远程多媒体交互软件的用户界面时，对于学龄前的小朋友来说，交互软件的功能应该尽量简单，界面卡通化，这样更能吸引小朋友的兴趣；对于年轻的网迷来说，交互软件的功能较为齐全，界面能够定制，迎合年轻人追求个性化的需求；对于计算机专

业人员来说则要求功能完整，界面风格可以较为严谨。

用户自身相关的情境信息有 User Profile、偏好 (Preference)、情绪 (Mood) 和角色 (Role) 等。User Profile 记录用户自身的基本信息，例如姓名、性别、年龄、生日、联络方式、个人主页、工作地址、家庭住址、Email 等等。用户的偏好记录了用户的个人喜好，例如应用程序字体的大小，程序界面的风格等。角色是用户重要的情境信息，例如在智能教室的应用领域中，用户角色分为教师、现场学生以及远程学生。不同角色的用户带有不同的使用权限，教师具有控制整个教学活动进程的权限，现场学生有回答教师提问的权限，而远程学生只有看整个教学活动的权限。不同的角色同时也决定了用户之间的关系，例如在智能会议室中，用户的角色分为讲演者 (Speaker) 和听众 (Audience) 两种，并且讲演者和听众的关系是分离 (owl:disjointWith) 的关系，讲演者有且仅有一个，因而当智能会议室中，系统通过传感器 (例如麦克风阵列、摄像头) 检测到同一个用户既是讲演者又是听众时，或存在两个或者两个以上的用户均是讲演者时，则会报告情境信息不一致的警告。情绪也是用户相关的重要情境信息，用户的喜怒哀乐影响用户的决策和行为。另外，在智能空间中，用户实体与活动实体的关系是参与关系 (engageIn)，用户实体与服务实体的关系是使用关系 (useService)。图 2.1 说明用户相关的属性以及用户实体与服务实体、活动实体的关系，其中 Teacher 和 Student 是用户实体针对智能教室应用领域的扩展本体。扩展本体的概念将在第 2.4.2 节进行简要说明。

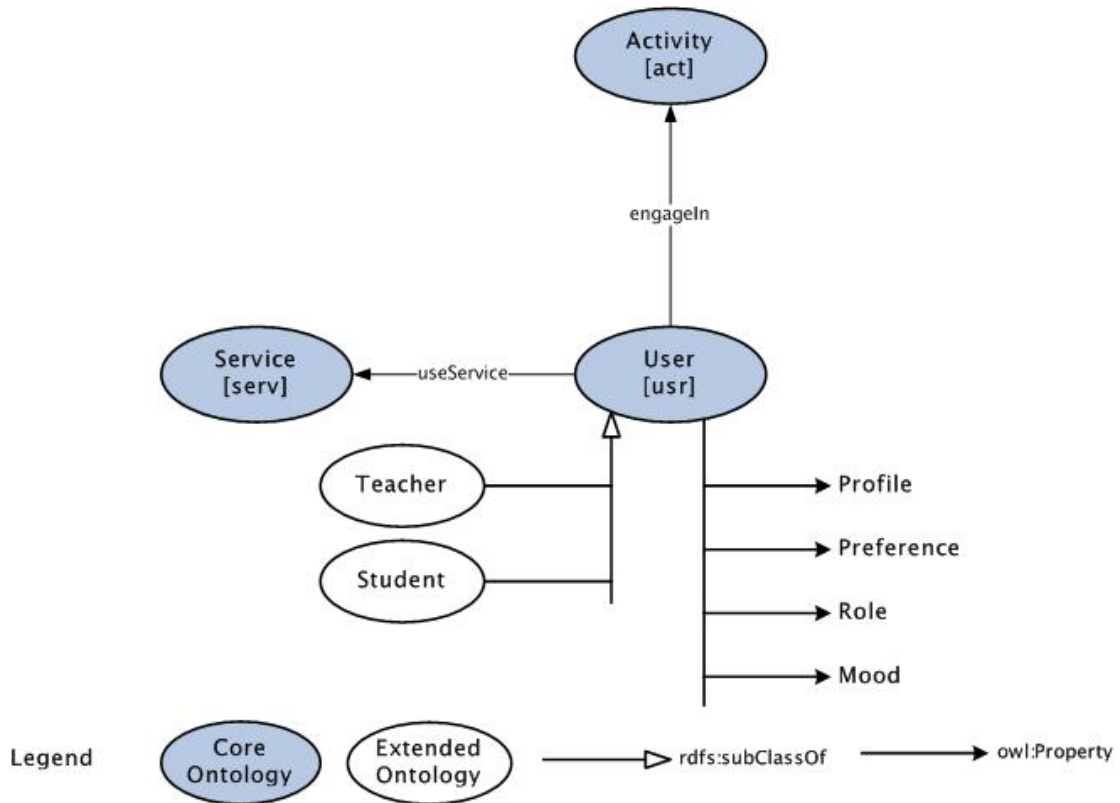


图 2.1 用户实体的关系图

利用 OWL 语言描述用户实体如 List 2.2 所示。

【List 2.2】

```

<owl:Class rdf:ID="User" />
<owl:DatatypeProperty rdf:ID="Role">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="&co;ss;User"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="Mood">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="&co;ss;User"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="firstName">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="&co;ss;User"/>
</owl:Class>
...
<owl:Class rdf:ID="Service" />
<owl:Class rdf:ID="Activity" />
<owl:ObjectProperty rdf:ID="engageIn">
  <rdfs:range rdf:resource="#Activity"/>

```

```

    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#User"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="useService">
    <rdfs:range rdf:resource="#Service"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#User"/>
</owl:ObjectProperty>

```

2.4.1.2 服务实体 (Service)

服务 (Service) 是具备提供某种特定功能的计算实体, 使用具有规范化定义的接口辅助用户完成自身任务或者活动。典型例子有, 当一个用户进入智能会议室后, 想将随身携带的 PDA 上的 PowerPoint 文档, 用会议室中的投影仪进行投放, 那么用户通过 I/O 设备来使用智能空间中提供的投影服务, 而投影服务则是由智能空间中的基础平台提供。如图 2.2 所示, 用户或者活动扮演着服务消费者 (Consumer) 的角色, 基础平台则扮演者服务提供者 (Provider) 的角色。同时 List 2.3 给出服务与用户、活动以及平台之间的关系。

【List 2.3】

```

<owl:Class rdf:ID="Service" />
<owl:Class rdf:ID="Platform" />
<owl:Class rdf:ID="Activity" />
<owl:Class rdf:ID="User" />
<owl:ObjectProperty rdf:ID="provideService">
    <rdfs:range rdf:resource="#Service"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Platform"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="useService">
    <rdfs:range rdf:resource="#Service"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#User"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="useService">
    <rdfs:range rdf:resource="#Service"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Activity"/>

```

</owl:ObjectProperty>

为了便于用户和活动使用各种各样的服务，服务实体应该具备具有语义信息的、计算机可解释（Computer-Interpretable）的描述形式。这里借用 OWL-S[49] 规范对服务本体的描述。虽然 OWL-S 广泛地应用于 Web Services 和 Semantic Web 研究领域，但是 OWL-S 提出了通用的标准化框架来描述服务本体。服务本体的概念利用三类概念进行解释，分别为 Service Profile、Service Model 和 Service Grounding。

- **Service Profile:** 用于描述服务的功能。也就是说，Service Profile 通过对服务输入输出的详细说明来提供服务功能的描述，提供的信息涵盖服务提供者的信息、服务质量的等级级别以及用于服务查找的其他属性。
- **Service Model:** 用于解释服务的处理过程。也就是说，Service Model 通过给出服务执行过程涉及的控制流和数据流的具体信息，来描述服务的处理过程。对于服务消费者而言，Service Model 的作用体现在：i) 进行深入的分析来判定服务是否满足服务消费者的需求；ii) 用于合成多个服务来完成特定任务；iii) 在服务设定过程中协调服务执行过程不同参与者的活动；iv) 监视服务的执行过程。
- **Service Grounding:** 用于说明服务消费者访问该服务的实现细节。典型地，Service Grounding 定义服务通讯采用的协议、消息格式以及服务特定的细节（例如服务交互过程中使用的端口号等）。

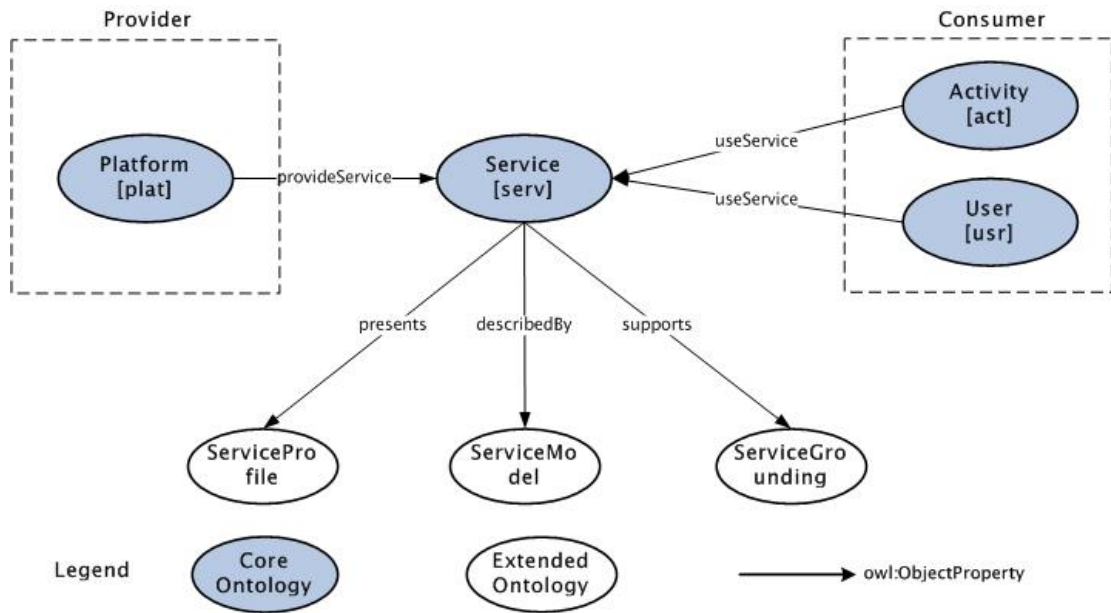


图 2.2 服务实体的关系图

用 OWL 语言对服务实体的描述如 List 2.4 所示。

【List 2.4】

```

<owl:Class rdf:ID="Service" />
<owl:Class rdf:ID="ServiceProfile" />
<owl:Class rdf:ID="ServiceModel" />
<owl:Class rdf:ID="ServiceGrounding" />
<owl:ObjectProperty rdf:ID="presents">
  <rdfs:domain rdf:resource="#Service"/>
  <rdfs:range rdf:resource="#ServiceProfile" />
  <owl:inverseOf rdf:resource="#presentedBy" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="describedBy">
  <rdfs:domain rdf:resource="#Service"/>
  <rdfs:range rdf:resource="#ServiceModel"/>
  <owl:inverseOf rdf:resource="#describes"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="supports">
  <rdfs:domain rdf:resource="&service;#Service"/>
  <rdfs:range rdf:resource="&service;#ServiceGrounding"/>
  <owl:inverseOf rdf:resource="&service;#supportedBy"/>
</owl:ObjectProperty>

```

2.4.1.3 环境实体 (Environment)

用户在智能空间中不可能是单独存在的，他通常是使用设备与周围环境或者周围其他用户进行交互的，周围的环境能够为用户提供有用的信息来辅助用户的决策。举例来说，在炎热的夏季，一位住户进入智能家庭中，通过房间中嵌入的温度测定传感器看到室温已经达到 35℃，因此用户使用随身携带的 PDA 控制房间中的空调，对室内进行降温。但是，存在的一个问题是在智能空间中环境产生的信息太宽泛了，并不是所有检测到或者测量到的数据都是有用的。从以用户为中心的角度出发，与用户活动或者移动性相关的环境信息才被认为是可用的情境信息。因此，我们认为环境实体分为三个子类：位置 (Location)、时间 (Time) 和环境的物理条件 (Physical Condition)，如图 2.3 所示。

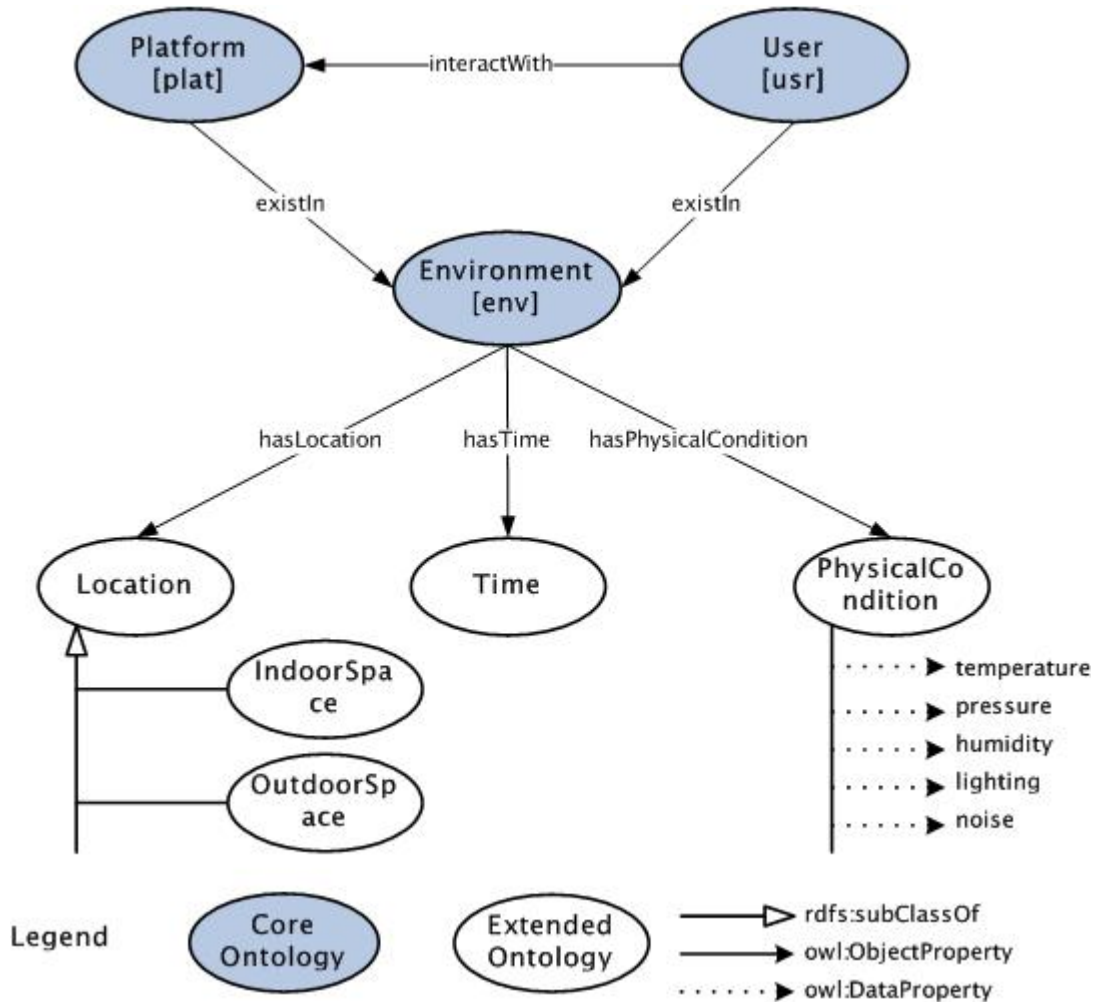


图 2.3 环境实体的关系图

根据应用领域的不同，位置实体可以分为 *IndoorSpace* 子类和 *OutdoorSpace* 子类，分别表示室内的空间位置概念以及户外的空间位置概念。环境的物理条件（*Physical Condition*）在智能家居的领域里则包含诸如温度、湿度、气压、光照、噪声等物理参数信息。

需要指出的是环境的信息一般都是通过传感器检测到的数据分析得出的，那么环境获得的数据信息具有实时性和不确定性的特点。不同传感器获得的数据之间存在发生冲突（*Conflict*）的可能，例如在智能家居中，卧室和客厅中的位置传感器都检测到用户的存在，因为两处的传感器都有可能收到用户携带位置传感器的信号，这样位置信息都发生了冲突。此时，就需要对传感器获得的数据的准确性进行可信度的推理才能决断用户可能的位置。另外，用户与环境之间并不是直接交互的，尽管用户存在与（*existIn*）环境中。用户则是通过访问平台（*Platform*）中的设备来获取环境信息的。

2.4.1.4 活动实体（*Activity*）

活动信息往往是智能空间中应用的重要线索，例如在智能会议室中，用户往往是在参与（*engageIn*）当前正在进行的会议活动中。如图 2.4 所示，活动实体用于描述活动的状态（*Status*）、出席者（*Attendee*）、地点（*Location*）、起始时间（*StartTime*）和中止时间（*EndTime*）等信息。活动实体可以认为主要分为两类：预定的活动（*Scheduled Activity*）和推断的活动（*Deduced Activity*）[42]。其中，预定的活动一般是指会议（*Meeting*）和上课（*Class*）等事先已经计划安排好的活动，活动的属性（例如出席者、地点、起始和中止时间等）都可以预先设定。另一类推断的活动是通过情境信息的推理得出的推断，具有一定的不确定性。以智能家居中的场景进行说明，我们通过客厅中布置的各种传感器检测的事实表明：i) 客厅中的电视机的状态是开（*On*）状态；ii) 通过压力传感器检测到用户正坐在客厅中的沙发上；iii) 位置传感器检测到沙发在电视机的可视范围以内，我们推断出用户当前参与的活动是看电视（*WatchingTV*）。用逻辑表示为：

$$\begin{aligned} & locateAt(?U, sofa) \\ & \wedge locateIn(sofa, VisualRange) \\ & \wedge locateIn(TVset, VisualRange) \\ & \wedge status(TVset, ON) \\ & \Rightarrow situation(?U, WatchingTV) \end{aligned}$$

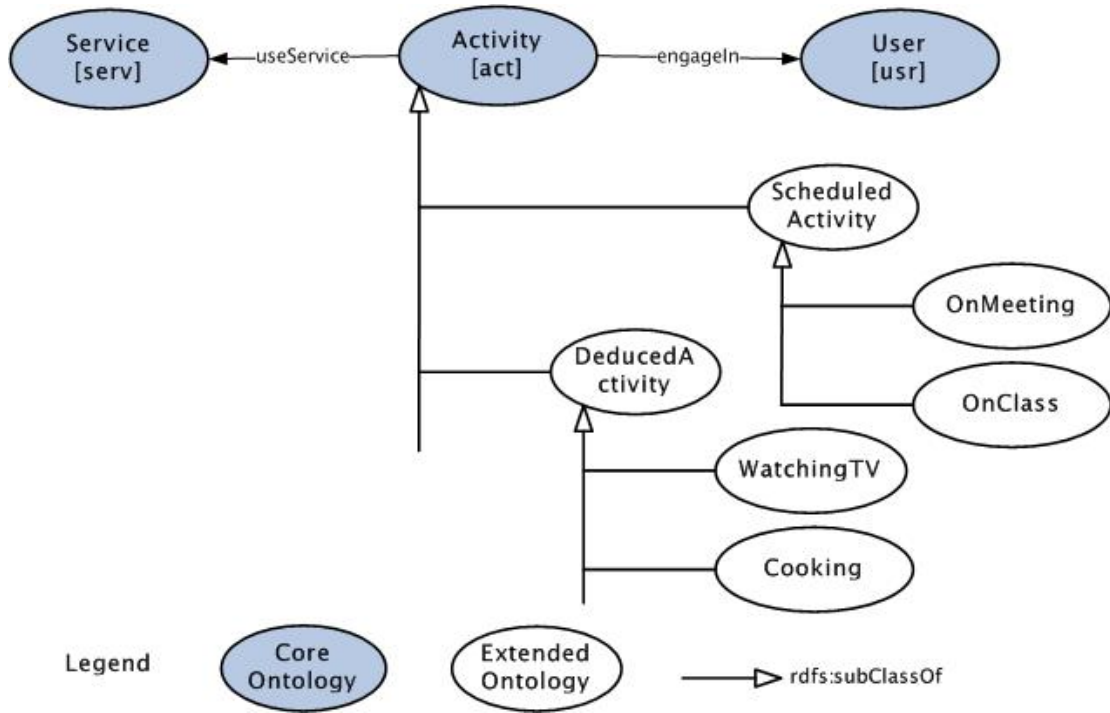


图 2.4 活动实体的关系图

2.4.1.5 平台实体（Platform）

智能空间的设计目的是使用户能够非常方便地在其中访问信息和获取计算机的服务来高效地进行单独工作与其他人的协作，这就要求智能空间嵌入强大的平台功能支持。如图 2.5 所示，在智能空间中，我们认为平台（Platform）实体提供了对平台中包含的软件（Software）实体和硬件（Hardware）实体的描述。

- 设备上的软件实体：设备上的软件实体主要是如下几种方式体现的：中间件（Middleware）、操作系统（Operating System）、虚拟机（Virtual Machine）等。我们定义设备上的软件实体规范，目的在于：i) 设备上运行的软件是智能空间中服务的提供者，在生成服务之前需要检测运行的软件功能是否满足服务功能的需求；ii) 实现跨平台的软件部署时需要检测软件平台是否支持该需求。举例来说，目前智能教室系统中是采用 Smart Platform[7]作为支持通讯和协调机制的软件中间件，服务是由在 Smart Platform 上运行的 Agent 所提供的。当需要迁移 Agent 时，就需要检测当前设备上运行的 Smart Platform 是否支持该 Agent 的运行，因为 PC 和 PDA 上运行的 Smart Platform 运行环境是不兼容的。

- 设备上的硬件实体：我们定义设备上的硬件实体包含的属性包括产品参数和设备的性能参数等，产品参数生产序列号、设备型号、生产商、生产日期等信息，性能参数则说明设备的硬件配置，例如 CPU 的主频、内存的大小、硬盘存储的大小、联网的带宽以及电池性能等。这些硬件性能参数对于移动设备而言是非常重要的，因为该硬件性能参数直接决定了应用服务是否能在该设备上运行。

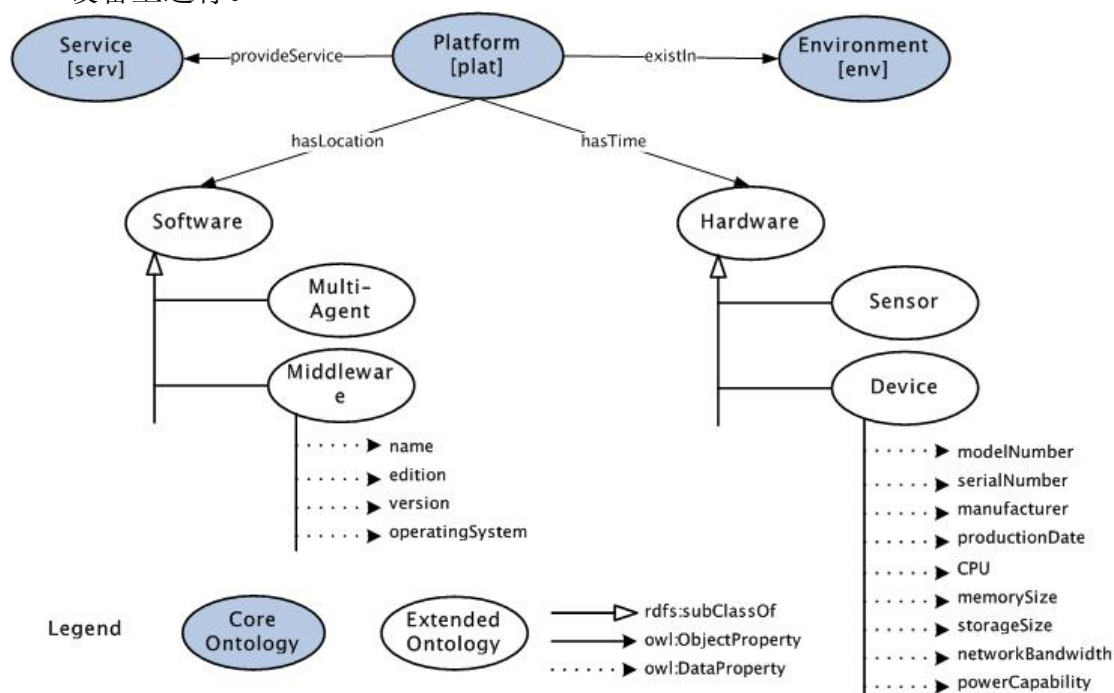


图 2.5 平台实体的关系图

2.4.2 扩展本体

扩展本体（Extended Ontology）是根据不同的应用领域（Domain）扩展核心本体，定义领域相关的本体细节和词表。举例来说，在智能教室应用领域中，用户本体根据角色的不同增加子类教师本体和学生本体，学生本体又可分为本地学生和远程学生，不同的角色划分取决于用户对智能空间系统的控制权：教师在智能教室中处于主导地位，可以向学生提问，控制教学活动的进程；本地学生可以回答教师的现场提问；而远程学生可能只是通过视频软件观看教学全过程。将本体分为核心本体和扩展本体的优点有：i) 根据领域的不同将本体进行层次的划分有利于减小情境知识的规模，构建本体知识库时只用加载领域相关的本体即可；ii) 提高本体的知识共享和复用，核心本体是所有领域所共享的

本体库，不同领域的扩展本体之间采用映射方式实现共享[38]。

2.4.3 实体间的关系

综上所述，智能空间中核心实体之间关系如图 2.6 所示，用户（User）进入智能空间后存在于（existIn）智能环境（Environment）中。当用户参与（engageIn）当前的活动（Activity）时，用户需要使用（useService）智能空间中提供的各种服务（Service），而该服务是由智能空间中的平台（Platform）所提供（provideService）的。当用户需要使用服务时，则需要通过与平台进行交互（interactWith）才能执行用户的任务。

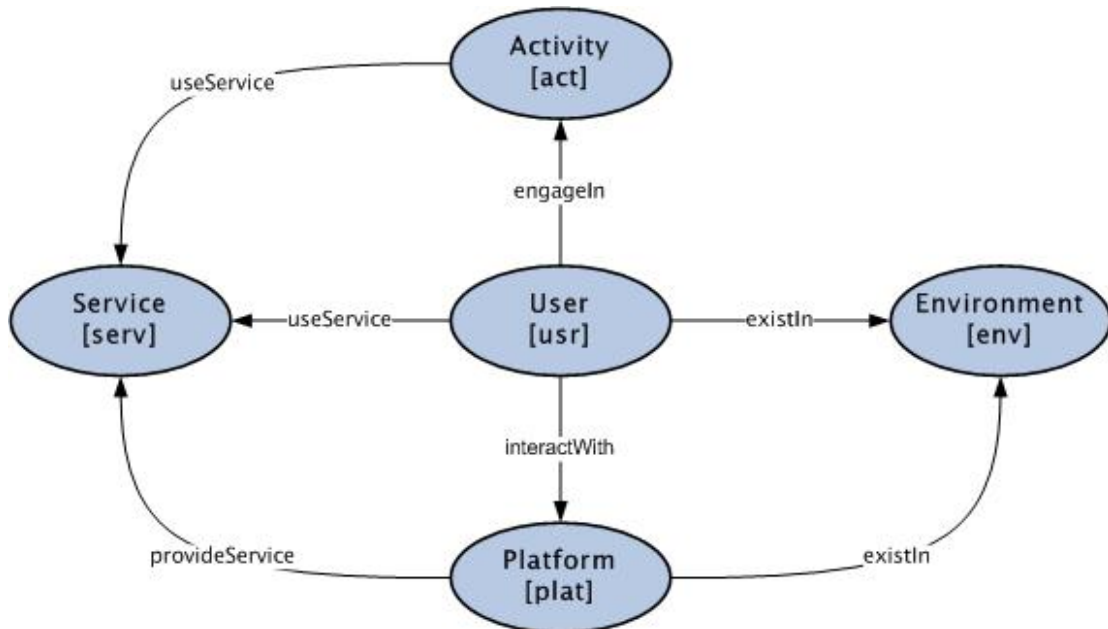


图 2.6 智能空间中核心实体之间的关系图

2.5 本章小结

智能空间的异构性、动态性以及其中模块间交互的自发性，决定着智能空间系统必须是一个开放的分布式系统。本体作为共享概念的形式化的规范说明”能够很好地描述智能空间中各种实体——用户、服务、环境、活动和平台之间的关系，为此本章先叙述了情境信息的分类以及情境建模的方法，然后简要介绍本体和描述逻辑的基本理论知识，然后采用本体论的方法形式化的抽象描述智能空间的核心本体以及领域相关的扩展本体，并给出各种实体之间的关系，

使得系统能够能很方便地将用户的任务实例化。本体的使用能够让不同的实体对特定领域中的各种概念有一个共同的认识，便于知识的共享和复用。

第3章 基于规则的情境推理机制

情境推理是情境感知系统的重要特征，情境感知系统的功能取决于情境推理的支持[38]。前文中我们已经提到，智能空间的设计目标是让智能空间中的用户透明地使用智能空间中的服务和计算资源，这要求智能空间系统不仅能够准确高效地响应用户提出的应用需求，更高的要求是智能空间能够“感知”用户的需求，主动地为用户提供服务和计算资源，进而前摄地（Proactively）为用户提供有用的服务。因此智能空间系统应具备根据当前用户和环境资源的情境来获取智能空间中存在的高层语义，也就是说智能空间系统应当具备情境推理的能力，通过感知智能环境中各种实体的情境或者状态来生成高层语义信息，其中基于规则的情境推理机制是情境推理中重要的方法，本章集中探讨基于规则的情境推理机制。

本章先给出情境推理的定义，然后阐明情境感知系统中推理的应用需求和任务，随后介绍情境推理的系统设计，最后详细描述几种基于规则的推理引擎的算法及其面向智能空间应用的示例。

3.1 情境推理的定义

智能空间中的情境推理具有相当重要的作用，在深入探讨智能空间的情境推理之前，我们首先介绍面向智能空间应用的情境推理的基本认识和定义。

面向智能空间应用的情境推理（Context Reasoning for Smart Space）是指从智能空间中存在的各种情境数据源信息（Context Data Source）到与智能空间中的用户（User）以及用户相关的应用（Application）的关联信息的推演过程。

这里我们需要强调的是生成的情境信息是与用户及用户相关的应用的关联信息，正如前章中对情境定义的分析，因为智能空间是一个高度动态、开放异构的分布式系统，并不是所有的信息都是有用的情境信息，我们推演出与用户相关或者与用户关联的应用相关的信息作为情境信息，才能增强智能空间系统对用户的可用性。举例来说，在智能教室环境中，教师的位置信息是一种重要的情境信息，因为通过检测教师在教室中的位置，感知教师的活动状况和分布，进而前摄地提供有用的服务，例如当上课开始时，智能教室检测教师进入教室

后，自动从教师个人主页上下载课程讲义；当教师准备开始讲课，站在讲台上时，系统将教师的教案投放在黑板上。这里情境信息是围绕智能空间中的用户和与用户相关的应用展开的，例如教师的位置信息、播放教师讲义的应用软件、教师讲台附近的大屏幕投影设备等。

从情境推理的定义，我们自然地将情境推理过程中涉及的情境信息按照情境推理处理流程分为两类：低层情境数据（Low-level Context）和高层情境信息（High-level Context）。低层情境数据是从传感器硬件或者软件中间件获得的原始数据，高层情境信息是根据应用需求确定的与用户及用户关联的应用相关的信息。情境推理的基本框架如图 3.1 所示，以定位系统为例，通过定位系统（例如 RFID、GPS 定位系统）获取用户的位置坐标数据<24m, 10m, 8m>，经过情境推理机制（例如坐标映射系统）得到用户的位置在 3 区 526 房间的信息，相对于原始坐标信息来说，推理出的位置信息具有房间概念的高层的语义信息，而其中推理机制的作用是从坐标系统到位置布局概念的映射过程。

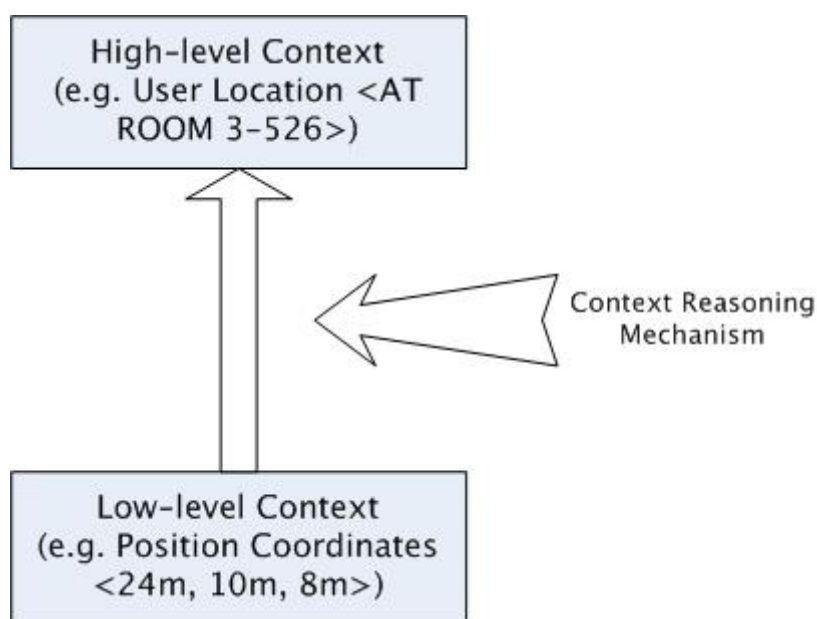


图 3.1 情境推理过程中低层和高层情境信息的角色

这里需要指出的是本章阐述的内容是在面向本体描述的情境信息模型的框架内，选取基于规则的推理方法来说明智能空间中情境推理的作用和应用方案。在此限制条件下，下文中将阐述情境推理的应用需求和任务，并提出基于规则的情境推理的设计框架和实现细节。

3.2 基于规则的情境推理的优点

情境推理 (Context Reasoning) 对于每个情境感知系统来说是重要的组成部分, 情境感知系统的功能取决于情境推理的支持。原有的情境感知系统, 如 Context Toolkit, 采用面向对象的程序设计语言 (如 Java、C++) 来定制情境推理模块的功能。面向对象的程序设计语言在接口层上保证了情境感知模块的复用, 但是在知识表示层上缺乏表示能力[35]。因此, 本文采用基于规则的逻辑推理的方法来实现情境知识的推理过程, 因为基于规则的情境推理具有如下几点优点:

- 具有清楚正规的情境推理规则表示。基于规则的情境推理具有清楚正规的情境推理规则的表示机制, 这样能够将高层的推理逻辑与低层的功能实现进行分离。通过将逻辑推理与功能实现的分离, 一旦情境推理的应用场景发生改变时, 开发者只需要更新应用场景所需的推理规则, 不需要进行大量的重编码的过程。
- 支持多种推理引擎和推理复杂度的控制。基于规则的推理方法已有多种逻辑模型来针对特定概念的推理, 例如时间和空间等一般概念。此外, 目前有多种推理模型和算法可以选择作为逻辑推理的方法, 例如基于描述逻辑 (Description Logics) 的算法、基于一阶概率逻辑 (First Order Logics) 的方法、基于贝叶斯网络 (Bayesian Networks) 的方法等等, 这类方法已有很多研究者分析和论证控制推理的复杂度等问题[32][42]。

3.3 基于规则的情境推理的任务

本文涉及的基于规则的情境推理机制的任务主要体现在两个方面: 一个是针对利用本体描述的情境信息的推理; 另一个是针对根据不同的应用领域所自定义规则的推理。

- 基于本体描述的情境信息的推理, 下文简称为“本体推理”。由于智能空间中不同实体均采用本体的方法进行刻画, 并利用 OWL 语言进行描述, 基于本体描述的情境信息的推理过程主要完成本体中的冲突检测、本体中的表达优化、本体的融合和验证知识的正确性等任务。
- 基于自定义规则的情境推理, 下文简称为“自定义规则推理”, 主要是指用户针对不同的应用领域自定义的逻辑规则的推理。用户可以根据应

用领域的不同自行添加推理机中的规则，这部分关系无法从对智能空间中不同实体的本体描述中获得，用户自定义规则保证了情境推理的灵活性和完整性[42]。

3.4 本体推理的分析和应用

3.4.1 本体语言的分析

情境建模是利用本体语言作为建模语言，它是以属性为中心，不同于传统的 C++、Java 等面向对象的编程语言。这些高级语言建模的出发点是对象，然后才是对象的各种属性。而本体语言建模的出发点是本体领域中的各种属性，然后使用这些属性来描述本体对象。本体对世界中某个对象的理解是从该对象所有的属性中得到的，所以在本体语言中往往可单纯地通过一个限制属性来定义一个类。本体推理首先要得到这些属性，然后使用这些属性蕴涵的推理规则。

从本体推理的角度出发，我们选择使用 OWL 语言来描述智能空间中的不同实体，基本的思想是在表达力和推理复杂度之间取得平衡，既要满足表达智能空间中情境信息的需求，又要控制推理复杂度，方便应用的开发。OWL 语言按表达力从低到高包括三个子语言：OWL Lite ,OWL DL 和 OWL Full。分析 OWL Lite, 发现 OWL Lite 降低了定义系统本身矛盾的可能性从而降低了推理的难度。首先,OWL Lite 去掉了枚举类和属性值限定使得类和属性的定义部分脱离了个体实例定义部分。其次，OWL Lite 不支持 Class 的并和补定义和 disjoint 关系使得定义 Class 层次时基本上不会产生冲突。最后，OWL Lite 限制基数限定为 0 或 1 使个体定义时受到较小的约束。因此，OWL Lite 尽管牺牲了部分表达力，但减少了定义本体中可能出现的矛盾,也更容易开发基于这些本体的推理工具。因而，本文中 choice 采用 OWL Lite 作为本体语言进行实体的描述。

3.4.2 本体推理的应用

对于基于本体描述的情境知识的推理来说，基本内容是由给定的情境信息获得隐含的知识，在本体中的推理从根本上说就是把隐含在显式定义和声明中的知识通过一种处理机制提取出来。然而，盲目的获得隐含的知识对建立和使用本体帮助不大，因而应该理清面向智能空间应用的本体推理的应用需求，才能够有效的组织推理机制。本体推理在智能空间应用有多方面的作用：本体的

冲突检测、表达优化、本体的融合和验证知识的正确性等。下面以智能家庭中用户身份的一个本体实例具体讨论，如 List 3.1 所示：

【List 3.1】

```

<owl:Class rdf:ID="User" />
<owl:Class rdf:ID="Man">
  <rdfs:subClassOf rdf:resource="#User" />
  <owl:disjointWith rdf:resource="#Woman" />
</owl:Class>
<owl:Class rdf:ID="#Woman">
  <rdfs:subClassOf rdf:resource="#User" />
</owl:Class>
<owl:Class rdf:ID="Parent">
  <rdfs:subClassOf rdf:resource="#User" />
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="#hasChild" />
    <owl:minCardinality
rdf:datatype="&xsd:NonNegativeInteger">1</owl:minCardinality>
</owl:Restriction></owl:Class>
<rdf:Property rdf:ID="hasChild">
  <rdf:domain rdf:resource="#User" />
  <rdf:range rdf:resource="#User" />
</rdf:Property>
<owl:Class rdf:ID="Father">
  <rdfs:subClassOf rdf:resource="#Parent" />
  <rdfs:subClassOf rdf:resource="#Man" />
</owl:Class>
<owl:Class rdf:ID="Mother">
  <rdfs:subClassOf rdf:resource="#Parent" />
  <rdfs:subClassOf rdf:resource="#Woman" />
  <rdfs:subClassOf rdf:resource="#User" />
  <rdfs:disjointWith rdf:resource="#Father" />
</owl:Class>
<Mother rdf:ID="Lily">

```

这个例子是用 OWL 语言定义了智能家庭中家庭成员的简单的本体，描述了六类：User、Man、Woman、Parent、Father 和 Mother，一个属性 hasChild 和一个实例 Lily。其中 Man、Woman 和 Parent 都是 User 的子类，Man 和 Woman 没有交集、Parent 必须至少有一个 hasChild 关系。Father 是 Man 和 Parent 的共同子类，Mother 是 Woman、Parent 和 User 的共同子类，并且 Father 和 Mother 没有交集。最后声明一个 Mother 的实例 Lily。这个本体定义在智能家庭应用中有重要的作用，比如用户看电视的场景中，智能空间系统通过对用户本体的认知

得到当前用户的身份，从而根据用户不同喜好来定制电视的节目菜单，例如 **Man** 一般喜欢收看体育类或新闻类节目，**Woman** 一般喜欢收看娱乐类或连续剧节目，**Child** 一般喜欢收看卡通节目，**Father** 和 **Mother** 可以通过身份控制 **Child** 禁止收看色情、凶杀和暴力类的少儿不良节目。

■ 本体的冲突检测

本体的冲突检测的重要作用是辅助建立正确的、一致的本体，因而推理机能够发现定义中隐含的逻辑矛盾，从而使用者才能在本体上正确的设计实例应用。例如，如果在 List3.1 上加入如下 List 3.2 所示的定义说明 Lily 是 **Man** 的实例。

【List 3.2】

```
<owl:Individual rdf:about="Lily">
  <rdf:type rdf:resource="#Man" />
</owl:Individual>
```

由于 Lily 是 **Mother** 的实例，并且 **Mother** 是 **Woman** 的子类，因而 Lily 必须是 **Woman**，又因为 **Man** 和 **Woman** 没有交集，所以出现了实例体系上的冲突。冲突检测的基本步骤是按照知识层次的顺序（可以是类和属性体系=>实例体系）逐级用相关的产生式规则推理出可能产生冲突的隐含的信息，检测其中的矛盾。在某一层次的推理无法产生新信息之后转入下一层次，直到验证完本体中所有信息。但因为实例系统通常十分庞大，而且在本体的使用过程中会不停的变化，再加上实例系统的冲突对本体整体影响较小，推理机为了保证效率可以不去专门遍历实例体系而等到使用中发现明显的冲突再作处理。

■ 表达优化

用这些语言建立本体的方法也在很大程度上影响着本体的使用效率。尤其是在智能空间应用中，本体的建立过程没有明确的规范。所以，除了检测冲突确保本体正确以外，推理机还要能够把建立者零散描述的概念、关系、实例整合起来，组成一个知识库，精简掉冗余的内容，并针对应用来优化本体的组织，使得本体在应用时更有效率。例如，List 3.1 中定义的 **Mother** 本体：

```
<owl:Class rdf:ID="Mother">
  <rdfs:subClassOf rdf:resource="#Parent" />
  <rdfs:subClassOf rdf:resource="#Woman" />
  <rdfs:subClassOf rdf:resource="#User" />
  <rdfs:disjointWith rdf:resource="#Father" />
</owl:Class>
```

其中声明 **Mother** 是 **People** 的子类，并且 **Mother** 和 **Father** 无交集就是多余的，因为 **Mother** 是 **Woman** 的子类就必然是 **People** 的子类，而且 **Mother** 和

Father 的父类 Woman 和 Man 是没有交集的，则它们的子类必然也没有交集。在一个本体中不可能声明所有的类似关系，因为那样就会使本体过于庞大难以处理。在实际应用中，类、属性和实例之间的关系是非常复杂的，推理机要能够把这些错综复杂的关系整理清楚，用符合应用需求的格式组织本体中的信息，例如，用树结构来描述类和属性的层次关系，用图来表达实例之间的联系等等。这样在获取本体信息时就可以使用成熟的、低复杂度的算法，从而提高效率。

■ 本体融合

在智能空间开放的环境中，相关的和相似的本体之间的融合显得非常必要，然而利用人工的方来进行本体融合的效率低下。推理机在这方面可以辅助人们完成工作，操作者可以先指定不同本体中的一些同义概念，由推理机针对概念的联系把本体合并。下面举一个例子，本体如 List 3.3 所示：

【List 3.3】

```
<owl:Class rdf:ID="Male User" />
<owl:Class rdf:ID="Female User" />
<owl:Class rdf:ID="Husband" />
  <rdfs:subClassOf rdf:resource="#Male User" />
</owl:Class>
<owl:Class rdf:ID="Wife" />
  <rdfs:subClassOf rdf:resource="#Female User" />
</owl:Class>
<Wife rdf:ID="Mary" />
```

List 3.3 所示的本体侧重描述人另一方面的概念的联系，定义了 Male People、Female People、Husband 和 Wife 四个概念。使用者要把 List 3.3 所示本体并入 List 3.1 所示的本体，则只需要说明 Male People 和 Man 以及 Female People 和 Woman 的等价。推理机就会自动产生如下信息：Husband 是 Man 的子类，Wife 是 Woman 的子类，Mary 是 Woman 的实例，Husband 和 Wife 没有交集等等。

■ 验证知识的正确性

验证知识的正确性是指使用者提供用本体语言描述的知识，系统返回这些描述在这个知识库中是成立、可能成立或不成立的判定。在使用本体时，使用者经常需要用本体的知识来作判断。举例来说，对于 List 3.1 所示的本体，使用者要知道 Lily 是否有孩子，推理机就生成如 List 3.4 所示的描述。

【List 3.4】

```
<owl:Class rdf:ID="WhoHasChild">
  <owl:sameClassAs><owl:Restriction>
```

```

    <owl:onProperty rdf:resource="#hasChild" />
    <owl:minCardinality
rdf:datatype="&xsd;NonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction></owl:sameClassAs>
</owl:Class>
<owl:Individual rdf:about="Lily">
    <rdf:type rdf:resource="#WhoHasChild" />
</owl:Individual>

```

根据本体描述和已有的知识，得出 Lily 是 Mother 的实例，也是 Parent 的实例，Parent 有 hasChild 关系，符合 WhoHasChild 类的条件，推理机就返回肯定的答案，从而验证知识的正确性。

3.5 自定义规则推理的分析

在智能空间应用中，开发者能够根据不同的应用场景定义规则。在讨论自定义规则推理之前，我们首先探讨利用一阶谓词逻辑系统来进行情境表示的方法，然后阐明自定义规则推理的方法。

3.5.1 基于一阶谓词逻辑的情境表示方法

在第2章中，我们主要探讨如何采用本体的方法来描述智能空间存在的不同实体，例如用户、服务、活动、环境和平台等概念，这里需要解决的问题是如何表达一般陈述性的事实。举例来说，我们已经使用本体描述了用户和位置的概念，然而从本体中我们无法描述“秦伟俊在 FIT 3-526 房间”的事实，其中“秦伟俊”是用户的一个实例，“FIT 3-526 房间”是位置概念的一个实例，因此这里我们采用一阶逻辑的方法来描述这个事实。一阶谓词逻辑系统的基本形式是 $Predicate(Subject, Value)$ ，其中：

- $Subject \in S^*$ ， S^* 是主语集合，例如 *User*、*Location* 等。
- $Predicate \in V^*$ ， V^* 是谓词集合，例如 *locationIn*、*hasStatus* 等。
- $Value \in O^*$ ， O^* 是主语的属性集合，例如 *Open*、*Close*、*FIT 3-526* 等。

利用一阶谓词逻辑系统，我们能否表达智能空间中的不同实体之间的相互关系，以几个智能家庭中的情境信息为例：

1. 通过浴室中的温度计传感器获得情境数据“浴室的温度是摄氏 30 度”的事实，利用一阶谓词逻辑形式化表示为 $temperature(Bathroom, 32^{\circ}C)$ ；
2. 通过定位系统获得情境数据“Lily 在浴室”的事实，利用一阶谓词逻辑形式

化表示为 $locatedIn(Lily, Bathroom)$;

3. 通过 Lily 的工作计划表获得情境“Lily 按照菜单做菜”的事实，利用一阶谓词逻辑形式化表示为

$$cook(Lily, Food_1) \vee cook(Lily, Food_2) \vee cook(Lily, Food_3)。$$

从上述例子我们可以看到 1 和 2 是利用一阶谓词逻辑表示的原子事实 (AtomFact)，3 是利用连接词 (例如 \vee) 表示的复合事实 (ComposedFact)。另外，我们同样能够利用 OWL 语言来表示上述的情境事实，不过我们需要将谓词扩展成 OWL 语言中的属性。我们以“Lily 在浴室”为例，首先利用本体和 OWL 语言描述 Lily 和浴室的概念，然后描述两者的关系，如 List 3.5 所示。

【List 3.5】

```
<owl:Class rdf:ID="Mother" />
<owl:Class rdf:ID="Room" />
<Room rdf:ID="bathroom" />
<Mother rdf:ID="Lily">
  <rdfs:locatedIn rdf:resource="#bathroom" />
</Mother>
```

这里描述 Mother 和 Room 两个概念，Lily 和 bathroom 分别是 Mother 和 Room 的实例，通过定义 rdfs:locatedIn 描述两个实例之间的关系，来表达“Lily 在浴室”的事实。

通过上述的描述，我们自然就有产生疑问，通过一阶谓词系统将情境事实进行刻画之后，如何产生新的知识呢？举例来说，我们设想一下智能家庭中的一个场景，通过各种传感器和多模态应用软件检测到 Lily 坐在客厅里的沙发上，客厅里的电视机距离沙发近，并且电视机正处于开的状态，那么我们有理由相信 Lily 目前正在看电视。这样通过情境的推理我们获得新的情境知识，这就是第 2 章中所提到的派生情境信息。我们可以看出上述的推理过程中推理的依据是我们自定义出来的，针对这类自定义规则的推理方法正是下文要阐明的。

3.5.2 自定义规则推理方法

用户自定义规则的推理方法有多种，这里介绍其中的两种常用的方法：演绎法或者前向链算法 (Forward Chaining) 和归纳法或者后向链算法 (Backward Chaining)。演绎法是从一个初始事实出发，不断地应用规则得出结论 (或者执行指定的动作)。而归纳法则是从假设出发，不断寻求符合假设的事实。

用户自定义规则的推理系统包括三个部分：规则库 (Rule Base)、工作存储

区 (Working Memory 或者 Fact Base) 和推理引擎 (Inference Engine)。演绎法 (Forward Chaining) 一般采用 RETE 算法[44], 是美国 CMU 的 Charles L. Forgy 于 1982 年发明的目前效率最高的一个 Forward-Chaining 推理算法, 其核心将分离的匹配项根据内容动态构造匹配树, 以达到显著降低计算量的效果。此处不详细说明 Forward Chaining 和 Backward Chaining 的基本原理。

利用 Forward Chaining 和 Backward Chaining 算法, 开发者在设计智能空间应用领域时预先定义推理规则或者添加规则, 以实现情境感知系统的智能性, 下面给出用 EBNF 范式表示自定义规则:

```

=====
Rule      := bare-rule,
           or [ bare-rule ] or [ ruleName : bare-rule ]

bare-rule := term, ... term -> hterm, ... hterm      // 演绎法的推理规则
           or term, ... term <- term, ... term       // 归纳法的推理规则

hterm     := term
           or [bare-rule]

term      := (node, node, node)                       // 三元组模式
           or (node, node, functor)                  // 扩展三元组模式
           or builtin(node, ... node)                // 调用产生式原语

functor   := functorName(node, ... node)             // 结构化文字

node      := uri-ref                                 // 例如 http://foo.org
           or prefix:localname                       // 例如 rdf:type
           or ?varname                                // 变量
           or 'a liter1'                             // 字符串文字
           or 'lex' ^^typeURI                        // 输入的文字
           or number                                  // 数字
=====

```

以智能家庭应用场景为例来作说明, 下面给出若干用于 Forward Chaining 推理的推理规则表达式:

1. 规则一: 当用户的位置在客厅中, 并且客厅里放置电视机, 电视机处于开状态时, 则判定用户正在看电视节目。

```

=====
(?user, rdf:type, coss:User) ^
(?user, rdfs:locatedIn, ?drawing_room) ^
(coss:TVset, rdfs:locatedIn, ?drawing_room) ^
=====

```

```
(coss:TVset, rdfs:status, 'ON')
=> (?user, rdfs:status, 'WATCHINGTV')
```

2. 规则二：当用户的位置在厨房中，并且厨房里放有电磁炉，电磁炉处于开状态时，则判定用户正在做饭。

```
(?user, rdf:type, coss:User) ^
(?user, rdfs:locatedIn, ?kitchen) ^
(coss:induction_cooker, rdfs:locatedIn, ?kitchen) ^
(coss:induction_cooker, rdfs:status, 'ON')
=> (?user, rdfs:status, 'COOKING')
```

3.6 情境推理的设计与实现

上文中我们给出情境推理的定义，以及基于规则的情境推理两个基本功能：本体推理和自定义规则的推理，本节将给出基于规则的情境推理的设计方法和相关的实现工具。

3.6.1 情境推理的总体流程

情境推理的总体流程可以用如图 3.1 所示的状态自动机说明。

步骤一：从传感器或中间件获取情境信息，并利用本体和 OWL 语言对情境信息进行描述。

步骤二：将获取的情境信息存入知识库中，用于持久性保存。

步骤三：获取情境信息的过程触发情境推理行为，判断输入的情境信息，如果情境信息是已知类型的情境信息，则将情境信息进行本体验证；否则将为新的情境信息选择推理引擎进行情境信息推理。

步骤四：输入新的情境信息时，选择合适的推理引擎对数据进行验证和推理，检测是否生成新的情境知识。

步骤五：对情境信息的验证，保证情境信息的一致性和正确性。

步骤六：如果发现输入的情境信息与已保存的情境信息发生冲突时，转入冲突处理过程，否则结束情境推理过程。

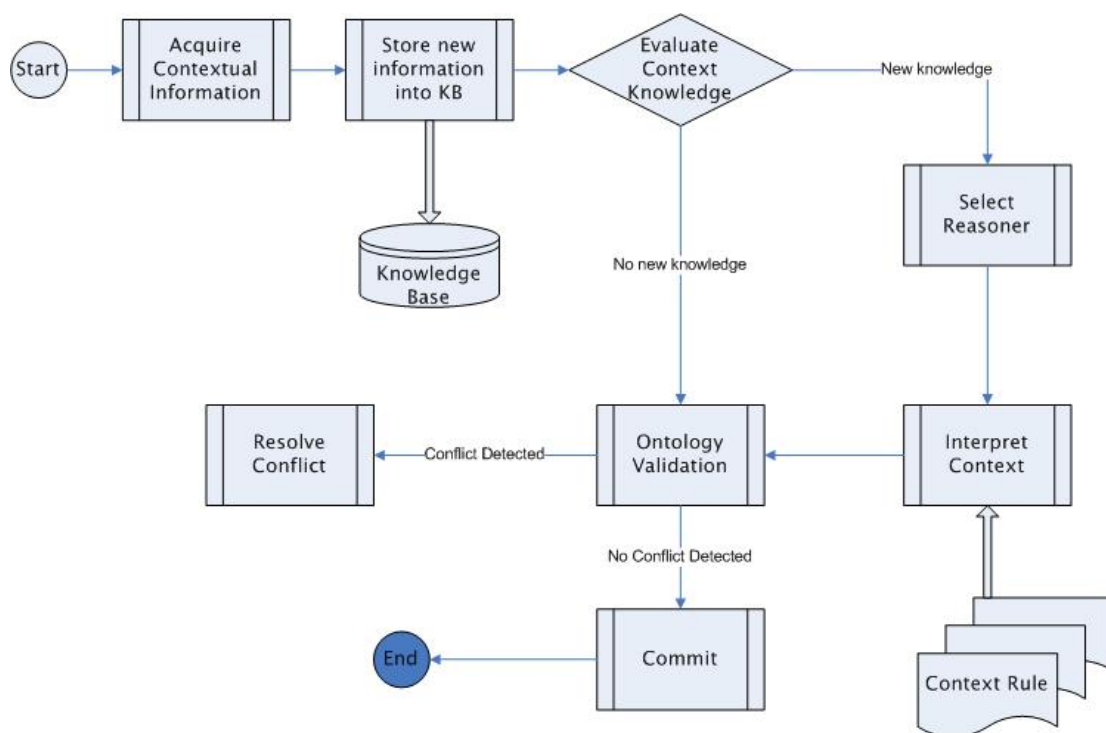


图 3.1 情境推理的总体流程图

3.6.2 情境推理的实现工具

情境推理的核心任务是本体的推理和自定义规则的推理，支持本体的推理以及自定义规则的推理工具很多，例如 Racer[45]、FaCT[46]、Jess[50]和 Jena[47]等。由于 Jena 软件包既支持本体知识的推理，又支持自定义的推理，因此我将介绍 Jena 的概况以及所使用的推理引擎。

Jena 是来自于惠普实验室语义网研究项目的开放资源，是用于创建语义网应用系统的 Java 框架结构，它为 RDF、RDFS、OWL 提供了一个程序开发环境。具体包括用于对 RDF 文件和模型进行处理的 RDF API、用于对 RDF、RDFS、OWL 文件（基于 XML 语法）进行解析的解析器、RDF 模型的持续性存储方案、用于检索过程推理的基于规则的推理子系统、用于对 Ontology 进行处理和操作的 Ontology 子系统、用于信息搜索的 RDQL 查询语言。Jena 的这些组成部分在解决语义网环境下语义检索中各司其职，起到重要的作用。

这里我将说明 Jena 的推理引擎，Jena 提供了基于规则的推理机（如 RDFS Reasoner、OWL Reasoner 等），包含了一般的推理功能，此外用户还可以注册第三方的推理引擎。如图 3.2 所示，推理机的工作原理是：推理机注册机制根据基

本 RDF 三元组描述（信息资源）和 Ontology（可选）创建出推理机,由此推理机可以生成包含推理机制的模型对象（Inference Graph, InfGraph），在 Jena 中，图（Graph）也被称为模型（Model），而表现形式为模型界面（Model Interface），然后可以使用 Model API 和 Ontology API 对此模型进行操作和处理，从而实现语义层面的信息检索。

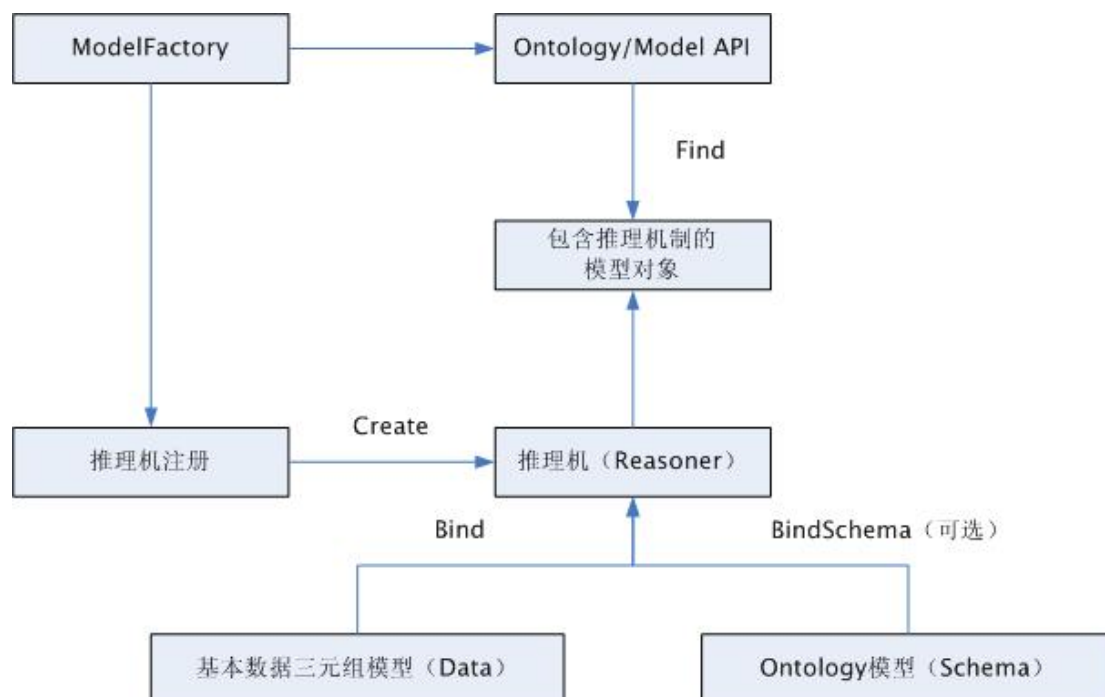


图 3.2 推理机的工作机制

在使用“推理机注册”创建推理机的时候，有三种方式可以使用：

1. 使用 Jena 自带的基于一般规则的推理机，主要有 RDFS Reasoner 和 OWL Reasoner，同时支持 DAML。这些推理机的推理规则都是基于一般用途的。
2. 使用基于自定义规则的推理机。推理机的内部是根据一定的触发机制通过对推理规则的解释从而实现推理的效果。具体的触发机制包括向前链引擎、向后链引擎和混合式规则引擎。
3. 使用第三方推理机。除了 Jena 自带的推理机机制之外，还有其他可以应用的推理机，这些推理机可以与 Jena 进行集成，所以对于 Jena 而言，称之为第三方推理机。比较典型的例如有 DIG 描述逻辑接口。

综上所述，Jena 提供了全面、灵活的推理机制和实现方法，比较好的解决了面向本体描述、基于规则的情境推理的难题。

3.7 本章小结

本章介绍了基于规则的情境推理机制。首先我们给出情境推理的一般化定义，然后阐明基于规则的情境推理的优点和任务，接下来我们就情境推理的两个重要作用进行了详细的分析：其中一个是针对利用本体描述的实体概念的推理，另一个是针对不同应用领域中用户自定义规则的推理，以及相关的推理机制。最后给出了情境推理的总体流程以及相关实现工具。

第4章 智能空间中的情境感知系统

情境感知系统开发利用环境中的情境信息来为用户提供有用的信息和服务。构建支持情境感知的软件平台仍然是情境感知研究中的重要课题。支持情境感知的软件平台用于收集和管理面向应用的情境信息，并对获取的数据进行抽象、验证和推理，最终向外部应用提供情境感知服务。本章提出基于本体描述的情境感知原型系统，然后给出情境感知在智能教室中的应用示例，用以说明基于本体描述的情境感知研究方法。

4.1 情境感知原型系统

4.1.1 应用需求

在第 2 章中，我们分析智能空间中的情境信息分类，从中我们可以看出智能空间中情境信息具有分布性和异构性的特点。为了能够利用情境信息，情境感知系统应该满足如下的应用需求[20][22][43]:

- 支持情境信息的高层抽象。智能空间布置有大量的传感器设备，情境感知系统的重要作用之一是将低层的传感器数据进行语义化的解释。例如温度传感器得到数据是 20，情境感知系统将数值标定语义信息，标定 20 的语义信息是温度是 20 摄氏度。因而，情境感知系统必须支持情境信息的高层抽象功能，将低层的数据转化为具有语义的高层信息。
- 支持情境数据的分布式合成、配置、存储和管理。智能空间系统是典型的分布式系统，因而情境数据具有明显的分布性特性，情境感知系统利用各种情境数据来进行处理分析，获取对智能空间中用户相关的信息，来增强系统的可用性，就必须能够合成分布式的情境数据，进而配置、存储和管理这类信息。
- 支持对于不确定性情境数据的情境解释或推理。从传感器中获得的数据由于测量的关系，数据具有不确定性和模糊性的特点。以定位传感器为例，目前实验室正在研发基于 RF 的定位传感器，在试验过程中，我们发现对于静止的物体，定位的精度在 10cm 以内，但是对于运动的物体

来说，定位精度在 1m 左右。在智能空间实际应用中，对于用户的位置信息是非常重要的。如果采用定位系统来对运动中的用户进行定位，获得位置信息的不确定性是很明显的。不仅如此，定位系统数据的获取还收到物理条件的影响，当空间中的障碍物过多时，定位系统中接受端收不到发射端的信号，这时无法得到位置信息，也就是说位置信息的获取不是连续的。

- 支持以事件驱动的机制来协助情境信息的收集、更新和推理。情境信息中包括变化频度比较频繁的实时的情境信息，按照情境信息的产生与时间的关系，又可以进一步划分为连续型情境信息和离散型情境信息。连续型情境信息采用流的方式从传感器中获得，离散型情境信息采用消息的方式从传感器中获得。情境信息的更新都是采用事件的形式发生的，情境感知系统支持以事件驱动的方式来辅助情境信息的收集和更新，进而驱动情境信息的推理。

这里需要指出的是本文给出的面向智能空间应用的情境感知原型系统没有涉及情境信息的不确定性。

4.1.2 设计原则

在设计情境感知原型系统时，我们考虑如下所示几点设计原则：

- 松散耦合
智能空间中的情境信息是高度动态和异构的。系统采用松散耦合的设计思想，使得系统能够选择合适的插件模块来满足情境感知应用的不同需求，提供对各种类型的情境信息的建模和推理功能，并且保证系统集成的耗损相对较低。
- 扩展性
组件化的抽象和封装保证了情境感知服务的扩展性。通过定制应用场景的配置文件和部署各种类型的传感器，情境感知系统能够获取智能空间应用中大量不同的情境信息来适应不同的应用需求。
- 透明性
系统设计上采用上层应用与底层服务的分离的思想。系统中提供情境信息的查询服务，对于上层应用来说，只需调用统一的查询接口就能够访问情境感知服务，底层的情境感知实现机制对于上层应用来说是透明的。

4.1.3 原型系统框架

我们提出了支持情境感知服务的原型系统，如图 4.1 所示，该系统是基于 Agent 实现的情境感知系统，系统框架有五部分组成：Context Wrapper Agent、Knowledge Base Agent、Context Provider Agent、Inference Engine Agent 和 Query Filter Agent 组成。

- **Context Wrapper Agent:** 用于从不同的信息源（例如各种传感器、设备、配置文件或者其他 Agent 等）获取和封装情境原始数据。
- **Context Provider Agent:** 用于抽象来自 Context Wrapper Agent 的原始数据，并利用本体来表示原始数据以便于知识的共享和重用。
- **Inference Engine Agent:** 提供多种推理引擎（例如 Forward Chaining 推理机、Backward Chaining 推理机、Jena 推理引擎包、Bayesian Network 推理工具等）对情境信息进行本体推理和自定义规则的推理。
- **Knowledge Base Agent:** 用于保存推理规则、观测结果和本体到数据库，以便于情境数据的持久性维护和管理。
- **Query Filter Agent:** 为上层应用提供查询接口和正则查询语言，使上层应用程序或者 Agent 能够查询或者订阅情境感知服务。

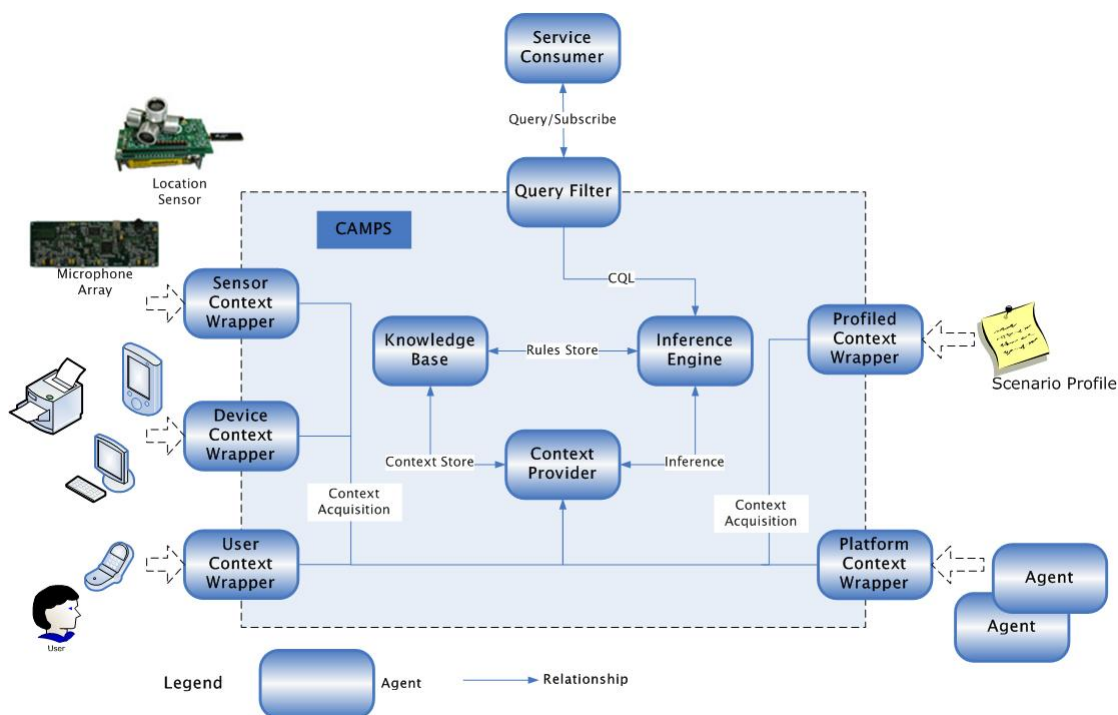


图 4.1 情境感知原型系统框架

4.2 情境感知在智能教室中的应用示例

智能教室是智能空间的典型应用，我们给出一个情境感知在智能教室中的应用示例——智能导播服务，其主要任务是在智能教室中提供转播机位的自动切换功能。我们首先给出智能导播的场景描述，然后利用本体来描述应用场景中的情境信息，接下来说明知识库的设计和情境推理的过程，最后说明情境信息查询的方法。

4.2.1 场景描述

我们首先给出智能空间中智能导播的应用场景，如图 4.2 所示。从图中我们看到，远程学生通过 Internet 参与到课堂中来。他们通过客户端软件，可以看到课堂上的情况，听到教师的讲解。同时，课堂的 PPT 投影通过截屏也显示在远程学生的电脑屏幕上。远程学生还可以像本地学生一样，申请发言，或是在电脑屏幕上对 PPT 投影进行标注。另外，课堂讲课的视频，通过课堂中的若干摄像头同时进行采集，系统通过分析当前环境的信息，选择出最佳的机位来显示给远程的用户。最后，整个课程会进行全程记录，以便今后的回放。

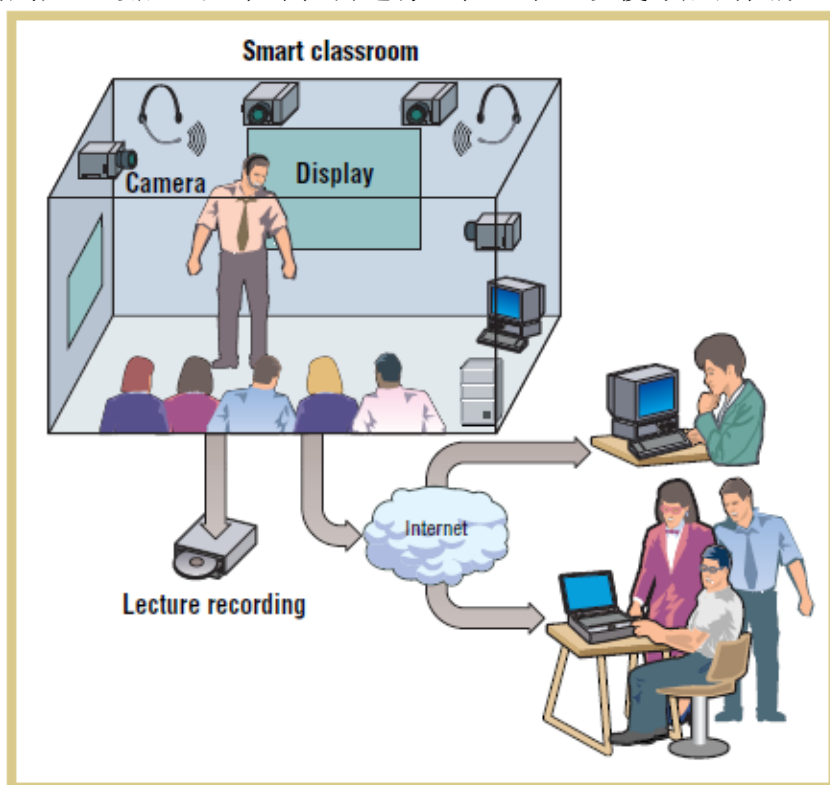


图 4.2 智能导播的应用场景

具体到课堂中，显示上使用一块 MediaBoard，它是一块电子白板，除了能够把投影显示在这块板上，教师还可以用不同颜色的触摸笔在板上书写。侧面墙上是 Student Board，远程学生的头像显示于这块板上。当他们中有多个人申请发言或者对 PPT 投影进行标注的时候，教师用激光笔指向某个远程学生的头像来选中是由谁发言或标注。

远程的学生需要能够看到智能教室中的情况，过程记录模块需要记录下当前的智能教室中的情况以便今后的回访，因此，当前教室的视频对于智能教室而

言是必不可少。我们的智能教室具有多种摄像头机位，它们同时采集视频。智能导播模块提供的功能有些类似于足球比赛的转播。当教室的焦点集中到黑板上时，智能导播会自动的把镜头切换到黑板机位；而当教师在拿起课本或者演示教具的时候，智能导播又会自动把镜头切换到演示教具的机位，依此类推。

智能导播模块可以跟踪当前智能教室内的视觉焦点所在，从而切换到相应的机位进行显示。我们经过分析，认为当前智能教室的视觉焦点主要由以下四个方面组成：

- 教师或者学生在黑板上书写。此时，教师或者学生是教学活动的焦点，摄像机应该给出黑板的近景，远程学生可以关注于教师和学生的活动。
- 教师正在讲台上讲课。此时，教师是教学活动的焦点，摄像机应该给出讲台的近景。
- 学生回答问题。此时，回答问题的学生是教学活动的焦点，摄像机应该给出学生的近景。
- 其他情况。此时摄像机可以给出教室的全景，让远程学生了解整个教学活动的全貌。

教师或者学生在黑板上书写，也包括黑板上演示的 PPT 进行了切换，概括而言，第一类的视觉焦点在于黑板；教师在讲台讲课，其中包括教师演示教具等，概括而言，第二类的视觉焦点在于给教师的特写；学生回答问题，当然，这第三类的视觉焦点就在回答这个问题的学生身上；最后，当视觉焦点不属于以上三类的时候，“其他”表示全景的镜头。

4.2.2 实现步骤

在介绍应用实现细节之前，我们给出利用情境感知系统来实现情境感知应用的总体步骤：

1. 情境信息的建模和描述。根据情境信息模型的定义，利用本体和 OWL 语言依据应用领域进行面向领域特定的本体建模。
2. 推理规则的定义。根据应用场景的不同，用户自定义推理规则，保存在知识库中。
3. 情境感知服务查询的描述。

4.2.2.1 情境信息描述

根据上述的应用场景，我们利用基于本体的描述方法和 OWL 语言来表示智能导播应用中使用的情境信息。关于智能导播服务相关的情境信息，我们总结如下：

➤ 用户实体

用户自身相关的情境信息有 User Profile、偏好 (Preference)、情绪 (Mood) 和角色 (Role) 等，其中 User Profile 记录用户自身的基本信息，用户的偏好记录了用户的个人喜好，情绪和角色是用户重要的情境信息。

在智能教室应用中，按照角色的不同，用户有教师 (Teacher)、本地学生 (Local Student) 和远程学生 (Remote Student) 等三类。在智能教室中，我们首先考虑教师用户的角色，教师的主要任务是授课，还有就是倾听学生的发言，通过 Microphone Array 传感器可以检测到教师的状态是讲课 (Speaking) 或倾听 (Silence) 两种状态，其次，教师讲课的时候除去授课，还可以在黑板上写字，还有时要翻动 PPT (通过语音控制或者计算机控制等)，因此还需要具有翻动 PPT (SlidingPPT) 的状态，而在黑板上写字的状态，可以通过 MediaBoard 板的消息检测到，因而教师有板书 (Writing) 的状态。教师还可以在讲课的过程中演示教具，因此教师还有演示教具 (ShowingRealia) 的状态。

考虑完教师之后，则需要考虑学生的状态。学生的状态比较简单，本地和远程学生基本类似，都需要具有发言 (Speaking) 和安静 (Silence) 两种状态，同时学生还有板书 (Writing) 的状态。有些不同的是，远程学生的发言是通过请求发言权后管理员批准后得到的，而本地则是直接站起来发言。

从上述的分析可以看出，用户实体对于智能导播的应用有用的属性主要集中在用户的状态 (Status) 和用户的角色 (Role) 上。

➤ 环境实体

环境实体主要包含位置实体、时间实体和环境的物理条件实体。在智能导播应用中，没有涉及到环境的物理条件，因而暂不考虑描述环境的物理条件。用户的位置信息事实上对于智能导播系统非常重要，因为在上课活动中，教师一般是上课活动的焦点，跟踪教师的位置有助于集中学生的注意力，同时用户之间的相对位置关系和用户与设备之间的相对位置关系是重要的情境信息，举例来说，教师的位置如果在 MediaBoard 的活动区域的话，教师进行板书的可能性是非常大的。另一个例子，教师与本地学生的位置较近时，教师向学生提问的

可能性就会增加。由于在智能教室中尚未考虑加入定位传感器的应用，所以没有考虑位置实体的使用。而时间本体在活动中得到应用，用于描述活动的起始和终止时间，同时也作为事件描述的属性之一。

➤ 活动实体

活动信息往往是智能空间中应用的重要线索。活动实体用于描述活动的状态（Status）、出席者（Attendee）、地点（Location）、起始时间（StartTime）和中止时间（EndTime）等信息。活动实体可以认为主要分为两类：预定的活动（Scheduled Activity）和推断的活动（Deduced Activity）。在智能教室应用中，教学活动（ClassActivity）可以认为是预定活动的子类。按照教学活动主题的区别，可以把活动状态分为讲课状态（Teaching）、教师与本地学生互动状态（Teacher2Local）和教师与远程学生互动状态（Teacher2Remote）等。

➤ 服务实体

服务（Service）是具备提供某种特定功能的计算实体，使用具有规范化定义的接口辅助用户完成自身任务或者活动。服务本体的概念利用三类概念进行解释，分别为 Service Profile、Service Model 和 Service Grounding。Service Profile 用于描述服务的功能，Service Model 用于解释服务的处理过程，Service Grounding 用于说明服务消费者访问该服务的实现细节。在智能导播应用中，尚未定义明确的服务应用。

➤ 平台实体

在智能空间中，我们认为平台（Platform）实体提供了对平台中包含的软件（Software）实体和硬件（Hardware）实体的描述。平台上的软件实体主要是如下几种方式体现的：中间件（Middleware）、操作系统（Operating System）、虚拟机（Virtual Machine）等。平台上的硬件实体主要是指设备上的硬件实体包含的属性包括产品参数和设备的性能参数等，产品参数生产序列号、设备型号、生产商、生产日期等信息，性能参数则说明设备的硬件配置，例如 CPU 的主频、内存的大小、硬盘存储的大小、联网的带宽以及电池性能等。在智能教室应用中，我们主要关心硬件的工作状态。在智能导播应用中，使用的硬件设备主要有 SmartBoard、Laser Pointer、Microphone Array、Projector 和 Camera 等。在上课教学活动中，这些硬件传感器应处于“开（On）”状态。

综上所述，针对智能导播应用，智能教室内的领域特定本体如图 4.3 所示，其具体利用 OWL 语言描述略。

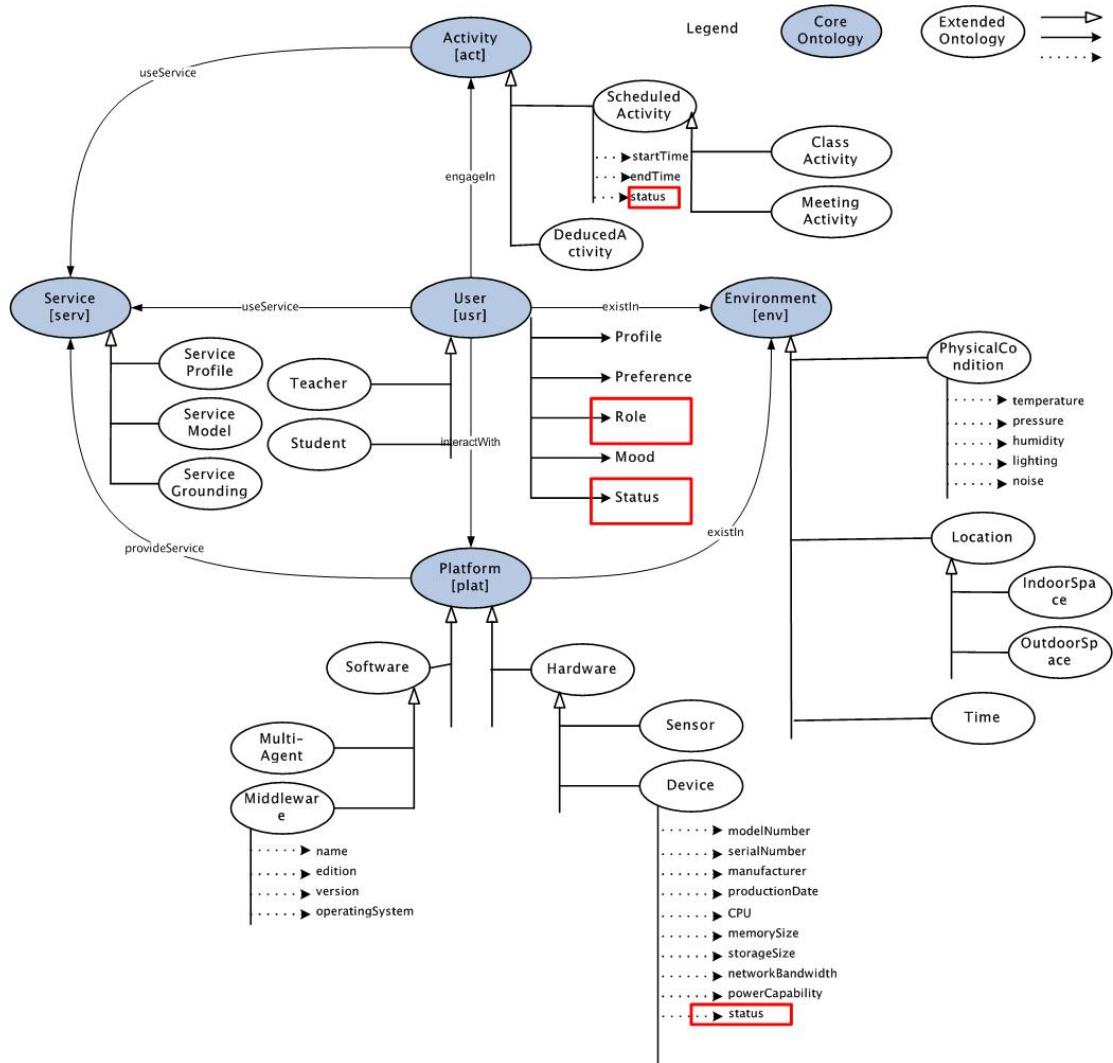


图 4.3 智能空间应用的部分领域特定本体关系图

4.2.2.2 推理规则定义

利用基于规则情境推理机制，不仅支持本体推理，还支持用户根据领域特定的自定义规则。在智能导播应用过程中，定义如下若干相对简单的自定义规则。基于如下所示规则，情境推理机采用前向链（Forward Chaining）算法进行推理。

1. 当教师正在板书的过程中，智能导播选择黑板的近景（BoardView）。

```
=====
(?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss#locatedIn,
```

```

http://media.cs.tsinghua.edu.cn/~pervasive/coss/Environment#SmartClassroom)
  ∧ (?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#hasRole,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#Teacher) ∧ (?user,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/Teacher#hasStatus,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/Teacher#Writing) ∧
(http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#SmartBoard
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#hasStatus,
'ON' )
⇒ (http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#Camera,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device/Camera#execute,
'BOARDVIEW' )
=====

```

2. 当本地学生正在板书的过程中，智能导播选择黑板的近景（BoardView）。

```

=====
(?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss#locatedIn,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Environment#SmartClassroom)
  ∧ (?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#hasRole,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#LocalStudent) ∧
  (?user,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/LocalStudent#hasStatus,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/LocalStudent#Writing) ∧
(http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#SmartBoard
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#hasStatus,
'ON' )
⇒ (http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#Camera,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device/Camera#execute,
'BOARDVIEW' )
=====

```

3. 当教师正在翻阅 PPT 的过程中，智能导播选择黑板的近景（BoardView）。

```

=====
(?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss#locatedIn,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Environment#SmartClassroom)
  ∧ (?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#hasRole,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#Teacher) ∧ (?user,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/Teacher#hasStatus,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/Teacher#SlidingPPT) ∧
(http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#SmartBoard
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#hasStatus,
'ON' )
⇒ (http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#Camera,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device/Camera#execute,
'BOARDVIEW' )
=====

```

4. 当教师正在讲台上讲课时，智能导播选择教师的特写 (TeacherView)。

```
(?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss#locatedIn,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Environment#SmartClassroom)
^ (?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#hasRole,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#Teacher) ^ (?user,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/Teacher#hasStatus,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/Teacher#Speaking) ^
(http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#MicrophoneArray
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#hasStatus,
'ON')
⇒ (http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#Camera,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device/Camera#execute,
'TEACHERVIEW')
```

5. 当本地学生回答问题时，智能导播选择本地学生的特写 (LocalStudentView)。

```
(?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss#locatedIn,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Environment#SmartClassroom)
^ (?user, http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#hasRole,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User#LocalStudent) ^
(?user,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/LocalStudent#hasStatus,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/User/LocalStudent#Speaking) ^
(http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#MicrophoneArray
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#hasStatus,
'ON')
⇒ (http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device#Camera,
http://media.cs.tsinghua.edu.cn/~pervasive/coss/Platform/Device/Camera#execute,
'LOCALSTUDENTVIEW')
```

4.2.2.3 服务查询描述

情境感知系统对上层应用提供统一的查询接口，在智能导播应用中，订阅关于 Camera 视角选择的查询，利用 OWL 表示为如下的查询描述。

```
<Device rdf:ID="Camera">
  <execute rdf:resource="#?">
```

```
</Device>
```

```
=====
```

用于表示查询当前 Camera 的视角。在情境感知系统内部，将该文件描述转化为情境查询语言，向情境信息数据库中查询该状态信息。

```
=====
```

```
SELECT ?view  
WHERE (<Camera>, <execute>, ?view)
```

```
=====
```

4.3 本章小结

本章主要给出了利用面向本体描述的情境感知系统应用的示例，首先分析了情境感知原型系统的应用需求和设计原则，并给出一个基于本体描述的原型系统设计框架。在此框架基础上，本章给出情境感知在智能教室中的典型应用——智能导播系统，首先定义场景的描述，然后利用本文介绍的情境信息模型对该应用场景进行情境建模，接下来描述应用于智能教室场景的自定义规则，最后给出智能导播利用情境感知服务的查询描述，整个流程目的在于阐明如何利用基于本体描述的情境感知系统来增强智能空间的应用过程。

第5章 总结和展望

5.1 论文工作的总结

智能空间是普适计算典型的应用实例，被普遍认为是研究普适计算理想的实验床，能够集中体现普适计算物理集成与自发互操作的本质特征及其关键技术。

智能空间研究的重要研究方面是情境感知。智能空间具备情境感知能力是指系统能够主动地获取和处理环境和用户相关的情境信息，前瞻地感知用户的意图，让用户透明地享受智能空间中的服务。作为智能空间智能性的集中体现，情境感知系统收集和管理面向不同应用领域的情境信息，抽象、验证和推理原始的情境数据以获取高层语义信息，最终向外部应用程序提供情境感知服务。

本文围绕面向智能空间的基于本体的情境感知模型中两个重要的子课题——情境信息建模和情境推理机制逐一展开，主要的研究内容总结如下。

5.1.1 基于本体的情境信息模型

移动设备在智能空间中广泛使用，用户可以随时随地进入或者离开智能空间，随时随地使用智能空间中存在的计算资源和服务，从而决定了智能空间环境是一个高度动态的、开放的、智能的分布式系统。为了让智能空间中用户透明地使用智能空间中的服务，智能空间应该具备获取和处理环境和用户活动相关的情境信息的能力。同时只有对智能空间中不同实体进行形式化的描述，才能便于计算机系统的处理，因而形式化描述智能空间中的实体，建立情境信息模型，成为近年来智能空间研究的重要课题。“本体是共享的概念模型的形式化的规范说明”是在语义层上描述智能空间中不同实体概念和关系的有力的工具。

情境感知的研究自从1992年Xerox Parc项目的研究工作起已有十余年的时间，目前成为普适计算研究的热点和重点，但是情境感知技术尚未达到非常成熟的阶段，研究仍然面临着许多的困难：1) 情境信息源的异构性；2) 缺乏软件支撑平台的支持；3) 缺乏通用的情境模型；4) 缺乏安全和隐私机制。

本文结合国际上已有相关工作的优秀思想的基础上，针对情境感知研究中缺乏通用的情境模型问题，我们的研究主要集中在如下几个方面：

- 1) 我们结合智能空间应用过程，分析面向智能空间应用的情境信息，并对

情境信息进行分类。根据情境的定义“情境是任何可以用来表现实体状态的信息。实体是与用户和应用之间交互相关的人、地点或者对象，包括用户与应用本身”，按照情境信息的类型划分，我们认为智能空间中核心的情境信息包含用户实体（User）、服务实体（Service）、环境实体（Environment）、活动实体（Activity）和平台实体（Platform）等基本类型。基于核心的情境信息描述，我们可以针对不同的应用领域进行扩展核心情境信息，并实例化不同实体的概念。

- 2) 为了形式化描述智能空间中的不同情境信息实体，我们采用本体的方法对情境信息进行形式化建模和描述，以便于智能空间中知识的共享和复用。智能空间中存在多种多样的情境信息，根据情境信息类型的不同，我们提出基于本体的情境信息分层模型，将智能空间的实体本体分为核心分体和扩展本体。在核心本体中，我们认为用户实体、服务实体、环境实体、活动实体和平台实体是核心本体的组成元素，并分析核心本体中各元素的相互关系；在扩展本体中，根据不同应用领域，我们继承和扩展核心概念，设计领域特定的本体概念。
- 3) 构建面向智能空间应用的本体库，利用 OWL 语言对情境信息模型中的不同实体进行表示，结合 Semantic Web 领域中针对 OWL 语言的应用与实现使得我们能够利用成熟和高效的软件模块（例如 Jena 等软件工具），提高了系统的开发和执行效率。

5.1.2 基于规则的情境推理机制

情境推理是情境感知研究中另一个重要的问题。智能空间的设计目标是让智能空间中的用户透明地使用智能空间中的服务和计算资源，这要求智能空间系统不仅能够准确高效地响应用户提出的应用需求，更高的要求是智能空间能够“感知”用户的需求，主动地为用户提供服务和计算资源，进而前摄地为用户提供有用的服务。因此智能空间系统应具备根据当前用户和环境资源的情境来获取智能空间中存在的高层语义，也就是说智能空间系统应当具备情境推理的能力，通过感知智能环境中各种实体的情境或者状态来生成高层语义信息。

目前的情境感知系统采用面向对象的程序设计语言（如 Java、C++）来定制情境推理模块的功能。面向对象的程序设计语言在接口层上保证了情境感知模块的复用，但是在知识表示层上缺乏表示能力，同时无法保证逻辑的完备性和

计算复杂度的可测性。

针对上述的问题，我们提出了基于规则的情境推理机制，研究主要集中在如下几个方面：

- 1) 由于我们使用本体来描述智能空间中的实体，利用本体语言来作为建模语言，因此在情境推理机制中我们利用情境推理来进行本体推理的功能，主要包括本体的冲突检测、表达优化、本体融合以及验证本体的正确性等功能。我们研究利用现有的 OWL 解析工具来进行针对 OWL 语言描述语法层上的验证。
- 2) 在智能空间应用中，开发者能够根据不同的应用场景定义规则。我们定义规范的规则描述语言来描述针对特定应用领域的自定义的规则库，并研究采用前向链和后向链的算法，生成带有高层语义的情境信息。

5.2 论文工作的贡献

本论文工作的贡献可以归纳为如下几点：

分析了智能空间中情境信息的类别，提出一种基于本体描述的情境信息模型，将智能空间中的实体分为核心本体和扩展本体。核心本体用以描述智能空间中抽象的、共有的实体，包括用户实体、服务实体、环境实体、活动实体以及平台实体。扩展本体针对不同的特定应用领域来继承和扩展核心概念，设计领域特定的本体概念。

构建了面向智能空间应用的本体库，利用 OWL 语言对情境信息模型中的不同实体进行表示，并存储在本体库中以备情境感知系统使用。我们结合了 Semantic Web 的现成技术来解析和使用基于本体描述的情境信息。

提出了基于规则的情境推理机制，将情境推理分为本体推理和自定义规则推理。使用本体推理来解决本体使用过程中本体的冲突检测、表达优化、本体的融合和验证知识的正确性等问题。使用自定义规则推理则根据不同应用领域中用户自定义的情景规则对底层的情境信息进行高层抽象和映射。

设计了情境感知的原型系统，利用基于本体描述的方法对智能空间中的典型应用——智能教室应用的不同实体进行本体建模，并利用规则描述语言来自定义智能导播应用中的规则，用以说明情境感知研究中采用本体描述设计方法的使用过程。

5.3 需进一步开展的工作

情境感知研究，我们计划继续开展如下几点研究：

- 1) 情境信息不确定性 (Uncertainty of Context) 的本体描述。不确定性是情境信息的典型特征，在从传感器设备中获取数据时，由于硬件精度等诸多因素导致获得的情境信息带有不确定的因素。在进一步的研究中，我们研究采用本体的研究方法来处理带有不确定性的情境信息。
- 2) 情境信息的质量 (Quality of Contextual Information)。情境信息涉及智能空间中不同的物理实体，在这类情境信息的收集、建模和推理的过程中，情境信息的错误率必然存在的。引入情境信息的质量，例如情境信息的新鲜度 (Freshness)、可信度 (Confidence) 等参数，有助于建立对情境感知服务质量的评估体系。

参考文献

- [1] Weiser M. The Computer for the 21st Century. *Scientific American*, 1991, 261(30): 94-104
- [2] Weiser M. The coming age of calm technology, *Beyond calculation: the next fifty years*, Copernicus, New York, NY, 1997
- [3] Kindberg T, Fox A. System software for ubiquitous computing. *IEEE Pervasive Computing*, 2002 1(1):70~81
- [4] NIST Smart Space Laboratory. <http://www.nist.gov/smartspace/>
- [5] MIT CSAIL AIRE (Agent-based Intelligent Reactive Environment) Project (previous Intelligent Room project), 2003. <http://www.ai.mit.edu/projects/aire/>
- [6] 谢伟凯. 智能空间关键支撑技术的研究: [博士学位论文]. 北京: 清华大学计算机科学与技术系, 2003
- [7] 毛雁华. 智能空间的软件平台及其资源管理的研究: [硕士学位论文]. 北京: 清华大学计算机科学与技术系, 2004
- [8] 陈恩义. 面向任务计算的研究: [博士学位论文]. 北京: 清华大学计算机科学与技术系, 2005
- [9] Microsoft Research, EasyLiving Project. <http://www.research.microsoft.com/easyliving/>
- [10] Stanford Interactive Workspaces Project. <http://iwork.stanford.edu/>
- [11] Shi Y C, Xie W K, Xu G Y. Smart Remote Classroom: Creating a Revolutionary Real-time Interactive Distance Learning System. In: Fong J (eds). *Proceedings The First International Conference on Web-based Learning (Hong Kong, Aug 2002)*. Berlin: Springer, 2002. 130-141
- [12] Warneke B, Last M, Liebowitz B, et al .Smart Dust: Communicating with a Cubic-Millimeter Computer. *IEEE Computer*. Jan 2001, 34(1), 44-51.
- [13] Jennings N R. On Agent-based Software Engineering. *Artificial Intelligence*, 2000, 117: 277-296.
- [14] Erickson T. Some problems with the notation of context-aware computing. *Communications of the ACM*, 2004. 102-104.
- [15] Dourish P. Seeking a foundation for context-aware computing. *Human-Computer Interaction*, 2001. <http://citeseer.nj.nec.com/dourish01seeking.html>
- [16] McCarthy J., Buvac S. Formalizing context (expanded notes). In Sasa Buvac and Łucja Iwanska, editors, *Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language*, Menlo Park, California, 1997. pages 99-135.

-
- [17] Castro P., Muntz R. Using context to assist in multimedia object retrieval applications. In Proceedings of the ACM Workshop on Multimedia Intelligent Storage and Retrieval Management, 1999.
- [18] Kindberg T., Barton J. A web-based nomadic computing system. *Computer Networks*, 2001, 35(4):443-456.
- [19] Schmidt A., Aidoo K. A., Takaluoma A. et al. Advanced interaction in context. *Lecture Notes in Computer Science*, 1999, 1707:89-??.
- [20] Dey A.K. Providing Architectural Support for Building Context-Aware Applications, PhD thesis, Georgia Institute of Technology, 2000.
- [21] Chen G. L., Kotz D. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, Computer Science, Hanover, NH, Nov 2000.
- [22] Dey, K. A., Abowd, G.D. Towards a Better Understanding of Context and Context-awareness. In the workshop on The What, Who, Where, When and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), Hague, The Netherlands, April, 2000. Also GVU Technical Report GIT-GVU-99-22. Submitted to the 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99), June 1999.
- [23] Schilit B., Adams N., Want R. Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, 1994.
- [24] Want R. Hopper A., Falcao V. et al. The active badge location system. Technical Report 92.1, Olivetti Research Ltd., ORL, 24a Trumpington Street, Cambridge CB2 1QA, 1992.
- [25] Wang, X. H., Zhang, D. Q., Gu, T., et al. Ontology Based Context Modeling and Reasoning using OWL. In Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications (PerCom2004), Orlando, FL, USA, March 2004. 18-22.
- [26] Gu, T., Wang, X. H., Pung, H. K., et al. Ontology Based Context Modeling and Reasoning using OWL. In Proceedings of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS2004), San Diego, CA, USA, January 2004.
- [27] Baader F., Calvanese D., McGuinness D. et al. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
- [28] Ranganathan A., McGrath R. E., Campbell R. H. et al. Ontologies in a Pervasive Computing Environment. In Workshop on Ontologies and Distributed Systems, Acapulco, Mexico, 2003
- [29] Thomson G., Terzis S., Nixon P. A. Towards Dynamic Context Discovery and Composition. The First UK-UbiNet Workshop, London, UK, 2003.
- [30] Hong, J. I., et al.: An Infrastructure Approach to Context -Aware Computing. *HCI Journal*, 2001, Vol. 16.

-
- [31] Ranganathan A., Campbell R. H. A Middleware for Context -Aware Agents in Ubiquitous Computing Environments. In ACM/IFIP/USENIX International Middleware Conference, Brazil, June, 2003.
- [32] Ranganathan A., Al-Muhtadi J., Campbell R. H. Reasoning about uncertain contexts in pervasive computing environments, IEEE Pervasive Computing, Volume: 3 Issue: 2 April-June 2004.
- [33] Chen H. An Intelligent Broker Architecture for Pervasive Context-Aware Systems, Phd Thesis, University of Maryland, 2004.
- [34] Borst W. N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse, Phd thesis, University of Twente, Enschede, 1997
- [35] Strang T., Popien C. L. A Context Modeling Survey. Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England, September 2004.
- [36] Henriksen K., Indulska J., Andry Rakotonirainy A. Modeling Context Information in Pervasive Computing Systems. In proceeding of 1st International Conference on Pervasive Computing, Zurich, Switzerland, 2002.
- [37] Schmidt, A., Laerhoven K. V. How to Build Smart Appliances. IEEE Personal Communications, August 2001.
- [38] Chen H., Finin T., Joshi A. An Ontology for Context-Aware Pervasive Computing Environments, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, May 31, 2004. 3(18):197-207.
- [39] 邓志鸿, 唐世渭, 张铭等. Ontology 研究综述. 北京大学学报, 2002, 38(5):930-938
- [40] 李善平, 尹奇纬, 胡玉杰等. 本体论研究综述. 计算机研究与发展, 2004, 41(7):1041-1052
- [41] Gruber T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing, Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, August 23, 1993
- [42] Gu T., Pung H. K., Zhang D. Q. A Service-Oriented Middleware for Building Context-Aware Services, Journal of Network and Computer Applications (JNCA) 28, no. 1 (2005): 1-18.
- [43] Fahy P., Clarke S. CASS: Middleware for Mobile Context-Aware Applications, MobiSys'04, 2004.
- [44] Forgy C. L. RETE: a fast algorithm for the many pattern/many object pattern match problem, artificial intelligence, 1982.
- [45] Volker Haarslev R. M. Description of the Racer System and its Applications. In proceeding of International Workshop on Description Logics (DL-2001), 2001.

- [46] Pan, J. Z., Horrocks, I. Reasoning in the SHOQ(D) Description Logic. Workshop on Description Logics (DL-2002), 2002.
- [47] Jena2: Semantic Web Framework: <http://www.hpl.hp.com/semweb/jena2.htm>
- [48] OWL Web Ontology Language, <http://www.w3.org/2004/owl/>
- [49] OWL-S, OWL-based Web Service Ontology, <http://www.daml.org/services/owl-s/>
- [50] Jess, the Rule Engine for the Java Platform, <http://herzberg.ca.sandia.gov/jess/>

致 谢

在硕士论文工作期间，导师史元春教授为我的研究工作创造了良好的条件，并在我困难的时候给了我许多鼓励，衷心感谢她对我的精心指导。同时她对严谨的治学、工作态度也是我的楷模。导师徐光佑教授对计算机科学发展方向的战略性眼光、和国际化视野给我很深的启示。他们的言传身教将使我终生受益。

感谢清华大学计算机系人机交互与媒体集成研究所普适计算小组的全体老师和通讯的帮助和支持！其中参与情境信息模型探讨的同学有索岳、李晶瑜、陈恩义等同学。

本课题研究受到 863 计划、教育部新世纪优秀人才支持计划等资助，特此致谢。

感谢计算机系老师和同窗们的关心和支持！感谢所有帮助过我的人们！



声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：

日 期：

个人简历、在学期间发表的学术论文与研究成果

个人简历

1981年7月22日出生于湖北省武汉市。

1999年9月考入清华大学计算机科学与技术系计算机科学与技术专业，2003年7月本科毕业并获得工学学士学位。

2003年9月免试进入清华大学计算机科学与技术系攻读博士学位至今。

发表的学术论文

- [1] Qin W J, Suo Y, Shi Y C, CAMPS: A Middleware for Providing Context-aware Services for Smart Space, International Conference on Grid and Pervasive Computing, 2006 (submitted)
- [2] 秦伟俊, 史元春, 相培峰. 大规模可扩展的网络教学系统. 清华学报 (已接受)
- [3] Xiang P F, Shi Y C, Qin W J. A Practical Software Architecture for Virtual Universities. In International Journal of Distance Education Technologies, vol.4, no.1, 2006
- [4] Xiang P F, Shi Y C, Qin W J, et al. CUBES: Providing Flexible Learning Environment for Virtual Universities, In Proceedings of the 3rd International Conference on Web-based Learning, Beijing, China, August, 2004