

# 整合 Agent 与语义 Web 服务<sup>\*</sup>

叶云 李舟军 李梦君 彭定

(国防科学技术大学计算机学院 长沙 410073)

**摘要** 目前的语义 Web 服务方法和标准提供了很多由 Agent 平台操纵的基础设施,这有利于 Agent 与语义 Web 服务的结合。本文分析了语义 Web 服务与 Agent 之间的关系,提出 Agent 的主要角色是用来封装用户的意图,协调用户的目标,并最终调用 Web 服务。我们利用当前的 Agent 技术与语义 Web 服务技术,设计出自己的平台,整合了 Agent BDI 模型与语义 Web 服务。

**关键词** Agent, 语义 Web 服务, BDI

## Integrate Agent and Semantic Web Services

YE Yun LI Zhou-Jun LI Meng-Jun PENG Ding

(School of Computer, National University of Defense Technology, Changsha 410073)

**Abstract** Integrating Semantic Web services and Agent can benefit from current Semantic Web services approaches and standards provide much of the general infrastructure currently handled by Agent platforms. This paper analyzes the relation between Semantic Web services and Agent, points that the main roles of Agent are encapsulating user's intentions, harmonizing user's goals, and ultimately invoking Web services. Taking advantage of current Agent technology and Semantic Web services technology, we design a platform that integrates Agent BDI model and Semantic Web services.

**Keywords** Agent, Semantic Web services, BDI

## 1 引言

为实现 Web 服务的发现、执行和组合自动化,借助语义 Web 的成果对 Web 服务进行研究,形成了语义 Web 服务。因为 Web 中的许多服务都体现出智能性的特征,例如对信息的搜寻,对服务的查找,满足用户的偏好,各个服务之间需要的交互,都正好体现出 Agent 的特点:自主性,协作性,适应性,社会性,所以越来越多的研究开始关注于 Agent 与语义 Web 服务的整合。用户跟个体 Agent 的连接可以是简单的网页形式,也可以是复杂的接口 Agent,并且,与传统的 C/S 模型相比,Agent 使用的 P2P 模型能更好的适应基于服务的 Web。

在本文中,我们讨论了语义 Web 服务与 Agent 的基本概念及其二者之间的关系,设计并实现了 Agent BDI 模型与语义 Web 服务的整合平台。

## 2 Agent 与语义 Web 服务

### 2.1 Agent

Agent 的标准化组织 FIPA (Foundation for Intelligent Physical Agents) 将 Agent 定义为: Agent 是某一领域中基本的行为者,它具有完成一项或者多项任务的能力,对外表现为一致性和完整性的执行模型,并能够与外部软件、操作人员和通信设施进行交互。Agent 具有感知性,能够对周围的环境进行感知,从而做出反应。Agent 的自治性使得一些由人工

干预的工作可以完全由 Agent 来替代。Agent 之间能够进行通信,因而可以构成一个 Agent 社团,加强对相关信息的交互。

Agent 体系结构定义和描述了组成 Agent 的基本成分及其作用,各成分之间的联系和交互机制,尤其是要描述如何通过感知到的内外部状态确定 Agent 应采取何种行动的算法,以及 Agent 的行为对其内部状态和外部环境的影响等等。根据对 Agent 内部数据结构的表示机制以及动作决策算法的差异,Agent 实现体系结构分为:反应式体系结构、知识型体系结构、认知型体系结构。反应式体系结构中 Agent 的动作决策不依赖于 Agent 的内部状态,而是一个从情景到动作的直接映射。知识型体系结构中 Agent 的动作决策依赖于 Agent 的内部状态,其内部状态被简单地定义为 Agent 所拥有的知识,Agent 的动作决策是一个基于其知识库的定理证明过程。认知型体系结构中 Agent 的动作决策依赖于 Agent 的内部状态,其内部数据结构由各种复杂的认知部件组成,Agent 的动作决策是对其内部多个相互关联的认知状态进行操纵的过程。本文主要关注于 Agent 的认知型体系结构。

### 2.2 语义 Web 服务

语义 Web 和 Web 服务的共同目标都是通过利用 Web 上人和机器都能够存取的内容,创建智慧自动服务及商务处理基础设施。因此考虑以上二者的结合,实现功能互补是一种自然的选择。将本体的概念和相应技术引入 Web 服务技术中,同时,由于本体具有丰富的语义和广泛的关系,它将变革

<sup>\*</sup> 本文受到国家自然科学基金项目(90104026, 60473057, 90604007)的资助。叶云 硕士研究生,主要研究方向为语义网、Agent 技术。李舟军 博士,教授,博士生导师,主要研究方向为进程代数理论,语义网与 Web 服务,安全协议的形式化验证,数据仓库与数据挖掘。李梦君 博士,讲师,主要研究方向为安全协议的形式化验证。彭定 硕士研究生,主要研究方向为网络路由器算法。

现有的 Web 服务,使之成为语义 Web 服务,使 Web 实现从自动化到智能化的转变成为可能。也就是说,通过合理的设计,可以充分发挥它们各自的优势,既结合语义 Web 的语义扩展,也结合 Web 服务的分布特性,最终提供一种基于语义的自动 Web 服务协作机制。

基于语义的 Web 服务可以做到:明确描述和推理事务之间的联系和规则;明确描述 Web 服务所执行的任务,实现 Web 服务发现的自动化;在 Web 服务所执行的过程中进行监控,并随时自动进行调整,实现执行自动化;将简单 Web 服务组合成为复杂 Web 服务,并实现自动化组合。

### 3 Agent BDI 模型与语义 Web 服务的整合

#### 3.1 Agent 与语义 Web 服务之间的关系

语义 Web 服务受益于 Agent 的动态创建,基于语义层的通信,判断和推理的能力,以及灵活性和健壮性等特性,从而使得语义 Web 服务能够更大程度、更全面地满足用户的需求。然而如何在更广泛的领域内将两种技术有效地结合起来仍然是一个严峻的课题。

有的研究认为,Agent 与 Web 服务之间是可以互相调用的,例如文[1]。他们认为通过提供合适的 WSDL(Web Services Description Language)到 ACL(Agent Communication Language)的映射,Web 服务调用 Agent 的能力是可行的,然而,因为 Agent 必须暴露它预先确定的行为,例如带有已知参数的操作名称,这与 Agent 的自治性是相背离的;如果 Agent 的行为不是确定的,那么 Web 服务必须具有类似 Agent 的行为去适应 Agent 自治性的响应,并且考虑到 Agent 代表的是用户的意图,所以 Agent 仅仅可以把它最普通的界面暴露给其它的 Web 服务,例如传递的消息和事件,然而这样的查询匹配是很不精确的。

本文认为 Agent 的主要角色是用来封装用户的意图,协调用户的目标,并调用 Web 服务。我们可以想象这样的一个例子:某用户需要从 A 地前往 B 地参加某国际会议,需要帮助用户定位用 OWL-S(Ontology Web Language for Services)描述的服务。对于用户的这一目标,是由很多的基本动作构成的,其中的交互过程也是基于很多的基本操作。用户需要自己将目标分解成所需的基本动作,然后交由相应的 Web 服务来执行,这给用户带来了极大的困难和不便,也没有体现出

语义 Web 服务自动化的特性。而 Agent 可以帮助用户完成目标的分解,经由简单的网页或者复杂的接口 Agent 的形式与用户进行交互,为用户提供便利,并最终调用相应的 Web 服务。目前,很多的研究,如文[2]利用 AI 规划来组合复杂的行为,当碰到一组目标时,这些规划代表用户执行。我们的工作是基于 Agent 中一类重要的认知型模型—BDI 模型展开的。

#### 3.2 BDI 模型介绍

Rao 和 Georgeff 于 1995 年提出了 BDI(belief-desire-intention)模型,这是一种重要的认知型模型。认知型体系结构的运作机理表明,具有该体系结构的软件主体不仅能够实现自己的行为决策,而且还能够表现出自发的行为特征。在该模型中,Agent 的体系结构包含了信念、期望和意图三个认知部件。信念定义了 Agent 对环境的理解和认识,它是 Agent 进行动作决策的基础。期望反映了 Agent 可能试图实现的任务和目标,体现了 Agent 的某种选择,但这种选择的约束是很弱的。Agent 的期望经过 Agent 承诺后上升为 Agent 的意图。意图是理解,分析和描述 Agent 行为决策的一个重要的认知部件,是对未来动作的合理选择,选择性是意图的本质属性。

BDI 模型能够充分揭示 Agent 的理性行为,这些高层、抽象的认知概念有助于我们研究 Agent 的状态、分析 Agent 状态与 Agent 行为之间的关系而不必具体了解 Agent 的内部实现细节。我们可以将这些认知概念作为一种自然、直观的科学抽象来理解、描述、规范、推理和预测 Agent 的体系结构、运行规律、内部状态变化以及 Agent 间的交互和合作,并且,BDI 模型具有良好的理论基础。

#### 3.3 BDI 模型与语义 Web 服务的整合

目前,在 Agent 领域,很多文献都花了大量的篇幅来描述 Agent 与 Web 服务应该如何整合。Web 服务提供的已有的基础设施,对于支持多 Agent 系统中的 Agent 交互十分理想。更加重要的是,Web 服务的基础设施被广泛接受并已被标准化,在不久的将来将会是主流的技术。另一方面,使用面向 Agent 的观点来看待 Web 服务,将为 Web 服务带来更大的发展和更多的方向<sup>[3]</sup>。

我们基于 Agent 技术与语义 Web 服务技术,设计并实现了一个 BDI 模型与语义 Web 服务整合的平台,如图 1 所示。

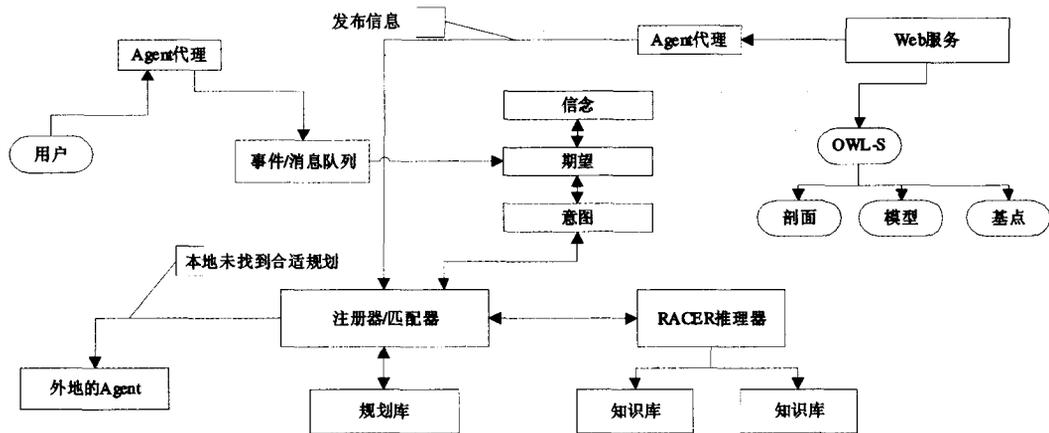


图 1 BDI 模型与语义 Web 服务整合的平台

Web 服务通过代理 Agent 向注册器注册,提供其基本的信息和调用的方式。这里,Web 服务并不是向传统的注册器

UDDI 中注册信息,而是向 Agent 平台中的注册器中注册,因为在传统的 UDDI 中,其注册的信息是基于语法的,不包含

语义的信息,并且在传统的 Web 服务中,消息的传输通过 SOAP 协议,在 Agent 平台中,需要提供 WSDL 到 ACL 的映射,其缺陷我们已在 Agent 与语义 Web 服务之间的关系中阐述。

用户需要某种服务时,由用户的代理 Agent 产生消息,放入事件消息队列中。每一个事件或消息是与一个期望所关联的,代表了用户想要完成的目标,为了推理 Agent 的性质和确保期望集合的相容性,目标应当显式地表示出来。Agent 为完成一个新的期望,就会产生一个意图与之对应。意图代表了当前执行下的一组规划。匹配器负责寻找和产生与用户目标最匹配的规划,使用 RACER 推理器进行推理,其过程是基于语义的。

当 Agent 在本地未找到合适的规划时,Agent 可以向其信任的 Agent 发送请求,当其信任的 Agent 有合适的规划时,则返回此规划,交由原 Agent 进行处理。如若没有,则返回失败的信息,整个过程结束。

下面,我们介绍各部件的功能和作用。

#### 知识库

里面存储了用描述逻辑形式化的语句,与 RACER 推理器连接。

#### 本体与 OWL-S

本体是描述概念及概念之间关系的概念模型,通过概念之间的关系来描述概念的语义。本体中最重要的概念是类,类在本体中描述某一领域相识个体的集合。为了创建本体,我们可以使用工具 Protégé。Protégé 是当前最流行的本体和基于知识的编辑器之一,提供了强大的函数来设计和实现本体,使用者可以利用 Protégé 的图形用户界面来编辑初始的和引入的内容。本体用来表达 Agent 特定的知识,在多元系统中 ACL 经常使用解析消息类型的公用属性并将它们归于不同的类的技术,我们借鉴这一技术,在通信中将本体领域相关的部分和领域无关的部分独立开来。这样,消息的意图和内容分离开来,当应用领域发生变化时,仅仅需要修改与领域相关的信息。

OWL-S 是指面向 Service 的 OWL(Web 本体语言),它通过服务的剖面(Profile)来发布和查找服务,服务的进程模型(Process model)描述一个服务操作的细节,而服务的基点(Grounding)提供了如何通过消息与一个服务互操作的细节。本体的推理可以通过 RACER 系统实现。

#### 信念

信念是 Agent 关于这个世界所具有的知识,是 Agent 进行推理和产生动作规划的基础。每个 Agent 都有一个知识库用来定义信念,信念是以断言的形式储存在知识库中的。

#### 期望

期望是用户想要完成的目标。Agent 从外部环境感知的事件产生的期望,与该事件具有相同的语法和结构,在此,我们不做区分事件与期望。期望与意图是一一对应的关系,Agent 为完成一个新的期望,就会产生一个意图与之对应,当期期望完成或者失败后,意图就会被删除。

#### 意图和规划

每一个意图对应一个期望,代表当前执行下的一组规划,控制 Agent 的行为。要想产生相关规划,Agent 先在本地规划库中寻找,规划的触发事件与事件队列中的事件进行匹配,如果产生了匹配的规划,则按照评估函数从中选择一个最优的规划。如果在本地规划库中没有找到合适的规划,Agent

向其信任的 Agent 发送请求,事件交付给它信任的 Agent 后,继续自己的动作,当相关规划的结果到达时,结果与此期望相关联。当没有活动的规划与最顶端的期望相关联时,意图被悬挂。

## 4 相关工作

C. Walton 在“Uniting Agents and Web services”一文中,给出了一个与 Web 服务相似等价的 Agent 的表示。Agent 在 Web 服务中有它们的副本,Agent 之间的通信采用了 Web 服务的通信方式,Agent 本体采用了 RDF/S 和 OWL。C. Walton 提出将 Agent 分解成 stub 和 body, stub 用来执行 Agent 交互协议和负责 Agent 之间的通信, body 封装推理进程,被编码为一组决定程序。Stub 和 body 的实现方式与 Web 服务相同。文[4]提出一种 BDI 风格的 Agent,它采用反应式规划,协调 Agent 的认知表示和 Web 服务的操作语义。这种协调的关键在于提供了被调用的 Web 服务的知识。目前的语义 Web 服务描述无法提供所有需要的知识,这就需要定义策略知识,Agent 可以用它来使得用户或者建议用户决定调用哪种服务。L. Bozzo, V. Mascardi 设计了 COOWS 平台,采用了 CooBDI 和 Web 服务技术,考虑到目前的 Web 服务技术在程序性知识方面存在缺陷, CooWS 给出了学习 Agent 及其运行环境的体系结构。Hong Zhu, Lijun Shan 在文[5]中,提出了一个 Web 服务的多元 Agent 概念模型,介绍了面向 Agent 的建模语言 CAMLE 中 Web 服务体系结构的图形模型。

**结束语** 本文讨论了语义 Web 服务与 Agent 的基本概念及其二者之间的关系,认为 Agent 的主要角色是用来封装用户的意图,协调用户的目标,并最终调用 Web 服务。我们利用当前的 Agent 技术与语义 Web 服务技术,设计出自己的平台,整合了 Agent BDI 模型与语义 Web 服务。与目前的 Agent 与 Web 服务整合技术相比,我们加入了语义的信息,使得匹配的过程不再是基于关键字的简单匹配,其匹配的结果更加精确。另外,当 Agent 在本地的规划库中没有找到合适的规划时,可以向其信任的 Agent 发送请求,这样,当用户的代理 Agent 无法完成用户的目标时,便可以委托其它的 Agent 完成,而不是立刻返回失败的信息,提高了平台的服务质量,增强了平台的灵活性。

## 参考文献

- Greenwood D, Calisti M. An Automatic, Bi-Directional Service Integration Gateway. In: Proc. Workshop on Web services and Agent-Based Engineering (WSABE'2004), 2004. URL: <http://www.Agentus.com/WSABE2004/program/7.pdf>.
- Sirin E, Parsia B. Planning for Semantic Web Services. In: Proc. Workshop on Semantic Web Services; Preparing to Meet the World of Business Applications, 2004. URL: <http://www.ai.sri.com/SWS2004/finalversions/SWS2004-Sirin-Final.pdf>
- Luck M, McBurney P, Shehory O, Willmott S. The AgentLink Community. Agent Technology: Computing as Interaction - A Roadmap for Agent-Based Computing. In: AgentLinkIII, 2005
- Dickinson I, Wooldridgel M. Digital Media Systems Laboratory, HP Laboratories Bristol. Agents are not (just) Web services; considering BDI agents and Web services: [Tech Report; HPL-2005-123]. July 2005
- Zhu Hong, Shan Lijun. Agent-Oriented Modelling and Specification of Web Services. In: WORDS, 2005. 152~159