

集成 Web Services 的多 Agent 分布式 计算机审计系统^①

A Distributed Multi-agent based Computer Audit System Integrated with Web Services

文巨峰 (南京审计学院计算机系 210029)
(东南大学计算机系 南京 210096)
汪加才 (南京审计学院计算机系 210029)
邢汉承 (东南大学计算机系 南京 210096)

摘要:本文提出了一种集成 Web 服务的多 Agent 分布式计算机审计系统,并分析了该模型中各模块的功能特点、Agent 行为的 WSDL 描述以及 Web 服务和 Agent 服务之间相互访问的消息转换。该系统具有良好的开放性、扩展性和自组织能力,为实现计算机审计智能化自动化提供了一种有效解决方案。

关键词:多 Agent 系统 Web 服务 计算机审计

1 引言

经济的数字化、网络化和全球化使得审计这项经济管理活动发生着革命性变化。然而,目前计算机审计难以适应这种变化以及审计发展需要。在审计实务范围日益广泛、各类审计规则日益繁多的审计环境下,合理分解审计实务构建不同功能模块来实现分布式审计模式,并设计实现自动化智能化程度较高的计算机审计系统是计算机审计的重要发展方向。

Agent 技术及多 Agent 系统^[1-2]是有效解决复杂分布式问题的计算模式之一。而且,FIPA-Agent 规范^[3]的出现,大大减少了构建多 Agent 系统的复杂性。Agent 独立性、多 Agent 系统的组织结构特点及其系统运作的拟人方式为设计实现审计平台这样一个复杂大系统提供了重要的模型参考。同时,为了使得构建的计算机审计系统支持先进的 Web 技术,能够实现与 Web Services 的相互访问,从而进一步扩展系统的能力和让系统中 Agent 作为共享的网络资源给非 Agent

客户端访问,本文提出了一个集成 Web Services 的开放式协同工作多 Agent 计算机审计系统结构模型。

2 相关的关键技术

2.1 FIPA-Agent 规范

传统的多 Agent 系统 MAS (Multi-agent System) 一般面向特定应用问题,采用紧耦合的体系结构。系统中各 Agent 需要拥有其他 Agent 的相关信息以供相互访问,而且 Agent 之间协作和交互是知识级,因而系统必须定义协同交互的全部细节,从而使得整个系统结构复杂而难以实现,并且难以适应应用需求的变化。单个 MAS 系统处理能力有限迫使多个系统之间交互增多。这样,Agent 以及系统之间访问接口和访问方式的标准化显得相当重要。因此,FIPA-Agent 管理规范得到了广泛的研究,并形成了系列标准。同时,这些

① 基金项目:江苏省高校自然科学基金项目(03KJD520106)

标准得到了广泛的应用和不断的发展^[4]。该标准使得网络上分布的各 Agent 具有“即插即用”特点, Agent 能够动态地查找和调用其他 Agent 的功能,从而使我们能够构建出具有动态的、松散耦合、高度适应性的多 Agent 系统。

该标准系统地定义了 Agent 管理参考模型 (Agent Management Reference Model)、Agent 命名、Agent 管理服务、Agent 管理本体 (Agent Management Ontology) 等几个重要方面。在多 Agent 系统平台中, Agent 采用包含 ACL (Agent Communication Language) 操作原语 (如, request、inform 等) 以及其他一些操作要素的 ACL 消息进行 Agent 的注册、Agent 服务的注册、查询和对其他 Agent 请求及响应。

2.2 Web Services

随着 Web 应用的不断发展,人们发现 Web 环境下不同平台之间的连接存在着巨大的困难。这时, Web Services 应运而生。Web Services 的主要目标就是在现有的各种异构平台的基础上构筑一个通用的与平台无关、语言无关的技术层,各种不同平台之上的应用依赖这个技术层实施彼此的连接和集成^[5]。Web Services 体系结构中主要三个组成部分分别为: SOAP、UDDI、WSDL。简单对象访问协议 SOAP (Simple Object Access Protocol) 定义了用户与 Web Services 通信的一种协议,其采用 XML 进行编码,经 HTTP 协议传送消息;统一描述、发现与集成协议 UDDI (Universal Description, Discovery and Integration) 是一种面向 Web Services 的信息注册中心的实现标准和规范,提供发布和查询服务描述的方法,使得用户能够访问 Web Services; Web 服务描述语言 (Web Services Description Language) 是描述 Web Services 的标准,它描述 Web Services 的接口和功能等信息。正因为 Web Services 提供了一套从服务描述、注册、发现、应用等一系列的标准,所以应用 Web Services 可以实现跨平台服务,为扩展应用系统的访问能力提供了标准。

在多 Agent 系统中集成 Web Services 的意义主要表现为: (1) 增强多 Agent 系统的开放性,提供非 Agent 接口访问服务,为审计资源的共享提供一条实用的途径; (2) 提供 Web 技术支持,这样多 Agent 系统中 Agent 可以访问网络环境中非 Agent 资源,提高系统综合处理能力。

3 集成 Web 服务的多 Agent 审计系统

充分考虑系统的智能性、开放性和系统持续可用的需要,本文提出一种集成 Web 服务的多 Agent 技术的分布式审计系统模型,如图 1 所示。总体来讲,系统分为五大部分,分别为人机接口、协调规划 Agent、审计 Agent 簇、Web Services Server、服务管理部分 (服务描述转换器、服务调用消息转换器)。其中,审计 Agent 簇由不同功能审计 Agent 组成。

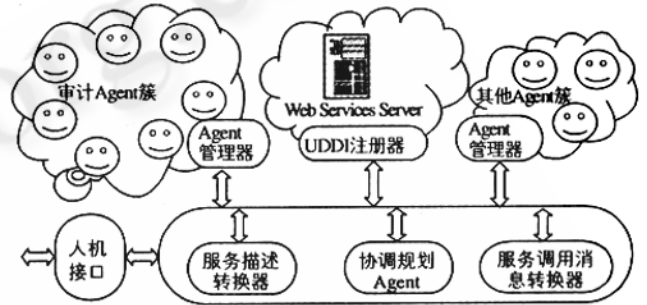


图 1 集成 Web Services 的多 Agent 审计系统模型

3.1 人机接口

人机接口是用户与分布式审计系统互操作的渠道,主要提交审计任务和完成相应审计任务初定服务筛选与静态流程设置。当服务选定及其参数设置完后,人机接口将审计任务转化为协调 Agent 能够识别的命令,并将这些命令、初始服务列表、参数设置、静态流程提交给协调 Agent。同时,人机接口还完成系统中 Agent 的知识和规则的更新。人机接口为非 Agent 客户端,可以是 Web 浏览器,这样大大简化了对客户操作要求。该客户端还可以被用于简单的审计对象或演示审计业务的一般操作流程等。

3.2 协调规划 Agent

协调规划 Agent 主要完成审计任务规划优化以及审计过程的总体协调,具体功能表现为:

(1) 任务规划优化。它结合 Agent 管理器提供的系统 Agent 的功能及状态、UDDI 注册器中提供的 Web Services 功能特点,依据人机接口提交的审计信息以及被审计单位的基本情况,规划出该项审计任务应有哪些 Agent 以及 Web Services 实体组成的动态联盟进行协作审计以及各个实体的任务列表、初始工作参数

等。当某些功能实体失效或出现故障时,它可以重新规划审计任务。

(2) 审计执行过程的协作协调。主要协调审计过程中的各个 Agent、Web Services 实体以及维护系统当前的运行状态信息等。整体上讲,协调 Agent 根据 Agent 各自的目标并充分考虑 Agent 之间的互利关系制定相应的协调方案,实现全局协调。当发生资源冲突或其执行任务无法进行下去时,Agent 就将相关事件提交给协调 Agent 进行处理,并将处理结果交给相应 Agent 强制执行,并向系统管理员通告这些异常情况和记录在系统的事件日志文件中。

3.3 Agent 管理器

Agent 管理器主要负责管理系统中 Agent 相关信息,为其他实体及协调 Agent 服务,主要功能有:

(1) 提供 Agent 目录服务。每个 Agent 采用 ACL 消息向 Agent 管理器注册、注销、修改、查询 Agent 服务。这样 Agent 和 Web Services 实体通过与 Agent 管理器交互就可以动态获取其他 Agent 的功能接口,从而与其他 Agent 进行通信以获取所需要的信息或调用其服务。

(2) 提供生命周期服务。Agent 可以动态地添加到多 Agent 系统中,但是需要在 Agent 管理器中进行注册。这样,Agent 管理器维持 Agent 标识 AIDs 目录和 Agent 状态信息。在状态改变时,Agent 管理器及时更新保存的状态参数值。

(3) 提供资源管理服务,负责系统中资源的注册服务。采用统一资源名字 URN (Universal Resource Name) 来命名系统中的各种资源,如文件、打印机等。同时,Agent 管理器保存有本地数据站点的有关信息,如本地数据的形式、数据库管理系统、业务数据库表结构、数据范围等。

3.4 审计 Agent

审计 Agent 簇中不同功能审计 Agent 分别运用审计算法完成协调规划 Agent 下达的审计子任务,并为系统提供异常分析数据。同时,审计 Agent 将自身的状态信息传递给 Agent 管理器,将审计分析结果和部分中间分析数据反馈给协调 Agent 以进行分析结果的融合。

为了有效地避免发生冲突,本模型中各审计 Agent 还负责自身具体审计活动的管理、调度和协调请

求以及 Agent 运行状态评估等工作,并做出审计活动的执行计划,将有可能引起冲突的计划交由协调 Agent 进行统一协调安排。审计 Agent 主要由四大模块组成,分别为:

(1) 通讯模块。主要将审计 Agent 的内部机制与外界隔离,通过收发信息来实现 Agent 以及与 Web Services 实体之间的交互。该模块还包含数据源的联接接口。一旦通信模块接收到消息,它就产生一个“事件收到”事件去触发能够处理该信息的部件。

(2) 环境感知模块。根据 Agent 的状态和通讯模块传送过来的消息确定事件的发生。它包含触发 Agent 事件的条件和结果。感知模块也是 Agent 从被审计单位获取电子数据的功能单元,当触发条件满足时,它被激活并接受数据流、格式化数据流,然后传递给审计模块。

(3) 审计模块。首先对业务审计数据进行规范化,然后依据审计算法对已经规范化的数据进行分析,确定这些数据中是否包含不符合操作规程的行为数据或异常情况。最后,将分析得到异常数据上传给协调 Agent,以便做进一步综合分析。为了提高审计 Agent 的自治能力,本系统的审计模块中包含由模糊神经网络组成的学习子模块,它接收和处理由一系列模糊值组成的模糊向量。通过对神经网络的训练,审计 Agent 可以根据不同的审计特征数据审计出不同的违规行为。

(4) 控制协调模块。控制任务的执行和系统各部件的运行。该模块采用基于时间的机制来协调和控制并发的各个事件。事件都带有时间、优先权等信息,它们可以在任务列表中进行排队,一旦某个事件被确认,就会给相应的执行部件发出消息。

审计 Agent 还含有知识库和状态库。知识库由协调 Agent 进行初始化,主要存放审计算法、审计模型、已有的知识和经过学习后获得的知识。状态库包括 Agent 标识 ID、运行状态的集合、与 Agent 执行相关的一些信息、原始数据、中间数据以及审计结果等信息。

3.5 Agent 行为描述与消息转换

如图 1 所示,Agent 以 Agent 服务的形式被 Web Services 或非 Agent 客户端调用,Web Services 也可以为 Agent 所用。为了实现用户或系统中功能实体及时有效地查询需求的系统服务,我们将 Agent 服务和

Web Services 分别在 Web Services 环境、多 Agent 环境中加以描述。同时,由于审计环境下 Web Services 实体和 Agent 采用不同的消息格式,因此系统需要中间转换模块来完成不同系统之间的数据传送和消息转换。集成 Web Services 服务的多 Agent 系统提供了两个模块,即服务描述转换器、服务调用消息转换器来实现异构系统的无缝连接。

(1) 服务描述转换器。服务描述转换器主要完成服务描述的转换,并提供注册信息。系统中 Agent 注册到 Agent 管理器中的服务注册信息经过该转换器转换为 UDDI 中的服务注册描述格式,并生成相应的服务描述文件(WSDL 文件);相反地,在 Web Services 在 UDDI 中的注册信息也相应地进行转换并注册 Agent 管理器中。其中,WSDL 文档主要包含 7 个重要构成元素,分别为 <definitions>、<types>、<message>、<portType>、<binding>、<port>、<services>,这些元素的具体功能详见文献^[4]。

(2) 服务调用消息转换器。服务调用消息转换器主要完成系统中 SOAP 消息与 ACL 消息之间的格式转换。其中,SOAP 消息主要由 SOAP 信封 Envelop 元素、SOAP 主体 Body 元素组成。ACL 消息主要包含特定的操作原语及其要素。对 GetProductSum() 接口访问的 SOAP 消息内容为:

```
< xml version = "1.0" encoding = "UTF - 8" >
< soapenv: Envelope... >
< soapenv: Body >
< GetProductSum soapenv: encodingstyle = " http://
schemas.xmlsoap.org/soap/encoding/" >
  < arg1 xsi: type = "xsd: string" > productname </
arg1 >
</ GetProductSum >
</ soapenv: Body >
</ soapenv: Envelope >
```

当 Web Services 客户端调用 Agent 服务时,发送的 SOAP 消息经由转换器转换为请求类(request) ACL 消息。AuditAgent 接受 request 消息后,进行内部处理,然后发送 inform 消息给出响应,该消息经消息转换器转换成 SOAP 消息并提交给 Web Services 客户端使用。转换后的请求类 ACL 消息为:

```
( request
```

```
: sender productclient@ sf. nau. edu. cn
: receiver AuditAgent@ sf. nau. edu. cn
: language fipa - slo
: protocol fipa - request
: ontology fipa - agent - management
: content " productname"
```

```
)
```

4 结论

本文提出的集成 Web Services 的多 Agent 分布式审计系统主要是从多 Agent 系统的角度来研究复杂审计任务的自动化智能化计算模型。该系统模型基于 FIPA - Agent 规范建构于与 FIPA - Agent 规范兼容且广为研究者亲睐的开放源代码 JADE 平台^[6]上,并集成 Java 环境下的 Web Services 技术。这些技术为设计实现切实可行的分布式审计系统提供了有力支持。由于审计环境是相当复杂的,审计规则也不断发生变化,本系统在借助人智慧实现人机协作审计来提高系统智能显得不足。下一步的主要工作进一步完善系统设计,并实现集成人类审计专家的两栖智能系统。

参考文献

- 1 Jennings N. R. , et al. A Roadmap of Agent Research and Development [J], Autonomous Agents and Multi - Agent Systems, 1998, (1).
- 2 Ferber J. Multi - agent Systems: An Introduction to Distributed Artificial Intelligence [M]. Harlow, England: Addison Wesley, 1999.
- 3 Foundation for Intelligent Physical Agents. FIPA Agent Management Specification [A]. <http://www.fipa.org/specs/fipa00023/>, June 2002.
- 4 Agentcities Task Force. Integrating Web Services into Agentcities Recommendation [A]. <http://www.agentcities.org/00006/actf-rec-00006a.pdf>, 2003.
- 5 柴晓路、梁宇奇编著,Web Services 技术、架构和应用 [M],北京 电子工业出版社,2003.1.
- 6 Collected Works. Exp: Special issue on JADE [J]. Telecom Italia Laboratories, September 2003, (3).