

基于 Agent 技术的 Internet 上构件服务框架研究^{*}

王 斌¹ 王建新¹ 夏东林² 张尧学² 陈松乔¹

(中南大学信息科学与工程学院计算机理论与软件研究所 长沙 410083)¹

(清华大学计算机系 北京 100084)²

Research on Agent Based Componentware Service Framework on Internet

WANG Bin¹ WANG Jian-Xin¹ Xia Dong-Lin², ZHANG Yao-Xue² CHEN Song-Qiao¹

(College of Information Engineering in Central South University, Changsha, Hunan 410083)¹

(Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084)²

Abstract With the development of component based development and network computing technology, people hope they can get components and component service quickly and accurately. After the analysis of traditional componentware provision approach and component mining technology, this paper proposes an agent based framework for component service discovering, provision and composition. The framework permits people to use componentware service directly or locate, understand componentware service on Internet and compose composite componentware service organically and dynamically with the help of agents. The composite componentware service can be used as new componentware service on Internet.

Keywords Component service, Composite component service, Semantic network, Agent

1 前言

软件生产脱离硬件的捆绑带来的变革,导致软件许可证发行方式的出现,这时软件生产者通过发行拷贝获得收益。而 Internet 技术和基于构件的开发方法带来了网络计算及网络化软件的发展,在这个趋势下软件生产者可以为用户提供有偿构件软件及其服务获得收益。随着 Microsoft 和 Sun 公司分别提出各自的分布式计算平台 .net 和 J2EE,以及 CORBA 组件规范的出现,这种软件发行方式已经趋向明朗,随之而来的是在 Internet 上分布大量的构件以及相关的网络服务技术。

构件挖掘是软件重用的重要研究方向,它主要研究在应用系统中发现和提取可能的构件并实现构件的自动组装。文[1]中将构件挖掘的过程描述为探索(Exploration)、设计(Excogitation)以及利用(Exploitation)三个阶段,这个过程旨在自动地、有组织地从一个富产构件的软件库中发现和抽取可用构件。文[2]将构件挖掘描述为软件产品及过程重用的活动。基于 Internet 这个巨大的软件库进行的构件挖掘是该研究领域中的一个研究方向,但是这项研究面临如下挑战:

①Internet 网页上一般分布着大量客户端构件,它们是 Internet 上构件挖掘的主要对象。由于构件发布方式没有遵循统一的描述规范以及网页开发技术不断更新,造成构件发现非常困难;

②大量的服务器端组件运行在应用服务器上,这些运行的组件对于 Internet 上的构件挖掘者来说是不可见的,这样就存在一个挖不到的死角;

③将基于不同运行环境的构件组装到同一个平台中要克

服巨大困难且会产生异常;

④将下载后的构件进行自动组装是难以实现的。根据 Post Correspondence Problem 的理论证明^[3],对于给定的一组构件,将它们进行自动组装,形成一个独立功能的构件是不确定的。

由于存在上述困难,本文寻求一种替代的解决方法——构件服务框架。软件重用在本质上是软件本身提供服务的重用。当我们将不同的构件组装成软件后,不同的构件彼此使用了对方的服务。另外,由于各种分布式计算平台的出现,人们有可能通过网络使用分布在 Internet 上运行的构件服务,所以可以把构件服务分为:构件之间彼此提供的服务及构件向人提供的服务。针对构件挖掘所面临的上述困难,本文提出的构件服务框架使人们可以直接使用发布在 Internet 上的构件服务,或通过 Agent 有组织地、动态地对分布在 Internet 上的各种构件服务进行准确定位、理解并组合成所需功能的组合构件服务。组合构件服务运行在分布、异构的计算环境中,为用户提供更强的功能和更大的灵活性。

本文还阐述了与构件服务研究相关的工作及其不足,给出了支持构件服务发现、发布及组合的框架模型,给出了构件服务框架的相关分析,最后是结论。

2 相关研究工作

与构件服务关系最为密切的研究活动集中在网络服务研究中。网络服务(Web Service)的研究旨在构造一种自包含(Self-Contained)、自描述(Self-Describing)的分布式 Internet 应用^[4]。构件服务从研究内容上涵盖网络服务的一部分,但是由于构件服务研究的服务粒度相对较小,而且构件服务具有

^{*} 本文研究由国家教育部重点课题(No. [2000]156 号)资助。王 斌 博士研究生,研究方向为 Agent 通信协作、数据挖掘。王建新 博士,研究方向为 QoS。夏东林 硕士研究生,研究方向为 Agent 通信。张尧学 教授,博士生导师,主要研究方向为多媒体网络服务质量控制、网络互联。陈松乔 教授,博士生导师,主要研究方向为软件工程。

更好的封装性,所以对构件服务的研究在某些方面具有与大粒度 Web Service 不同的特点。

构件服务框架的研究目标是构件服务能够被软件 Agent 理解,并实现构件服务间的自动互操作。在网络服务的应用研究中,Microsoft,IBM,Sun 等公司都提出了自己网络服务的体系结构^[5]。但是由于这些体系结构都紧紧地绑定在各公司的主导产品上,所以在本质上存在各种网络服务体系不能互操作的问题。另外,不同的 Web Service 体系结构都使用各自标准化后的术语和词汇,并没有考虑对服务的语义网络描述,以致于将绝大多数网络服务限制在人类操作而不是被 Agent 访问^[6,7]。网络服务应用研究中涉及到的相关技术,如:UDDI (Universal Description, Discovery and Integration), WSDL (Web Service Description Language) 对网络服务的互操作以及动态的服务能力匹配具有良好的支持,但是由于没有语义信息的支持,不能实现网络服务的自动互操作。在网络服务的理论研究中 DAPA 提出了能够基于语义对概念进行定义的 DAML-S (DAPA Agent Markup Language-Semantic) 语言。文[6]指出,在 DAML-S 应用模式中没有包含居中 (Intermediary) 角色,如代理等,所以 DAML-S 在实际应用中带来了体系结构不可靠的缺点。

根据上面的分析,本文在构件服务的框架中将结合网络服务的两方面研究优点,实现基于 Ontology 进行构件服务的语义描述、提供构件服务代理和构件服务注册功能、能够被人 and Agent 理解和使用的开放服务结构。

3 构件服务框架

下面将从 3 个方面给出构件服务框架结构:首先给出构件服务语义网络概念,构件服务语义网络是 Agent 进行构件服务发现、理解构件服务功能及构件服务间关系的重要依据。其次给出基于居中 Agent 的构件服务发现/发布模型,这个模型是基于 Agent 的构件服务搜索、服务协商、服务组合以及服务执行容错的基础结构。在前 2 个方面的基础上,框架的第三部分给出构件服务的动态组合模型,它是构件服务研究的最终目标。

3.1 构件服务语义网络

构件服务的标准描述是构件服务发布/发现的基础。网络服务应用中的 UDDI 支持在线网络服务注册、服务发布和动态服务发现。由于 UDDI 本身并不支持服务的语义描述,所以 UDDI 的服务搜索依靠预定义的服务分类关键词而不是网络服务内容的语义描述。尽管 Agent 可以搜索 UDDI 的服务注册库并且可以检索服务描述,但是需要用户来确认服务描述的意义以及对接口进行编程。为此,在一些网络服务体系结构中,WSDL 作为一种内容语言和 UDDI 绑定在一起。虽然 WSDL 在功能上类似于 DAML-S,但是 WSDL 中对服务的抽象程度要低于 DAML-S,而且对于服务存取的前置条件 (Preconditions)、后置条件 (Postconditions) 以及效果 (Effects) 并没有在 WSDL 中说明。针对构件服务语义网络,研究给出如下原则:

①构件服务的研究对象不同于网络服务中商业领域大粒度服务,它们不局限于某个特定领域,粒度相对较小。尤其在基于智能 Agent 技术的构件服务框架中,必须考虑基于不同领域 Ontology 的构件服务语义描述;

②需要以较高度度的抽象对构件服务进行描述,保证服务描述普遍适用;

③应该有合适的逻辑推理语言能够对构件服务间的依赖关系进行推理。

根据上述原则,我们将构件服务描述分为 2 部分:构件服务能力描述以及构件服务间依赖关系的描述。构件服务能力描述包括构件服务的功能、构件服务的服务执行模式及构件服务执行条件,而对构件服务间依赖关系的描述构成了构件服务的语义网络。这样就形成了构件服务描述的一个类层次结构,在这个类层次结构下具体的构件服务是这些类的具体实例,多个构件服务聚集形成了基于构件服务语义网络的构件服务库。图 1 是构件服务 Ontology 的抽象类层次结构图。

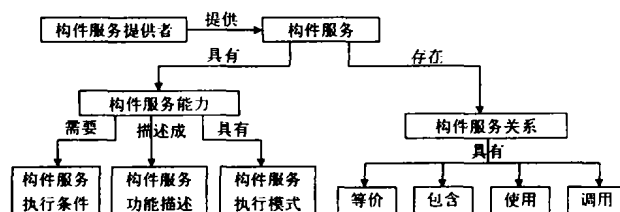


图 1 构件服务 Ontology 类图

构件服务库是构件服务提供者进行构件服务信息发布以及用户实施构件服务查询操作的重要软件结构。当一个新的构件服务被发布到构件服务库中时,构件服务的描述中并不包含该构件服务与构件服务库中的构件服务间依赖关系的描述。所以,构件服务库中需要提供一种能够确定构件服务依赖关系的算法以及遍历构件服务语义网络的算法,来快速定位所需的构件服务,并可对构件服务进行推理等操作。

3.2 构件服务发现/发布模型

一般使用构件的方法是首先发现构件,将其下载,然后进行组装。由于存在前述困难,构件发现中纯粹靠一些搜索技术在 Web 上以“拉”的方式获取构件收效甚微。而构件服务框架既可以利用目前存在的搜索技术,又能以一种“推”的方式为人或 Agent 提供构件服务。“推”的方式主要采取将构件服务主动发布到 Internet 上的某个中介,便于搜索技术更有效地发挥作用。构件服务框架采用“推”、“拉”相结合的原则进行工作。

Agent 作为软件实体具有一定程度的自治性,并且对具体的事件做出反应。由于 Agent 能够对消息进行处理,并能够以某种约定的语言进行交互,所以 Agent 可以实现较高度度的软件过程自动化,尤其在 Internet 上的信息搜索和信息发布中可以扮演重要角色。在构件服务框架中,采用 Agent 技术来克服 DAPA 的 DAML-S 不具备居中 (Intermediary) 角色的缺点。基于 Agent 构件服务发布和发现以及后述构件服务动态组合采用如下 2 种 Agent 协作模式,下面对其做具体描述。

图 1 中构件服务提供者将自己能够提供的构件服务描述注册到居中 Agent 中,用户 Agent 或者其他 Agent 将构件服务请求发送到居中 Agent,居中 Agent 匹配相应的构件服务,然后将构件服务请求发送到相应的构件服务提供者。构件服务提供者对构件服务请求做出处理后,将处理结果发送到居中 Agent,最后由居中 Agent 将处理结果再转发到构件服务请求 Agent。在这个过程中,居中 Agent 起到了消息代理的功能。

图 2 中构件服务提供者将自己能够提供的服务描述注册到居中 Agent 中,用户 Agent 或者其他 Agent 将构件服务请

求发送到居中 Agent, 居中 Agent 匹配相应的构件服务, 然后将匹配的构件服务提供者的相关信息发送到用户 Agent。用户 Agent 根据接受到的信息, 向构件服务提供者发送服务请求信息。在这个过程中, 居中 Agent 起到了服务推荐者的功能。

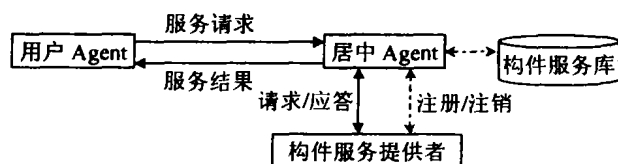


图2 构件服务代理模式

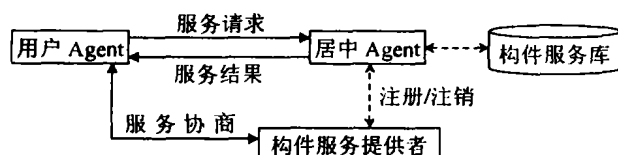


图3 构件服务推荐模式

在代理服务模式中, 由于服务请求和应答的 Agent 间彼此并没有直接进行通信, 所以它可以应用于一些对安全要求较高的个性化服务中, 且适合信息量较小的、服务模式相对固定的构件服务。在推荐服务模式中, 由于需要服务请求和应答双方的 Agent 进行通信, 所以这种服务模式更适合 Agent 间交互式、通信量较大的通信场合。所以, 代理服务模式更适合构件服务的发现, 而推荐服务模式更适合 3.3 节中的构件服务动态组合模型。

3.3 构件服务动态组合

构件服务组合分为静态组合和动态组合。静态组合是在组合构件服务时将构件服务预先进行绑定, 并写入组合构件服务文档中的过程。动态组合是在组合构件服务文档中预留个别构件服务没有具体指明, 而是在组合构件服务执行的过程中通过 Agent 之间的协作来动态确定, 或者预先绑定的构件服务在运行中发生“失效”时, 通过 Agent 的协商来重新确定一个替代构件服务保证组合构件服务顺利执行。构件服务动态组合过程模型如图 4 所示。

构件服务的动态组合过程分为 3 个阶段: 第一个阶段是构件服务需求阶段, 第二个阶段为组合构件服务阶段, 第三个阶段为构件服务重用阶段。具体的组合过程描述如下:

第一阶段 构件服务需求阶段。当用户有某个具体任务要完成时, 可以将该任务定义为由某个构件服务来实现, 然后直接利用用户 Agent 到 Internet 上去搜索。如果有合适的构件服务返回, 则直接执行构件服务; 如果没有合适的返回结果, 则将任务进行分解, 以形成组合构件服务描述文档。具体见图 4 中的(1)和(2);

第二阶段 组合构件服务阶段。对组合构件服务描述文档中的构件服务分别进行发现和服务协商, 动态完善组合构件服务描述文档。具体见图 4 中(3)、(4)和(5);

第三阶段 构件服务重用阶段。当成功地完成第二步后, 用户已经有了一个完备的构件服务描述文档, 这个文档中构件服务执行语义没有歧义, 对于构件服务的引用是准确的。用户可以将这个组合构件服务描述文档注册到某个构件服务库中, 为 Internet 上其他用户提供服务。具体见图 4 中(6)。

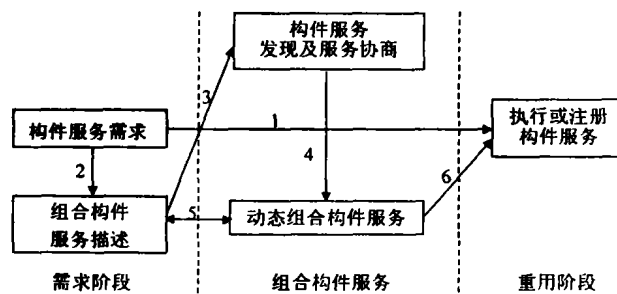


图4 构件服务动态组合过程模型

4 构件服务框架分析

4.1 构件服务库的一致性分析

当已经被确定能够成功实现某项功能的组合构件服务被注册到构件服务库中后, 构件服务库中被该组合构件服务使用的构件服务应该在构件服务关系描述中增加相应的描述。当该组合构件服务在执行的过程中由于某个构件服务的“失效”而不得不重新寻找替代的构件服务, 那么组合构件服务的描述将被修改, 相应的构件服务库中对应构件服务的信息将被修改。另外, 构件服务注销时也要对相关构件服务的描述进行修改。所以, 在构件服务注册、注销以及动态执行过程中, 必须考虑到构件服务库中构件服务关系的一致性。

4.2 组合构件服务的执行分析

独立构件服务的执行过程很简单, 而组合构件服务的执行需要进行任务调度。一般可以将组合构件服务的执行方式分为 2 种:

①集中控制模式。由一个构件服务提供者作为组合构件服务调度者, 其他构件服务提供者只能得到它们承担的具体任务描述, 而不必知道整个组合构件服务的描述。当某个构件服务提供者执行完它所承担的任务后, 将任务的执行结果返回到组合构件服务调度者, 然后由组合构件服务调度者将下一步的任务描述发送到另一个构件服务提供者, 直到任务完成。

②分布控制模式。与组合构件服务相关的构件服务提供者都得到完整组合构件服务描述文档, 它们将自己承担的任务完成以后就按照文档的要求, 将整个组合构件服务描述文档发送到下一步的构件服务提供者。当全部任务执行完成后, 由执行最后一步任务的构件服务提供者将结果返回到任务的提交者。

组合构件服务的执行过程要求必须为组合构件服务指定专门的构件服务提供者, 并且构件服务提供者要实施组合构件服务的注册、注销及执行容错管理。

4.3 构件服务框架中的协议

构件服务框架中通信协议是构件服务框架中各个软件体相互协作的重要依托。Internet 上通用的协议, 如 HTTP、TCP 等, 可以作为服务框架中信息的传输协议。

框架中第二种重要的协议是构件服务协商协议。一般构件服务协商发生在如下 3 种情况: 当用户通过用户 Agent 向某个构件服务提出请求时, 代表用户的用户 Agent 和代表构件服务的提供者要进行相应的服务协商; 当用户进行构件服务组合的时候, 对于组合构件服务中要确定的每一个构件服务, 都需要和提供该构件服务的提供者进行协商, 以

(下转第 21 页)

始刷新普通队列中的视图。

以上只是一个初步的 Web 仓储系统视图刷新原则。实际上,还有许多因素需要进一步考虑,如维护数据一致性时需要对系统进行关联性分区,不同分区的视图并行刷新,每个分区内部再如上考虑。另外,由于返回给用户的查询结果不一定是最新内容,所以用户希望数据的“过期时间”越短越好。这里就有一个数据年龄的概念。一般地说,物化视图的年龄等于其基础数据发生变化后至今的这段时间,而系统的年龄等于视图年龄按查询频率的加权平均值。若考虑数据年龄,则 Web 仓储系统视图刷新的目标就变为“在系统刷新能力和刷新时间的限制下,希望以较小的代价,获得最大的系统时新性,并且使系统年龄尽量小”。

结论 本文介绍了 Web 仓储在系统维护的过程中如何进行物化视图的刷新,并给出了相关算法。这些算法已经在我们的 973 项目的原型系统中得到初步使用和验证,取得了较好的效果。未来的工作,除了需要在我们的系统中进一步发展和完善相关算法外,还希望能够在大型、实用的 Web 集成环境中验证算法的可靠性和有效性。

参考文献

1 Brian E. Brewington and George Cybenko: How dynamic is the

(上接第 17 页)

决定构件服务是否可用;在构件服务执行的时候,当发生个别构件服务“失效”的情况下,构件服务提供者要能够向居中 Agent 进行服务查询,并和替代构件服务的服务提供者进行协商。

框架中的第三类协议是构件服务的调用协议。构件之间彼此能够通信是跨越 Internet 使用构件服务的基本要求。文[8]对主流的构件规范中存在“笨拙”总线的问题进行了详细的分析,提出采用移动 Agent 来替换传统的基于 RPC 的“笨拙”总线。但是,由于移动 Agent 对客户端和服务端端的软件环境都有特殊的要求,而且涉及到很多安全问题及多种通信协议,所以实现将需要复杂的环境支持^[9,10]。文[9]中实现了基于 SOAP(Simple Object Access Protocol)的服务代理结构,可以完成基于移动 Agent 的对象间异步通信方式,而且 SOAP 已经是业界接受的标准,所以在构件服务框架中我们采用 SOAP 作为构件服务间相互调用的标准协议。

结论 本文提出了一个 Internet 上的构件服务框架,它具有如下优点:①框架中的各软件体基于 Internet 上通用的协议及标准进行协同工作,使得框架的设计及实现简单;②该框架模型可以充分与人们对软件的需求相适应,为人们便利地获取各种构件提供的服务以及改善软件重用提供了新的方法。

参考文献

1 Spinellis D. Explore, Excogitate, Exploit: Component Mining.

Web? Computer Networks, 2000, 33: 257~276

- 2 Cho J, Garcia-Molina H. Synchronizing a database to improve freshness. In: Proc. 2000 ACM Int. Conf. on Management of Data (SIGMOD), May 2000
- 3 Cho J, Garcia-Molina H. Estimating frequency of change. Submitted for publication, 2000
- 4 Cho J. Crawling the Web: Discovery and Maintenance of a Large-Scale Web Data. [Ph. D. thesis]. Stanford University
- 5 Hirai J, Raghavan S, Garcia-Molina H, Paepcke A. WebBase: A Repository of Web Pages. In: Proc. of 9th Int. World-Wide Web Conf., 2000, 277~293
- 6 Liu Haifeng, Ng W-K, Lim E-P. Keeping a very large website up-to-date: Some feasibility results. In: Proc. of the 1st Intl. Conf. on Electronic Commerce and Web Technologies (EC-Web2000), Greenwich, UK, Sep. 2000
- 7 George B. Thomas Jr. Calculus and analytic geometry. Addison-Wesley, 4th edition, 1969
- 8 Taylor H M, Karlin S. An Introduction To Stochastic Modeling. Academic Press, 3rd edition, 1998
- 9 Xyleme L. A dynamic warehouse for XML data of the Web. IEEE Data Engineering Bulletin, 2001

IEEE Computer, 1999, 32(9): 114~116

- 2 Mili A, Mili F, Mili A. Reusing Software: Issues and Research Directions. IEEE Transactions on Software Engineering, 1995, 21(6): 528~562
- 3 Arora S. Probabilistic Checking of Proofs and Hardness of Approximation Problems, 1995: [A technical report based on the author's Ph. D thesis]. Contains a proof of the PCP theorem
- 4 Martin J. Web Services: The Next Big Thing, xml Journal 2. <http://www.sys.com/xml>
- 5 Myerson J M. Web Service Architecture. <http://www.webservicesarchitect.com/content/articles/webservicesarchitectures.pdf>
- 6 Sollazzo T, et al. Semantic Web Service Architecture-Evolving Web Service Standards toward the Semantic Web. <http://www.aifb.uni-karlsruhe.de/WBS/sha/papers/swobis-flairs.pdf>
- 7 McIlraith S, Son T C, Zeng H. Semantic Web Services. IEEE Intelligent Systems, Special Issue on the Semantic Web, 2001, 16(2): 46~53
- 8 LU Jian, ZHANG Ming, LIAO Yu, TAO Xian-ping. Research on Componentware Framework Based on Mobile Agent Technology. Journal of Software, 2000, 11(8): 1018~1023
- 9 Shin Nakajima. A SOAP-based Infrastructure for Service Broker. <http://pizza.cs.ucl.ac.uk/xse01/ready/5.pdf>
- 10 Wooldridge M J, Jennings N R. Pitfalls of Agent-Oriented Development. In: Proc 2nd Int. Conf. on Autonomous Agents (Agents-98), Minneapolis, USA, 1998. 385~391