

面向服务组合的主动 Agent 聚合框架及仿真实验分析

叶荣华 魏山山 钟发荣

(浙江师范大学计算机科学与技术研究所 金华 321004)

摘要 Web 服务的组合需要聚合足够多的能满足服务请求的元服务,由服务请求者自己来查找这些需要的服务是目前 SOA 的常用方法,但这种方法没有考虑服务本身作为主体也有主动推销服务的意愿。为此提出了一个面向服务组合的主动服务聚合框架,引入 Agent 作为 Web 服务的代理可以将服务抽象地封装成一个可以主动发现服务请求的智能实体,通过“意图-行为-实现”机制实现了服务能力与服务请求的匹配。最后通过一个仿真程序上的几组实验数据分析讨论了聚合框架的性能,验证了该方法的可行性。

关键词 服务代理,意图,行为,实现,聚合

中图分类号 TP393 **文献标识码** A

Web Services Composition-oriented Framework of Active Agent Aggregation and Simulation Experiment Analysis

YE Rong-hua WEI Shan-shan ZHONG Fa-rong

(Institute of Computer Science and Technology, Zhejiang Normal University, Jinhua 321004, China)

Abstract Web services composition need to aggregate adequate meta services to meet the service request. In SOA, the task to find these services is commonly fulfilled by the service requestor, but this method does not take into account the wish of service provider, who want to promote the sale of its services. To this end, an active service aggregation framework of Web service composition-oriented was proposed, which introduce Agent as a Web Service proxy. By Agent, services can be packaged into abstract entities which are able to find service requests actively. Through the “Intention-Behavior-Achievement” mechanism, the service capabilities and the service requests are matched. Finally, in order to verify the feasibility of the method, the performance of aggregation framework was analyzed and discussed by several sets of experimental data on a simulation program.

Keywords Service agent, Intention, Behavior, Achievement, Aggregation

1 引言

众所周知,1989 年 Tim Berners Lee 发明的 World Wide Web (WWW)^[1]极大地改变了之后人们获得和访问信息的方式。但 WWW 只考虑了信息源的分布,而不能集成分布的功能^[2],Web 服务技术的出现,改变了这一现状,因为它可以将分布于 Internet 上的异构系统(或组件)以一种松耦合的方式集成在一起。实践中,为了简化逻辑和提高服务的可重用性,单个 Web 服务的功能往往设计比较简单,而具体应用又相对复杂,所以需要多个简单服务组合成一个复杂服务^[3]。在传统 SOA 下的 Web 服务组合过程中,Web 服务本身始终处于被动的地位,它被动地等待服务请求者的发现和调用,而实现生活中,既有服务需求者主动去发现满足要求的服务,也有服务提供者主动去推销服务。因此,通过在 SOA 中引进 Agent 来实现主动服务的思想,并提出了一种面向服务组合的 Agent 主动聚合框架。

2 相关研究

Semantic Web Services (SWS)^[4]的提出为 Agent 技术提

供了一个新的应用领域,目前国内外有许多研究结合了 Agent 和 SWS 两种技术。Nicholas Gibbins 等^[5]用将通信消息类型描述与消息内容的领域含义描述相分离的方法,很好地结合了 Agent 和 Web 服务,并由此实现了一个基于 Agent 的 Web 服务系统。Francisco 等^[2]提出一个基于本体的智能 Agent 与语义 Web 服务集成框架,通过一种以本体为中心的方法,将 Agent 的自主性和灵活性与 Web 服务的开放性有机地结合在了一起。Yasmine Charif^[6]将 Agent 与 Web 服务完全融合为一个整体,并利用 Agent 的自我推理能力,提出和实现了一个动态服务编排的方法。这些方法虽然将 Agent 技术引入到了 Web 服务领域,但本质上仍只是由一个 Agent 被动地管理一个 Web 服务,只有当服务的请求者与之联系时才能发挥 Agent 的作用,而不能为 Web 服务的提供发挥主动作用。文献^[7]提出了一种需求驱动的主动网构实体聚合方法,一方面通过功能本体实现了 Agent 向需求聚合的“磁石效应”,另一方面用 AMD(Automated Mechanism Design)^[8]解决了 Agent 之间的协作问题,但其本质上更像是一种多 Agent 的协作机制,而非面向 Web 服务的方法。为了充分发挥 Agent

到稿日期:2009-08-26 返修日期:2009-11-11 本文受国家自然科学基金(60873234),浙江省自然科学基金项目(Y107625)资助。

叶荣华(1971-),男,副教授,主要研究方向为 Agent 技术、语义 Web 服务、面向对象软件工程等,E-mail: rhye@zjnu.edu.cn; 魏山山(1986-),男,硕士生,主要研究方向为智能网络计算等;钟发荣(1963-),男,博士,教授,主要研究方向为移动进程演算、Web 服务复合等。

的主动性,提出了一种面向服务组合的 Agent 主动聚合框架及相关模型,并用仿真实验初步验证了该聚合方法的可行性和有效性。

3 Agent 主动聚合框架

当一个服务请求不能由一个单一的 Web 服务完成时,首先要根据这个服务请求查找到足够多的能提供子功能的 Web 服务,然后将这些子 Web 服务组合成一个组合服务。在传统 SOA 中子服务的查找一般是由服务请求者完成,但单纯使用这种方法可能导致大量 Web 服务不能被有效利用。为此提出了主动服务的概念,即 Web 服务可以主动去发现服务请求,以便自身可以最大程度地被利用。

3.1 聚合过程概述

可以利用 Agent 来实现主动 Web 服务,每个 Agent 代表一个 Web 服务,作为服务的代理它理解服务的能力,作为 Agent 它可以与其它 Agent 交互以完成服务向服务请求的聚合,将这里的 Agent 称为 Web Service Agent(记为 WSA)。图 1 所示环境由一个服务请求(由实线椭圆表示)和大量 WSA(由小圆表示)组成,一个 WSA 发现了一个能够为之提供服务的请求时,就将其聚合在它周围,虚椭圆内的黑色小圆是已经聚合成功的 WSA,而白色的小圆表示还没有找到服务请求的 WSA,而半黑白小圆表示已发现服务请求,但还没有完成聚合的 WSA。

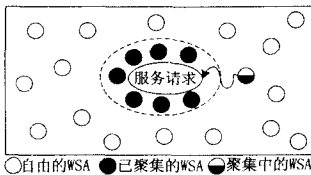


图 1 面向 Web 服务组合的 Agent 聚合过程

通过主动聚合的方法实现 Web 服务组合的首要任务是实现 WSA 向服务请求的聚合,即 WSA 找到服务请求,并成功向其登记。为了使 WSA 能找到自己能提供服务的服务请求,必须要有一套服务与服务请求匹配的方法。提出了一种基于环境本体^[9]的“意图-行为-实现”机制^[10],以较好地解决自主服务与服务请求的匹配问题。

3.2 “意图-行为-实现”机制

本节只简单介绍基于“意图-行为-实现”机制的服务实体与服务请求匹配方法,更详细的论述可以参考文献[10]。

环境本体有别其它本体的最主要的特征是可以很好地描述环境实体的状态变化,即可以形式化描述一个实体在某个原因驱动下的属性变化过程。图 2 是“意图-行为-实现”机制及其与环境本体的关系。

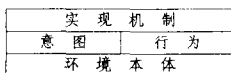


图 2 “意图-行为-实现”机制及其与环境本体的关系

首先在环境本体中引进了两个概念意图和行为。

定义 1(意图, Intention) 定义为一个五元组 (r, s_0, I, s_r, O) , 其中:

- r 是一个环境本体中定义的一个环境实体;
- s_0 是 r 的初始状态;
- $I = I_b \cup I_w$ 是输入信息的有限集,表示为了实现意图需

要提供的信息。其中, I_b 为基本输入集,它的每个输入的值直接给出, I_w 为等待输入集,它的每个输入的值要通过后续的消息交换得到;

- s_r 为 r 的目标状态,表示意图实现后资源所处的状态;
- O 为一个输出集,表示意图实现的过程中必须得到的输出信息。

Intention 表示期望在给定输入集的情况下,环境资源能通过某种状态转换关系从初始状态经某些中间状态到达目标状态,并得到一个输出集。

定义 2(行为, Behavior) 定义为一个七元组 $(r, s_0, I, S_m, s_r, O, T)$, 其中:

- r 是一个环境本体中定义的一个环境实体;
- s_0 是 r 的初始状态;
- I 输入信息的有限集;
- S_m 为行为作用下 r 的中间状态集;
- s_r 为行为作用下 r 的目标状态;
- O 为行为作用下 r 的输出集;
- $T \subseteq \Omega \times \Omega (\Omega = \{s_0\} \cup \{s_r\} \cup S_m)$ 为行为作用下 r 的状态转换关系。

Behavior 表示在环境本体的语义下,作用于环境实体,并使其按某种目标变化的能力。

接着来定义实现这个概念。

定义 3(实现, Achievement) 如果一个意图(记为 Int)和一个行为(记为 Beh)满足下列 3 个条件,则称行为 Beh 实现了意图 Int ,记为 $Achieve(Beh, Int)$:

- $r^{Int} = r^{Beh}$ (上标 Int 和 Beh 分别表示该符号代表是 Intention 中和 Behavior 中的相应标记,下同);
- $s_0^{Int} = s_0^{Beh}, s_r^{Int} = s_r^{Beh}$;
- $I^{Int} \supseteq I^{Beh}, O^{Int} \subseteq O^{Beh}$ 。

由上述定义可知,意图本质是期望一种环境实体在给定输入下能够从初始状态向目标状态转变并得到一个输出,而行为是环境本体中描述一个环境实体层次状态机的有意义子集,当一个行为与一意图满足一定的条件时,可以把行为看成是意图的一种实现手段。

3.3 WSA 能力与服务请求的匹配

可以用意图来描述服务请求,用行为描述 WSA 的能力,然后用实现机制完成 WSA 能力与服务请求的匹配。

定义 4(服务请求, Service Request) 定义为一组 Intention 的有限集,记为 $SR = \{Int_i | 0 < i \leq n\}$ 。

定义 5(WSA 能力) 定义为一个二元组 (B, MX) , 其中: $B = \{Beh_i | 0 < i \leq n\}$ 是一组 Behavior 的有限集合; $MX \subseteq B \times B$ 是服务实体内两个行为之间的消息交换关系的有限集。

定义 6(WSA 能力与服务请求的匹配) 给定 WSA 的能力 C 和服务请求 R , 如果对于 $\forall Beh \in C, B, \exists Int \in R$, 使得 $Achieve(Beh, Int)$, 则称此 WSA 能力 C 与服务请求 R 匹配。

由以上定义可知,服务请求可以定义为意图的集合,一个请求可能很复杂,可能期望多个环境实体产生各种变化。同理,服务能力可以定义为一组行为及其之间的关系。如果一个服务的每一个行为都正好能实现服务请求的一个对应意图,则可以认为一个 Web 服务能为一个服务请求提供服务。图 3 展示了服务能力与服务请求的匹配过程。

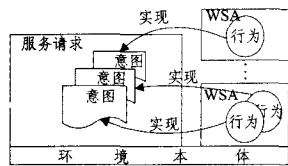


图3 WSA能力与服务请求的匹配

3.4 聚合成功的条件

在一个面向 Web 服务组合的 WSA 聚合框架中,聚合成功完成的标志就是要保证聚合的子服务最终可以组合成一个组合服务来满足服务请求,因此 WSA 的聚合必须满足以下两条件:

- (1)生成一个非空的 WSA 集 S ;
- (2)对任意服务请求中的意图 I ,都至少存在 S 中的一个 WSA 有这样一个行为 B ,使得行为 B 能实现意图 I 。

第 4 节的仿真实验就是基于这个条件来判别向一个服务请求的聚合平均在多少时间内完成以及环境中服务请求的平均完成率。对一个服务请求来说,聚合的 WSA 可能刚好达到要求,也可能超过需要的服务总数,这就需要为聚合的服务进行选择。本文主要关注聚合的基本框架以及在其下 WSA 的聚合效果,而聚合服务的选择不是本文研究的重点。

4 仿真实验及分析

为了说明聚合框架的可行性和有效性,本节通过几组仿真实验来度量该方法在不同参数下进行聚合时的时间开销、服务请求完成率和 Web 服务利用率。需要指出的是这个实验是用一种简化的方式来实现服务请求与服务能力的匹配,而不是真正基于“意图-行为-实现”机制的服务请求与服务能力的匹配,这里重点关注聚合框架在宏观上的效果。

4.1 实验准备

由于目前构建具有大量服务请求和 WSA 的环境有一定困难,因此本实验是通过在一台计算机上的仿真程序完成的。仿真实验程序由 Visual C# 开发,运行在 Windows XP Professional SP3 下的 .Net Framework 2.0 上,CPU 为 Intel Pentium(R)1.7GHz,内存为 512M。

仿真程序用一组随机产生的相同范围内的正整数来模拟服务请求的意图和 WSA 的行为。假设数值 v 代表行为 B ,而数值 u 代表意图 I ,如果 $v=u$,则称行为 B 实现了意图 I 。程序中有 5 个参数可调,它们的意义分别如表 1 所列。

表 1 仿真程序参数的含义

参数	含义
服务请求个数	环境中服务请求的数目
最大意图数	服务请求最多可以包含的意图个数
WSA 个数	环境中 WSA 的数目
最大行为数	WSA 最多可以包含的行为个数
不同意图(或行为)数	环境中最多包含的意图(或行为)个数

4.2 实验数据及分析

根据不同的参数,做了 3 组实验,其中每一组实验的数据都是重复 100 次后的平均值。首先,选择表 1 中 5 个参数分别为 10,10,100,3 和 20 作为基准值。当调整参数时,分析聚合效果的变化规律。设计了 3 个指标说明框架的效果,它们分别是完成向一个服务请求聚合的耗时、服务请求完成率和 WSA 利用率。

表 2 显示了在保持其它参数不变的情况下改变服务请求的个数后,时间开销、服务请求完成率和 WSA 利用率的变化情况。通过分析数据可以发现随着服务请求个数的增加,耗

时没有明显改变,而服务请求的完成率成缓慢下降趋势,而 WSA 的利用率成缓慢上升趋势,表明在 WSA 数量较小的情况下,单纯增加服务请求数量不能明显提高性能。

表 2 服务请求个数对聚合效果的影响

服务请求个数	完成向一个服务请求聚合的平均耗时(ms)	服务请求平均完成率(%)	WSA 平均利用率(%)
10	174.26	13.5	3.77
20	168.24	8.05	4.01
30	195.13	6.63	4.63
40	206.22	7.13	6.71
50	211.63	5.26	5.62
60	225.71	4.683	6.1
70	215.59	4.357	6.4
80	227.94	3.76	6.22
90	227.78	3.856	7.03
100	246.08	3.62	7.39

表 3 显示了在保持其它参数不变的情况下,改变 WSA 的个数时,时间开销、服务请求完成率和 WSA 利用率的变化情况。通过分析数据发现随着 WSA 个数从 100 增加到 1000,需求的完成率成明显的上升趋势,而耗时和 WSA 利用率出现不规则变化,但两者依然在可接受的范围内,并且当 WSA 个数为 1000 时,服务请求完成率达到 99.5%,暗示增加 WSA 的数量在不明显影响其它性能下可以有效地提高服务请求的完成率。

表 3 WSA 个数对聚合效果的影响

WSA 个数	完成向一个服务请求聚合的平均耗时(ms)	服务请求平均完成率(%)	WSA 平均利用率(%)
100	173.29	13.5	3.71
200	336.94	36.9	6.159
300	488.76	57.8	6.91
400	663.3	74.1	7.04
500	666.69	82	6.134
600	794.66	88.4	5.55
700	123.92	94.4	5.21
800	118.74	96.8	4.59
900	207.26	97.8	4.1
1000	502.37	99.5	3.87

表 4 显示了其它参数调整时聚合效果的变化。通过分析知道当服务请求个数和 WSA 个数都较大时,聚合可以得到较理想的服务请求完成率和 WSA 利用率,如表 4 第 2 行所列,当需求为 100 个而 AWS 为 1000 个时,需求的平均完成率为 81.02%,而 WSA 的平均利用率为 24.645%。另外,最大意图数,最大行为数和不同意图(或行为)数的增加会降低服务请求的完成率和 WSA 的利用率。

表 4 其它参数对聚合效果的影响

服务请求个数	最大意图数	WSA 个数	最大行为数	不同意图(或行为)数	完成向一个服务请求聚合的平均耗时(ms)	服务请求平均完成率(%)	WSA 平均利用率(%)
10	10	100	3	20	174.26	13.5	3.77
100	10	1000	3	20	702.73	81.02	24.645
10	10	100	3	30	189.71	5.5	1.15
10	5	100	3	20	202.97	18.6	3.65
10	20	100	3	20	202.71	7.3	2.53
10	10	100	5	20	230.28	5.4	1.23

通过以上实验数据分析可知,当参数在一个比较合适范围内时,聚合框架的聚合可以达到比较好的聚合效果。并且本实验只考虑了在确定服务请求数和 WSA 个数下的一次完全遍历的聚合效果,实际应用中当一个服务请求不能完成聚合时会解散已经聚合的 WSA,这些服务请求和 WSA 又可以重新进入新一轮聚合。由此可见,本文提出的面向服务组合的 Agent 聚合框架可以达到预期的聚合效果。

结束语 Web服务的组合一般有两种方式:编制(Orchestration)和编排(Choreography)^[11],但不管是哪一种,都必须找到一定数量的整体能满足服务请求的元服务。在目前的SOA体系中这一工作基本上是由服务提供者(或其代理)完成的,这样一种架构不能满足服务提供者主动推销服务的需要。本文将Agent作为Web服务的代理,利用Agent的智能性和自治性提出了一个面向Web服务组合的主动服务聚合初步框架,并通过仿真实验的性能分析很好地说明了该框架的可行性。

使用该方法进行服务聚合的基础是环境本体上的“意图行为-实现”机制,但基于该机制的服务能力与服务请求的匹配本质是服务与请求的功能匹配,并没有考虑服务的QoS因素,下一步的研究工作将会考虑这一点。

参考文献

- [1] Berners-Lee T, Cailliau R, Luotonen A, et al. The World-Wide Web[J]. Communications of the ACM, 1994, 37(8): 76-82
- [2] García-Sánchez F, Valencia-García R, Martínez-Béjar R, et al. An ontology, intelligent agent-based framework for the provision of semantic web services[J]. Expert Systems with Applications, 2009, 36(2): 3167-3187
- [3] 岳昆, 王晓玲, 周傲英. Web服务核心支撑技术: 研究综述[J]. 软件学报, 2004, 15(3): 428-442
- [4] McIlraith S, Son T C, Zeng H. Semantic web services[J]. IEEE

(上接第125页)

火墙对系统性能的制约,提出了一种基于代理的P2P-CDN混合式体系框架。

2) 目前BT类型流媒体系统的研究还缺乏实际部署的检验,现有研究多数采用仿真、建模分析等手段,而实际性能分析的研究极为少见。文献[2]编码实现了一个测试平台,但规模极小,用来验证具有海量访问需求的流媒体系统显得缺乏说服力。未来在实际环境中检验系统性能是研究工作的重点。

3) 未来BT文件共享系统和BT类型流媒体系统将共同存在,如何实现两者的互通和联动是一个有趣的研究课题。

参考文献

- [1] Vlavianos A, Iliofotou M, Faloutsos M. BiToS: Enhancing BitTorrent for Supporting Streaming Applications[C]//9th IEEE Global Internet Symposium, Barcelona, Spain, 2006
- [2] Choe Y R, Schuff D L, Dyaberi J M, et al. Improving VoD server efficiency with bittorrent[C]//Proc. of the 15th international conference on Multimedia, Augsburg, Germany, 2007
- [3] Cai Q C, Zhang X J. An Effective Approach for Live Media Streaming Based on BitTorrent[J]. Journal of Communications, 2008, 3(6): 11-18
- [4] Janardhan V, Schulzrinne H. Peer assisted VoD for set-top box based IP network[C]//Workshop of Proc. of ACM SIGCOMM, August 2007
- [5] Shah P, Paris J-F. Peer-to-Peer Multimedia Streaming Using BitTorrent[C]//IEEE IPCCC'07, 2007
- [6] LaFortune R, Carothers C D, Smith W D, et al. Simulating Large-Scale P2P Assisted Video Streaming[C]//HICSS'09, 2009: 1-10
- [7] Hwang K-W, Misra V, Rubenstein D. Stored Media Streaming in BitTorrent-like P2P networks[M]. Columbia University, New York, Number cucs-024-08, April 2008
- [8] Bharambe A, Herley C, Padmanabhan V. Analyzing and Improving a BitTorrent Network's Performance Mechanisms[C]//

Intelligent Systems, 2001, 16(2): 46-53

- [5] Gibbins N, Harris S, Shadbolt N. Agent-based Semantic Web Services[C]//Budapest, Hungary. Proceedings of the 12th international conference on World Wide Web, New York USA: ACM, 2003: 710-717
 - [6] Charif Y. Dynamic Service Choreography based on Introspective Agents' Coordination[D]. English, Paris VI (version of April the 10th), 2008. <http://www-poleia.lip6.fr/~charif/Papers/YasmineCHARIF-PhDSummary.pdf>
 - [7] 郑丽伟, 金芝. 需求驱动的主动网构实体聚合[J]. 软件学报, 2008, 19(5): 1083-1098
 - [8] Sandholm T. Automated mechanism design: A new application area for search algorithms[C]//Kinsale, Ireland. Principles and Practice of Constraint Programming-CP 2003. Heidelberg: Springer, 2004: 19-36
 - [9] Wang Puwei, Jin Zhi, Liu Lin, et al. Building toward Capability Specifications of Web Services Based on an Environment Ontology[J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(4): 547-561
 - [10] 叶荣华, 金芝, 等. 一种需求驱动的自主Web服务聚集方法[J]. 软件学报(已录用)
 - [11] Peltz C. Web Services Orchestration and Choreography[J]. IEEE Computer, 2003, 36(10): 46-52
-
- [9] Esposito F, Matta I, Michiardi P, et al. Seed Scheduling for Peer-to-Peer Networks[C]//IEEE NCA09. Cambridge, MA USA, 2009
 - [10] Bindal R, Medved J. Improving Traffic Locality in BitTorrent via Biased Neighbor Selection[C]//Proc. of ICDCS, 2006
 - [11] Carlsson N, Eager D L, Mahanti A. Peer-assisted On-demand Video Streaming with Selfish Peers[C]//Proc. IFIP/TC6 Networking'09, Aachen, Germany, 2009: 586-599
 - [12] Carlsson N, Eager D L. Peer-assisted On-demand Streaming of Stored Media using BitTorrent-like Protocols[C]//Proc. IFIP/TC6 Networking'07, Atlanta, GA, 2007
 - [13] Skevik K-A, Goebel V, Plagemann T. Analysis of BitTorrent and its use for the Design of a P2P based Streaming Protocol for a Hybrid CDN[R]. University of Oslo, June 2004
 - [14] Dana C, Li D, Harrison D, et al. BASS: BitTorrent Assisted Streaming System for Video-on-Demand[C]//IEEE MMSP, Oct. 2005
 - [15] Erman D. Extending BitTorrent for Streaming Applications[C]//Proc. of the 4th Euro-FGI workshop on "New Trends in Modeling, Quantitative Methods and Measurements", 2007
 - [16] Parvez N, Williamson C, Mahanti A, et al. Analysis of BitTorrent-like Protocols for On-demand Stored Media Streaming[C]//Proc. ACM SIGMETRICS'08, Annapolis, 2008: 301-312
 - [17] Magharei N, Rejaie R, Guo Y. Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches[C]//INFOCOM, 2007: 1424-1432
 - [18] Seibert J, Zage D, Fahmy S, et al. Experimental Comparison of Peer-to-Peer Streaming Overlays: An Application Perspective[C]//Proc. of INFOCOM, 2007
 - [19] Zhang X, Liu J, Li B, et al. CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming[C]//Proc. IEEE INFOCOM'05, Miami, FL, USA, March 2005
 - [20] Zhao Q, Lui JCS, Chiu DM. Exploring the Optimal Chunk Selection Policy for Data-Driven P2P Streaming Systems[C]//The 9th International Conference on Peer-to-Peer Computing, Sept. 2009