第18卷 第3期
2008年3月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol. 18 No. 3
Mar. 2008

基于强化学习的Web服务组合

汤萍萍，王红兵
（东南大学 计算机科学与工程学院，江苏 南京 211189）

摘 要：单个Web服务的能力有限，如何使企业内或企业间的诸多业务服务有机地集成，提供更有价值的服务，是目前基于Web服务的应用集成的核心问题。Web服务组合就是将已有服务按照一定的逻辑顺序组织起来构成新的服务，从而实现更强大的功能。这里提出一种基于强化学习的服务组合方法，它在于能够实现动态的Web服务组合。

关键词：Web服务；服务组合；动态服务；机器学习；强化学习
中图分类号：TP393 **文献标识码**：A **文章编号**：1673－629X(2008)03－0142－03

Web Service Composition Based on Reinforcement －Learning

TANG Ping-ping, WANG Hong-bing
(School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

Abstract: The ability of single Web service is limited, so how to make the services that are in the enterprises or among the enterprises organically integrated, and to provide more valuable service, is the core of application－integrated problem based on Web service. Web service composition is to have the existed services to compose new ones according to a certain logic order, thus to realize the bigger and powerfuler function. In this article, suggest a method that is some kind of Web service composition based on reinforcement－learning, which will realize dynamic composition.

Key words: Web service; service composition; dynamic composition; machine learning; reinforcement learning

0 引 言

Web服务组合就是将已有服务按照一定的逻辑顺序组织起来构成新的服务，从而实现更强大的功能[1]。服务的组合已经在相关研究领域获得重视，相关的技术方法已经出现。组合服务在不改变原有服务组件的基础上根据用户需求重新组合产生新的服务，以期给系统带来更大的灵活性，以及针对用户的可定制个性化服务，服务的组合潜在地缩短了开发时间，减少了开发新应用的工作量。

组合服务分为静态组合服务和动态组合服务。静态组合服务是在设计时或装载时进行的。虽然性能上比较稳定，但缺乏定制灵活性。动态组合服务是在运行时进行组合的，它能够给系统带来更大的灵活性，也是Web服务研究领域的最终目标。文中所实现的正是这种动态的Web服务组合。

机器学习是人工智能应用的重要研究领域，是研究如何使用机器来模拟人类学习活动的一门学科。强化学习reinforcement learning，是一种重要的机器学习方法[2]。强化学习基本原理是：如果Agent的某个行为策略导致环境正的奖赏（强化信号），那么Agent以后产生这个行为策略的趋势便会加强。Agent的目标是在每个离散状态发现最优策略以使期望的折扣奖赏和最大。文中将把强化学习的重要算法引入Web服务组合，从而真正实现动态组合。

1 一个具体实例

场景分析：在Internet上有一个电子政务的门户网站，该网站负责提供申请失业保险的Web服务，现有某市民在此电子政务的门户网站上申请失业保险服务。门户网站服务代理的“任务接受器”接收到此信息后，将通过“服务组合产生器”自动生成对应的流程图，这里是一个复杂的过程，期间需要去语义UDDI服务注册中心发现可供使用的服务[3]，并且进行描述的转换[4]。接下来业务服务引擎调用并执行各个基本服务即可。

例如，当流程图（如图1所示）中指出下一步是调用“确定劳动关系”的Web服务时，门户网站服务代理的业务服务引擎就转去调用专门提供“劳动状况服务”

收稿日期：2007－06－24
作者简介：汤萍萍（1981－），女，安徽芜湖人，硕士研究生，研究方向为Web服务、Java技术开发、电子商务；王红兵，博士生导师，教授，研究方向为Web服务、Java技术开发、电子商务。

的服务代理,当然这个"劳动状况服务"是由别的权威的门户网站所提供。调用结束后将会反馈回必要的信息,接着再调用提供"银行帐号服务"的服务代理,等等。

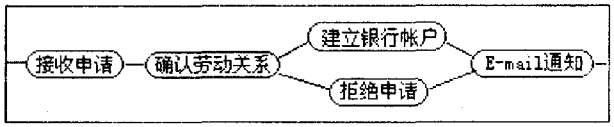


图 1　组合流程图

需要特别指出的是:"组合服务产生器"产生流程图的过程,在目前都只是停留在手动设计阶段,也就是说,有专业人员事先绘制好几种流程图,再根据用户不同的需求调出相应的图。所以说这是相对静态的服务组合,而文中所提出的动态组合,是无需人的参与,完全由服务代理自己产生,而且随着环境的变化,服务代理自己可以学会如何产生出最优的组合方法[5]。

Web 服务提供的是一个外部可被调用的接口函数,如[WebMethod(Description = "函数的描述信息")]。为简化起见这里用 S0,S1,S2 等等来表示这些服务的接口。现在假定在 UDDI 上发布了 6 个基本服务(如图 2 所示):劳动状况服务、获取基础信息服务、劳动信息查询服务、劳动状况评估服务、银行帐号服务、E-mail 服务。

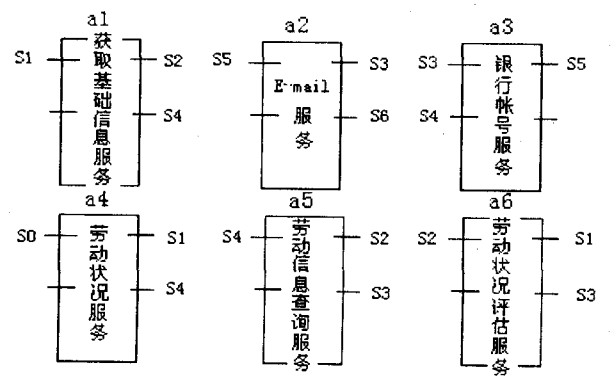
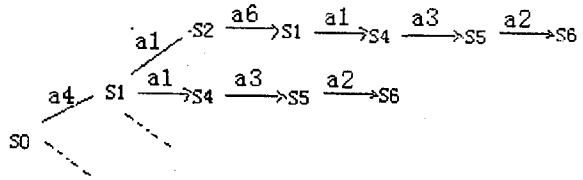


图 2　UDDI 上发布的 6 个服务

做以下重要的约定:把服务输入输出作为状态,服务本身作为动作。假定某用户输入初状态为 S0,要求目标状态为 S6。

2　动态的 Web 服务组合

针对上例,开发的智能服务代理的搜索过程如下:



下面具体阐述所设计的改进后的强化学习算法是如何实现以上过程的[6]。

2.1　马尔可夫决策过程(MDP)

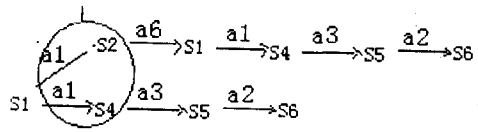
马尔可夫决策过程是由四元组 $< S,A,R,P >$ 定义,包含一个环境状态集 S,系统行为集合 A,奖赏函数 R 和概率转移函数 P。

记 $R(s' \mid s,a)$ 为系统在状态 s 采用 a 动作使环境状态转移到 s' 获得的瞬时奖赏值:

$$R^a_{ss'} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\} \quad \text{for all } s,s' \in S, a \in A(s)。$$

记 $P(s' \mid s,a)$ 为系统在状态 s 采用 a 动作使环境状态转移到 s' 的概率:

$$P^a_{ss'} = Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad \text{for all } s,s' \in S, a \in A(s)。$$



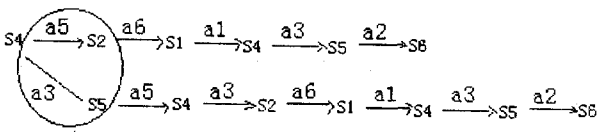
所以,根据前面的重要约定,系统在状态 $s1$ 采用动作 $a1$ 使环境状态转移到 $s2$ 或者 $s4$ 的概率就由 $P(s' \mid s,a)$ 来决定。

2.2　值函数 $V^\pi(s)$

它是一个迭代函数,表示从初始状态 s 按照策略 π 直到进入终止状态所得的累积奖赏值:

$$V^\pi(s) = \sum_a \pi(s_1 a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + V^\pi(s')]$$

这里,策略 π 是一组随机的规则,它指导系统的下一步动作。



系统在状态 $s4$ 采用动作 $a3$ 还是动作 $a5$ 由策略 π 决定。$\pi(s,a)$ 是指在状态 s 下所选取的策略,若 $\pi(s,a) = 0$,表示该策略没有选取 a 动作;若 $\pi(s,a) = 1$,表示该策略选取了 a 动作。

2.3　函数改进

在这个实例中,其实每次只会发生一个动作,即对于 $\pi(s,a)$(对于所有 a 属于 $A(s)$),只有一个值为 1,所以值迭代和策略迭代可以简化为:

$$V^*(s) = \max_{a \in A(s)} \sum_{s'} P^a_{ss'}[R^a_{ss'} + V^*(s')]$$

$$\pi'(s) = \operatorname{argmax}_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + V^\pi(s')]$$

这里,$\max\{f(x,y): x \text{ 属于 } X\}$ 是 $f(x,y)$ 在 X 上能达到的最大值。

$\operatorname{argmax}\{f(x,y): x \text{ 属于 } X\}$ 是 $f(x,y)$ 在 X 上达到最大值时的 x 值。

3 动态服务组合算法

根据以上结论可以得出以下算法的描述:设计一个函数 function composition(state s0),也就是说只要给定一个初始状态,系统可以自动生成"一套组合策略"。

3.1 值迭代算法

Initialize $V(s)$ arbitrarily
Loop until policy good enough
  Loop for $s \in S$
    loop for $a \in A$
      if s': = 终态 $T(s,a) = 0$ and $T(s,a,s') = 1$
      $Q(s,a)$: = $R(s,a)\sum_{s' \in S} T(s,a,s')V(s')$
    $V(s)$:$\min_a Q(s,a)$ and $Q(s,a)! = 0$
  end loop
end loop

$V(s)$数组中记录的是从状态 s 按照策略 π 直到进入终止状态所得的"最大"累积奖赏值,设定初始值全部为 1。当然,根据不同的应用需要,这里的最优条件可以变化,例如取"最大"累积奖赏值,相应的算法语句为:$V(s)$: = $\max_a Q(s,a)$。

三维数组 $T(s,a,s')$ 记录了系统在状态 s 采用 a 动作使环境状态转移到 s' 的概率,假定这里使用平均概率,如果 $T(s,a,s') = 0$,则表示系统在状态 s 采用 a 动作不可能使环境状态转移到 s',或者是系统在状态 s 不可能、不可以采用动作 a。

此外在本例子中 $R(s,a) = 1$,既任意一步行动都会得到 1 的奖赏值。

3.2 策略迭代算法

以下是先对策略进行初始化操作,经过循环迭代,最终生成数组 $\pi(s)$,表示的是在状态 s 下的一组最优策略。

choose an arbitrary policy π'
loop
  π: = π'
  compute the value function of policy π:
    solve the linear eguations
    $V_\pi(s) = (R(s)) + \sum_{s' \in S} T(s,\pi(s),s')V_{\pi'}(s')$
  improve the policy at each state:
    $\pi'(s)$: = $\text{argmin}_a(R(s,a) + \sum_{s' \in S} T(s,a,s')V_\pi(s'))$
  until π: = π'

4 相关工作

Web 服务的应用正在逐步推广,提供相同功能的服务会越来越多,于是出现的新问题是:如何动态地从众多 Web 服务中选择出最适合用户需求的。这就是基于 QoS 约束的 Web 服务发现和选择模型[7]。QoS 由一系列具体属性来描述,如服务的价格、性能、可靠性、安全性、响应时间等[8]。上面的算法需要进一步扩展,以实现基于 QoS 约束的 Web 服务发现和选择模型。

此外,强化学习方法存在的主要问题:

1)当状态空间较大时,算法收敛前的实验次数可能要求极多[9]。

2)多目标的学习:大多数强化学习模型仅针对单目标的问题学习相应的决策策略,难以适应多目标、多策略的学习需求[10]。

3)许多问题面临的是动态变化的环境,其问题求解目标本身可能也会发生变化。

下一步的工作将主要围绕强化学习方法存在的主要问题,认真研究,以改进组合的方法。服务组件或基本服务的定位、协调、通信及调用策略,服务执行结果的评估,正确性验证等等,这些都是有待解决的问题。更好地进行有监督的 Web 服务组合,实现更加准确和高效的 Web 服务搜索和 Web 服务发现,也是正在进行的研究工作。最终,是要建立一个能够实现完全意义上的动态组合的原型系统。

5 结 论

目前对于服务的组合还停留在静态组合阶段,虽然已经实现了诸多原型系统,但基本上都是"半自动"的,即开发人员要事先将各种流程放到系统中,然后用户根据不同需要去调用某种特定流程。服务组合的目标其实是要实现系统的自动组合,这也是 Web 服务领域非常热点的课题之一。达到动态的组合不仅需要依赖已有的技术,如服务发现、绑定、远程调用的一系列协议,更重要的是借助于各种有效的算法实现动态特性。强化学习方法一直以来多应用于机器人研究(如小人踢足球等)、五子棋等游戏比赛[11]。文中通过对强化学习算法的改进而提出的这种新的算法,它可以很好地实现服务的动态组合。

参考文献:

[1] Campbell A T. A quality of service architecure[D]. UK: Lancaster Univ., 1996.

[2] Barto A G. Reinforcement learning in the real world[C]//In Proceedings of IEEE International Joint Conference on Neural Networks(IJCNN). Budapest, Hungary: [s. n.], 2004.

[3] Massimo P, Katia S. Autonomous Semantic Web Services[J]. IEEE Internet Computing, 2003, 11(5): 34 - 41.

[4] Liu Yutu, Ngu A H H, Zeng Liangzhao. QoS Computation and Policing in Dynamic Web Service Selection[C]//In Proceedings of the 13th international World Wide Web conference on

(下转第 164 页)

个 UUID 号、群信任状及群标识属性，β 取值为 1，α 分别取不同整数值，且 $\alpha > \beta$。实验结果如图 3 和 4 所示。

从图 3 的实验结果可以看出，随着管理节点的增加，社群的搜索效率不断提高，当管理节点达到 10 个左右时，社群搜索效率趋于稳态。图 4 的结果表明，当管理层被赋予较大的权重值时，系统的安全预警时间缩短，使恶意节点对社群的危害减轻，对恶意节点的删除效率提高；但权重取值过大又会对系统有其他不良的影响，因为如果恶意节点进入管理层，而被给予较大的权重值，那么社群无法对其进行有效的监控与管理。

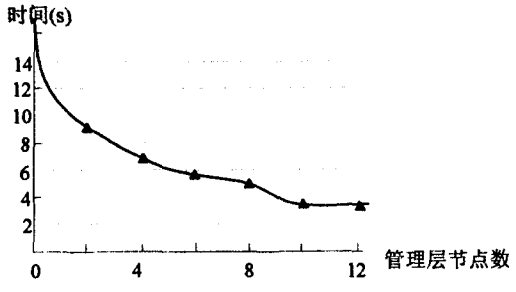


图 3 节点搜索时间

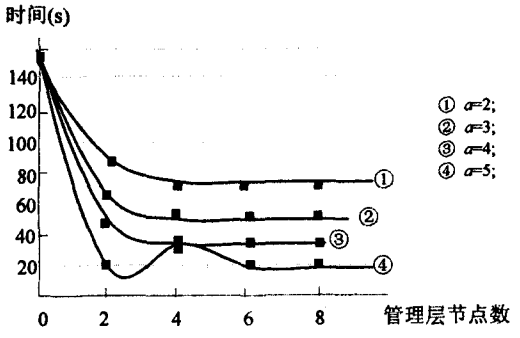


图 4 清除恶意节点所需时间

上述实验结果表明，兴趣社群的组建缩短了 P2P 网络节点的搜索时间，有效地提高了节点搜索资源的效率，降低了搜索路径长度；而增加管理层的控制，及时有效地清除恶意节点，可提高 P2P 网络的可靠性和稳定性。

4 结 语

通过模拟人类社会区域自治的管理体系，给出一种基于兴趣社群的 P2P 网络节点自组织管理方法。该方法将大规模的 P2P 网络划分为各个小规模的社群，在社群内实现分层的自治管理，管理层在社群内有较大的权重值，普通节点也赋予了一定的管理与监督功能，从而促进了整个社群能向安全有序的方向发展。实验结果表明，兴趣社群结构的 P2P 网络自组织管理方法能提高 P2P 网络搜索的性能，对恶意节点能进行有效监控与管理，对最终形成长期协同合作的 P2P 社群起到了良好的促进作用。

参考文献：

[1] Watts D J, Strongatz S H. Collective dynamics of 'small-world' networks[J]. Nature,1998, 393: 440-442.

[2] 许 骏,柳泉波,史美林.协作社群形成与演化机制——理论与算法[M].北京:科学出版社,2005.

[3] 唐九阳,张维明,肖卫东,等. 类人类社会基于社区的对等网自组织构造[J].计算机研究与发展,2006,43(8):1383-1390.

[4] Lei Guo, Song Jiang, Li Xiao, et al. Exploiting content localities for efficient search in P2P systems[C]//Proceedings of the 18th Int'l Symposium on Distributed Computing (DISC 2004). Amsterdam, Netherlands:[s.n.],2004:349-364.

[5] Kobayashi H, Takizawa H, Inaba T. et al. A self-organizing overlay network to exploit the locality of interests for effective resource discovery in P2P systems[C]//The 2005 International Symposium on Applications and the Internet (SAINT 2005). Trento, Italy:[s.n.],2005:246-255.

[6] 杨文俊.P2P网络系统中节点自组织管理机制[J].计算机技术与发展,2006,16(7):57-60.

[7] Pouwelse J A, Garbacki P, Wang J, et al. Tribler: A social based Peer-to-Peer system[J]. Concurrency and Computation: Practice and Experience,2007,19:1-11.

(上接第 144 页)

Alternate track papers & posters. New York, USA:[s.n.], 2004:66-73.

[5] 欧毓毅,郭荷清,许伯桐.Web服务动态组合的研究[J].计算机应用研究, 2006(4):14-22.

[6] Wang Ben-Nian, Gao Yang, Chen Zhao-Qian. LMRL: a multi-agent reinforcement learning model and algorithm [C]//Proceedings 3rd International Conference on Information Technology and Applications(ICITA). Sydney, Australia:[s.n.],2005:303-307.

[7] Menasc D A. QoS issues in Web service[J]. Internet Computing, 2002, 6(6): 72-75.

[8] Mcilraith S A, Son Tran Cao, Zeng Honglei. Semantic Web Services[J]. IEEE Intelligent Systems, 2001, 16(2): 46-53.

[9] Dietterich T G. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition[J]. Journal of Artificial Intelligence Research,2000,13:227-303.

[10] Gullapalli V. A comparison of supervised and reinforcement learning methods on a reinforcement learning task[C]//Proceedings of the 1991 IEEE International Symposium on Intelligent Control. Virginia, USA:[s.n.],1991:394-399.

[11] 张汝波,顾国昌,刘照德,等.强化学习理论、算法及应用[J].控制理论与应用,2000,17(5):633-642.