



基于 SOA 的服务动态扩展机制研究 *

樊程, 蒋静, 戴洪, 潘振宽

(青岛大学信息工程学院 青岛 266071)

摘要

针对 SOA 应用系统中的服务动态扩展问题进行研究,提出了基于 SOA 的服务动态扩展系统模型,该模型在 SOA 架构的基础上,引入了服务资源池和服务动态扩展思想,把系统可用服务放入资源池中供系统调用,通过调整资源池的服务组成和规模实现服务动态扩展。本文给出了系统模型框架和服务动态扩展的功能逻辑,并使用 ASML 语言描述模型中的动态扩展规则,最后使用 Spec Explorer 对系统模型进行仿真测试,验证了系统的可行性和有效性。

关键词 服务动态扩展;服务资源池;SOA;ASML

1 引言

SOA(service-oriented architecture)是一种面向服务的架构模型,旨在降低服务代理之间的耦合度。在 SOA 框架中,IT 资源被构造为一系列可重用的服务,这些服务是松散耦合的,与平台和实现无关,SOA 通过这些服务之间定义良好的接口和契约将它们联系起来,并通过网络对这些服务进行分布式部署、组合和使用,服务在整个构架中以一种统一和通用的方式进行交互^[1]。SOA 使企业应用摆脱了面向技术解决方案的束缚,企业可以将业务封装为服务并对外发布以供消费者使用,这样能更加精确地表示业务模型,更好地支持业务流程^[2]。

SOA 系统的核心服务层旨在提供服务定位、发现和绑定,实现服务共享与协同,因此通常把研究重点放在实现服务的互通与重用上,搁置了服务的动态扩展问题。随着 IT 技术的发展和用户需求规模的增长,系统不能为用户

动态扩展服务的问题逐渐暴露出来。如果不能很好地解决这个问题,SOA 系统对用户的友好性将会大打折扣,因此服务动态扩展问题已经成为制约 SOA 技术发展的瓶颈^[3]。

服务动态扩展主要包括计算能力的扩展、存储服务的扩展以及 Web 服务的扩展等。对于计算能力和存储服务的动态扩展,已经在云计算中得到商业实现^[4],如 Amazon 推出的弹性计算云(elastic compute cloud,EC2)可以为用户提供计算能力的动态扩展;简单存储服务(simple storage services,S3)则实现了存储服务的动态扩展。对于 Web 服务的动态扩展,目前正处于探索阶段,尚未见到成功的商业案例。本文针对 Web 服务的动态扩展问题进行研究,提出了一个基于 SOA 的服务动态扩展机制模型,模型中所涉及的服务扩展即指 Web 服务扩展,并给出设计框架,同时对扩展规则进行描述,最后进行仿真验证。

2 相关概念及系统模型

2.1 相关概念定义

服务动态扩展要求系统能够按照用户的需求,提供所

* 青岛市科技计划基金资助项目(No.06-2-2-JCH),国家支撑计划基金资助项目(No.2006BA111B07)

需的服务,即使系统内不存在此项服务,系统仍能动态地为用户从系统外扩展类似服务。下面给出与服务动态扩展相关的概念和定义。

定义 1 (服务) 服务由一个或多个具有内在联系的服务组件组成,以实现某种具体的业务逻辑。每个服务都是一个独立的部署单元,这给应用部署带来很大的灵活性,只要保持服务接口不变,就可以通过部署新服务实现新的业务逻辑,同时不影响系统其他部分。

定义 2 (服务资源池) 通过虚拟化方法^[5],将系统可用服务虚拟成一个可以任意组合和分配的集合,称为资源池。资源池的组成和规模可以动态改变,分配给用户的服务可以动态地添加和收回。

定义 3 (系统默认服务) 用户登录系统后可以直接调用的服务,即系统初始化时资源池内可以直接被系统使用的服务。

定义 4 (服务扩展标志) 每个服务对不同的访问用户都会产生一个服务扩展标志 (service extension flag, service_EF),该标志反映当前用户与服务的关系,用来指示用户是否已经扩展此项服务,它是一个二元组 (ServiceX_EF, UserIDn)。其中,ServiceX_EF 是一个布尔型变量,表示对于标识为 UserIDn 的用户,服务 ServiceX 是否被扩展到资源池中,若 UserIDn 用户尚未扩展此服务,ServiceX_EF 为 False,当 UserIDn 用户扩展此服务之后,其将被置为 True。

定义 5 (服务动态扩展事件) 当用户请求系统默认服务外的服务时,由系统自动产生。服务动态扩展事件 (service dynamic extension event) 的目的是向系统请求将资源池外的服务迁移到池内,以满足用户的需求。

定义 6 (服务动态扩展) 当用户请求服务产生服务动态扩展事件时,系统根据相应规则,在资源池外查找相关服务,并将其扩展到资源池中,注册并供系统使用,同时用户 ID 对应的扩展标志置为 True,这一过程称为服务动态扩展 (service dynamic extension, S-DE)。

定义 7 (服务集与扩展服务集) 服务集 (service set) 是指系统默认服务的集合,表示为 ServiceSet (Service 1, Service 2, ..., Service N); 扩展服务集 (eservice set) 是指用户在使用系统过程中通过扩展能够被系统使用但尚未成为系统默认服务的已扩展服务集合,可以形式化地表示为 EServiceSet (Eservice 1, Eservice 2, ..., Eservice N)。

2.2 系统模型

基于 SOA 的服务动态扩展系统就是在 SOA 构架下,实现服务的动态扩展,其系统模型总体架构如图 1 所示。

该模型在传统 SOA 架构的基础上,把系统分为 3 个大层次:SOA 应用层、面向服务组织和扩展层以及现有系统和资源层。下面介绍模型中各层次的特点和功能。

(1) SOA 应用层

SOA 应用层表示用于访问业务流程、服务和应用程序的通道,建立在系统提供的服务之上,负责将用户的请求发送给 SOA 系统并等待响应。

(2) 面向服务组织和扩展层

服务组织是指服务代理 (service agent) 通过服务发现和绑定 (service discover & binding) 对 Web 服务进行统一的管理和调度。服务动态扩展是本文工作的核心所在,主要工作是在传统 SOA 架构的基础上加入服务资源池和服务动态扩展控制器 (S-DE control)。服务资源池用来存放可

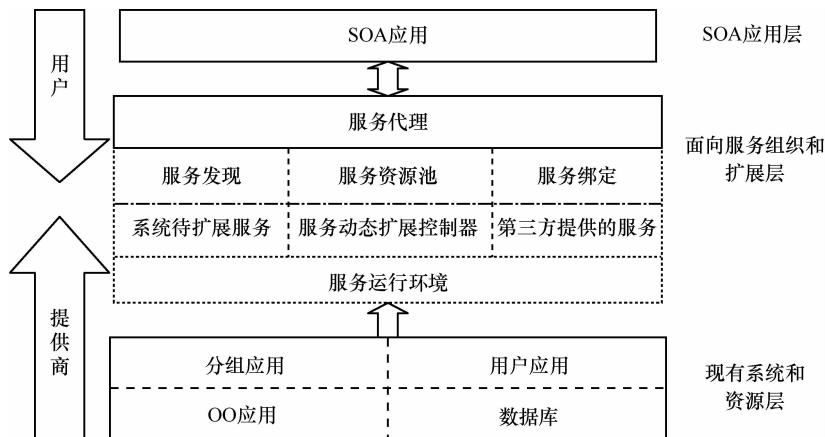


图 1 基于 SOA 的服务动态扩展系统模型架构



以被系统直接发现和绑定的服务,服务扩展控制器用来捕获并处理扩展事件,主要是在系统待扩展服务(underSEWS)和第三方提供的服务(third-party WS)中查找匹配服务,并将其迁移到资源池中,实现系统对服务的扩展。服务运行环境(services runtime environment)为 Web 服务提供一种服务的表示、访问和运行机制,并提供 Web 服务的安全保障、服务生命周期管理等。

(3) 现有系统和资源层

现有系统和资源层指系统现有的 IT 资源和数据,包括一些系统应用和企业内外部的资源,SOA 构架可以很好地利用这些 IT 资源,将其利用到新的 SOA 系统中,减少系统开销。

3 服务动态扩展模块设计

3.1 动态扩展实现机制

图 2 给出了服务动态扩展系统的功能逻辑结构,它由 6 部分组成:服务代理、服务发现和绑定模块、服务动态扩展控制器、服务资源池、待扩展服务集和第三方服务集。

服务资源池中的服务集合是动态可变化的,系统初始化时的默认服务都存放于服务资源池中,可以直接被系统调用。系统待扩展服务集隶属于此系统,但由于成本、使用频率等问题并不能直接被系统使用,需要通过扩展系统处理之后才能被系统调用。第三方服务集是指当系统内服务不能满足用户需求时,系统向合作伙伴请求的系统外服务。所有服务只有被扩展到资源池中才能被系统直接调用,系统调用资源池外的服务时,会产生服务扩展事件。当产生服务动态扩展事件时,服务扩展控制器捕获事件,并

按照相应的扩展规则处理扩展事件,最终实现服务动态扩展。图 2 所示的服务动态扩展系统的交互过程如下。

(1)服务代理将用户请求提交给系统,并与服务发现与绑定模块协商,对所需服务进行发现和绑定。

(2)服务发现和绑定模块在服务资源池中查找所需服务,若找到,则返回与服务代理协商绑定;若未找到,则将信息反馈给服务发现和绑定模块,同时产生服务动态扩展事件。

(3)服务动态扩展控制器捕捉服务动态扩展事件,并按照扩展规则对其进行处理。

(4)服务动态扩展控制器在待扩展服务集和第三方服务集中查找满足扩展条件的服务。

(5)服务动态扩展控制器将需要扩展的服务迁移到服务资源池中,按照扩展规则对服务进行相应的处理,完成服务的扩展。新扩展的服务能够被发现和绑定。

3.2 服务动态扩展规则描述

抽象状态机(abstract state machine,ASM)是一种用于系统建模和分析的通用形式化方法,可以对任何系统建模,并且可以在不同的抽象层次上描述一个系统。基于 ASM 的软件模型具有精确性和自然性。抽象状态机语言(abstract state machine language,ASML)^[6]是一种基于 ASM 的软件规范描述语言,用 ASML 描述模型得到的是人易读、机器可执行的规范。本文主要通过 ASML 对服务动态扩展规则进行形式化描述,下面给出用 ASML 描述的服务动态扩展中的几个重要规则。

规则 1 服务注册 Register(SERVICE)

If $\exists \text{ServiceX} \in \text{SERVICE}$

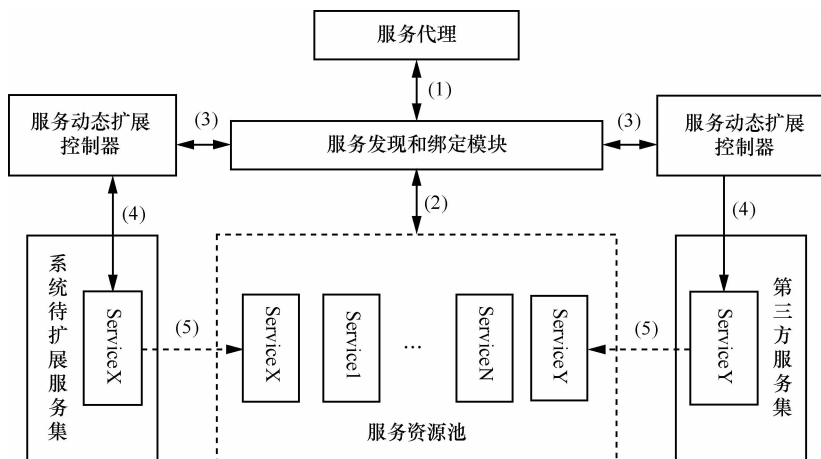


图 2 服务动态扩展系统的功能逻辑结构

$\text{Status}(\text{ServiceX}) = \text{unregistered} \wedge \text{Event}(\text{ServiceX}) =$
 register
 Then
 SubmitInformation(ServiceX)
 If SubmitInformation(ServiceX)=true
 Then
 $\text{Status}(\text{ServiceX}) := \text{registered}$
 其中: $\text{Status}: \text{SERVICE} \rightarrow \text{STATUS}$

Event: $\text{SERVICE} \rightarrow \{\text{register}, \text{unregister} \dots\}$

SubmitInfromation: $\text{SERVICE} \rightarrow \text{BOOL}$

在系统使用过程中,如果调用一个尚未注册的服务事件,会产生服务注册事件,系统会将服务注册信息提交,从而完成服务的注册。

规则 2 池外服务的查找与定位

If $\exists \text{ServiceX} \in \text{SERVICE}$ RequestUsableService(ServiceX,
 user) = true $\wedge$
 $\text{MappedArea}(\text{GetApplication}(\text{ServiceX})) = \text{undef}$
 If $\exists \text{ServiceX} \in \text{SERVICE SET}$ Then
 Search ServiceX in SpareService where
 $\text{Compatiable}(\text{Attri}(\text{ServiceX}), \text{Attri}(\text{user}))$ Then
 $\text{RequestUsableComp}(\text{ServiceX}, \text{user}) := \text{false}$
 $\text{MappedArea}(\text{GetApplication}(\text{ServiceX})) := \text{AreaLocate}$
 (ServiceX)
 Endif
 If $\exists \text{ServiceX} \notin \text{SERVICE SET}$ Then
 Search ServiceX in CoopService where11 Compatiable
 (Attri(ServiceX),Attri(user)) Then
 $\text{RequestUsableComp}(\text{ServiceX}, \text{user}) := \text{false}$
 $\text{MappedArea}(\text{GetApplication}(\text{ServiceX})) := \text{AreaLocate}$
 (ServiceX)
 Endif
 If $\text{MappedArea}(\text{GetApplication}(\text{ServiceX})) \neq \text{undef}$
 Then
 $\text{MappedPool}(\text{ServiceX}) := \text{Locate}(\text{ServiceX})$
 $\text{ServiceSE}(\text{ServiceX_EF}, \text{user}) := \text{true}$
 $\text{Use}(s, \text{Pool}(\text{ServiceX})) := \text{true}$
 Endif
 其中: $\text{RequestUsableService}: \text{SERVICE} \times \text{USER} \rightarrow$
 BOOL
 $\text{GetApplication}: \text{SERVICE} \rightarrow \text{APPLICATION}$

$\text{MappedArea}: \text{APPLICATION} \rightarrow \text{AREASYSTEM}$
 $\text{Attri}: \{\text{USER}, \text{SERVICE}..\} \rightarrow \text{ATTRIBUTE}$
 $\text{Compatible}: \text{ATTRIBUTE} \times \text{ATTRIBUTE} \rightarrow \text{BOOL}$
 $\text{AreaLocate}: \text{COMPONENT} \rightarrow \text{AREAAGENT}$
 $\text{Locate}: \text{COMPONENT} \rightarrow \text{Pool}$
 $\text{MappedPool}: \text{SERVICE} \rightarrow \text{Pool}$
 $\text{ServiceSE}: \text{SERVICE} \times \text{USER} \rightarrow \text{BOOL}$
 $\text{Use}: \text{SERVICE} \times \text{Pool} \rightarrow \text{BOOL}$

系统调用服务资源池外的服务时,系统会在资源池外对相应的服务进行查找和定位,并对服务进行映射,以便使服务进行迁移和扩展。若请求的服务在系统提供的服务列表中,则在系统待扩展的服务集中进行查找定位;若不在服务列表中,则在合作伙伴提供的第三方服务集中查找和定位。

规则 3 服务动态扩展

If $\exists \text{ServiceX} \in \text{SERVICE}$ $\text{userX} \in \text{USER}$
 $\text{Status}(\text{user}) = \text{Call}(\text{ServiceX}) \wedge \text{MappedPool}(\text{ServiceX}) =$
 undef
 Then
 $\text{Search}(\text{ServiceX})$
 $\text{Locate}(\text{ServiceX})$
 $\text{Event}(\text{ServiceX}) = \text{Extend}$
 $\text{WantToExtend}(\text{ServiceX}) := \text{true}$
 If $\text{WantToExtend}(\text{ServiceX})$ Then
 $\text{MappedPool}(\text{ServiceX}) := \text{Locate}(\text{ServiceX})$
 $\text{AddtoESERVICESET}(\text{ServiceX})$
 $\text{Register}(\text{SERVICE}) := \text{true}$
 $\text{ServiceSE}(\text{ServiceX_EF}, \text{user}) := \text{true}$
 $\text{Usable}(\text{ServiceX}) := \text{true}$
 $\text{CallService}(\text{ServiceX}) := \text{true}$
 Endif
 Endif
 其中: $\text{WantToExtend}: \text{SERVICE} \rightarrow \text{BOOL}$
 $\text{Event}: \text{SERVICE} \rightarrow \{\text{Extend}, \text{Curtail}, \text{Nothing} \dots\}$
 $\text{MappedPool}: \text{SERVICE} \rightarrow \text{POOL}$
 $\text{Substituted}: \text{COMPONENT} \times \text{COMPONENT} \rightarrow \text{BOOL}$
 $\text{Usable}: \text{SERVICE} \rightarrow \text{BOOL}$
 $\text{CallService}: \text{SERVICE} \rightarrow \text{System}$

当用户请求资源池中不包含的服务时,系统会产生动态扩展事件,并在待扩展服务集和第三方服务集中查找匹



配的服务,对其进行定位,然后将服务迁移到资源池中,并授权给服务代理,同时将服务加入服务集,更改服务与用户相应序列的扩展标志,将其置为扩展可用,则系统可以调用新扩展的服务。

规则 4 服务动态撤销

If $\forall$ ServiceX $\in$ ESERVICESET Status (ServiceX) = Finished $\wedge$

RequestUsableService(ServiceX,user) = false Then

Check: Check Status (ServiceX,user)

If Event(ServiceX,user)=Curtail Then

DeleteESERVICESET(ServiceX)

LeavePool(ServiceX)

Register(SERVICE):=undef

ServiceSE (ServiceXSE,user):=false

Usable (ServiceX) := false

ElseIf Event(ServiceX,user)=Wait Then

Wait()

Goto Check

ElseIf Event(ServiceX,user)=Keep Then

DeleteESERVICESET(ServiceX)

AddtoSERVICESET(ServiceX)

EndIf

EndIf

其中: Status: SERVICE $\rightarrow$ STATUS {running, pause, finished}

CheckStatus : SERVICE $\times$ USER $\rightarrow$ STATUS {curtail, wait, keep...}

当扩展的服务被调用完之后,系统会向用户询问扩展服务的后续使用情况,若用户想裁减此项服务,则把此服务从扩展服务列表和资源池中删除,注销其注册,扩展标志位置为 False,服务成为系统不可用;若用户想保留此服务,则将服务从扩展列表中删除,加入服务列表中,使其成为系统默认服务;若用户想等待一段时间后再做决定,则系统会等待相应时间后再次询问用户。

4 仿真测试

使用 Spec Explorer 对服务动态扩展系统模型进行仿真测试。Spec Explorer 是微软公司针对 ASM 模型开发的一种软件开发工具,为 ASM 模型的测试提供支持,用户可以在 Spec Explorer 中验证 ASM 模型的可行性和有效性^[7]。

假设用户登录并使用服务动态扩展系统,向系统请求

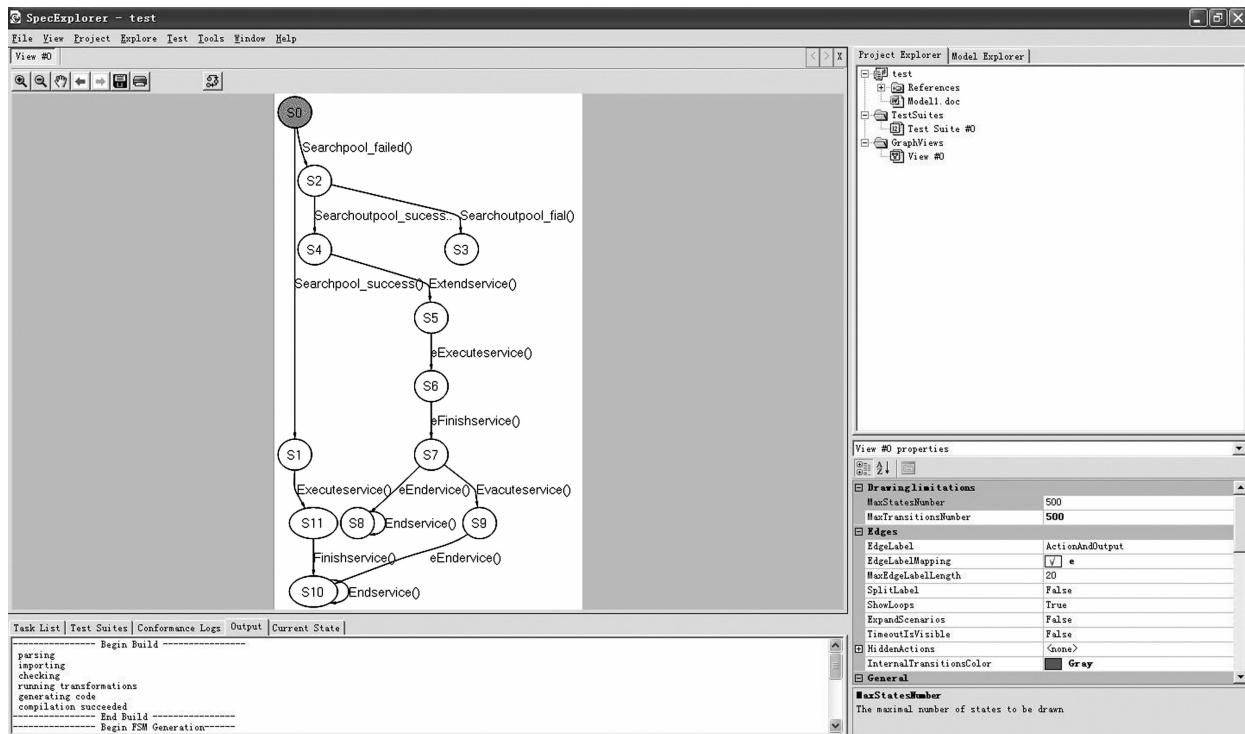


图 3 基于 SOA 的服务动态扩展系统模型仿真结果

一个服务,请求的服务类型未知,即不确定用户所请求的服务是否为系统默认服务。用 ASML 描述系统模型,同时模拟上述测试情景,并将其装载到 Spec Explorer 中,运行生成服务使用过程的有限状态机,并产生测试序列,如图 3 所示。图 3 中显示了用户向系统请求服务后,系统为用户进行服务发现和绑定的全过程,其中也包含了对服务资源池外服务的动态扩展,仿真测试结果与模型设计期望一致,从而验证了模型的可行性和有效性^[8]。

5 结束语

通过对 SOA 系统中 Web 服务扩展问题的研究,本文提出了基于 SOA 的服务动态扩展系统模型,并通过仿真验证了模型的正确性和可行性。本文提出的模型不仅有助于 SOA 系统更好地适应网络环境,满足用户动态变化的需求,同时也为建立高可用、高性能、可扩展的系统和服务提出了一种新的设计思想和实现方法,使系统服务对用户更加友好。

参考文献

- 1 Papazoglou M P, Georgakopoulos D. Service-oriented computing introduction. *Communications of the ACM*, 2003, 46(10): 24~28
- 2 Papazoglou M P. Service-oriented computing: concepts, characteristics and directions. *Proceedings of the 4th International Conference on Web Information System Engineering*, Roma, Italy, 2003: 3~10
- 3 David S Linthicum. *Cloud Computing and SOA Convergence in Your Enterprise: a Step-by-Step Guid*. Beijing: Posts & Telecom Press, 2011
- 4 Chen Kang, Zheng Weimin. Cloud computing: system instances and current research. *Journal of Software*, 2009(5):1337~1348
- 5 Ren Lei, Zhang Lin, Zhang Ya-bin, *et al*. Resource virtualization in cloud manufacturing. *Computer Integrated Manufacturing Systems*, 2011, 7(11):511~518
- 6 Li Hui, Li Minglu. Abstract state machine based system design and analysis for grids. *Acta Electronica Sinica*, 2003(12): 2096~2100
- 7 Borger E. The origins and development of the ASM method for high level system design and analysis. *Journal of Universal Computer Science*, 2002, 8(1):2~74
- 8 Giampolo Bella, Elicinia Riccobene. Formal analysis of the Kerberos authentication system. *Journal of Universal Computer Science*, 1997, 3(12):1337~1381

Research on Service Dynamic-Extension Mechanism Based on SOA

Fan Cheng, Jiang Jing, Dai Hong, Pan Zhenkuan

(College of Information Engineering, Qingdao University, Qingdao 266071, China)

Abstract Aiming at the challenge to realize the service dynamic extension in service-oriented architecture (SOA), a service dynamic expansion mechanism based on SOA is proposed. Services pool and service dynamic extension idea are introduced to this service dynamic expansion system model, and the services are designed into the form of services pool to be called by the system, and the system realizes the service dynamic extension by administrating the resource pool. The critical processes of the service dynamic extension rules are written in ASML (abstract state machine language). Last spec explorer is used to test the model, validating its feasibility and validity.

Key words service dynamic-extension, service resource pool, SOA, ASML

(收稿日期:2012-07-28)