

# 基于生物网络的 Web 服务突现模型研究与实现<sup>\*</sup>

孙宏彬<sup>1,3</sup> 丁永生<sup>1,2</sup>

(东华大学信息科学与技术学院 上海 201620)<sup>1</sup> (数字化纺织服装技术教育部工程研究中心 上海 201620)<sup>2</sup>  
(长春工程学院电气信息学院 长春 130021)<sup>3</sup>

**摘要** 生物神经内分泌和免疫系统通过复杂的调节,构成了一个智能的生物网络。本文借鉴系统实现的自组织、自聚集和自协作特性及复杂的双向调节机制,提出了 Web 服务突现的研究方法。基于移动 Agent 设计具有免疫行为的生物实体并代理 Web 服务,服务请求是由突现形成的生物实体网络提供,生物实体通过自组织协作完成服务的动态合成和自主管理等一系列工作。Web 服务突现模型由 Java 开发的功能模块和平台组成,并利用 Linear Logic 设计了实体的协商。

**关键词** 神经内分泌免疫系统,突现,Web 服务,移动 Agent

## Study and Implement of Web Service Emergent Model Based on Biological Network

SUN Hong-Bin<sup>1</sup> DING Yong-Sheng<sup>1,2</sup>

(College of Information Sciences and Technology, Donghua University, Shanghai 201620)<sup>1</sup>  
(Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Shanghai 201620)<sup>2</sup>  
(School of Electrical Engineering and Information, Changchun Institute of Technology, Changchun 130021)<sup>3</sup>

**Abstract** Biological neuroendocrine and immune system establishes an intelligent biological network through complicated regulation. Inspired by the characteristics of system emergence, such as self-organization, self-cluster, self-cooperation and bidirectional regulation, a method of Web service emergence is proposed. Mobile Agent is designed as a bio-entity with biological immune operation, which applied to represent Web service, the requests of Web services are provided by emergent network of bio-entities. Bio-entities complete a set of relevant works from dynamic composition to self-management of Web service. Web service emergent model is composed of functional software modules and platforms developed by Java, and the negotiation of bio-entities is achieved by linear logic.

**Keywords** Neuroendocrine-immune system, Emergence, Web services, Mobile agent

## 1 引言

Web 服务采用 WSDL、UDDI 和 SOAP 等基于 XML 的标准和协议,具有互操作性、跨平台性和松耦合的特点。语义,特别是本体的提出,为 Web 服务的自动发现、合成提供了一种新的思想。在语义 Web 服务的方式下,服务的发现、执行和合成不需要由人来完成,而是可以由 Agent 来自动完成用户提出的复杂任务。服务数量的增加使集中管理变得非常困难,同时大量的服务更新、移动服务、普适计算使服务一直在变化中,所以 Web 服务合成和动态管理是非常复杂的。

Agent 虽然成功用于服务发现、合成<sup>[1]</sup>,但服务管理过程如何利用 Agent 自主性形成自动服务,以及数量巨大的 Agent 的产生、消亡、迁移、稳定、通信还没有成熟的管理方法。典型系统如: eFlow 系统<sup>[2]</sup>、DynFlow<sup>[3]</sup> 和 SELF-SERV<sup>[4]</sup>。SHOP2 服务合成方法<sup>[5]</sup>,在互联网中,这种系统会很快形成瓶颈。对于下一代的 Web 服务动态管理框架, Hai Zhuge<sup>[6]</sup> 总结出其具有的主要特征:服务访问控制的分布性、自动的合成策略、服务变化的自适应、自扩展及对服务合成的自调整等。

生物体的组织是一个复杂的巨系统,其免疫服务突现和神经内分泌调节给我们研究 Web 服务合成和管理重要的启示。已经证实,细胞因子和受体构成交互的生物网络通用语

言,通过双向信息传递和相互作用使神经内分泌系统与免疫系统之间构成一个有机的整体智能系统。免疫系统通过免疫突现来清除外来抗原(Ag,即病原体),巨量的免疫 B 细胞在神经内分泌的调节下,由一些功能简单的个体通过独特型识别作用形成具有特定功能的网络组合<sup>[7]</sup>,完成了抗原的识别和清除的任务。免疫层是主要的工作平台。神经内分泌层感知免疫层变化,通过复杂的调节机制,使系统构成一个稳定、动态平衡的生物网络系统。整个系统设有中心控制节点,完全是分布式、自组织完成。

生物网络系统的重要特性是智能协调作用的突现。突现是指生物个体遵循一组简单的行为规则,而一群个体则展示复杂的突现行为,即局部相互作用能突现全局行为。突现计算有三个重要方面:自组织、聚集现象和协作行为。在生物网络计算框架<sup>[8,9]</sup>的基础上,本文借鉴生物组织的整体特性机理,进一步提出了 Web 服务突现的方法。服务的突现表现为一个自发的、协作的服务聚集的过程。服务是由突现形成的超级实体或进一步形成的超级实体网络提供的。显然,Web 服务动态合成和管理框架的特征与神经内分泌、免疫形成的生物组织智能系统及其特性非常相似。

Web 服务突现的研究方法是利用移动 Agent 代理 Web 服务,移动 Agent 设计成为具有免疫生物行为的单元,简单的 Web 服务单元通过移动 Agent 自组织、自进化、自聚集协作

<sup>\*</sup> 基金项目:国家自然科学基金重点项目(60534020)、国家自然科学基金(60474037)、教育部新世纪优秀人才支持计划(NCET-04-415)。孙宏彬 博士研究生,副教授,研究方向为网络智能自动化与网络智能计算;丁永生 博士,教授,博士生导师,研究方向为智能系统、网络智能、DNA 计算和生物网络结构。

形成新服务的过程,可以自动完成动态服务合成和管理等一系列工作。

在生物网络平台<sup>[8-10]</sup>基础上,本文提出 Web 服务突现的研究方法,并设计和部分实现了 Web 服务突现的模型。提供了 Web 服务自组织管理的动态合成一种新的思路。仿真数据说明了模型的进化和适应性。

## 2 Web 服务突现的模型设计及实现

### 2.1 Web 服务突现模型的设计

Web 服务突现模型是一个应用 Java 语言开发的软件功能模块结构,如图 1 所示。它架构在分布式网络节点的 Java 虚拟机(JVM)上(包括工作站、服务器、计算机设备)。生物实体(移动的 Agent)生存环境相当于免疫系统,每个移动 Agent 都生存在 Agent 生存环境中,生物实体代理简单的 Web 服务,同时生物实体被设计为具有简单免疫细胞的单元,遵循能量控制机制和生物行为规则(迁移、再生、变异和死亡等),而一群生物实体展示了复杂的行为(如适应性、进化、安全性等)。Web 服务请求相当于抗原,多个生物实体的突现形成实体网络(完成 Web 服务的合成),这种方式下,系统自动完成合成到管理的工作。Web 服务突现模型主要功能模块如图 1。

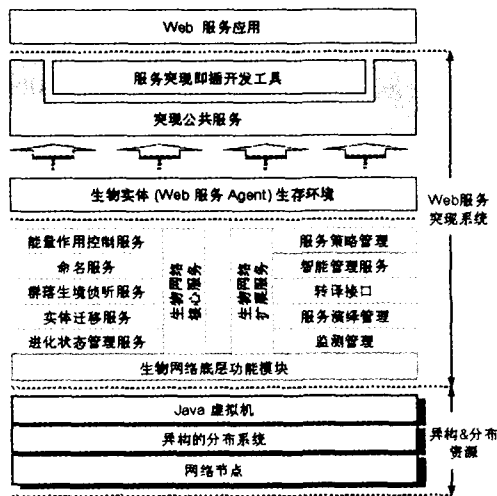


图 1 Web 服务突现模型的设计

(1)生物实体。是网络应用中的最小组件。它是一种自治的移动 Agent,用于形成网络服务与应用。生物实体由属性、行为和功能组成(见表 1)。属性描述关于生物实体的信息。行为实现生物动作,主要有能量交换和存储、生物实体间通信、迁移、复制/再生等。功能实现生物实体提供的服务,包括与服务相关的操作\资料,如数据、应用代码、用户信息。

(2)网络底层资源服务模块。作为生物网络容器提供了底层操作来维持生物网络平台。通常生物实体不能直接访问容器。容器提供的服务主要有:实体注册/取消注册、实体激活/释放、资源管理、实体参考管理、接收/发送请求。

(3)生物网络核心服务模块。是基于生物免疫机理的服务单元,包括:能量作用控制服务、实体迁移服务、进化状态管理、命名服务、群落生境侦听服务等,并把到来的消息分配给运行在本地平台上的生物实体。它们提供基本的生物免疫机理的核心服务,通过这些服务可以完成免疫机理的各种任务。

(4)生物网络服务扩展模块。是对核心模块的调节单元及系统的功能扩展单元,定义了与神经、内分泌系统相对应的调节功能,主要作用是对 Web 服务生存环境动态管理、调节

生物服务实体操作优化,引导实体的自治行为和自主决策,形成群体智能同时保持系统的稳定。

1)监测管理:收集生物服务实体在发现、更新、突现的消息,同时监测实体的网络状况及实体的生境的信息。

表 1 生物实体属性和行为

属性	描述	类型
UID	全局唯一标识	String
Service type	提供服务内容	
Energy	能量,用来标识能力	Numeric
State	活跃、挂起、停止	Numeric
Relationship	生物实体间联系强度	Numeric
Platform identity	系统平台标识	String
行为	描述	
Energy exchange	生物实体进行操作时产生	
Replication	复制生物实体	
Reproduction	进化操作产生新的生物实体	
Interaction	相互作用控制	
Announcement	对系统的信息发布	

2)服务评估策略管理:分析监测生物实体收集的消息(服务质量、服务信誉、网络环境的变化),对生物服务实体的服务管理做智能调节。调节过程是通过对生物服务实体的能量调整、生物激励等方式来完成。

3)智能服务管理:管理用户 Agent 的服务请求,同时管理生物服务实体对应 Web 服务的语义描述和模块的语义服务支持。

4)转译接口:传递生物服务实体的服务协作要求,同时支持实体间进行 Web 服务的 Linear Logic 推演交互。

5)服务推演管理:对生物服务实体 Web 服务的突现过程管理,完成 Web 服务突现的工作流转换。

(5)生物实体的生存环境。生物服务实体工作在生物网络核心模块和扩展模块构成的分布式容器上。Web 服务发现和服务突现是生物服务实体在环境平台上相互作用完成的。每个带有自治行为的生物实体,除了具备与自身个性服务和应用相关的功能外,都遵循着生物演化法则,且可以通过生物网络结构的变异和自然选择机理进化到期望的行为。生物服务实体遵守能量管理机制,生物实体的生物操作和服务都由能量进行调节,实体的迁移、再生、变异和死亡,都伴随这能量的增加和减少。能自治地适应异构和动态变化的环境条件环境即是移动 Agent 的生物生存环境,基于它们所在环境的本地信息进行自治决策,它们都拥有生物组织系统的智能特征。

生物实体如果代理提供优质服务,会拥有较高的能量,而较差的服务会使生物服务实体能量降低,直至死亡。随着时间的推移,实体代理会趋向更有效的服务,从而按照“优胜劣汰”的自然选择机理达到服务管理的自进化。同时,对于服务的环境变化,生物实体会按照生物行为对代理的 Web 服务知识进行相应的合并、分离等生物操作,动态地适应环境。通过这种方式来构建分布的 Web 服务管理,生物服务实体设计相对简单,而系统的自进化、自组织、自适应能力是智能调节下多个生物服务实体共同作用的结果。在实体的协同作用中,Web 服务智能管理调节层会进行动态调节,但没有中心机构控制着所有的生物服务实体。

(6)突现公共服务。由生物实体相互作用突现 Web 服务管理的所需要的服务。显然,这个服务实现了 Web 服务的资源定位、合成、管理、任务分配、认证的工作。

(7)服务突现即插开发工具。提供了 Web 服务应用的开发环境,包括底层的功能开发、生物实体的生成、远程协作、协商支持等。

(8)生物网络间通信。生物网络间的通信有两种:生物实体之间通信和生物网络平台之间通信。通信时,除了重用 FIPA ACL 的部分通信动作(Request, Agree, Refuse, Not-Understand, Inform, Failure, Query-Ref, Query-If)外,根据生物网络中服务内容的需要,BNCL 定义了一些新的通信动作,如 Defray(支付能量给一个提供服务的生物实体)、Advertise(向邻近的实体通报自己的信息如元数据、实体提供的信息等,将实体加入到生存环境中)和 Convene(通报发送者正在寻找某种特点的实体)等。生物网络使用 RMI-IIOP(Remote Method Invocation over Internet Inter-Orb Protocol)技术作为消息的传输协议。生物网络平台采用基于 RMI-IIOP 技术的生物网络通信架构来传送具有生物实体语义的 BNCL 语言的消息。可以用 Java 语言来快速地进行分布式应用开发

对于复杂的互联网中的 Web 突现的应用,这种模型作为平台可以布置在各个 Web 服务的网络节点的 Java 虚拟机上,如图 2 所示。

通过这种方式来构建 Internet 网络 Web 服务和应用比较简单,因为仅需要设计生物实体层次上的相对简单的行为。而其它要求,如适应性、安全性和自救能力是多个平台共同作用的突现行为。Web 服务突现模型可以应用到各种复杂的系统中完成网络应用服务,对解决分布式、并行、大规模及互联网服务提供了一种新的研究方法。

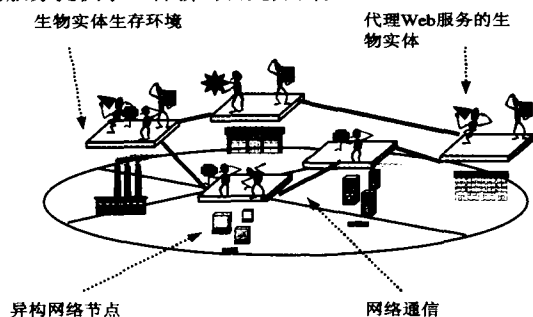


图 2 Web 服务突现模型的分布配置

## 2.2 Web 服务突现的实现

Java 支持 Sockets、URL 通讯、分布式对象协议(RMI)和面向对象的程序设计技术,用 Java 编程很容易实现分布 Agent 间的通讯、实体和功能模块的编程。生物实体的属性、行为和能被设计成为相应的类,组成能量控制的简单生物操作功能也设计在生物实体中,通过阈值条件控制执行。为了实现 Web 服务突现,生物实体不仅要具有生物机制的设计,重要的是提高 Web 服务之间关系的推理能力。为了解决该问题,在 Java 语言设计的生物实体中采用的基于 Linear Logic(LL)<sup>[11]</sup>协作推演研究方法。LL 是一种功能非常强大的推演方法,已应用于 Web 服务合成<sup>[12]</sup>。我们将 LL 设计在多生物服务实体(Agent)的协商过程之中,使生物服务实体利用 LL 的规则对服务合成问题进行自协商解决。

突现实现的问题主要有三个:Web 服务转换为 LL 的单元和序列形式问题、协商过程实现和执行效率问题。生物实体代理的 Web 服务信息是采用 OWL-S 本体描述语言,主要通过 Service Profile、Service Model 和 Service Grounding 这三个类来描述语义。Web 服务的内容由 Service Profile 类描述的,分为功能属性和非功能属性两个部分。Service Profile 的

功能属性包括 Inputs, Outputs, Preconditions, Effects 和 Exceptions,对主要功能进行说明。非功能属性包括有 Consumable Quantitative Attributes, Qualitative Constraints 和 Qualitative Results,非功能属性是相关参数约定,主要用来评估和选择服务。

功能属性和非功能属性转换都要通过本体与 LL 规则的映射完成。典型转换过程 LL 表示如下(参数 1 和参数 2 满足则执行操作):

$$\begin{aligned} &|-! \text{ onto} \# \text{Parameter1}(\text{Model} \# \text{pa1}) \otimes \\ & \quad ! \text{ onto} \# \text{Parameter2}(\text{Model} \# \text{pa2}) \otimes \\ & \text{ onto} \# \text{Not} \text{ } \circ \text{ } \text{process1} \# \text{return} \\ & \quad ! \text{ onto} \# \text{Yes}(\text{Model} \# \text{Yes}) \otimes \\ & \text{ onto} \# \text{process2} \end{aligned} \quad (1)$$

非功能属性转换中 Consumable Quantitative Attributes 作为参数限定服务资源,Constraints 可以转换为:

$$\Delta_1 = \text{Consumable} \text{ } \otimes ! \text{Constra int} \quad (2)$$

Qualitative Results 可以转换为:

$$\Delta_2 = ! \text{Fact} \quad (3)$$

按照 LL 的协商规则,Web 服务需求表示为如下公式:

$$\Gamma_a \Gamma_b ; \Delta_i |- ((I \otimes P) \text{ } \circ \text{ } (O \otimes F) \otimes E) \otimes \Delta_n \quad (4)$$

存在  $\Gamma_a$  和  $\Gamma_b$  等由生物实体代理的 Web 服务,给定的非功能性的属性限定  $\Delta_i$ ,在通过输入参数 I 和输出参数 O 匹配,使 Web 服务满足从状态 P 到 F 的转变,并满足  $\Delta_n$  的相关要求,

单纯的 LL 协商执行效率较低,这里我们进一步使用了局部推演的方法,描述如式(5)所示:

$$\begin{aligned} &S; B \rightarrow B \\ &S(a) = \begin{cases} S((R \text{ } \{r\} + \{r'\}, P + \{p'\})) & (*) \\ a & \text{其他} \end{cases} \quad (5) \\ &(*) \text{ if } P \neq \phi \text{ and } \exists p \in P \text{ and } \exists r \in R \\ & \text{ such that } S_R(r, p) = (r', p') \text{ and } r' \neq r. \end{aligned}$$

B 是生物实体。可以表示为  $a = (R, P)$ , 其中 R 是 LL 接口规则, P 生物实体的状态变化。通过局部推演是生物实体的 LL 表示变得简化,提高协商速度。下面是 LL 的接口规则:

$$\frac{S|-B \otimes C}{S|-A \otimes C} R_b(L_i(\lambda)) \quad (6)$$

$$\frac{A \otimes C|-G}{B \otimes C|-G} R_f(L_i(\lambda)) \quad (7)$$

$$\frac{! A \otimes ! A \otimes B|-C}{! A \otimes ! B|-C} R_a \quad (8)$$

$$\frac{A \otimes B|-C}{! A \otimes B|-C} R_{Li} \quad (9)$$

$$\frac{B|-C}{! A \otimes B|-C} R_{wi} \quad (10)$$

式(6)是前向推演规则,式(7)是后向推演规则,式(8)、(9)、(10)为非限定资源的接口规则。A, B, C 是 LL 公式,  $L_i$  是生物服务实体 LL 表示单元,  $R_b(L_i)$  和  $R_f(L_i)$  使  $L_i$  从初始状态变为简化状态,  $B \otimes C$ ,  $A \otimes B$  和  $A \otimes C$  代表状态变化条件。

为了在 Java 环境中实现生物实体的 LL 功能和局部推演,在生物实体设计中利用 Jena 完成 OWL-S 和 LL 的转换,同时使用由标准的 Prolog 构成的 Lygon 语言(0.7 版本)编写接口规则<sup>[13]</sup>,并嵌入 Java 语言编写的生物实体程序中。在生物实体对服务需求进行协商推演时,生物服务实体通过分布转译接口进行交互,利用 LL 规则判定合作伙伴,进行服务之间的联合协作(图 3),形成服务的自组织、自聚集即形成 Web 服务的突现。

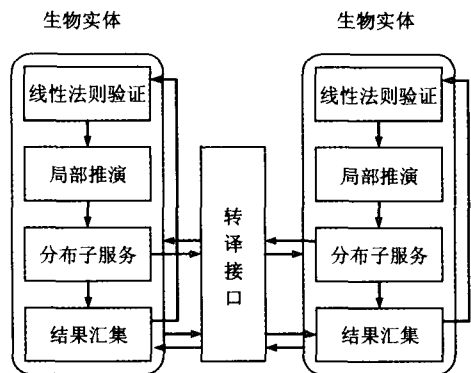


图3 生物实体的协商模式

基于整体智能的 Web 服务突现不仅完成了传统的 Web 合成服务,并能实现分布式、自适应的动态管理,它具有以下特点:1)Web 服务动态合成是自组织、自聚集地协作完成;2)Web 服务管理是自进化的。生物实体代理服务是动态的,当有服务失败及更新时,生物实体按照生物激励和简单的行为法则,可以动态进化选择最合适的服务;3)动态服务突现中使用了 LL 的智能推演,利用了语义的描述,服务的匹配考虑了功能性属性和非功能属性,更加准确;4)服务突现中提出同步的智能决策调节因素,使服务的实现过程更加优化。

### 3 仿真结果及性能分析

利用 Web 服务突现模型作为仿真平台可以对复杂的 Web 服务合成进行研究,本仿真试验采用了 Pentium 4 处理器的 (2.4 GHz 和 512 MB RAM) 的计算机,模拟了多节点的服务请求和服务突现,实验选择了具有 96(8×12) 个节点的网络结构。假设每个虚拟机的网络节点上都运行有仿真平台,网络平台上的资源花费能量的数量相同。访问的服务请求频率不随时间的变化,设定用户的网络服务需求是 10 次/s。系统设置为生物实体能量为:1000,节点时延迟为 0.01s,仿真的时间为 6h。

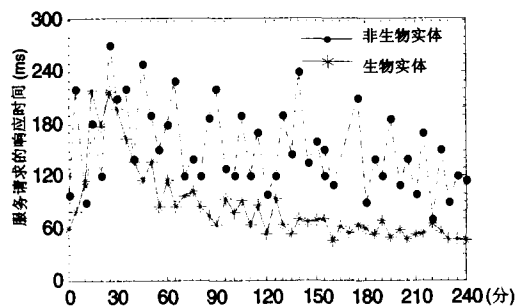


图4 服务请求响应时间(ms)

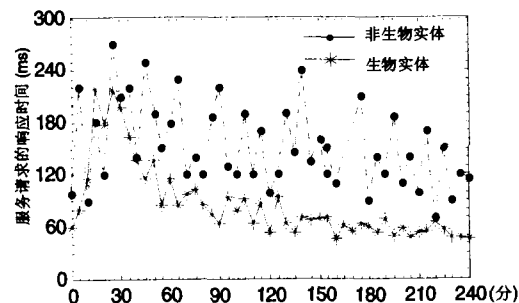


图5 每次服务请求的跳数

仿真中不考虑 LL 协商和组合的时间,每次服务请求由 3

个服务组合成,服务发现时间均等,本实验中将服务请求和服务随机地分配到 96 个节点之上,二个主要参数即:服务请求响应时间和完成每次服务请求的跳数作为测量指标。仿真结果主要用来定性的分析系统的适应性和进化性。仿真试验配置了运行生物实体(设计了变异和交叉,复制,死亡等生物操作)和运行非生物实体 Agent 的两种系统状态,分别作相关的数据仿真。仿真重复三次。取平均数据。

用户 Agent 则根据用户请求对满足其要求的服务提供能量报酬。如果单个生物实体不能满足要求,为了获得更多的能量,多个生物实体利用以上阐述的突现机制自发地形成超级实体来提供服务。

服务响应时间和每次服务请求的跳数分别是服务质量、服务费用评定的主要指标。由仿真结果可以看出(图 4,图 5),非生物实体在响应时间参数趋势没有变化。生物实体的相应数据在最初数据较高,达到 240ms;在 1h 以后,数据指标开始下降在 80 到 100 之间。这是因为生物实体在开始阶段要存储能量,进行迁移、变异和交叉操作生物操作以完成突现服务,如果生物实体代理优质的服务,则会有较高的能量,随着时间的推移,生物实体很快响应提供服务。Web 服务突现模型自适应、自进化地满足用户的需求。同样,对服务请求跳数的数据分析也可得到同样的结论。以上仿真实验结果表明,具有生物个体和移动 Agent 特点的生物实体所构建的 Web 服务突现模型具有自适应、自进化、松耦合等功能特点,能够满足未来 Internet 服务的需求。

**结束语** Web 服务是一个正在迅速发展的研究领域,服务动态管理复杂性的特点与生物系统相似。本文在研究生物智能的基础上,借鉴突现机理,结合分布式网络理论、Agent、Web 服务、语义 Web 的有关原理,设计了面向 Web 服务突现的智能模型,可为下一代 Web 服务管理的动态合成、自进化、自组织、自适应等设计提供一种新颖的研究途径和思路。其进一步的工作是扩充 Web 服务突现模型的功能,完善智能管理调节设计,并研究其在 Web 信息智能服务的应用。

### 参考文献

- 1 Sycara K, Paolucci M, Soudry J, et al. Dynamic Discovery and Coordination of Agent-Based Semantic Web Services [J]. IEEE Internet Computing, 2004, 8(3): 66~73
- 2 Casati F, Shan M-C. Dynamic and adaptive composition of e-services. Information Systems 2001, 26(3): 143~163
- 3 Meng J, Su W, Lam H, et al. Achieving Dynamic Inter-Organizational Workflow Management by Integrating Business Processes, Events, and Rules. In: Proceedings of the 35th Hawaii International Conference on System Sciences, Hawaii, USA, 2002, 10
- 4 Sheng Q-Z, Benatallah B, Dumas M, et al. SELF-SERV: A Platform for Rapid Composition of Services using Web Services in a Peer-to-Peer Environment. In: Demo Session of the 28th International Conference on Very Large Databases, Hong Kong, China, 2002. 1051~1054
- 5 Sirin E, Parsia B, Wu D, et al. HTN planning for Web Service composition using SHOP2, Web Semantics. Science, Services and Agents on the World Wide Web, 2004, 1(4): 377~396
- 6 Hai Zhuge. The Future Interconnection Environment [J]. IEEE Computer, 2005, 38(4): 27~33
- 7 Jerne N K. Towards a network theory of the immune system. Annual Immunology, 1974, 125C: 373~389
- 8 丁永生,任立红.一种基于免疫突现计算的生物网络结构的设计[J].控制与决策,2003,18(2):185~189
- 9 任立红,丁永生.一种新颖的基于生态网络计算的仿真平台[J].系统仿真学报,2002,14(11):1497~1499,1503
- 10 Gao L, Ding Y-S, Ren L-H. A Novel Ecological Network-based Computation Platform as Grid Middleware System [J]. Int J Intelligent Systems, 2004, 19(10): 859~884.
- 11 Kungas P, Matskin M. Symbolic Negotiation with Linear Logic. In: Proceedings of the Fourth International Workshop on Computational Logic in Multi-Agent Systems, LNCS, 2004, 3259: 71~88
- 12 Rao J-H, Kungas P, Matskin M. Composition of Semantic Web Services Using Linear Logic Theorem Proving. Information Systems, 2006, 31(4-5): 340~360
- 13 Winikoff M, Harland J. Implementing the Linear Logic Programming language Lygon. In: Proceedings of the International Logic Programming Symposium, Portland, USA, 1995. 66~80