

基于语义网的动态工作流互操作^{*}

郁书成 姜进磊 史美林

(清华大学计算机系 北京 100084)

摘要 动态工作流互操作是工作流互操作领域中的一个重要问题。在动态工作流互操作中,如何表示和理解工作流的语义是关键问题。此外,新兴的电子商务使得很多的工作流之间需要跨越 Internet 进行互操作,因此,如何在 Internet 上实现工作流互操作成为另一个问题。通过对工作流互操作进行分析,提出了一种以 Web 服务为服务接口、以语义网技术为服务发现机制的可以跨越 Internet 的动态工作流互操作模型。

关键词 工作流, 互操作, Web 服务, 语义网, 本体

Semantic Web Based Dynamic Workflow Interoperation

YU Shu-Cheng JIANG Jin-Lei SHI Mei-Lin

(Department of Computer Science & Technology, Tsinghua University, Beijing 100084)

Abstract Dynamic workflow interoperation is an important issue in the workflow interoperation area. The key problem of dynamic workflow interoperation is how to represent and understand the semantic of workflow. In addition, the emerging e-business makes workflow interoperation across Internet. Therefore, it becomes another problem to realize Internet-scale workflow interoperation. In this article, by analyzing the characteristics of workflow interoperation, a dynamic workflow interoperation model is proposed, which takes Web service as the interoperation interface and semantic Web technology as the service discovery mechanism and can support interoperation across Internet.

Keywords Workflow, Interoperation, Web service, Semantic web, Ontology

1 概述

工作流管理系统作为一种良好的定义、管理和执行业务流程的工具,被广泛使用于电子商务等环境中。经济一体化进程的加速使得企业之间的合作日益密切,企业往往需要突破组织边界部署其业务流程。作为业务流程执行和管理的工具,工作流系统往往需要跨越组织边界进行互操作。按照 WfMC 的定义^[1],工作流互操作是指“两个或者多个工作流引擎进行通信和交互的能力,从而使过程实例可以跨越这些工作流引擎执行和进行协调”。互操作可以发生在两个和多个工作流引擎之间、同一工作流执行服务中的两个或者多个工作流引擎之间以及不同工作流执行服务当中的两个或者多个工作流引擎之间。目前,异构平台工作流之间的互操作成为工作流互操作需要研究的重要问题。

电子商务的发展使得工作流互操作需要跨越 Internet,工作流互操作不仅仅需要考虑紧耦合的情形,还需要考虑松耦合的互操作。此外,在电子商务中,新的服务随时可能出现,旧的应用随时需要被代替,因此合作伙伴可能会经常变化。同时,企业的需求也可能会不断地变化,因此其业务流程也会经常变化。在这种情况下,工作流的各个子流程可能会不断地变更,为了适应这种动态特性,工作流需要实现子流程的动态发现和绑定功能,而解决此问题的关键是对工作流语义的描述。

Web 服务技术的出现加速了工作流互操作技术的发展,通过将工作流包装成 Web 服务、使用 XML 格式互操作消息,Internet 上各种工作流之间可以进行跨平台、松耦合的互操作。然而,Web 服务缺乏对于服务语义的描述,WSDL^[2]只能描述 Web 服务的接口语法和通信协议,UDDI^[3]也只能对

Web 服务进行简单的文字描述。在动态环境下,要实现基于语义的服务动态发现,还需要一种描述 Web 服务语义的机制。

最近兴起的语义网技术可以解决 Web 上数据的语义表示问题。它基于 RDF^[4]语言实现对对象本体的描述,使用基于各种逻辑的推理引擎实现基于语义的查询。目前,语义网技术在知识本体开发方面已经发展得比较成熟,并且已经出现了比较成熟的开发本体的语言,如 DAML + OIL^[5] 和 OWL^[6]。在 Web 服务方面,比较成熟的本体有 DAML-S^[7],但是,DAML-S 的服务特征本体只提供对于 Web 服务通用特征语义的描述,为了适应动态工作流互操作的需要,还需要对此特征本体进行一定的扩展。

本文将研究如何利用现有的语义 Web 服务技术实现工作流的动态互操作,并且考虑与现有的工作流互操作和电子商务标准相一致。第 2 节分析动态工作流互操作的特点及需要解决的问题,第 3 节给出相关的技术介绍;第 4 节提出一种基于语义 Web 服务技术的工作流互操作方案;第 5 节介绍相关工作;最后是结论和进一步工作。

2 动态工作流互操作

2.1 简介

动态工作流互操作是企业为了适应动态电子商务环境而提出的一种互操作方案。跟静态互操作不同的是,动态互操作并不在设计阶段定义好所有的子流程,而是将那些需要经常变化的子流程留到运行时进行动态发现和绑定^[8]。

2.2 需要解决的问题

要实现子流程的动态发现和绑定,需要清楚地描述子流

^{*} 本文受国家自然科学基金(批准号 60073011)以及清华大学 985 项目“高速网络协同工作环境及软件开发”资助。郁书成 硕士研究生,主要研究方向为计算机支持的协同工作(CSCW)、工作流管理等。姜进磊 博士研究生,主要研究方向为计算机支持的协同工作(CSCW)、工作流管理、高级事务处理等。史美林 教授,博士生导师、主要研究领域为计算机支持的协同工作(CSCW)和计算机网络。

程的语义,并根据此语义进行搜索、匹配和绑定。

在静态的工作流互操作中,参与互操作的双方需要事先进行相关约定,这包括互操作消息格式(如 Wf-XML^[9]等)、通信协议(如 HTTP、SOAP 等)、业务协议(如 ebXML、cXML 等)和安全策略等。因此,在动态互操作中,必须清楚地描述这些方面的语义。另外,在静态互操作中,合作双方事先知道对方能够提供什么样的服务,而在动态互操作中,必须明确的、以机器可以理解的方式表示出子流程的功能性。

有了语义的描述,另一个需要解决的问题就是根据查询条件进行语义查询。由于对于同一事物可能有多种描述方式,因此,语义查询必须根据一定的逻辑进行推理。

3 相关技术简介

3.1 语义网技术体系结构

语义网体系结构如图 1 所示。最下面两层提供通用语法,RDF 可以被看作语义网的第一层。根据 W3C 的建议^[10],RDF 是“处理元数据的基础,它为在 Web 上交换机器可理解信息的应用之间提供互操作性”。它描述三种类型实体:资源、属性和声明。RDF 的上一层是本体(Ontology)词汇层,本体是“对一个概念共享理解的明确形式化表示”^[11]。根据 Berners-Lee 的意见,本体的上一层是逻辑层(Logic),它提供各种逻辑推理方法,而当今的研究通常将本体和辑层一起考虑。体系结构最上面两层是 Proof 和 Trust 层,这两层的功能是验证声明的正确性,但是目前这方面的研究很少。想了解关于语义网的详细介绍请参见文[12]。

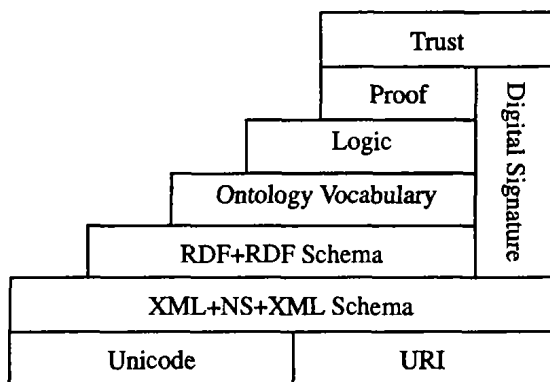


图 1 语义网体系结构

3.2 语义 Web 服务

目前对于 Web 服务语义研究最具代表性的为 DARPA 开发的 Web 服务本体 DAML-S,它是基于 DAML+OIL 语言的。DAML-S 描述了 Web 服务的服务特征(service profile)本体、过程本体和 Grouding 本体,其中,服务特征本体告诉我们“此服务能做什么”,它描述了 Web 服务的如下特征:

- 1) 服务提供者信息。包括联系信息等;
- 2) 功能描述。包括输入输出参数、前提条件和效果;
- 3) 其它特性。包括服务类别、服务质量、服务参数。

过程本体告诉我们“此服务是如何工作的”,它描述的是一个服务是如何实现的。过程可以是原子过程,也可以是复合过程。Grouding 本体则告诉描述如何访问服务,它给出诸如通讯协议、端口号等信息。

4 基于语义网技术实现动态工作流互操作

4.1 概述

根据上面的讨论,方案不仅需要支持跨平台和松耦合的

互操作,而且需要具备子流程动态发现和绑定功能。Web 服务由于对于跨平台和松耦合互操作的良好支持,成为我们首选的互操作平台,提供服务的工作流系统以 Web 服务的方式发布其服务。同时,为了支持子流程的动态发现,我们采用语义 Web 服务技术来描述子流程的语义,并实现基于语义的查询和匹配。由于在我们的方案中,子流程服务相当于一个原子过程,为简化起见,我们将服务的接口特性和调用特性都使用服务特征(service profile)本体进行描述,而省略了过程本体和 Grouding 本体。当然,为了包含服务接口信息和调用信息,此特征本体还应该进行相应的扩展,这将在后面进行讨论。

除此之外,我们认为一个好的系统还应该结合现有的标准。现有的互操作标准有多种,采用较多的是 WfMC 制订的 Wf-XML 互操作标准,目前使用的是 Wf-XML 1.0 标准,它定义了如下几种互操作调用:

CreateProcessInstance:创建远程过程实例;

ChangeProcessInstanceState:改变远程实例状态(启动、挂起和终止远程实例);

GetProcessInstanceData:请求得到远程工作流实例运行时状态数据;

ProcessInstanceStateChanged:远程实例状态改变通知。

我们的系统支持 Wf-XML 作为互操作消息。当然,考虑到目前业界对于 Wf-XML 的使用还没有完全达到 Wf-XML 标准的要求,因此,我们的方案也只考虑一些常用的 Wf-XML 特性如上述几种调用接口、Context 数据等。

4.2 系统结构

系统的结构如图 2 所示,系统使用 DAML 提供的查询语言 DQL^[13]进行查询。图中各模块的功能描述如下:

SDA(Service Discovery Agent):服务发现代理。它负责服务的搜索、消息的传递和转换以及安全策略等问题。

DQL 服务器:使用支持 DAML+OIL 规则(Rules)的推理机(Reasoner)对知识库(KB)进行推理,找出符合查询条件的服务。可供选择的推理引擎如 JTP^[14],它支持 DAML+OIL 以及其他多种推理机,并支持 DQL 查询语言。

服务知识库:保存描述服务特征的知识库。使用 DAML 解析器(Parser),将使用 XML 语法描述的服务特征本体转化为推理机所需要的知识库三元组,以供推理机进行推理。可供选择的本体解析工具有 Jena^[15]等。

系统的工作流程如下,各流程的顺序见图中标注:

- ①本地工作流发出调用请求
- ②SDA 发送包含服务特征本体属性的 DQL 查询语句至 DQL 服务器

③DQL 服务器使用推理机对服务特征知识库进行推理

④DQL 服务器返回结果集给 SDA

⑤SDA 调用搜索到的服务,并得到结果

⑥SDA 将结果传给内部工作流

在这个体系结构中,服务发现代理(SDA)承担了内部工作流与外部所有通讯的任务。SDA 的功能模块结构如图 3 所示,图中各模块的功能描述如下:

服务查询模块:负责接收内部工作流的服务查询信息,并将此信息转化为 DQL 查询语言发送至 DQL 服务器。然后从 DQL 服务器接收查询结果,并将此查询结果转化为内部数据格式发送给其他内部模块。

消息转换模块:负责消息的封装和接封。它将内部消息封装到 Wf-XML 消息或从 Wf-XML 消息中提取数据等。

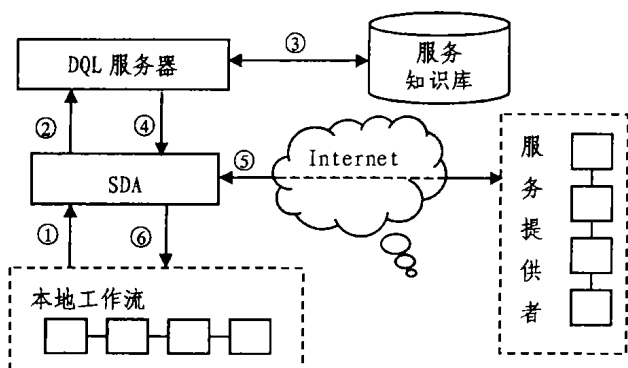


图 2 系统体系结构

安全策略模块:负责消息的加密和解密。采用哪一种方式进行加解密要根据服务查询模块查询到的对方安全策略决定。

服务调用模块:根据查询结果中的服务 URI,进行实际的服务调用。

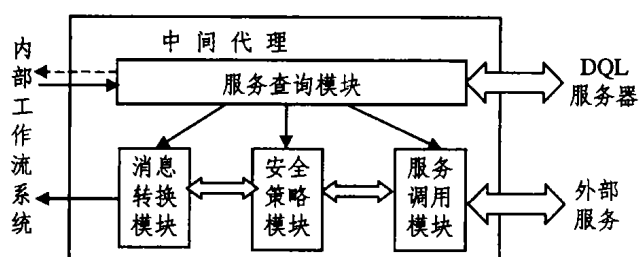


图 3 SDA 内部结构

当内部工作流系统需要调用外部服务时,先将所需服务的信息(服务种类、服务的前提和效果、服务质量要求、返回参数类型等)以及输入信息发送至 SDA 的服务查询模块,服务查询模块将这些信息转化为包含我们所扩展的特征本体(见下节)属性的 DQL 查询语句,并在查询模式(query pattern)中指明返回集,然后将 DQL 查询请求发送给 DQL 服务器。

如果没有满足条件的服务,服务查询模块返回查询失败信息给内部工作流系统(如图 3 虚线所示)。否则,它将消息格式、安全策略信息分别发送到消息转换模块和安全策略模块,这两个模块对输入信息进行格式转换和加密。最后由服务调用模块根据资源 URI 对服务进行调用。

服务调用返回时返回结果由服务调用模块返回至安全策略模块和消息转换模块进行解密并转换成内部工作流使用的消息格式,最后由消息转换模块将结果数据返回给内部工作流系统。

4.3 Web 服务本体扩展

DAML-S 特征(profile)本体所描述的只是 Web 服务最基本的特征信息。在工作流互操作应用中,只提供上述信息是不够的,服务使用者还需要关于服务提供者更多的语义信息。因此,还需要对服务特征本体进行扩展。根据 2.2 节的讨论,我们需要增加消息类型、通信协议、业务协议和安全策略等语义信息。

消息类型指定了采用何种互操作消息,在我们的解决方案中,采用的是 Wf-XML 消息。当然,将来也可以对此方案进行扩展而支持其它消息格式,这也是消息类型属性的功能所在。

业务协议是针对具体应用而采用的各种电子商务协议,如 ebXML、cXML 等,也可以是某业界采用的其它电子商务标准。由于我们采用 Wf-XML 作为互操作消息,因此,业务协议消息应该被封装于 ContextData 内。

安全策略包括用户身份认证策略和基于消息的加密策略。

另外,由于我们没有使用过程本体和 Grounding 本体,因此,在我们的特征本体中还应该加上 Web 服务的调用特征,如接口特性、消息特性、端口特性、外部 XML Schema URI 等信息,以供服务调用模块使用。

根据上面的想法我们将服务描述本体进行扩展,其 UML 图如图 4 所示。图中虚线以下的部分是我们所扩展的特征属性。

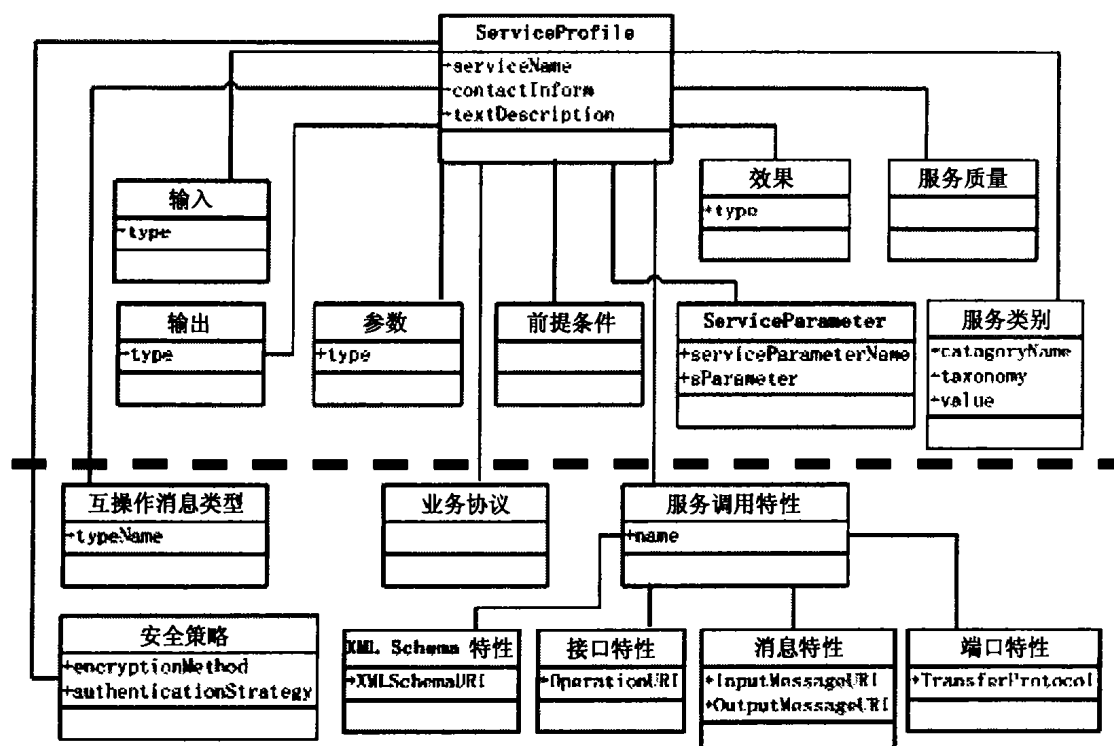


图 4 扩展后的服务描述 UML 示意图

所扩展的各项特性用 DAML+OIL 语言描述如下:

```
<daml:Class rdf:ID="Profile" />
<daml:Property rdf:ID="wfmessageType">
  <daml:domain rdf:resource="#Profile" />
  <daml:range rdf:resource="#WfServiceMessageType" />
</daml:Property>
<daml:Property rdf:ID="safeStrategy">
  <daml:domain rdf:resource="#Profile" />
  <daml:range rdf:resource="#ServiceSafeStrategy" />
</daml:Property>
<daml:Property rdf:ID="businessProtocol">
  <daml:domain rdf:resource="#Profile" />
  <daml:range rdf:resource="#Thing" />
</daml:Property>
<daml:Property rdf:ID="invokeProperties">
  <daml:domain rdf:resource="#Profile" />
  <daml:range rdf:resource="#invokeProperties" />
</daml:Property>
<daml:Class rdf:ID="ServiceSafeStrategy" />
<daml:DatatypeProperty rdf:ID="encryptionMethod">
  <daml:domain rdf:resource="#ServiceSafeStrategy" />
  <daml:range rdf:resource="#Thing" />
</daml:DatatypeProperty>
<daml:DatatypeProperty rdf:ID="authenticationStrategy">
  <daml:domain rdf:resource="#ServiceSafeStrategy" />
  <daml:range rdf:resource="#Thing" />
</daml:DatatypeProperty>
<daml:Class rdf:ID="invokeProperties" />
<daml:Property rdf:ID="XMLSchema">
  <daml:domain rdf:resource="#invokeProperties" />
  <daml:range rdf:resource="#XMLSchema" />
</daml:Property>
<daml:Property rdf:ID="OperationProperties">
  <daml:domain rdf:resource="#invokeProperties" />
  <daml:range rdf:resource="#OperationProperties" />
</daml:Property>
<daml:Property rdf:ID="MessageProperties">
  <daml:domain rdf:resource="#invokeProperties" />
  <daml:range rdf:resource="#MessageProperties" />
</daml:Property>
<daml:Property rdf:ID="PortProperties">
  <daml:domain rdf:resource="#invokeProperties" />
  <daml:range rdf:resource="#PortProperties" />
</daml:Property>
```

各项特征的属性由于篇幅的原因这里不再一一给出,下面只给出互操作消息类型特征中 typeName 属性的描述方式,其他特征的属性描述方法类似:

```
<daml:Class rdf:ID="WfServiceMessageType" />
<daml:DatatypeProperty rdf:ID="typeName">
  <daml:domain rdf:resource="#WfServiceMessageType" />
  <daml:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</daml:DatatypeProperty>
```

经过扩展的 Web 服务本体不仅包含了 Web 服务的一般特性,更包含了工作流互操作所需要的几种关键语义特征,因此适用于工作流互操作的需要。

有了扩展的 Web 服务本体,就可以使用此本体描述 Web 服务,使用 DAML 解析工具将其存放到服务知识库中,供 DQL 服务器进行查询。

需要注意的是,本方案并不依赖于 DAML+OIL 语言,任何能够描述本体的语言都可以被用来描述我们所需要的 Web 服务本体。

5 相关工作

当前对于动态工作流互操作的研究主要两种模式:基于合同的互操作和基于多代理的互操作。

在基于合同的解决方案系统中,参与协作的各方需要一份各方都认可的合同模板,并基于此模板在注册服务器上注册自己的服务,使用者通过查询注册服务器得到其期望的合作方,并根据合同规定进行互操作。基于合同的解决方案有 WISE^[16]、CrossFlow^[17]等。

在基于多代理的解决方案中,参与协作的各方需要使用一致的代理通信语言(ACL)进行通信,每个代理通过一个注

册服务器注册自己的服务以供使用者查询。基于代理的解决方案有 ADEPT^[18]等。

基于合同的解决方案要求协作各方共享一份合同模板,由于缺乏语义描述,这种合同只能被各参与协作者理解,而无法被广泛使用于互联网上。传统的基于多代理的解决方案依赖于代理间互相能够理解的代理通信语言(ACL)进行通信,因此只适合于组织内部,对于跨组织甚至是在 Internet 上实现动态互操作,传统的代理解决方案并不适合。

结论和进一步工作 本文讨论了基于语义网动态工作流互操作问题,通过分析工作流互操作,特别是动态工作流互操作的特点,给出了一种基于语义 Web 服务技术的解决方案,解决了子流程的动态发现和调用问题。

基于语义 Web 服务的动态发现机制能够很好地适用于动态协作环境。随着技术的发展,各种本体之间将可以方便地进行语义转换,从而可以跨越服务知识库进行服务查询,因此可以将互操作扩展到整个互联网上。

但是,这种方案由于技术的限制还存在以下几点需要发展的地方:其一是当前语义网技术还在发展,现在的工作只是在本体描述方面比较成熟,而且各个领域的本体还没有完全形成标准;其二是虽然利用语义 Web 服务能够解决动态服务发现和调用,但是将工作流作为 Web 服务进行调用只能实现粗粒度的互操作,也就是 WfMC 所定义的嵌套子过程互操作模型。要实现细粒度动态工作流互操作,还需要描述服务提供者工作流的内部结构,并根据此结构进行查询和协商,这些都是以后工作所需要解决的问题。

参考文献

- 1 WfMC. Workflow Standard - Interoperability Abstract Specification. Workflow Management Coalition, WfMC-TC-1012, Nov. 1999
- 2 W3C. WSDL Recommendation. World Wide Web Consortium, March 2003. <http://www.w3.org/TR/2003/WD-wsdl12-20030303/>
- 3 <http://www-900.ibm.com/developerWorks/cn/webservices/ws-theme/ws-uddi.shtml>
- 4 W3C. RDF Specification. World Wide Web Consortium. <http://www.w3.org/RDF/#specs>
- 5 <http://www.daml.org/2001/03/daml+oil>
- 6 W3C. OWL Web Ontology Language Recommendation. World Wide Web Consortium, Feb. 2004. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
- 7 <http://www.daml.org/services/daml-s/0.7/>
- 8 Medjahed B, et al. Business-to-business interactions: issues and enabling technologies. The VLDB Journal, 2003, 12(1): 59~85
- 9 WfMC. Workflow Standard - Interoperability Wf-XML Binding. Workflow Management Coalition, WfMC-TC-1023, May. 2000
- 10 W3C. RDF Model and Syntax specification. World Wide Web Consortium, Feb 1999, <http://www.w3.org/TR/REC-rdf-syntax-19990222/>
- 11 Gruber T R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino N, Poli R eds. Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993
- 12 W3C. Semantic Web. World Wide Web Consortium, <http://www.w3.org/2001/sw/>
- 13 <http://www.daml.org/2003/04/dql/>
- 14 <http://www.ksl.stanford.edu/software/JTP/>
- 15 <http://www.hpl.hp.com/semweb/jena.htm>
- 16 Lazcano A, Alonso G, Schultdt H, et al. The WISE approach to electronic commerce. International Journal of Computer Systems. Science & Engineering, 2000, 15(5): 345~357
- 17 Grefen P, Aberer K, Hoffner Y, et al. CrossFlow: Cross-Organizational workflow management in dynamic virtual enterprises. International Journal of Computer systems, Science, and Engineering, 2001, 15(5): 277~290
- 18 Jennings N R, Norman T J, Faratin P, et al. Autonomous agents for business process management. International Journal of Applied Artificial Intelligence, 2000, 14(2): 145~189