维普资讯 http://www.cqvip.com

支持 Web Services 语义组合的框架 IRS-II:分析与讨论*)

吴步丹[1] 何丕廉[1] 金 芝[2]

(天津大学计算机科学与技术系 天津 300072)[1] (中国科学院数学与系统科学研究院 北京 100080)[2]

摘 要 Web Services是基于Internet进行分布式计算的基本元素,面向Service的计算将成为未来计算技术发展的趋势。但如何能够自动发现需要的Service资源,提高Web Services组合的智能化程度,是近来的研究热点之一。IRS-II首先提出基于任务本体的Service发布、发现和组合方法,可以说是将Service发现和组合从语法的层次提升到了语义的层次。本文首先深入分析了IRS-II提出的方法及其体系结构,然后对Web Services的智能检索和服务组合的问题进行讨论。

关键词 Web Services,领域任务模型,IRS-II,UPML,Service发现和组合

Analysis and Discussion of a Framework for Semantic Web Services Composition

WU Bu-Dan[1] HE Pi-Lian[1] JIN Zhi[2]

(Computer Department, Tianjin University, Tianjin 300072)[1]

(Academy of Mathematics and System Sciences, The Chinese Academy of Sciences, Beijing 100080)[2]

Abstract Web Services are essential in internet-based distributed computing, Service-oriented computing will become the direction of computing technology. The problems of how to discover proper Services automatically and how to enhance the intelligent composition of Web Services are within the hot topics recently. IRS-II is a framework for Web Services composition. It proposes a task ontology-based methodology of Service publishing, discovery and composition, which enhances the level of Service discovery and composition from syntax to semantics. This paper firstly investigates the methodology and architecture of IRS-II in depth. And then some deeper topics in the intelligentized Web Services discovery and composition have been discussed.

Keywords Web Services, Domain task models, IRS-II, UPML, Service discovery and composition

1 引言

随着Internet技术的发展和广泛应用,Web Services越来越受到人们的重视。但什么是Web Services呢?目前仍没有统一的定义。W3C曾给出这样一个定义,即它是能被URI标识的一个软件应用,它的接口和绑定能够由XML程序来定义、描述和发现,并且通过基于Internet协议,采用基于XML的消息支持与其它软件应用的直接交互。这个定义有点过于依赖当前的实现平台,从中抽取出其关键特征,可以得出Web Services是Internet使能的、自含的计算实体,是完全封装好的一段计算机程序。分布在Internet上的各种各样的Web Services已经成为一类重要的Web资源。设想当这样一些Web Services能够在Internet上真正实现自由的互联、互通、协作和联盟的时候,就形成了一种与当前的信息Web类似的软件Web。一种理想的情况是,这些Web Services能够自主感应它们所处的环境中的需求,并实现这一需求。而且,在单个Web Services不能实现全部需求的情况下,多个小粒度的Web Services能够自主组合,形成更大粒度的Web Services,最终实现需求。这些为了实现新的需求自发组织起来的、在软件Web上运行的系统,将能够感知外部网络环境和系统需求的变化,并随着这种变化按照一系列指标进行静态调整和动态演化,尽可能使自己能够具有更高的用户满意度。

目前,Web Services的支撑平台结构如图1所示。

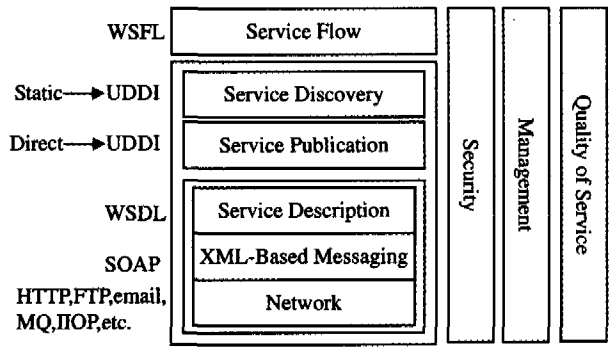

图1 Web Services的平台层次结构

如果把Web Services出现之前访问网络资源和后来使用Web Services都看成对Internet资源的访问,那么Internet上对资源的描述方法就从单一的URL,HTML,HTTP发展到SOAP,WSDL,UDDI,WSFL与原有描述方式的结合。SOAP是在分布式环境中通过XML编码进行通信的一种简单网络协议。WSDL是定义如何使用XML语法描述Web Services

*)(部分)受到国家重点基础研究发展规划的资助(编号:2002CB312004)。**吴步丹** 硕士研究生,主要研究领域为语义Web服务、自然语言处理、人工智能;**何丕廉** 教授、博导,主要研究领域为人工智能与计算机辅助教育、自然语言处理、Web检索与挖掘;**金 芝** 研究员、博导,主要研究领域为需求工程、基于知识的软件工程、知识工程和人工智能。

维普资讯 http://www.cqvip.com

的规范。UDDI是统一描述、发现和集成协议的简称。WSFL是描述服务的组成和流程的服务流语言。SOAP，WSDL，UDDI，WSFL的出现，为Web Services提供了建立在原有底层传输协议之上的不同层次的描述。新的描述更准确、丰富地描述了Web Services，而且带来了描述上的灵活性。一个可以使用的Web Services，可按照需要选用若干层次的功能，而无需所有的特性。

但是，这些描述只支持手工检索和手工组合，目前主要有直接发送服务描述、利用Web技术发送、使用WSDL文件仓库和利用UDDI四种服务发现机制[1]。

要实现自动的智能检索和智能服务组合，起码需要有下面两个方面的支持：第一，能刻画Web Services的语义。在这方面，DAML-S[2]和WSMF[3]做了一些较为深入的工作。第二，也是目前最具挑战性的问题，是能支持Web Services语义理解和语义组合的框架。

IRS-II[4]是英国Open University提出的在支持Web Services语义理解和语义组合的一个尝试。它将基于知识的方法应用于Web Services的检索和组合之中，并以UPML框架[5]为基础，显式地刻画了领域知识（即领域任务模型）和抽象的问题求解方法，并通过一定的推理手段来实现任务和问题求解方法之间的映射，在一定程度上展现出对语义Web Services的支持。

本文首先对IRS-II提出的方法及其体系结构进行深入的分析，然后针对IRS-II方法在实现Web Services的智能检索和智能服务组合上的问题进行讨论。目的在于揭示基于知识的方法在Web Services的语义理解和智能组合方面所能起到的作用，从而进一步促进这个方面的研究。论文分为4节：第2节分析IRS-II体系结构；第3节讨论IRS-II应用案例；第4节是问题与讨论；最后是结语。

2 IRS-II体系结构

2.1 UPML框架

IRS-II是以UPML（Unified Problem-solving Method Description Language）为基础构建起来的。UPML框架由6种元素组成：任务、领域模型、问题求解方法、本体、桥、精化器。它们之间的关系如图2所示。

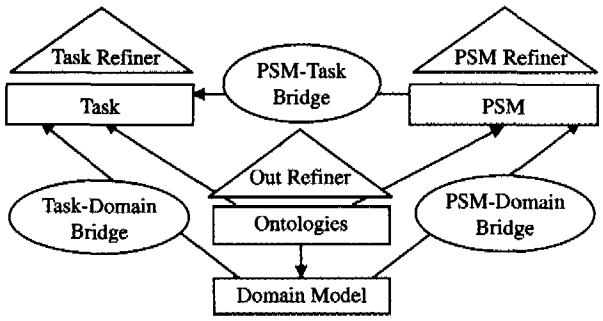

图2 UPML框架

• 本体（ontologies）

本体是概念化的明确的规范说明。在UPML框架中，本体用来定义任务，问题求解方法和领域模型的术语及其属性。比如exchange-rate-provision这个任务的定义中就包括了money-ontology这个领域本体。一个任务的定义可能涉及到很多个领域本体。

• 任务（tasks）

对任务的描述主要有两大部分：一是对任务要达到目标的描述，二是对领域知识的前提假设和任务输入的前提条件的描述。对象来看，关于领域知识的前提假设的定义和任务输入的前提条件都是系统的输入，但任务描述将它们区分开了：前提条件是动态输入上的条件，前提假设的定义是推理器所参照的知识推理中的前提条件。另外，任务可以是一种高层次的对象任务，例如classificationv2；也可以是具体的任务，例如exchange-rate-provision。

• 领域模型（domain models）

领域模型描述了定义问题求解方法和任务所需要的领域知识。UPML中用三种元素来定义领域模型：领域知识特征描述、领域知识、外部领域模型的假设。领域模型中的元素是对象层次较高的应用，在实际应用中可以只实现某应用的一个子应用。

• 任务-领域桥（bridges）

桥建立任务和领域的对应，因为对任务的描述是独立于领域的，可以复用的。构成UPML的每一种元素都可以采用适合于自己的本体，所以需要桥来在不同的模型间建立对应关系。比如任务exchange-rate-provision采用了ontology A，问题求解方法exchange-rate-provider-ontology采用了ontology B。桥建立对应关系后，一个本体下的问题求解方法可解决不同本体下的任务。

• 问题求解方法（PSMs）

问题求解方法分为两种：复杂PSMs和简单PSMs。复杂PSMs将任务分解成子任务；简单PSMs用领域知识的假设进行推理，实现一定粒度的任务所要求实现的功能。简单PSMs用来解决任务，一个粒度足够小的任务对应一个简单PSM。问题求解方法可以是抽象高层次的，例如Heuristic-Classification[6]；也可以是具体的可以解决某一细粒度任务的，例如exchange-rate-provider-ontology就可以解决exchange-rate-provision这个任务。

• PSM精化器（refiner）

精化器增强PSM的适用性，提供更加精化而且可复用的PSMs，以避免一个任务请求过多的PSMs构件的组合爆炸问题。当一般性PSMs的所有精化版本都不能与任务和领域特定的环境相匹配的时候，就用一般性PSMs来完成任务的某一功能。

在UPML中，任务与问题求解方法的关系是实现知识推理的重要基础。如图3所示，待解决的问题可由复杂PSM分解为大任务，大任务由复杂PSM继续分解成小任务，呈树状结构。当子任务粒度分解到足够小时，就可用某一简单PSM来解决。研究IRS-II的PSM model和task model发现，task model中都是粒度很小的任务，在PSM model中都可找到直接的问题解决方法来解决。一个任务下可能有多个问题解决方法，例如exchange_rate_provision这个任务下就有三个问题解决方法：European-exchange-rate-provider，non-european-exchange-rate-provider，exchange-rate-provider，它们都可以解决exchange_rate_provision这个task，取决于具体应用的要求。所以问题解决方法间可形成树的结构，European-exchange-rate-provider和non-european-exchange-rate-provider都是exchange-rate-provider的孩子结点。图3所示的是任务-PSM关系的一般结构[7]。树根是高层次的抽象任务，例如诊断或设计任务，用任务类型表示。比如树根是设计这一问题类型，则根据领域本体可能有多种设计问题：旅游路线设

维普资讯 http://www.cqvip.com

计、房屋结构设计、执行程序设计等等，不同的根问题由不同的复杂 PSM 来分解。图 3 所示从树根引出的虚线表示不同问题的分解。比如 Decomposition-Method-1 表示对旅游路线设计问题的分解，Decomposition-Method-n 表示对执行程序设计问题的分解。问题分解成任务后，任务要么由 Primitive-Method 也就是简单 PSM 直接完成，要么由 Decomposition-Method 也就是复杂 PSM 分解成子任务。这样的解决问题方式是自顶向下，逐步求精。面向任务的问题求解方法，也是近来研究 PSMs 和知识系统的常用方法。

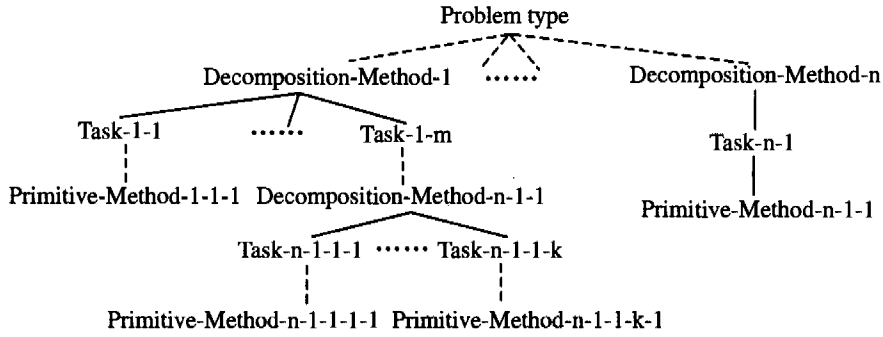

图 3　任务-PSM 关系的一般结构

2.2　IRS-II 的结构

图 4 所示是 IRS-II 的体系结构，它由三大部分组成：IRS 服务器、IRS 发布器和 IRS 客户端。这三部分通过基于 SOAP 的协议进行交互。

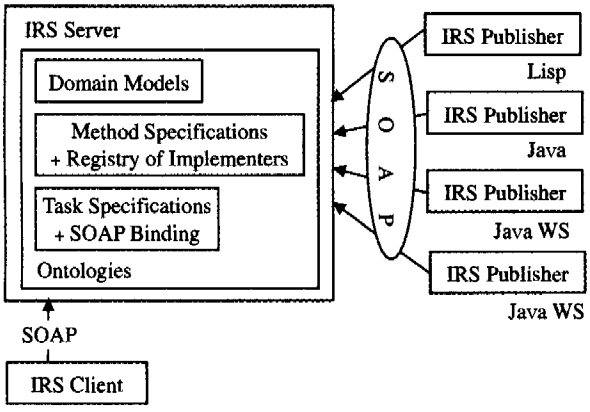

图 4　IRS-II 体系结构

• IRS 服务器

IRS 服务器提供本体支持，Web Services 的语义描述，这个本体包括三部分：领域模型、方法规格说明和任务规格说明。领域模型描述了定义问题求解方法和任务所需要的领域知识。方法规格说明描述用于解决任务的方法并提供解决任务的推理。任务规格说明描述任务的前提条件、完成条件、要达到的目标等。IRS 服务器中有两个不同层次 Web Services 语义描述：知识层次的描述和映射机制描述。服务器中的本体构成了第一层次知识层次的描述，使用 OCML[8] 语言。第二层次的映射机制描述将服务功能描述和具体的 Web Services 对应起来，使用 DAML-S 中的 grounding 概念。

• 任务规格说明

图 5 是一个任务的描述定义。该任务有两个 input-role、一个 output-role。支持性定义（如 currency 和 positive_number 定义）在与该任务相关联的本体或该任务所包括的本体中。图 5 没有对任务的前提条件的描述，实际上 input-role 暗示了前提条件。Web Services 的协调和组合通过任务描述中的前提和目标表达式实现。

```
(def-class exchange_rate_provision(goal-specification-task)
  ? task
  ((has-input-role :value has_source_currency
                   :value has_target_currency)
   (has-output-role:value has_exchange_rate)
   (has_source_currency:type currency:cardinality 1)
   (has_target_currency:type currency:cardinality 1)
   (has_exchange_rate:type positive_number)
(has-goal-expression
   :value(kappa (? psm? sol)
                (=? sol(the official_exchange_rate
                        (role-value? psm has source_currency)
                        (role-value
                        ? psm has target currency)))))))
```

图 5　exchange_rate_provision 任务的定义

每一个任务的描述都对应一个 soap-bindings，如图 6 所示。

```
(def-irs-soap-bindings
 exchange_rate_provision_ontology //ontology name
 exchange_rate_provision //task name
 ((has-source-currency"xsd:symbol")//source currency
  (has-target-currency"xsd:symbol")) //target currency
 "xsd:float") //output
```

图 6　exchange_rate_provision 任务的 soap-bindings

它用来实现 Web Services 描述和语义描述（如任务描述）的整合，将 input-role 和 output-role 映射到 soap 类型。任务规格说明是本体中的描述，本体描述明确了所需要和产生的知识，是知识层面的描述。Soap 类型将这些知识转化为通信层的编码符号，形成符号层面的描述。一个 Web Services 必须同时符合知识层和符号层描述的要求，才能解决某一个任务。

• 问题求解方法规格说明

方法规格说明（PSMs）和实现方法的注册 Web Services 都存在 IRS 服务器中。如图 7，方法规格说明中也有 input-role 和 output-role，但是 source-currency 和 target-currency 都是具体 european_currency。每个描述中还有 precondition 和 postcondition，查找并调用 Web Services 由前提和事后条件来支持。图 7 所示的服务能否解决兑换外汇的任务，取决于方法规格说明中 ECB 提供的兑换率是否与任务要求的一致，或者匹配代理[9] 是否认为 PSM 和任务的要求是否足够相似。

```
(def-class MM_Bank exchange rate provider (primitive-
 method)
 ? psm
 ((has-input-role
```

维普资讯 http://www.cqvip.com

```
    :value has-source-currency
    :value has-target-currency
(has-output-role
 :value has-exchange-rate)
(has-source-currency: type european currency: cardinality
1)
(has-target-currency: type european-currency: cardinality 1)
(has-exchange-rate: type positive_number)
(has-precondition
    :value(kappa(? psm )(stock-available
                    ( role-value? psm has-target-cur-
                    rency))
(has-postcondition
    :value(kappa(? psm? sol)
          ( =? sol ( the-European-Central-Bank-exchange-
                    rate
                    (role-value? psm has-source-currency)
                    ( role-value? psm has-target-curren-
                    cy)))))))
```

图 7 解决任务 exchange_rate_provision 的 PSM 描述

· IRS 发布器

IRS 发布器有两项功能：

一是关联 Web Services 和其存在于 IRS 服务器中的语义描述。一条语义描述可以对应多个实现该描述的服务，一个服务也可对应多条语义描述。

二是为单独的 java 或 lisp 代码生成一系列包装，使它们发布成由问题求解方法抽象描述的 Web Services。发布器可以为这样的代码自动生成 Web Services 端点。

在 IRS 发布器上发布标准的 Web Services，使用如图 8 的界面。

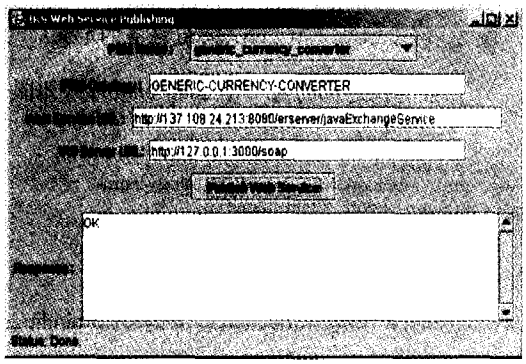

图 8 IRS-II 中发布一个 Web Services 的界面

将单独的 java 和 lisp 代码发布成 Web Services，需要使用一个简单的 API。如果是 lisp 代码，需要使用宏 irs-method-registration。

· IRS 客户端

IRS-II 有一个重要特征：它是功能驱动的。IRS 客户端提供一个界面和一系列有关任务的 API 来支持功能驱动。用户在 IRS-II 的界面上选择一个服务请求相应任务后，点击“achieve task”按钮，IRS-II 就会定位到相应的各问题求解方法并调用相应的 Web Services。这种功能驱动的调用也可以通过编程来实现。

3 IRS-II 应用案例

研究者们已经/正在为 IRS-II 开发多种应用，本节用“病人运送”作为例子来说明其服务发布和组合服务的过程。该案例的背景是：英国有一项医疗政策，为了减少一些不是很危急病人在某一医院的等候诊治时间，国家给这种病人选择到英国本土其他医院接受治疗的机会。因此，一个病人可能在本地医院进行诊断后，需要转送到英国的其他医院进行治疗。在这个过程中，病人需要查找可以提供相应治疗的医院，查询医院的治疗费用、运送病人的开支等。

为了支持这个服务，IRS-II 首先需要分析其中可能蕴涵的服务，通过分析得出如下五类 Web Services：

· 诊断和推荐治疗方案的服务。相应 PSM：Arthritis-Diagnosis-Service，Arthritis-Therapy-Service。

· 查找可以实施所需手术的电话黄页服务。相应 PSM：Medical_Service_Locator。

· 医院中答复某手术可行性和价格的服务。相应 PSM：The_Hippy_Hospital_Bed_Service，Not_The_Hippy_Hospital_Bed_Service，Hip_Hip_Hospital_Bed_Service，Another_Hippy_Hospital_Bed_Service。

· 可以提供护送病人价格的救护服务。相应 PSM：Rail_Ambulance_Service，Air_Ambulance_Service。

· 兑换本地货币和手术实施地货币的外汇兑换服务。相应 PSM：Exchange-Rate-Provider，Generic_Currency_Converter。

其次，再建立从总任务到这些基本的问题求解方法的分解过程，得到任务分解树，如图 9 所示。其中，圆点表示任务分解，非叶矩形节点为任务，树的叶节点则为具体的问题求解方法。

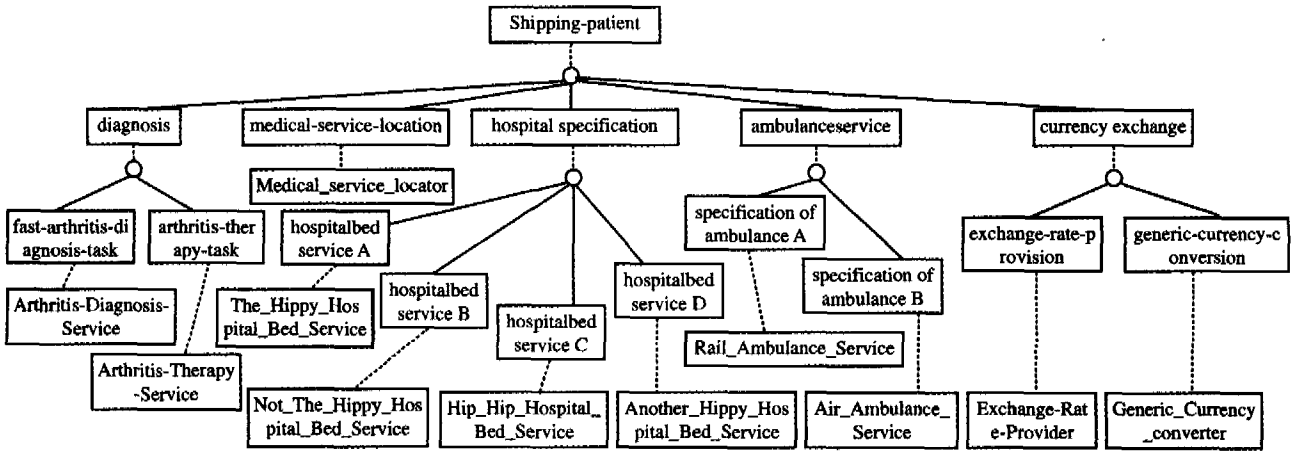

图 9 病人运送任务分解树

维普资讯 http://www.cqvip.com

仅有这个只含任务名字任务分解树还不够，IRS-II 还需要为每个任务和问题求解方法建立各自的描述（见本文图 5 和图 6 的任务描述，以及图 7 的问题求解方法描述）。至此，领域知识库的构建工作得以完成。

有了这个领域知识库，就可以开始用这个平台进行 Web Services 的发布和组合工作。当然，在提供组合服务之前，需要有人发布基本的服务（服务发布界面如图 8 所示）。假设上述任务分解树中所有叶节点上的问题求解方法都有相应的 Web Services 被发布，并假设有一个服务请求 patient shipping。下面的过程展示了服务组合的过程。

1. 服务器接收病人填写的症状并发送给问题求解方法：shipping-patients-service，该方法处理完后由客户端将处理完成消息发送给服务器。服务器将症状发送给 arthritis-diagnosis-service，该方法将诊断结果 severe-osteoarthritis 通过客户端发送给服务器。服务器启动任务：arthritis-therapy-task，以 severe-osteoarthritis 为条件，调用相应方法 arthritis-therapy-service。方法处理结果是 Arthroplasty theropy，需要 hip-replacement，并发送给服务器。

2. 服务器启动任务 medical-service-location，任务的输入角色是 has medical service。服务器调用相应方法 medical-service-locator，该方法以 has medical service hip-replacement 为输入值，将处理结果 the-hippy-hospital 发送给服务器。

3. 服务器启动任务 hospital-specification，调用方法 hip hip hospital bed service，处理结果可以手术的时间（month-of 6）（year-of 2003）发送给服务器。服务器调用方法 another hippy hospital bed service，处理结果（day-of 13）（month-of 8）（year-of 2003）发送给服务器。服务器调用方法 the hippy hospital bed service，处理结果 alendar-date（day-of 7）（month-of 7）（year-of 2003）发送给服务器，服务器确认 has hospital name the-hippy-hospital，has medical service hip-replacement，result：（date 2748 calendar-date（day-of 7）（month-of 7）（year-of 2003））。

4. 服务器启动任务 ambulanceservice，该任务的输入是：has currency Euro，has destination-location Paris，has start-location Milton-Keynes。服务器调用相应方法 air-ambulance-service，处理结果 has currency Euro，the result 3000 发送给服务器。服务器调用方法 generic-currency-conversion，输入是 has target currency "pound"，has source currency "euro"，has amount 3000。服务器调用方法 exchange-rate-provider，输入是 has target currency "pound"，has source currency "euro"，处理结果 the result 0.695482 发送给服务器。服务器将 the result 0.695482 发送给方法 generic-currency-conversion，处理结果 1918.6446 发送给服务器。

5. 服务器将结果发送到方法 shipping-patient-service，结果内容为：20 of June，2003. It will cost 1919 Pound to ship the patient.

该案例表明：

• IRS 服务器决定各个任务及相应方法的调用顺序，这种顺序是作为知识预先存在服务器中的。例如，在完成了 diagnosis 任务后，服务器就自主启动下一个任务 medical-service-location，调用顺序完全依照服务器中预先存好的任务启动顺序。

• Service 组合由问题求解方法的前提条件和后置条件来支持。每个问题求解方法都直接接收由服务器发送来的消息，这些消息符合方法规格说明中的前提条件，各方法就可以顺利进行自己的处理，处理完的结果符合该方法的后置条件。服务器将上一个 Web Services 的处理结果发送给下一个问题求解方法，并且该结果符合其前提条件，使得各个 Web Services 能够无缝地组合起来。

• IRS-II 的服务调用是功能驱动的。一个服务完成后，客户端发送"achieve-task"消息给 IRS 服务器，服务器调用能提供下一个功能的 Web Services。例如，当方法 air-ambulance service 给出救护车费用 3000Euro 后，客户端获取到方法 air-ambulance service 完成的消息，并发送"achieve-task"的消息给 IRS 服务器，服务器依照预先存好的任务启动顺序得知需要将 3000Euro 转换为英磅，因此接着调用方法 generic-currency-conversion。服务器的每一次服务调用是由接收到客户端发出的"achieve-task"消息触发，而不是由 Web Services 将本服务的处理结果发送给服务器来触发，如图 10。

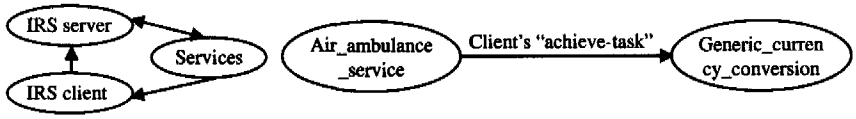

图 10 IRS 的功能驱动

4 问题与讨论

4.1 IRS-II 的优点

第一，领域知识在 IRS-II 中起到了重要作用，IRS-II 客户端软件中可视化地给出了领域任务模型。KMI 对领域知识的定义[10]是：Knowledge about a particular domain，as distinct from task，method or application knowledge。IRS-II 的领域任务模型解决以下问题：

• 应用（application）的需求分析过程就是任务分解过程。IRS-II 工具中的 Applications 定义了任务和问题求解方法可以解决的问题；Task models 为某一领域内新应用的开发提供了可复用的任务模块，任务和问题求解方法提供可复用的软件资产。

• IRS-II 的领域任务模型提高了用户的自主性、灵活性和应用的可扩充性。当某一领域内的新应用出现时，有权限的用户可以自行编写任务、问题求解模型来扩充领域任务模型，注册实现问题求解模型的 Web Services 之后就可以实现新的应用。因此 IRS-II 是一个实现以领域任务为模型、面向任务的应用的软件平台。

第二，IRS-II 用问题求解方法（PSM）来抽象地表示 Web Services，向任务模型屏蔽了具体的实现，从而使系统的搭建更为灵活。在 IRS-II 中，简单 PSM 可以解决一定粒度的任务。假如去掉 PSM，直接将 Web Services 与一定粒度的任务相对应，则会产生一系列的问题：

• 一个任务要有多个接口来对不同语言编写的 Web Services 进行识别。而有了问题求解方法，一个任务只需要

维普资讯 http://www.cqvip.com

对应一个问题求解方法或少数几个问题求解方法(PSM 间有父子继承关系时),而让不同语言编写的 Web Services 内部来处理其与问题求解方法的接口。

• Web Services 没有统一管理。一个任务下的问题求解方法可以有父子继承关系,例如 exchange-rate-provider 这个父 PSM 有两个孩子 PSMs:european-exchange-rate provider 和 non-european-exchange-rate-provider。这种继承关系反映 PSM 解决同一任务的强弱能力,用户可以根据应用的不同,用不同强弱版本的 PSMs 及其相应的 Web Services 来解决具体应用中的任务。

第三,IRS-II 采用 UPML 中任务 task 和问题求解方法 PSM 区分的机制。任务是聚集 Services 的基本机制,一个任务可以聚集多个 Service。只要找出一个 Service 在哪个 task 下,就可以得知这个 Service 的类型,而不需要查询服务提供者来得知该 Service 的类型。在 DAML-S 中,一个任务被定义成一个服务类,具体解决任务的 Service 就是该类的一个对象。这样,Service 和任务间是 is-a 的继承关系,我们无法清楚地区分任务与 Service 的区别。而实际上,is-a 的继承关系并不总是有效的,一个问题求解方法可能只能解决一个 task 的弱版本,用一个弱版本的方法来解决其任务是不完全的。任务和方法的区分为匹配代理对二者的匹配程度进行推理提供了基本框架。

第四,IRS-II 在对 domain ontologies, task models, PSM models 的描述中增加了与本模块描述相联系的模块描述,使得任务推理更加灵活准确。domain ontologies, task models, PSM models 之间关联成一个复杂的网络,当一个需求生成时,IRS-II 就面向任务依照网络自动分解任务进行推理,并找到需求所要求的其它任务,这是 IRS-II 中服务的检索方式。直到将所有需求所要求的任务都分解到可以用简单 PSM 解决的粒度后,IRS 服务器调用各个 PSM 相应的 Web Services 运行整个应用,由 IRS 服务器负责服务的组合。因此,一个应用能实现的前提条件不仅是 IRS-II 中存在完成该应用的所有 task,PSM 和注册的 Web Services;还必须在 domain ontologies, task models, PSM models 之间存在关联网络。这样,IRS-II 才能依照网络进行推理、分解任务,找到相应的 Web Services。图 11 给出 arthritis-diagnostic-fast 这个 application 的部分 applications, domain ontologies, task models, PSM models 网络推理图,图中箭头的正指向代表 include 关系。

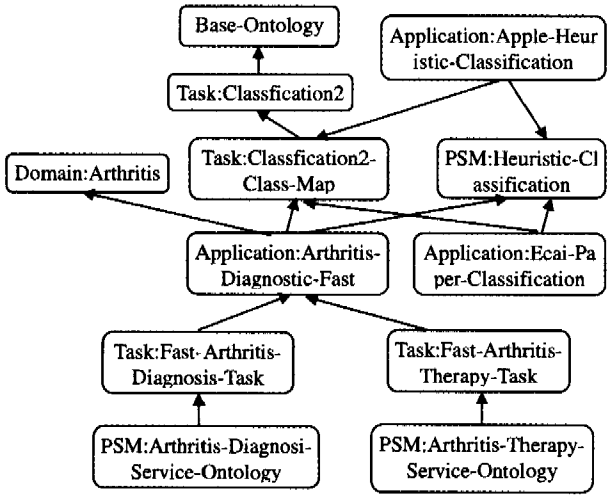

图 11 arthritis-diagnostic-fast 的网络推理图

4.2 进一步的问题

分布式计算是实现网络环境中资源共享和平衡负载的技术。随着 Internet 的普及和延伸,Internet 环境上跨平台、跨语言和松耦合的 Web Services 分布式计算模式应运而生,成为了分布式计算的新热点。它融合了网格计算中"无处不在的、集成的计算与资源环境"的思想,将 WWW(World Wide Web,万维网)上静态的、无结构的 Web 页面扩展到 Internet 上自主行为的、具备执行任务能力的服务(Web Services),试图实现分布式系统间的跨平台、跨语言和松耦合。实现自主自治的 Web Services 发现、调用和组合,是该领域的最终研究目标。

IRS-II 告诉我们,领域知识将在自主自治的 Web Services 发现、调用和组合中起到重要作用,恰恰是当前大部分关于 Web Services 的研究所忽略的地方。IRS-II 中的领域任务分解策略和领域任务模型是支持 Web Services 发现、调用和组合的核心,是服务请求理解、服务分解和服务发现及调用整个流程的依据和支撑。同时,IRS-II 用问题求解方法抽象地表示 Web Services,向任务模型屏蔽了具体的实现,从而使服务的组合,更为灵活。IRS-II 中的丰富的本体库资源支撑了这种基于任务的推理。

当然,IRS-II 的领域模型是面向任务的,服务请求的分析过程实际上就是任务分解过程。这种结构的缺点是:对任务变化的适应性不强;领域任务模型限制了它能够满足的服务请求;这种固定的任务分解策略局限了面向 Service 的计算的能力。原因是仅有的任务模型不能很好地描述软件体系结构。IRS-II 的领域特征模型即任务模型的作用非常大,基本上是确定了任务模型,再找到相应的 Web Services,就可以构造一个可执行的应用。这样的任务模型方便并加快了应用的生成执行,却限制了任务模型中还未涉及的新应用。对于新应用,则需要根据新的领域知识而不仅仅是任务知识来对原领域任务模型进行剪裁和扩充。因为,一方面新应用可能不需要复用领域模型内的所有信息,另一方面新应用可能具有一些领域模型所没有覆盖的特殊需求。

结合需求工程领域中对需求语义的研究[11],即需求的语义将落实到所期望的软件环境上,是所期望的软件将对其环境发生的作用。我们认为,Web Services 的语义也应该落实到其作用的环境上,构件的环境由可控的操作资源、不可控的触发、构件依赖的资源组成。这里的构件指由一个 Web Services 或由几个 Web Services 组合而成的能完成特定功能的可复用的软件资产。在建立环境本体的基础上,我们初步提出基于环境本体的构件建模方法。当一个需求到来时(这里的需求用初始状态和终止状态表示),头构件分析需求,得到目标模块,并到构件库中查找合适的构件来进行组合。查找合适的构件不一定是一次查找就能成功的,而是依照从环境本体中分析得出的可能的从初始状态到终止状态的资源变迁。如果找到的某一个实际构件与一条变迁路径中的某一变迁所表示的构件粒度不符合,或找不到能完成变迁所表示构件功能的实际构件,则选择另一条变迁路径,直到可以完成构件的组合,使整个应用得以运行。

结束语 我们认为,对 Web Services 作用环境的建模将成为基于 Service 计算平台的领域模型的另一个重要方面。在环境模型基础上的 Web Services 语义刻画,以环境及其作用相似度为度量的 Web Services 理解和发现,以及以环境合成为约束的Service组合,将成为实现自主自治的WebServices

维普资讯 http://www.cqvip.com

综述:知识系统的V&V技术*)

曾庆田 梁永全 段 华
(山东科技大学信息科学与工程学院 青岛 266510)

摘 要 知识库的异常是影响整个知识系统性能的重要因素之一,因此必须对获取的知识进行校验。本文综述了知识库异常检测和验证的相关研究,给出了异常知识的分类及其危害性,分析了知识库验证困难的原因,介绍了用于知识库验证的静态和动态方法,列举了国际上几个著名的知识库验证工具,并对知识库验证的研究进行了展望。

关键词 知识系统,知识异常,知识表示,知识验证

A Survey on Verification and Validation of Knowledge-based Systems

ZENG Qing-Tian LIANG Yong-Quan DUAN Hua
(College of Information Science and Technology, Shandong University of Science and Technology, Qingdao 266510)

Abstract Knowledge abnormities including inconsistency, redundancy and incompletities are the main reasons to affect the knowledge-based system efficiencies, so the knowledge verification and validation (V&V) are necessary. The related works about knowledge V&V are summarized in this paper. The knowledge abnormities are classified and their harms to knowledge-based system are discussed. However, the knowledge V&V are usually difficult since knowledge representation, update and scale. The static and dynamic methods for knowledge V&V are listed respectively, and some known tools for knowledge V&V are compared. Finally, the future research for knowledge V&V is discussed.

Keywords Knowledge-based system, Knowledge abnormities, Knowledge representation, Knowledge verification, Knowledge validation

1 引言

知识工程在许多领域得到了广泛的应用,但不同开发工具和应用平台使系统的通信、共享、重用性等性能受到了限制,新的知识系统不能有效利用现有的知识系统,不断开始"重新设计",造成人力、物力、才力的大量浪费。20世纪80年代中期,世界上出现了知识共享(Knowledge Sharing)的研究热潮[1~3]。这一研究持续到了20世纪末期,到目前势头有增无减。之所以如此,根本原因就是人们已经越来越认识到两点:第一,计算机能力的进一步提高不能完全依赖于它的运算速度;对复杂的问题,计算机需要大量的与问题背景相关的知识。第二,知识的价值体现在它的可共享性上,计算机所获得的知识如果不能被别的系统或个人所共享,那么它的价值就不能充分发挥出来。与知识共享相关的一个重要问题就是已有知识的获取问题。自从20世纪60年代以来,知识获取成为了计算机研究领域的一个难题。但是,人们的研究重点是开发有效的知识获取方法和技术,能让计算机获取领域专家的经验,用于一些专家系统,使得专家系统具有专家的问题求解能力。

随着专家系统的不断开发,其实践和技术已变得日益完善和成熟。其中的知识获取工作是一项较为困难而又不可缺少的环节,是专家和知识工程师们不能回避的关键问题。知识获取所引起的知识库不一致性是影响整个系统性能的重要因素之一。在知识获取的过程中,不可避免地存在一些错误的知识或矛盾的知识,常见的原因包括:(1)知识源错误:由于我们获取的知识是来源于文本的,所在知识源本身存在难以避免的疏漏和差错。(2)知识获取时无心出错:知识工程师在知识的规范化表示时,由于失误或是其他疏忽容易产生错误。(3)知识描述不准确:在文本中有大量的知识由于自然语言表达的随意性和含糊性而导致语义的多样性,而知识工程师往往根据自己的理解来表示知识,因此造成同一条知识被解释为不同的语义的情况。(4)知识表示错误:知识工程师在进行知识表示时,由于采用了不恰当的表示方法造成错误,或存在对相同知识的前后表示不一致。(5)知识推理的非单调性:由

*)本文工作得到自然科学基金的资助(#60073017和#60273019)和科技部重大基础项目基金(#2001CCA03000和#2002DEA30036)的资助。

发现、调用和组合的一个全新的解决方案。

参 考 文 献

1 李安渝. Web Services技术与实现. 北京:国防工业出版社,2003
2 DAML-S 0.9 Draft Release (2003). DAML Services Coalition. http://www.daml.org/Services/daml-s/0.9/
3 Fensel D, Bussler C. The Web Services Modeling Framework WSMF. http:// informatik. uibk. ac. at/users/c70385/wese/wsmf. bis2002. pdf. 2002
4 Motta E, Domingue J, Cabral L, et al. IRS-II: A Framework and Infrastructure for Semantic Web Services. In: ISWC 2003, LNCS 2870. 306~318
5 Fensel D, Benjamins V R, Decker S, et al. The Component Model of UPML in a Nutshell. WWW Proceedings of the 1st Working IFIP Conference on Software Architectures (WICSA1), San Antonio, Texas, USA, February 1999
6 Clancey W J. Heuristic Classification. Artificial Intelligence, 1985,27:289~350
7 Fensel D, Motta E. Structured Development of Problem Solving Methods. IEEE Transactions on Knowledge and Data Engineering, 2001,13(6): 913~932
8 Motta E. Reusable Components for Knowledge Modelling. Amsterdam, The Netherlands: IOS Press, 1999
9 Sycara K, Lu J, Klusch M. et al. Matchmaking among Heterogeneous Agents on the Internet. In: Proceedings of the 1999 AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford University, USA, 22~24 March 1999
10 http://kmi. open. ac. uk/projects/irs/glossary. html
11 Zave P, Jackson M. Four Dark Corners of Requirements Engineering. ACM. 1996