

基于本体的 Web 服务可靠性模型

王喜凤¹ 王广正¹ 金玲玲²

(安徽工业大学计算机学院 马鞍山 243002)¹ (海南师范大学数学与统计学院 海口 571158)²

摘要 可靠性是 Web 服务选择和组合的一个重要度量标准。针对 Web 服务发现机制中存在效率低下和查准率不高的问题,提出了一种新的 Web 服务可靠性评估方法——OntoRel。该方法基于本体技术,分析了可靠性属性的度量标准及其关系,建立了 Web 服务可靠性本体模型并评估了 Web 服务可靠性。该方法有助于可靠性知识域的管理和开发、Web 服务的可靠性评估和预测、Web 服务的自动选择和组合。

关键词 Web 服务,本体,可靠性

中图分类号 TP311.5 **文献标识码** A

Ontology-based Web Services Reliability Model

WANG Xi-feng¹ WANG Guang-zheng¹ JIN Ling-ling²

(School of Computer Science, Anhui University of Technology, Ma'anshan 243002, China)¹

(School of Mathematics and Statistics, Hainan Normal University, Haikou 571158, China)²

Abstract Reliability plays an important role in the selection and combination of Web services. With the popularization of Web services, researches on minimizing the discovery duration and improving precision ration are getting more important. This paper proposed a new ontology-based approach, called OntoRel, to evaluate Web services reliability. The OntoRel approach analyzes the measurement criteria and relation of reliability attributes, builds the ontology model and measures the reliability of Web services. The approach is characterized by the management and development of reliability knowledge domain, the evaluation and prediction of Web services reliability and the automatic selection and combination of Web services.

Keywords Web services, Ontology, Reliability

1 引言

Web 服务是由 URL 标识的软件应用程序,其接口和绑定能用 XML 来定义、描述和发现,通过基于 Internet 的协议以 XML 消息交换的方式,与其他应用程序进行直接交互^[1]。Web 服务可以在网络中被描述、发布、查找以及调用,从而扩展了应用程序的功能,实现了软件的动态提供。在 Web 服务选择和组合的质量标准中,可靠性是一个非常重要的度量标准,因此 Web 服务的可靠性评估成为一个重要研究方向^[2,3]。因为自然语言存在二义性,现有的基于关键字的 Web 服务发现方法存在效率低下和查准率不高两个问题,基于语义的 Web 服务发现可以解决这些问题^[4,5]。所以,本文提出了一个新的基于本体的 Web 服务可靠性评估方法——OntoRel,来描述和评估 Web 服务的可靠性。

软件可靠性是指在一段特定的自然单元或时间间隔内,在特定的环境下软件无失效运行的概率^[6]。软件可靠性研究成果已很多,如文献[6-9]等。Web 服务可靠性是 Web 服务正常运行的一个度量,一般用失效的百分比标识其值^[3]。目

前 Web 服务的可靠性研究处于起步阶段,研究文献有可靠性建模^[10,11]、可靠性评估和预测^[12-18]、可靠性和服务质量关系^[19]等。一些文献研究了基于语义的 Web 服务发现,如文献[4,5]等,提出了各种服务发现方法。

本文提出了一种新的可靠性评估方法——OntoRel,方法使用本体描述 Web 服务及其可靠性的主要概念和关系,并对 Web 服务的可靠性进行预测。OntoRel 方法有助于 Web 服务自动发现和组合、可靠性知识域的管理和开发、Web 服务的可靠性评估和预测。

本文第 2 节讨论了 Web 服务可靠性相关内容的研究情况;第 3 节给出了 Web 服务可靠性本体及组合服务可靠性计算方法;第 4 节给出了 OntoRel 方法的实例研究。

2 相关性研究

在近几年,很多研究小组已经研究了基于本体的 Web 服务发现问题,主要探讨了 Web 服务及其 QoS 的描述和匹配问题。在这部分,我们列出与本工作相关的方法,并指出各方法的相似点和不同点。

到稿日期:2009-11-24 返修日期:2010-01-05 本文受国家自然科学基金项目(60473142)和安徽省教育厅自然科学基金项目(2006KJ238B)资助。

王喜凤(1980-),女,博士生,讲师,主要研究方向为 Web 服务的可靠性等,E-mail:wxf80106@163.com;王广正(1980-),男,硕士,讲师,主要研究方向为可靠性。

2.1 Web 服务本体

在文献[4,20-22]中,均对 Web 服务的功能部分进行了描述。文献[4]从输入、输出、前置条件和后置条件 4 个方面对 Web 服务功能进行了描述,提供了基于描述逻辑本体的匹配;文献[20,21]从接口、实现、服务域和是否免费 4 个方面对 Web 服务功能进行了描述,并提供了匹配算法;文献[22]从输入、输出、前置条件和效果 4 个方面对 Web 服务功能进行了描述,并提供了匹配算法,但同文献[20,21]的语义匹配度计算方法不同,每项匹配的权重不同。从上述可知,对 Web 服务功能的各种描述项基本相同,但匹配方法不同。

2.2 QoS 本体

QoSOnt 本体^[23,24]为语义 Web 服务的 QoS 定义了一个模型。QoSOnt 本体包括多个本体,这些本体分为 3 个层次:基层、属性层和特定领域层。基层由基 QoS 本体和单位本体构成,基 QoS 本体由一些常用 QoS 概念构成,单位本体以时间为例进行说明;属性层的本体定义了特别 QoS 属性及其度量,有可依赖(dependability)本体和性能(performance)本体;特定领域层对特定类型系统与底层建立关联,如网络系统或 Web 服务系统,特定类型系统具有领域专有 QoS 属性。但 QoSOnt 本体不支持说明 QoS 特性中的 QoS 剖面,不支持 QoS 关系;支持 QoS 度量单位转换,但不支持 QoS 参数间映射。

QoS-MO^[25]是一个顶层本体,包括的概念有:Web 服务的 QoS 特性(QoSCharacteristic)、约束(constraints)和层次(levels),使用 OWL-S 描述 Web 服务。当创建一个 Web 服务的 QoS 描述时,一个本体也被创建,导入 QoS-MO 本体,然后根据需求对概念进行整理并进行个体实例化。QoS-MO 本体包括下列类:QoSCharacteristic 是 QoS-MO 本体的主要类,用来定义服务的可量化特性;QoSCategory 用来对相关 QoS 特性分组;QoSDimension 用来对 QoS 特性度量建模;QoS-Context 可以表达多个 QoS 特性的组合质量;QoSDimension-Mapping 提供 QoSDimension 间映射;QoSValue 表示一个 QoSDimension 的值;QoSConstraints 有 3 类:QoSOffered, QoSRequired 和 QoSConstruct,分别代表什么角色对服务质量的要求;QoSlevels 说明一个服务支持的不同 QoS 模型。在 QoS-MO 本体中,提供者提供了多种服务质量级别,用户可以按照需求进行选择。QoS-MO 本体支持提供者和请求者间相互依赖的 QoS 需求,但不支持 QoS 单位、QoS 值类型等。

Tran^[26,27]首先研究了现有 QoS 本体的优缺点,在此基础上建立了 Web 服务的 QoS 本体 WS-QoSOnto,它支持 QoS 信息的详细描述,支持度量、单位、单位类型、影响方向、QoS 转换和角色,支持不同类型参与者表达不同级别 QoS 需求。WS-QoSOnto 本体由 5 个部分组成:QoS 角色、描述、级别和分组,QoS 特性的主要属性,QoS 关系,QoS 度量和主要 QoS 特性。但 WS-QoSOnto 对主要 QoS 特性的分类不够准确。

文献[28]为地理信息服务的质量建立了一个本体,该本体整合了数据质量和 QoS,由顶层本体(Upper ontology)、数据本体(Data ontology)、QoS 本体(QoS ontology)和应用本体(Application ontology)构成,但仅提出一个框架。

文献[29]提出一种基于本体的 Web 服务质量评估方法,本体由类(Class)、属性(Attributes)、度量(Metrics)和加权概念树(Weighted concept tree)构成,介绍了 Web 服务质量的

计算和原型系统,但整体缺少新意。

2.3 Web 服务可靠性

Cortellessa^[10,15]提出了一个可靠性预测方法,原型系统 RelServ 支持组合服务可靠性计算。Cortellessa 认为服务提供者对组合服务的透明度策略分为 3 种:不透明(No transparency)、半透明(Partial transparency)和完全透明(Total transparency),不同策略决定服务请求者对组合服务内部结构的了解程度,即是否需要迭代算法计算组合服务的可靠性及迭代的深度。但 Cortellessa 提出的预测方法和其他常见方法不同,且没有数据证明其方法的实用性,缺少实际参考价值。

文献[11]提出了一种动态评估 Web 服务可靠性的模型。首先利用群测试和多数投票原则计算原子服务的可靠性,然后利用原子服务的可靠性、运行剖面和服务间结构计算组合服务的可靠性。但该方法需要强大计算能力的 Web 服务开发环境来完成服务的测试和选择功能,集成和运行代价较高。

文献[19]详细列出影响 Web 服务可靠性的相关因素,主要有正确性、容错性、易测性、互操作性、性能和可用性,通过测试得到这些因素的相关值,最后用这些因素的复合函数表示可靠性。该方法分析了 Web 服务的可靠性的影响因素,缺点是函数的不确定性和测试的不完备性。

文献[16]提出了一个基于进程代数进行可靠性预测的方法,该方法同时考虑了操作剖面和体系结构。在操作剖面中,认为输入是一个随机变量,并定义了变量的概率密度函数,从而得到原子服务的可靠性;采用的体系结构引用了 OWL-S 中现有的结构。但输入随机变量满足的分布具有不确定性。

在 ATAM, DUSA 和 AEM 3 种方法的基础上,文献[13]提出了可靠性体系结构评估并研究了一些体系结构机制,但没考虑其他质量,如可维护性、安全性等的影响。

文献[2,30,31]提出了基于可靠性的软件体系结构开发,利用本体对软件体系结构和质量需求建立关联。在这种方式下,需求可以发生变化,但目标是以体系结构的开发来获得高质量的产品,且可靠性知识域中的概念都是在体系结构下建立起来的,缺少广泛性。

3 Web 服务可靠性本体

3.1 可靠性顶层本体

可靠性顶层本体表述了 Web 服务和可靠性的关键概念及其关系,各类参与者都可以表达对 Web 服务的功能和质量(可靠性)的描述,并进行了可靠性知识域的管理和开发,详见图 1。

Web 服务功能本体:有关 Web 服务的功能本体,本文参考了文献[4,20-22]中有关服务功能的描述,结合作者对 Web 服务理解和实际需求,采用接口、实现、是否付费及付费方式 4 个方面来描述 Web 服务功能。

可靠性本体,首先确定可靠性度量的子特性,即除可靠性值外,实际常用哪些标准来衡量可靠性的好坏;其次确定度量过程中涉及到的概念,如角色确定哪类参与者描述可靠性信息,方向确定子特性值是越大越好还是越小越好,ValueType 确定子特性值类型和值,Unit 确定子特性值单位转换;再次确定这些概念间的关系,如 Reliability 类通过 hasDirection 对象属性与 Direction 类相关联,表示 Reliability 度量标准所对

应的方向。

角色(Role):在服务合作中,存在着不同类型的参与者,如服务提供者、请求者(用户、系统中介)和第三方(认证授权机构),分别代表提供质量者、查询质量者和认证质量者。

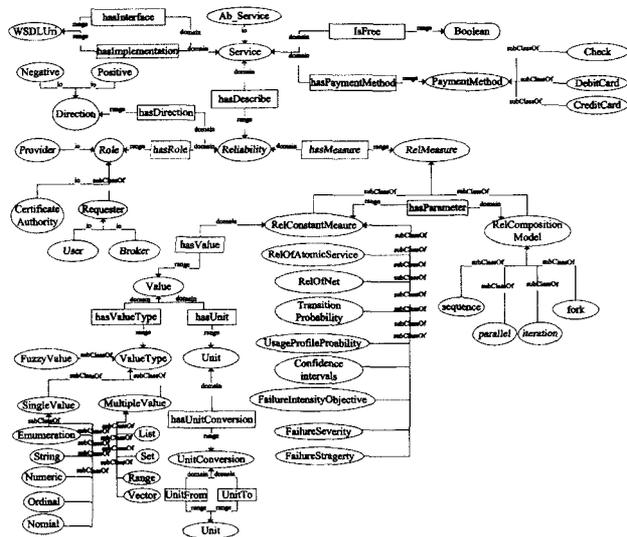


图1 Web服务可靠性本体

方向(Direction):可靠性度量标准的方向。正表示度量标准值越大越好,如 MTBF、避免失效、抵御误操作、可用性、易修复性,其中避免失效、抵御误操作、可用性、易修复性越接近 1 越好;负表示度量标准值越小越好,如平均恢复时间。

可靠性测量(RelMeasure):包括可靠性常量(RelConstantMeasure)和可靠性组合模型(RelCompositionModel)两部分。可靠性常量是指可靠性度量的子特性或同可靠性度量相关的子项;可靠性组合模型根据原子服务可靠性和服务间逻辑关系计算组合服务的可靠性,即可靠性组合模型的计算需要原子服务可靠量子项值作为参数或自变量。可靠性常量部分的子类有原子服务可靠性(RelOfAtomicService)、网络可靠性(RelOfNet)、转移概率(TransitionProbability)、使用剖面概率(UsageProfileProbability)、置信区间(ConfidenceIntervals)、失效强度目标(FailureIntensityObjective)、失效严重程度(FailureSeverity)、失效策略(FailureStrategy)。

原子服务可靠性(RelOfAtomicService):原子服务是指由服务提供者所提供的服务体,不调用其他服务,是不可再分的单元。原子服务可靠性可以通过分析计算历史用户日志而得,或根据失效数据利用某种模型计算而得,或假设已知,或将上述方法互相结合得到。

网络可靠性(RelOfNet):在访问 Web 服务时网络的可靠性度量。

转移概率(TransitionProbability):从某一服务或状态转移到另一服务或状态的可能性。

使用剖面概率(UsageProfileProbability):在使用剖面中操作出现的概率。

置信区间(Confidence intervals):原子服务可靠性的可信范围。

失效强度目标(FailureIntensityObjective):期望 Web 服务在发布到市场之前应该达到的失效强度。

失效策略(FailureStrategy):失效策略分为错误避免(Fault prevention)、错误移除(Fault removal)和容错(Fault

tolerance)3种。

3.2 可靠性子特性

在软件工程产品质量标准 ISO/IEC 9126^[32-35]中,可靠性度量分为成熟性度量、容错性度量、易恢复性度量和可靠性的依从性度量 4 个部分(见图 2),并对每一部分度量子项目进行列表解释,详见文献[32-35]。

根据 Web 服务的特性,本文选择标准 ISO/IEC 9126 中部分项目对可靠性进行度量,分别为:

成熟性度量:平均失效间隔时间(MTBF);

容错性度量:避免失效(Failure avoidance)和抵御误操作(Incorrect operation avoidance);

易恢复性度量:平均恢复时间(Mean Recovery time)、易修复性(Restorability)和可用性(Availability)。

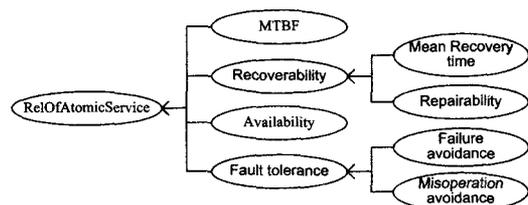


图2 可靠性子特性

3.3 组合服务可靠性度量

根据原子服务可靠性及服务间关系计算组合服务的可靠性。假设 R_i 表示原子服务 W_i 的可靠性, Net_i 表示访问原子服务 W_i 的网络可靠性。对于组合服务 W ,由 n 个原子服务 W_1, W_2, \dots, W_n 组成,对于不同的结构关系,组合服务计算公式如下:

(1)顺序结构: W_1, W_2, \dots, W_n 按顺序结构构成,则组合服务 W 的可靠性为 $R_w = \prod_{i=1}^n Net_i R_i$ 。

(2)分支结构:在分支 W_1, W_2, \dots, W_n 中,设选择 W_i 的概率为 Q_i ,则组合服务 W 的可靠性为 $R_w = \sum_{i=1}^n Net_i Q_i R_i$ 。

(3)并行结构: W_1, W_2, \dots, W_n 按并行结构构成,则组合服务 W 的可靠性为 $R_w = \prod_{i=1}^n Net_i R_i$ 。

(4)迭代结构: W_1 迭代执行, p 为重复的 Web 服务 W_1 执行结束后继续循环的概率,则组合服务 W 的可靠性为 $R_w = (1-p)R_1 Net_1 / (1-pR_1 Net_1)$ 。

4 实例研究

4.1 原子服务

本体建模的结果最终是要机器可读。我们运用本体建模工具 Protégé 及其 OWL-plusin 组件所构建的 Web 服务可靠性本体的部分源代码如图 3 所示。

```

<owl:Class rdf:ID="RelConstantMeasure">
  <rdfs:subClassOf rdf:resource="#RelMeasure"/>
  <owl:disjointWith rdf:resource="#RelCompositionModel"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasValue">
  <rdfs:domain rdf:resource="#RelConstantMeasure"/>
  <rdfs:range rdf:resource="#Value"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="RelOfAtomicService">
  <rdfs:subClassOf rdf:resource="#RelConstantMeasure"/>

```

```

<owl:disjointWith rdf:resource="# RelOfNet"/>
<owl:disjointWith rdf:resource="# TransitionProbability"/>
<owl:disjointWith rdf:resource="# UsageProfileProability"/>
<owl:disjointWith rdf:resource="# Confidenceintervals"/>
...
</owl:Class>
...
<owl:Class rdf:ID="RelMeasure">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="# ConstantQualityMeasure"/>
        <owl:Class rdf:about="# FunctionQualityMeasure"/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
...

```

图3 原子服务可靠性本体

4.2 组合服务

根据原子服务可靠性的描述,计算组合服务可靠性。以银行卡取款为例:用户输入银行卡号和密码,得知卡内余额,设卡内余额为 y 元;然后从卡内取钱,每次从卡内取 2000 元,直到卡内余额不足 2000 元。上述组合服务涉及两个原子服务:查询余额(Check)和取钱(Withdraw)。此组合服务可表示为 Sequence {Check; while($y > 2000$) {Withdraw}}; 则银行卡取款服务的可靠性为 $R = Net_{check} R_{check} (Net_{withdraw_i} R_{withdraw_i})^{quotient(y, 2000)}$ 。

4.3 可靠性匹配算法

基于本体的可靠性匹配算法,有助于 Web 服务自动发现和组合。假设 Web 服务的可靠性匹配结果分为两种:成功(1)或失败(0)。现以原子服务的可靠性为例,则请求服务 WS_r 和原子服务 WS_a 的匹配算法如下:

```

int RelOfAtomicServiceMatch(WS_r, WS_a)
{
  int relOfAtomicMatchDegree=0;
  if ((WS_r.MTBF<=WS_a.MTBF)&&(WS_r.Repairability<=
    WS_a.Repairability)&&(WS_r.Availability<=WS_a.Avail-
    ability)//值越大越好
    &&(WS_r.Failureavoidance<=WS_a.Failureavoidance)&&
    (WS_r.Misoperationavoidance<=WS_a.Misoperationavoid-
    ance)//值越大越好
    &&(WS_r.FailureIntensityObjective<=WS_a.FailureInten-
    sityObjective)&&(WS_r.RelOfNet<=WS_a.RelOfNet)//
    值越大越好
    &&(WS_r.MeanRecoverytime>=WS_a.MeanRecovery-
    time)&&(WS_r.Confidenceintervals>=WS_a.Confiden-
    ceintervals)//值越小越好
    &&(WS_r.FailureStragerty==WS_a.FailureStragerty)),
  {
    relOfAtomicMatchDegree=1
  }
  return relOfAtomicMatchDegree;
}

```

在可靠性度量量子项中,子项值越大越好的有 MTBF, Re-

pairability, Availability, Failureavoidance, Misoperationavoid-
ance, FailureIntensityObjective, RelOfNet; 越小越好的有
MeanRecoverytime, Confidenceintervals, 而 FailureStragerty希
望相同。

结束语 本文针对 Web 服务选择标准及发现和组合中
存在的问题,提出了一个基于本体的 Web 服务可靠性模型,
该模型旨在:

研究可靠性度量标准及其关系,对可靠性知识域进行管
理和开发,以便于各类参与者交流;

建立组合服务可靠性计算方法,对 Web 服务可靠性进行
评估和预测,为 Web 服务选择提供依据;

基于本体的 Web 服务和可靠性描述,有助于 Web 服务
自动的发现和组合。

参考文献

- [1] Web Services Architecture; W 3 C Working Group Note [EB/OL]. <http://www.w3.org>, 2004-02-11
- [2] Zhou Jiehan, Niemel E, Evesti A, et al. Pekka Savolainen: OntoArch Approach for Reliability-aware Software Architecture Development[C]//COMPSAC. 2008; 1228-1233
- [3] Zo Hangjung, Nazareth D L, Jain H K. Measuring Reliability of Applications Composed of Web Services[C]//HICSS. 2007; 278
- [4] Lutz M. Ontology-based descriptions for semantic discovery and composition of geoprocessing services[J]. Geoinformatica, 2007, 11(1): 1-36
- [5] 孙萍, 蒋昌俊. 利用服务聚类优化面向过程模型的语义 Web 服务发现[J]. 计算机学报, 2008, 31(08): 1340-1353
- [6] Musa J D. Software reliability engineering[M]. China Machine Press, 2003
- [7] Lyu M R. Software Reliability Engineering: A Roadmap[C]//FOSE. 2007; 153-170
- [8] Immonen A, Niemel E. Survey of reliability and availability prediction methods from the viewpoint of software architecture[J]. Software and System Modeling, 2008, 7(1): 49-65
- [9] Gokhale S S. Architecture-based Software Reliability Analysis: Overview and Limitations [J]. IEEE Trans. Dependable Sec. Comput, 2007, 4(1): 32-40
- [10] Cortellessa V, Grassi V. Reliability Modeling and Analysis of Service-oriented Architectures[C]//Test and Analysis of Web Services. 2007; 339-362
- [11] Tsai Wei-Tek, Zhang Dawei, Chen Yinong, et al. A software reliability model for web services[C]//IASTED Conf. on Software Engineering and Applications. 2004; 144-149
- [12] Griman A, Valdoser L, Mendoza L, et al. Issues for Evaluating Reliability in Software Architectures[C]//AMCIS 2005 Proceedings. 2005; 189
- [13] Grimán A, Pérez M A, Mendoza L E, et al. A method proposal for architectural reliability evaluation[C]//ICEIS. 2007; 564-568
- [14] Tsai Wei-Tek, Huang Qian, Xu Jingjing, et al. Ontology-based Dynamic Process Collaboration in Service-Oriented Architecture [C]//SOCA. 2007; 39-46
- [15] Grassi V, Patella S. Reliability Prediction for Service-Oriented Computing Environments[J]. IEEE Internet Computing, 2006, 10(3): 43-49
- [16] Huang Ning, Wang Dong, Jia Xiaoguang. An Algebra-based Reliability Prediction Approach for Composite Web Services[C]//ISSRE. 2008; 285-286

(下转第 160 页)

```

(connector03.connector_type.interface_element)
MetaConnector03?.metaconstructure.Changeable
=connector03.metaconstructure.Changeable
MetaConnector03?.connectorbehavior.Changeable
=connector03.connectorbehavior.Changeable
MetaConnector03?.metaconbehavior.SpecificationMetaInfo
=MetaConnector03?.metaconconnectorbehavior.setSpecification
(connector03.connector_type.operation)
MetaConnector03?.metaconconstraint
=MetaConnector03?.setConstraint(connector03.connector_type.invariant)
MetaConnector03?.metaconproperty
=MetaConnector03?.setProperty(connector03.connector_type.state)

```

所以,定义增加组件操作的完整性描述模式 T_AddComponent 为:

$$T_addComponent \triangleq (addComponent_One \wedge Success) \vee (addComponent_Two \wedge Success)$$

4 相关工作

对于在软件设计阶段软件体系结构的重用,国内外研究机构提出了很多不同的方法,目前具有代表意义的研究成果和本文提出的基于反射式软件体系结构软件重用的对比研究有:面向领域的体系结构重用需要针对特定领域,而本文的软件体系结构重用方法具有通用性;体系结构设计知识的重用仍然面临着许多重大的技术障碍;软件框架可以将体系结构的设计方案连同其实现代码一起进行重用是支持实现阶段而非设计阶段的重用,本文基于反射式软件体系结构的软件重用是针对软件设计阶段。

本文提出的基于反射式软件体系结构软件重用是一种更通用、更便捷的体系结构制品本身的重用方法,它具有统一的体系结构建模方法的信息,分别利用反射机制和 PMB 协议来屏蔽和完成设计阶段软件体系结构制品的重用,所以本文以及后续的研究内容在一定程度上可以部分解决软件体系结构重用存在的主要问题。

结束语 本文的主要贡献在于:给出了支持软件体系结

构设计时重用的反射式软件体系结构,描述了反射式软件体系结构的元级软件体系结构和基级软件体系结构之间进行交互和互操作的协议 PMB,基于软件规格语言 Object-Z 对 PMB 协议进行了形式化描述。

我们的研究工作与已有研究不同的地方在于:(1)提供了一种在设计阶段支持软件体系结构重用的反射式软件体系结构。(2)对反射式体系结构的 PMB 协议做了部分定义。

作为今后的工作,我们将对反射式软件体系结构的元信息模型不断丰富和完善;完善 PMB 协议的定义;给出反射式软件体系结构的元级体系结构和基本级软件体系结构的一致性性质的定义;给出经过重用操作后,反射式软件体系结构的元级和基本级软件体系结构的一致性性质的证明方法和过程。

参考文献

- [1] Binns P, Engelhart, Vestal M. Domain-Specific Software Architectures for Guidance, Navigation, and Control [J]. *Software Eng. and Knowledge Eng.*, 1996, 6(2):1011-1017
- [2] Shaw M. Some Patterns for Software Architecture, *Pattern Languages of Program Design* [M] // Vlissides, Coplien, Kerth, et al., eds. 1996:255-270
- [3] Schmerl B, Garlan D. AcmeStudio: Supporting Style-Centered Architecture Development [C] // *Proceedings of International Conference on Software Engineering*, Edinburgh, Scotland, May 2004
- [4] Froehlich G, Hoover H J, Liu Ling, et al. Designing object-oriented frameworks, In *CRC Handbook of Object Technology* [M]. CRC Press, 1998
- [5] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [6] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [7] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [8] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [9] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [10] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [11] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [12] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [13] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [14] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [15] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [16] 罗巨波, 应时. 一种支持软件体系结构重用的反射机制及其形式化[J]. *计算机科学*, 2009, 36(8):145-148
- [17] Coppolino L, Romano L, Mazzocca N, et al. Web Services workflow Reliability Estimation Through Reliability Patterns [C] // *SecureComm*. 2007
- [18] Abramowicz W, Kaczmarek M, Zyskowski D. Duality in Web Services Reliability [C] // *AICT/ICIW*. 2006:165
- [19] Zhang Jia, Zhang Liang-Jie. Criteria Analysis and Validation of the Reliability of Web Services-oriented Systems [C] // *ICWS*. 2005:621-628
- [20] Maximilien E M, Singh M P. Toward autonomic web services trust and selection [C] // *ICSOC*. 2004:212-221
- [21] Maximilien E M, Singh M P. A Framework and Ontology for Dynamic Web Services Selection [J]. *IEEE Internet Computing*, 2004, 8(5):84-93
- [22] Liu Chuanchang, Peng Yong, Chen Junliang. Web Services Description Ontology-Based Service Discovery Model [C] // *Web Intelligence*. 2006:633-636
- [23] Dobson G, Lock R, Sommerville I. QoSOnt: a QoS Ontology for Service-Centric Systems [C] // *EUROMICRO-SEAA*. 2005:80-87
- [24] Dobson G. OWL and OWL-S for Dependability-Explicit Service-Centric Computing [C] // *Service-Oriented Computing: Consequences for Engineering Requirements, SOCCER*. 2006:4
- [25] Tondello G F, Siqueira F. The QoS-MO ontology for semantic QoS modeling [C] // *SAC*. 2008:2336-2340
- [26] Vuong Xuan Tran. WS-QoSOnto: A QoS Ontology for Web Services [C] // *SOSE*. 2008:233-238
- [27] Vuong Xuan Tran, Hidekazu Tsuji, Masuda R. A new QoS ontology and its QoS-based ranking algorithm for Web services [J]. *Simulation Modelling Practice and Theory*, 2009, 17(8):1378-1398
- [28] Onchaga R, Widya I, Morales J, et al. An ontology framework for quality of geographical information services [C] // *GIS*. 2008:64
- [29] Liu Gaoyong, Wang Huiling. A Method for Evaluating the Service Quality of E-commerce Websites Based on Ontology [C] // *ICIII '08*. 2008:293-297
- [30] Zhou Jiehan, Niemel E. Ontology-based software reliability modeling [C] // *Software and Services Variability Management-Concepts, Models and Tools*. 2007
- [31] Evesti A. Quality oriented software architecture development [C] // *Espoo. VTT Publications* 636, 2007:79
- [32] ISO/IEC. ISO/IEC 9126-1 International Standard; Software engineering. Product quality [S]. Part 1: Quality model, 2001:25
- [33] ISO/IEC. ISO/IEC 9126-2 Technical Report; Software engineering. Product quality [S]. Part 2: External metrics, 2003:86
- [34] ISO/IEC. ISO/IEC 9126-3 Technical Report; Software engineering. Product quality [S]. Part 3: Internal metrics, 2003:62
- [35] ISO/IEC (2004) ISO/IEC 9126-4 Technical Report; Software engineering. Product quality [S]. Part 4: Quality in use metric, 2004:59