

一种支持构件服务质量的构件管理框架

赵俊峰, 王亚沙, 谢 冰, 杨芙清

(北京大学信息科学技术学院软件研究所, 北京 100871)

摘 要: 软件构件库是有效地管理软件构件的基础设施. 分布式构件技术的发展, 使得构件库除了管理传统下载使用的构件外, 还必须能够管理日益增多的在线使用构件, 并为在线构件服务质量提供相应的质量保证与支持. 针对这一问题, 本文首先引入了构件服务质量模型及相应的质量属性量化方法, 同时给出相应的规约描述, 然后提出了一种基于上述模型和规约的支持构件服务质量的构件管理方法, 并描述了该方法在青鸟构件库管理系统(JBCLMS)中的实现.

关键词: QoS; 构件库; 主观反馈; 客观反馈

中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2004) 12A-165-04

A Management Framework of Component Supporting QoS of Component

ZHAO Jun-feng, WANG Ya-sha, XIE Bing, Yang Fu-qing

(Institute of Software, School of Electronics Engineering & Computer Science, Peking University, Beijing 100871, China)

Abstract: Software component library is the most effective way to manage the software component. With the development of the distributed component, the component library manages not only the traditional off-line components, but also on-line components. In addition to this, the component library must support QoS (Quality of Service) of component. In order to tackle the problem, first a component quality model, the ways of quality attributes measurement and corresponding specification are introduced in this paper. Then a method to support the QoS of component based on the model and specification mentioned above are put forward. At last, the implementation of the method in Jade Bird component library management system (JBCLMS) is described.

Key words: quality of service(QoS); component library; subjective feedback; objective feedback

1 引言

软件复用是提高软件生产效率和软件质量的切实可行的解决方案之一, 软件构件库是软件复用的核心机制之一. 随着构件化软件生产的不断发展和以 CORBA、DCOM 和 EJB 为代表的分布式构件技术的发展, 尤其是近几年来 Web Services 技术的兴起, 涌现出了大量分布式构件, 使得构件库中存储的可复用构件数量不断增加, 构件库中的构件使用从原先的下载使用扩展到了在线使用. 构件库除了管理传统下载使用的构件外, 还必须能够管理日益增多的在线使用构件, 并为在线构件服务质量提供支持和保证. 这为构件管理带来了新的问题和挑战. 构件服务质量(Quality of Service, QoS)已成为影响构件可用性的重要因素之一. 针对上述问题, 本文给出了一种描述构件服务质量的质量模型, 并给出相应的质量属性的量化方法及描述规约. 同时给出了管理构件服务质量的方法, 即利用主观反馈和客观反馈机制相结合的方法, 在构件库中为构件服务质量提供支持. 该方法可以有效地管理不同使用方式的构件, 辅助用户高效地、准确地检索到满足功能及性能需求的

构件.

本文第二节介绍了构件服务质量模型, 并给出构件服务质量属性及其量化方法, 同时给出相应的规约描述; 第三节给出了支持构件服务质量的构件管理框架; 第四节论述了在青鸟构件库管理系统中, 如何实现支持构件服务质量的构件管理框架; 最后在第五节对本文工作进行了总结和展望.

2 构件服务质量模型及其规约

2.1 构件服务质量模型

为了对构件进行更好的管理, 我们将构件分为两类, 一类称之为离线构件, 指的是需要获得构件实体才能进行软件开发的构件, 另一类是在线构件, 即那些无需获得构件的实体, 直接通过网络(如 Internet、Intranet 等)与构件实体进行通讯和互操作的构件. 不论是在线构件还是离线构件, 都需要构件对所提供的服务质量进行保证, 这样才能有效地进行基于构件的软件开发.

构件服务质量是指构件的非功能性属性(如性能、可靠性、可用性和安全性等), 用以定义用户使用服务的满意程

度^[1-3]. 它对于构件的正常运行和有效复用是非常重要的因素, 特别地, 对于在线构件而言, 由于构件运行体和复用构件的应用程序处于分布的环境中, 复用在线构件的应用程序对在线构件服务质量就更为关注, 甚至是复用时在功能需求之外要考虑的另一个主要需求.

为了更好地刻画构件服务质量, 我们采用一种多层次的且易于扩充的构件服务质量模型(CQSM, Component QoS Model)对构件服务质量进行描述. 如图1所示, CQSM是由一组描述构件服务质量的构件服务质量维构成, 维是可嵌套的, 即可以由下层维对上层维进行更详细的描述. 而一个服务质量维最终是由一组服务质量属性构成, 服务质量属性是刻画构件服务质量的原子属性, 该属性不可再分, 即服务质量属性是叶结点, 而质量维是分支结点. 对于模型中的每个结点, 我们采用三元组 $\langle I, W, M^{(i)} \rangle$ 对其进行描述, 其中 I 为该结点的量化值; W 为在该模型下该结点的权重值; $M^{(i)}$ 代表计算公式, 当为 M' 代表该结点对其下层所有子结点信息进行计算汇总的计算公式(如加权平均法), 当为 M 时代表对于本结点的计算公式.

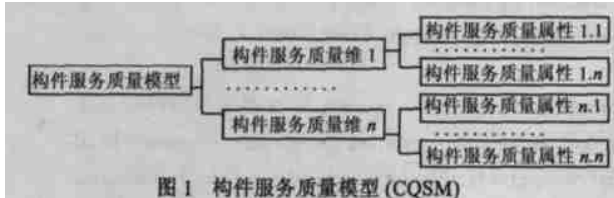


图1 构件服务质量模型(CQSM)

离线构件服务质量信息需通过用户下载使用完该构件后, 才能获知其是否满足用户的期望, 并由用户给出评判信息. 我们可以采用4.1节所述的方法对离线构件服务质量信息进行获取, 这一方法同样适用于在线构件. 但是在线构件主要是通过网络实时使用的, 因此需要对其质量属性信息进行更客观地收集, 以确定其服务质量信息.

由于对在线构件的管理及使用目前均处于研究探索阶段, 因此对于在线构件服务质量属性的选择及定义我们参照了目前国际上的一些研究工作^[1-5], 并结合了实际的需求. 它们是: (1) 可靠性; (2) 可用性; (3) 执行时间; (4) 吞吐量. 为了更好地对这些属性进行管理, 还需对这些属性进行量化, 才能对服务质量进行评判. 关于质量属性的量化方法目前国际上也有一些相关研究, 我们参照了惠普实验室等提出的量化模型^[5-8], 结合实际的实验, 制定了对在线构件质量属性量化的方法. 下面我们给出这些属性的定义及量化方法:

■ 可靠性——表示维持构件服务质量的程度. 每天、每月或每年的失效次数是衡量构件所提供服务的可靠性的尺度. 我们采用式(1)进行计算:

$$Q_{reliability}(S) = 1 - F_I \setminus S_I \quad (1)$$

其中, 在时间间隔 I 内, F_I 代表未完成的服务请求, S_I 代表到达的总服务请求, $F_I \setminus S_I$ 为构件 S 提供服务的平均失败率.

■ 可用性——指的是构件是否已就绪可供立即使用. 可用性表示服务可用的可能性. 我们采用修复时间TTR(Time-to-Repair)对该属性进行量化. 可用性无法直接度量, 必须通过计算平均故障间隔时间(MTBF)和修复并恢复全部系统功能的

平均时间(MTTR)指标得出.

我们采用式(2)进行计算:

$$Q_{availability}(S) = \left[1 - \frac{MTTR}{MTTR + MTBF} \right] \times 100\% \quad (2)$$

其中MTBF与MTTR计算如下:

$$MTBF = \sum \left(T_{f(n)} - T_{r(n-1)} \right) \setminus n \quad (3)$$

$$MTTR = \sum \left(T_{r(n)} - T_{f(n)} \right) \setminus n \quad (4)$$

$T_{r(n-1)}$ 代表构件 S 上一次失效后恢复正常的时刻, $T_{f(n)}$ 代表构件 S 本次失效时刻, $T_{r(n)}$ 代表构件 S 本次恢复正常运行时刻, n 代表时间间隔 I 内构件 S 发生故障的次数.

■ 执行时间——指的是构件完成所请求的任务所花费的平均时间. 我们采用式(5)进行计算:

$$Q_{execTime}(S) = \frac{\sum \left(T_{q(n)} - T_{a(n)} \right)}{n} \quad (5)$$

其中, $T_{q(n)}$ 代表请求完成时刻, $T_{a(n)}$ 代表请求到达时刻, n 代表时间间隔 I 内对于构件 S 的请求次数, 在此执行时间的单位以秒为单位.

■ 吞吐量——代表了在给定的时间区间内, 构件所能满足的最大请求数量, 即时间间隔 I 内构件 S 的吞吐量为构件 S 在该时间间隔所能处理的服务请求的最大值, 我们用公式(6)表示:

$$Q_{throughput}(S) = \text{MAX}(S_{succ}) \quad (6)$$

其中, S_{succ} 代表时间间隔 I 成功处理的服务请求数.

由此, 可采用一个矢量来表示构件 S 的服务质量, 如式(7)所示:

$$Q(S) = (Q_{reliability}(S), Q_{availability}(S), Q_{execTime}(S), Q_{throughput}(S)) \quad (7)$$

采用CQSM模型描述可将上述所讨论的在线构件服务质量模型描述为如下形式, 设有构件 S 则其服务质量如下:

$$\langle \langle \text{ReliabilityValue}, 0.3, Q_{reliability}(S) \rangle, \langle \text{AvailabilityValue}, 0.3, Q_{availability}(S) \rangle, \langle \text{ExecTimeValue}, 0.2, Q_{execTime}(S) \rangle, \langle \text{ThroughputValue}, 0.2, Q_{throughput}(S) \rangle \rangle$$

第一个元素代表计算出来的值, 第二个元素代表该属性的权重, 第三个属性为上述所示的对应于该属性的量化公式.

2.2 构件服务质量的描述

不同构件模型的定义和描述方法核心机制是一致的, 即将构件分为接口和实现两部分, 其中接口用来表示构件与外界交互信息, 而实现用来表示可被直接复用的构件实体. 由于构件接口信息是稳定信息, 不宜加入像构件服务质量信息这样会随着时间推移而变化的动态信息, 我们在构件接口规约仅加入对构件质量属性的引用标记, 由该引用标记引向服务质量信息, 这样构件服务质量的信息独立于构件接口信息, 而不影响构件使用. 我们引入如下来描述:

$$\begin{aligned} \langle \text{ComponentQoSProperties} \rangle &::= \langle \text{ComponentQoS ID} \rangle \langle \text{Component ID} \rangle \\ &\langle \text{QoSModel} \rangle \\ \langle \text{QoSModel} \rangle &::= \langle \text{QoSModel ID} \rangle \langle \text{QoSModelName} \rangle \langle \text{QoSModelValue} \rangle \\ &\langle \text{CompQoSPropertiesSet} \rangle \end{aligned}$$

```

<CompQoSProperties Set> ::= <QoSProperties> <QoSValue> {, <QoSProp-
erties> <QoSValue>}
<QoSProperties> ::= <QoSPropertiesName> <QoSWeight> <QoSValue>
<QoSModelValue> ::= QoSModelValue[ <ValueUnit> ]
<QoSValue> ::= QoSValue[ <ValueUnit> ]
<ValueUnit> ::= TTFI TTRI SECOND| YUAN| .....

```

首先, <Component QoS ID> 与 <Component ID> 一起唯一标识构件反馈信息。考虑到良好的扩展性, 在描述一个构件服务质量时可采用不同的服务质量模型(<QoSModel>) 进行描述, 用 <QoSModelID> 和 <QoSModelName> 来标识描述构件服务质量的模型, 用 <QoSModelValue> 来描述构件在某一服务质量模型下的模型值。我们采用 <CompQoSPropertiesSet> 对构成模型的质量维或属性进行描述, 对于 <QoSProperties> 我们用属性或维名称、权重值和属性或维量化值(<QoSValue>) 进行描述。

3 构件服务质量管理框架

采用主观反馈和客观反馈机制相结合的管理方法对构件进行管理。通过收集用户使用构件的信息和构件运行状态信息, 对构件服务质量进行综合评判, 并将该信息作为描述构件的必须信息之一。这样用户在检索构件时就能获取构件服务质量信息, 根据该信息用户可更好选择符合其所需的构件。另外, 反馈信息还有助于对构件、构件的分类模式及构件库自身等方面进行改进, 并作为对构件库中的构件进行评价的依据之一。

我们将反馈信息分为主观信息和客观信息, 主观信息是用户提供的有关使用构件的反馈信息, 主观信息评价的重点在于用户使用完构件后的主观判断信息, 客观信息是通过一定机制收集的在线构件服务时的服务质量信息, 重点在于在线构件的运行状态信息。

主观反馈虽然可以反映构件的质量信息, 但是主观反馈所收集的信息依赖于用户主观判断, 并且对于所收集的反馈信息是否能公正、准确及正确地反映构件的服务质量尚有不稳定性。另外, 随着在线构件的数目越来越多, 在线构件的运行情况成为用户选择构件的必要条件, 尤其是随着越来越多的构件供应商出现, 会有大量功能相同而性能不同的在线构件出现, 如何在其中选择具有最优性价比的构件成为亟需解决的问题之一。由于客观反馈可以向用户提供切实可靠的构件运行状态信息, 它是对主观反馈的必要补充。另一方面, 客观反馈虽然可以较客观地反映构件的运行情况, 但其所能采集的服务质量信息数据有限, 如构件的易用性这种信息就无法通过客观反馈系统采集。因此我们同时引入主观反馈与客观反馈, 让这两种机制相辅相成, 以更好地评价构件的质量, 从而为用户有效地选择构件提供条件。

我们选用青鸟构件库管理系统作为我们的支撑平台, 分别实现了主观反馈及客观反馈子系统, 下面我们就介绍一下构件服务质量管理框架的实现。

4 构件服务质量管理框架的实现

4.1 主观反馈管理框架的实现

主观反馈子系统收集的是构件的主观反馈信息。由于不同领域对于构件服务质量属性的关注点不同, 因此不同领域

的用户针对同一构件会有不同的反馈信息, 如金融领域的用户会更关注于安全方面。因而在构件库系统中采用固定的构件服务质量模型则不能很好地体现不同用户的关注点, 因此需提供可以定制构件服务质量模型的功能。另外, 构件库需采用定性的方式和定量的方式保存用户反馈信息。为此, 采用 2.1 节中的 CQSM 构件服务质量模型, 可以容易地将权重信息加入到实际使用的服务质量模型中, 并对每一质量维及质量属性打分, 这样就可以得到量化的构件服务质量信息。

在实际的实现中, 我们采用“基础模型-模型-实例”的量化机制对构件的主观反馈信息进行管理。其中基础模型和模型对应于一个采用 CQSM 定义的服务质量模型, 基础模型用来定义构件服务质量模型的构成框架, 但不包含权重信息, 而模型是具有权重值的构件服务质量模型。这是由于根据不同领域的要求, 不同质量维和质量属性的重要程度是不一样的。如表 1 斜线部分所示的即为按照某种领域特性制定的基础模型, 表 1 属性权重列和维权重列表示, 用户可以根据自己所在领域对构件质量的不同要求, 确定哪些属性是关键属性, 哪些属性是次关键属性, 分别给予不同权重值, 这样就能灵活地适应不同用户不同领域的需求, 更好地反映构件服务质量。

用户在进行构件质量信息反馈时, 可以根据所在领域特性选择一个特定的模型发布自己的量化反馈信息, 发布量化反馈信息的形式就是对质量属性进行打分(如可规定分值的范围为 0—100)的过程。如表 1 所示, 用户根据复用构件的情况对构件进行评价打分, 然后将这些信息进行相应的处理(用户可以自定义处理方式), 如表 1 示例采用的是加权平均值方法, 这样可以得到该模型下用户的总评价信息, 并将这些信息作为构件信息发布到青鸟构件库中, 当用户在检索构件时, 这些质量信息可作为用户准确选择构件的依据。

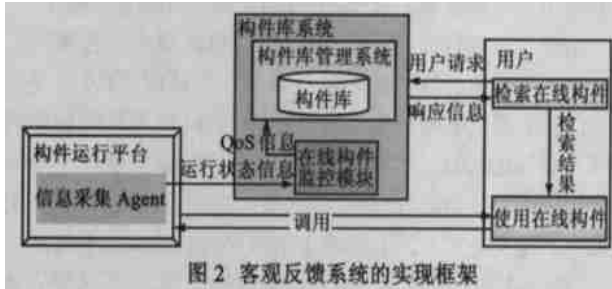
表 1 构件主观反馈质量模型示例

	维	属性			权重 (维)	分值 (0~100)	总分
		属性名	权重(属性)	分值(0~100)			
构件 S A	功能性	准确性	0.2	75	0.4	25.8	66.95
		适合性	0.3	70			
	互操作性	移植性	0.2	60			
		兼容性	0.3	55			
M P L	可靠性	成熟性	0.4	70	0.3	19.65	
		容错性	0.3	65			
		易恢复性	0.3	60			
E	易用性	易理解性	0.5	70	0.2	14.5	
		易操作性	0.5	75			
	可维护性	适应性	0.4	70	0.1	7	
		文档完备性	0.6	70			

为了帮助用户准确地找到其所需的构件, 我们还需建立一套构件服务质量评价体系, 即对采用同一模型进行评估的构件划分类别, 如采用表 1 示例模型的构件规定其服务质量满分为 100 分, 那么可分为四个等级—优秀、良好、中等、差, 其对应分值分别为 90~100、80~90、60~80、0~60。用户可在其检索条件中加入对构件服务质量的约束条件, 从而缩小了用户检索范围, 提高了检索构件的查准率, 从而帮助用户更好地选择其所需的构件。

4.2 客观反馈管理框架的实现

客观反馈系统主要收集并分析在线构件运行状态信息,从而客观评价在线构件服务质量.目前在青鸟构件库中,客观反馈系统主要收集的是有关构件可靠性、可用性、执行时间和吞吐量这四个质量属性信息,这几个质量属性的量化方法见2.1节.图2所示的是客观反馈系统的实现框架,为了得到在线构件服务质量信息,我们在构件库系统中加入了一个监控模块,使其实时收集和分析构件运行状况,并将分析结果提交给构件库管理系统,作为构件的描述信息之一.信息采集Agent运行在构件运行平台之上,它的主要目的是采集构件的运行信息,如为了计算可靠性需采集构件未完成的服务请求数,并将这些运行信息送到构件库中的监控模块,供监控模块分析构件的运行情况.用户可以通过自己的要求定制相应的检索条件,如加入构件执行时间不大于30ms这样的条件,当有符合用户需求的构件存在时,构件库将结果返回给用户,用户可以根据检索结果中的信息直接访问并使用在线构件,这样构件库系统可以更好地为用户的功能和性能要求服务,提高了构件的可用性.



5 结束语

本文论述了一种支持构件服务质量的构件管理方法,即将主观反馈与客观反馈的机制有机地结合起来管理离线构件与在线构件.本文给出了刻画构件服务质量的CQSM模型及对质量属性的量化方法,并给出了相应的规约描述.接着论述了如何利用反馈机制进行构件服务质量的保证,并给出了基于反馈的构件服务质量的管理体系,并结合青鸟构件库管理系统实现了该框架.由于在线构件管理尚处于发展阶段,因此如何选择合适的质量属性对在线构件进行更精确的描述,如何更准确地度量在线构件的质量属性值,如何利用构件的历史信息给用户使用构件提供指导等问题均是未来研究方向.随着上述问题的解决,用户会更加方便地使用各类构件,促进基于构件的软件开发的技术发展.

参考文献:

[1] Anbazhagan Mani, Arun Nagarajan. Understanding quality of service for

Web Services [Z]. http://www-900.ibm.com/developerWorks/cn/welsservices/ws-quality/index_eng.shtml, 2002-01.

- [2] Rajesh Sumra, Anilazi D. Quality of Service for Web Services—Demystification, Limitations, and Best Practices[Z]. <http://www.developer.com/java/web/article.php/2248251>, 2002-05.
- [3] ITU-T Recommendation 1.350. General Aspects of Quality of Service and Network Performance in Digital Network, including ISDNs[S]. ITU-T, 1993-03.
- [4] Aoyama M, Weerawarana S, Maruyama H, Szyperki C, Sullivan K, Lea D. Web services engineering: Promises and challenges[A]. Proceedings of 24th International Conference on Software Engineering[C]. Orlando, Florida, USA, IEEE Computer Society, 2002-05. 647-648.
- [5] Zeng L Z, Benatallah B, Dumas M. Quality driven web services composition[A]. WWW2003[C]. Budapest, Hungary, 2003-05. 411-421.
- [6] Xavier Franch. Systematic formulation of non-functional characteristics of software[A]. In Proceedings of 3rd International Conference on Requirements Engineering (ICRE) [C]. USA: Colorado springs, IEEE Computer Society, 1998.
- [7] Akhil Sahai, Jinsong Ouyang, Vijay Machiraju, Klaus Wurster. Specifying and Guaranteeing Quality of Service for Web Services through Real Time Measurement and Adaptive Control[R/OL]. HP Labs, <http://www.hpl.hp.com/techreport/2001/HPL-2001-134.html>.
- [8] Svend Frolund, Jari Koistinen. Quality-of-Service Specification in Distributed Object Systems[Z]. HP Technical Report, HPL-98-159. 1998-09.
- [9] 潘颖, 赵俊峰, 谢冰. 构件库技术的研究与发展[J]. 计算机科学, 2003, 30(5): 90-93.
- [10] 杨燕燕. 基于数据仓库技术的构件库系统研究[D]. 北京: 北京大学计算机科学技术系, 1999.

作者简介:



赵俊峰 女, 1974年生于福建泉州, 博士生, 讲师, 主要研究领域为软件复用技术, 构件技术, 大型分布式构件库系统等. E-mail: zhaof@sei.pku.edu.cn.



王亚沙 男, 1975年生于湖南湘潭, 博士生, 主要研究领域为软件复用与构件技术, 软件复用过程, 构件库技术等. E-mail: wangys@sei.pku.edu.cn.