

预见式连接件本体模型研究

王 华

(浙江科技学院 信息与电子工程学院,杭州 310023)

摘 要: 由软件生存环境变化引起的软件体系结构调整,需要通过连接件的重新配置来实现,迫切需要预见式的自适应连接件对系统进行运行时演化。针对当前被动式连接件的不足,于是借助面向方面的思想构造预见式连接件本体模型,以实现构件连接的修改和体系结构的动态配置,提高软件体系结构自适应性,并借助影响因子价值去量化选择调整政策的依据。本方法应用于一个实际运营的在线健康评估系统之后,提高了系统的服务质量。

关键词: 预见式自适应;连接件;面向方面;本体模型

中图分类号: TP311.5 **文献标志码:** A **文章编号:** 1671-8798(2018)01-0032-05

Research on proactive connector ontology model

WANG Hua

(School of Information and Electronic Engineering, Zhejiang University
of Science and Technology, Hangzhou 310023, Zhejiang, China)

Abstract: Adaptation of software architecture triggered by changes of software living environment is realized by reconfiguration of the connector, which is urgently required to be proactively self-adaptive to activate the system evolution at runtime. Considering the shortcoming of the existing reactive connector, the proactive connector ontology model is established on the basis of the aspect-oriented concept. The model could modify component connection and configure software architecture dynamically and thus enhance self-adaptive capability of software architecture. The impact factor value is employed to quantifying the choice basis of adaptive policies. After being applied to an operating online health evaluation system, the proposed method has improved the service quality of the target system.

Keywords: proactive self-adaptation; connector; aspect-orientation; ontology model

收稿日期: 2017-06-20

通信作者: 王 华(1978—),男,浙江省金华人,副教授,博士,主要从事自适应系统、软件体系结构研究。E-mail: fhfwh@qq.com。

在软件体系结构(简称为体系结构或 SA)的研究中,用连接件表示构件间复杂的语义关系。将连接件视为与构件等同的实体有助于体系结构的研究^[1-2],因为连接件主要负责构件间的通信,这就将构件的计算功能独立化。研究者可以专注于软件生存环境对体系结构的影响并将之建模为连接件,通过连接件修改或替换构件,实现基于体系结构的自适应性。软件生存环境驱动体系结构的重配置可以借助于连接件的适应性调整来实现,然而体系结构描述语言^[3-5]中非显式的构件连接阻碍了体系结构自适应能力的提高。

在自适应的连接件设计中,软件生存环境是一个重要的驱动力。把生存环境视为方面(aspect)进行建模是构建连接件的有效方法^[6-7]。Shin 等提出了分布式环境中安全软件体系结构的可重用连接件的设计方法^[8]。Ringert 等提出了将平台独立的连接件模型转换为平台相关的连接件模型的方法,通过将构件-连接件的结构化扩展组合为构件行为来建模^[9],并提供了连接件裁剪方法^[10]。Khorasani 等借助于 xADL 体系结构描述语言和基于规则的系统实现了软件体系结构的重配置^[11]。这些研究在连接件模型的自适应配置上存在不足,这就迫切需要面向方面的预见式连接件(aspect-oriented proactive connector, APC, 以下简称预见式连接件)以使自适应系统具有更强的可预见性和更智能的行为,从而提高客户服务质量。针对传统被动式连接件的不足,笔者借助面向方面的思想来构造预见式连接件本体模型,以实现构件连接的修改和体系结构的动态配置,提高软件体系结构的自适应性,并借助于影响因子价值去量化选择作为调整政策的依据。通过一个实际运营的在线健康评估系统进行测试,结果表明,本方法为应用提供了预见式的自适应配置机制,提高了应用的服务质量。

1 预见式连接件本体模型设计

1.1 方面连接件设计

Medvidovik 等指出了描述连接件所具备的 6 个特征:接口、类型、语义、约束、演化和 NFR(non-functional requirement, 非功能性需求)属性^[12]。预见式连接件可以方便地定制、调整交互关系,灵活修改构件。一方面,软件体系结构设计师往往关注于功能性的构件,而忽略了生存环境中描述非功能属性的横切关注点(crosscutting concerns),如带宽、安全性、需求变化、分布性等。这些横切关注点的产生会将构件和连接件复杂化,使计算功能、通信功能和非功能性需求“纠缠”在一起。另一方面,体系结构的演化主要是由于横切关注点引起,而借助于 AOP(aspect-oriented programming, 面向方面编程)^[13]思想正好能够把这些横切关注点封装成方面并作为体系结构的构件,称之为方面构件(aspect component, AC),通过一个特殊的连接件对方面构件和基础构件(primary component, PC)进行通信协调。这个特殊的连接件称为预见式连接件。由基础构件完成功能性需求,由方面构件完成对横切关注点的建模。通过预见式连接件可以方便地在方面构件和普通构件之间进行路由,实现体系结构的自适应演化。其中,方面构件则可以实现对横切关注点的重用。

图 1 显示了如何通过预见式连接件实现基础构件和演化因素驱动的方面构件之间的隔离,以此为基础实现体系结构的预见式自适应。从图中可以看出,预见式连接件使用不同的角色将不同的体系结构元素联系在一起,而体系结构的元素就表现为基础构件、方面构件、角色和编织政策(weaving policy)。编织政策可以通过扩展的 ECA(event-condition-action)规则进行描述,笔者在之前的工作中已经开发这个轻量级

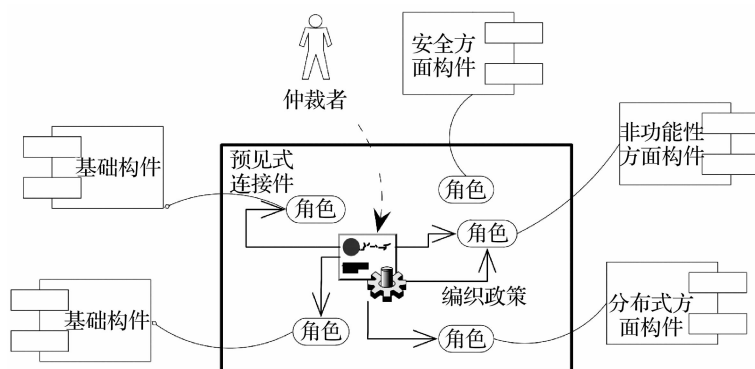


图 1 预见式连接件模型

Fig. 1 APC model

的自适应语言^[14]。仲裁者的作用是接受体系结构描述语言中的体系结构演化脚本。

1.2 预见式连接件本体设计

综上所述,预见式连接件的组成元素包含了角色和编织政策。从体系结构的角度,笔者认为编织政策指明了角色之间相互关联和动态协作的体系结构的自适应行为,将参与体系结构配置和连接的所有角色进行了相关的定义和规约描述。编织政策成为指导体系结构演化的“剧本”,即对体系结构重配置的基础信息。由此可以得出结论,预见式连接件中的所有角色不是直接相连的,它们只能被编织政策使用并通过编织政策中的定义进行信息交互。角色集合反映了在构建一个系统过程中作为子系统的约束条件,即能够连接到预见式连接件上所有构件的接口抽象。

控制转移机制和数据转移机制是预见式连接件的基本构造块的 2 个原语。通过操作这 2 个原语,可以在基础连接件之上构造复合连接件。通过控制转移机制和数据转移机制反映预见式连接件状态的变化,具体表现为预见式连接件所提供的服务语义、互联机制和通信协议等。借助于本体论^[15]和框架表示法^[16]得到如图 2 所示的预见式连接件本体模型图。

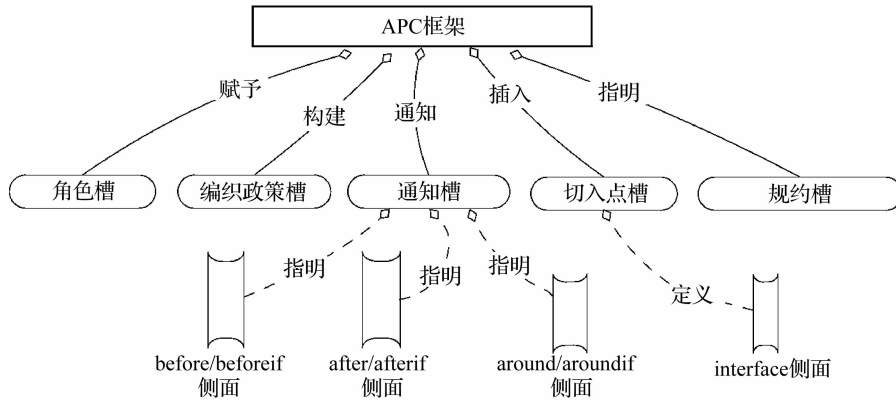


图 2 体系结构的预见式连接件本体模型

Fig. 2 APC ontology model of software architecture

图 2 中的切入点槽表示方面构件的横切面,表现为经由基础构件中的接口传递而来的各种事件,以此触发体系结构的重配置。规约槽规定了预见式连接件的相关约束条件,如网络带宽的限制等。预见式连接件本体模型也扩展了 AOP 中的通知,这包括了基本侧面 before、after 和 around 以及扩展侧面 beforeif、afterif 和 aroundif。应用相应的侧面可以表示调用方面构件或基础构件的触发条件,其具体含义如下:

- 1) before, 将请求传递至所要请求的基础构件之前触发方面构件的动作。
- 2) after, 将请求传递至所要请求的基础构件,接着调用经由编织政策构造的方面构件提供的操作接口。
- 3) around, 取代基础构件的操作,而直接调用经由编织政策构造的方面构件提供的操作接口。
- 4) beforeif(condition), 仅当 condition 为真时按照 before 的方式执行。
- 5) afterif(condition), 仅当 condition 为真时按照 after 的方式执行。
- 6) aroundif(condition), 仅当 condition 为真时按照 around 的方式执行。

1.3 影响因子价值函数

为了表示当前生存环境下用户对系统运行状态的满意程度,本研究设计了一个影响因子价值函数 (impact factor value, IFV),作为预见式自适应动作的评价标准。IFV 的计算公式如下:

$$\text{IFV}(c) = \prod_{d \in \text{dim}(s)} (U_d^{\text{weight}} \times p_{c \rightarrow c'}^{\text{dgr}}). \quad (1)$$

式(1)中,对于构件 c 的每个质量属性 d , $U_d^{\text{weight}} \in [0..1]$ 表示用户对 c 在质量属性 d 上的偏好程度。在某个 d 的 U_d^{weight} 等于零,表示即使其他质量属性权重很高,但 c 对该用户无意义。另外,预见式连接件的重新配置会影响客户对服务的感受,用 p 表示同步代价。例如在线视频服务可能会根据预测的带宽情况动态调整图像编码算法,调整时需要通知客户以保持服务的连续性,这会引起客户的分心而导致服务

质量的下降,于是可以通过同步代价进行度量。这里, $dgr \in [0..1]$ 表示当构件 c 替换成构件 c' 时用户的满意程度,IFV(c)数值越大表示同步代价越大。

2 实例研究

为了评估本方法的可行性而进行了一个案例研究分析。目标系统是一个健康管理子系统——健检助手系统(HES, health evaluation assistant system, 以下简称本系统),网址是 <http://www.joyhar.com>,即通过数据挖掘给出相关的健康促进书。从体检结果、生活习惯、饮食、生活环境、职业行为等方面出发,提供对个人最重要的个性化健康信息,对身体状况进行全面分析,结合疾病分析模型对慢性病进行早期预警,并提出系统性的健康促进建议。该评估系统已运行 3 年,获得了大量的数据,客户总数为 55 027 个,生成的《健康促进书》中有效的总份数为 103 429 份,笔者抽取了客户体检项目中的 23 891 项。本系统通过算法模型对健康问卷和该次体检的结果数据进行分析,生成个人健康促进书,包括个人动态健康档案、个人疾病解读、个人异常指标、个人生活方式指导、个人饮食指导、个人运动指导、个人心理指导、高发疾病风险提示、个人健康趋势及规划和体检报告 10 个模块。

根据本系统的运营规则,不同客户可以根据自己的需要订购不同质量的服务,如金卡用户或者银卡用户。随着越来越多的客户使用系统提供的服务,本系统经历了一系列事件,如请求者需等待更长的时间才能得到一份健康评估报告,在体检高峰期某些金卡客户无法得到服务。这些事件可以作为驱动体系结构演化的横切面来考虑,即软件生存环境的变化。本系统将它们编织为方面构件和预见式连接件驱动体系结构演化。

2.1 预见式连接件构建过程

本系统预见式连接件构建过程如下。

1) 初始构件模型如图 3 所示。Client 构件和 HealthReport 构件是简化后构件,分别表示客户请求构件和健康评估报告构件。

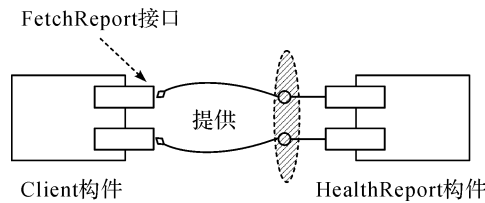


图 3 HES 初始构件模型

Fig. 3 Initial component model of HES

2) 为了提供更好的服务质量,对网络带宽进行前向预测,把检查带宽看成是本系统的横切面,并构造成为一个方面构件 PredictResource。该构件至少提供了分别表示预测带宽和 CPU 负载的接口——PredictBandwidth 和 PredictCPU。本系统还构建了计算影响因子价值的方面构件 CalculateIFV,为计算总影响因子价值,CalculateIFV 提供计算服务。

3) 把 PredictResource 构件织入本系统,根据获得的预测带宽值预见式调整构件的服务配置,把来自 Client 构件的 FetchReport 接口作为 PredictResource 方面构件的切入点。带宽度量值分为 3 个等级: LOW、HIGH 和 MEDIUM,分别用小于 10 Kbit/s、10~100 Kbit/s 和大于 100 Kbit/s 来表示,可以得到图 4 所示的编织政策。

```
on Client.fetchReport if PredictBandwidth=LOW triggers CalculateIFV.service[aroundif(format=GRAPHIC)]HealthReport.report
on Client.fetchReport if PredictBandwidth=HIGH triggers CalculateIFV.service[afterif(format=TEXT)]HealthReport.report
on Client.fetchReport if PredictBandwidth=HIGH triggers CalculateIFV.service[beforeif(format=GRAPHIC)]HealthReport.report
on Client.fetchReport if PredictBandwidth=MEDIUM triggers CalculateIFV.service[before]HealthReport.report
```

图 4 HES 的编织政策

Fig. 4 Weaving policy of HES

4)对 PredictResource 进行动态编织建模,这需要构建一个预见式连接件——ResourceApcc,用来协调 CalculateIFV、Client 和 HealthReport 三者之间的交互,根据编织政策决定是否调用 CalculateIFV 构件及何时调用 CalculateIFV 构件。本系统演化后的连接件模型如图 5 所示。

2.2 实验结果分析

通过预见式连接件的动态调整后,HES 系统的响应时间与未优化时相比较的结果如图 6 所示。

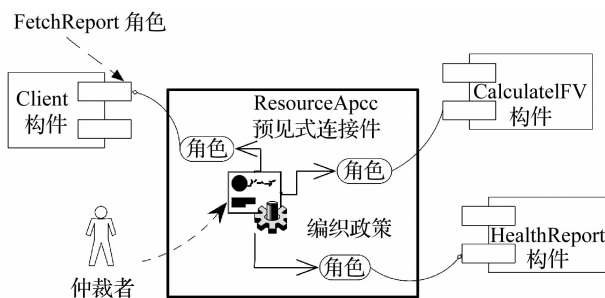


图 5 HES 演化后的连接件模型

Fig. 5 Connector model of evolved HES

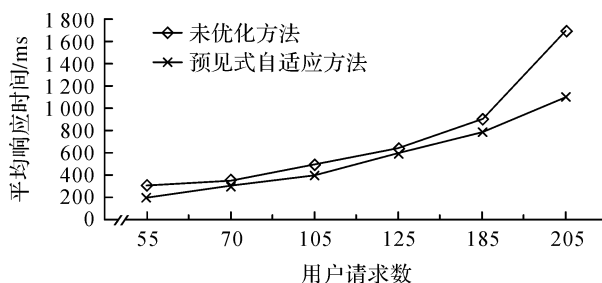


图 6 未优化方法和预见式自适应方法的性能比较

Fig. 6 Comparison of performance between the non-optimized method and the proactive self-adaptive method

从图 6 可以看出,随着用户请求数的增加,用户请求的平均响应时间也在增加。当系统有 125 个用户请求时,自适应优化方法比未优化方法稍好;随着用户请求数的不断增加,平均响应时间增加比较快。此时,预见式自适应优化方法显示出了较好的性能。并且请求数越多,性能的改善更加明显。

3 结 论

软件生存环境的可变性和不确定性对体系结构的自适应性提出了新的要求,传统连接件的自适应性是被动式的,即必须预先设置规则(政策),通过生存环境的条件改变触发自适应政策的执行。为了克服这种局部性和被动式的调整方式,笔者提出了面向方面的预见式连接件模型,并将影响因子价值量化以调整政策,以此作为选择政策的依据。经过对一个在线的健康评估助手系统进行改造使之具备自适应的能力,实验结果表明,当使用本文的预见式连接件模型时提高了系统的性能,因而具有应用可行性。

参考文献:

- [1] OZKAYA M. What is software architecture to practitioners: a survey[C]//5th International Conference on Model-Driven Engineering and Software Development. Porto: MODELSWARD,2017:677.
- [2] 梅宏, 陈锋, 冯耀东, 等. ABC:基于体系结构、面向构件的软件开发方法[J]. 软件学报,2003,14(4):721.
- [3] BLOM H, CHEN D J, KAIJSER H, et al. EAST-ADL: an architecture description language for automotive software-intensive systems in the light of recent use and research[J]. International Journal of System Dynamics Applications,2016,5(3):1.
- [4] ZUNIGAPRIETO M, INSFRAN E, ABRAHAO S. Architecture description language for incremental integration of cloud services architectures[C]//IEEE Symposium on the Maintenance and Evolution of Service-Oriented Systems and Cloud-Based Environments. Toronto: IEEE,2016:16.
- [5] SARI B, REUSS H C. A model-driven approach for the development of safety-critical functions using modified architecture description language (ADL)[C]//International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles and International Transportation Electrification Conference. Toulouse: IEEE, 2017:1.
- [6] SHESHASAYEE A, JOSE M R. A theoretical framework for the maintainability model of aspect oriented systems [J]. Procedia Computer Science,2015,62:505.

(下转第 64 页)