

基于方面构件模型及其组装与实现

谢武平¹, 薛锦云^{1,2}, 万松松¹

(1. 江西师范大学 计算机信息工程学院, 江西 南昌 330022;

2. 中国科学院 软件研究所 计算机科学开放实验室, 北京 100080)

摘要:将面向方面编程技术融入传统的基于构件开发方法,提出了一种基于方面的构件模型。同时探讨了基于方面构件的组装策略以及利用XML语言描述组装、编织方面化构件的处理逻辑。新的构件模型采用更加模块化的构件实现方式,从传统构件模型中提取出方面元素,很好地解决了传统的基于构件开发方法中构件间关注点横切所导致系统混乱的问题,提高了构件的可复用性以及软件的开发效率。

关键词:面向方面编程;构件组装;关注点

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2009)04-0160-03

Aspect - Based Component Model and Its Assembly and Implementation

XIE Wu-ping¹, XUE Jin-yun^{1,2}, WAN Song-song¹

(1. College of Computer Information Engineering, Jiangxi Normal University, Nanchang 330022, China;

2. Laboratory for Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: Integrating aspect-oriented programming technology into the traditional component-based software development methodology, and raises aspect-based component model. The relevant assembly strategy is discussed, and using XML to describe the processing logic of assembly and weaving of aspect-based component. The new component model adopts a more modular component to implement, distills the aspect element from traditional component model, and satisfactorily resolves the problem of chaotic systems caused by cross-cutting concerns in the traditional development methods, which can enhance the reusability and development efficiency of software.

Key words: aspect-oriented programming; component assembly; concern

0 引言

近年来随着软件系统规模的激增,系统的开发效率、可扩展性、可维护性等显得极为重要。软件复用得到了人们越来越多的关注。而其中的软件构件技术更是被认为是解决软件危机、提高软件生产效率和质量的现实可行方法^[1]。

基于构件的软件开发(component-based software development, CBSD)通过将可复用构件运用一定的集成机制组装成为完整的系统^[2],从而缩短软件的开发周期、增强软件的可维护性。但该开发方法将实现系

统的各个模块同等看待,从而导致系统实现中非核心业务模块(如:身份验证、存储管理等)与核心业务模块交织在一起,造成非核心业务横跨系统中的多个模块,使得模块内部的代码混乱。为了更好地将系统模块化,提出了基于方面(aspect-based programming, AOP)^[3]的开发方法,对原有模块进行重组,抽取与系统业务无关却为多个模块所通用的非核心业务处理逻辑,封装成为各个“方面”。

基于方面的开发方法,将整个系统看作由多个关注点构成,其中完成系统核心业务的模块称为核心关注点^[4]。核心关注点的实现只关注自身业务逻辑,并不关心与其横切的方面。每个核心关注点的实现更加独立,降低了模块的耦合度。

CBSD通过集成可重用构件提高了软件的开发效率。面向方面技术区别对待系统中关注点,实现横切方面的分离,提升了软件的内聚度。在文献[5]中,通过对构件的方面化,提出了一种方面化的构件模型,该

收稿日期:2008-07-24

基金项目:国家自然科学基金项目(60273092);科技部973重大基础研究发展前期专项资助项目(2003CCA02800)

作者简介:谢武平(1984-),男,江西丰城人,硕士研究生,研究方向为软件形式化和自动化、计算机软件与理论;薛锦云,教授,博士生导师,研究方向为软件形式化和自动化、并行分布式计算、智能教学软件、软件工程。

择 P 或 Q 的需求。

并行组装:核心构件 P、Q 并行组装,所得的复合构件 R 的外部行为表现为 P、Q 的并行执行行为。R 构件的对外需求表现为 P、Q 构件的需求。

以上各种组装机制中,R 的需求不仅包括功能,接口需求,还包含接入点需求。

(2)核心构件与方面构件的编织。方面构件定义为非功能性构件,所以将核心构件与方面构件进行编织,并不会使原核心构件在功能上有所增强或削弱。

核心构件 P 与方面构件 Q 进行编织,所得复合构件 R 的外部行为表现为核心构件 P 的外部行为。但由于方面构件的织入,满足了核心构件的部分对外接入点需求,R 构件的对外接入点的需求将为 P 的接入点需求集合减去 Q 实现的接入点集合。

(3)方面构件与方面构件的组装。将方面构件 P 与方面构件 Q 进行组装时,P、Q 为非功能性构件,所得复合方面构件 R 对外表现为 P 的对外提供方面服务与 Q 的对外提供方面服务的组合。

3 构件组装的实现

基于方面构件组装实现的关键是要实现构件模型中的编织器。因 XML 语言有强大的描述机制及可展性的特点,此处根据上一节的组装策略,选用 XML 描述生成的构件组装逻辑。

核心构件由一个构件名(name)、参数属性、构件的方法和构件的接入点等构成。核心构件的定义以<component>标记开头,以</component>结尾。其中接入点属性包括描述横切方面的位置信息。

方面构件由一个构件名(name)、方面、访问控制属性等构成。方面构件的定义以<aspect>标记开头,以</aspect>标记结尾。

(1)核心构件与核心构件的组装^[9]。

选择组装:

```

<component name = " " ... > //新构件名称、属性等
<composition> //核心构件与核心构件的组装
<condition exp = " " > //判断类型,及选择次数
<true> //判断为“真”组装构件
<component name = " " ... > </component> //为真时组装执行构件
.....
</true>
<false> //判断为“假”组装构件
<component name = " " ... > </component> //为假时组装执行构件
.....

```

```

</false>
..... //可以实现递归,从而实现循环
</condition>
</composition>
.....
</component>
顺序组装:
<component name = " " ... > <sequence> <component name = " " ... > </component>
.....
</sequence>
</component>
并行组装:
<component name = " " ... >
<parallels msg = > //msg: 发送给并行执行的构件的消息
<component name = " " ... > </component>
.....
</parallels>
</component>

```

(2)核心构件与方面构件的编织。该编织过程见图 2,方面构件为非功能性构件,将其编织到核心构件中,并不影响核心构件的功能,但将减少核心构件的约束需求。

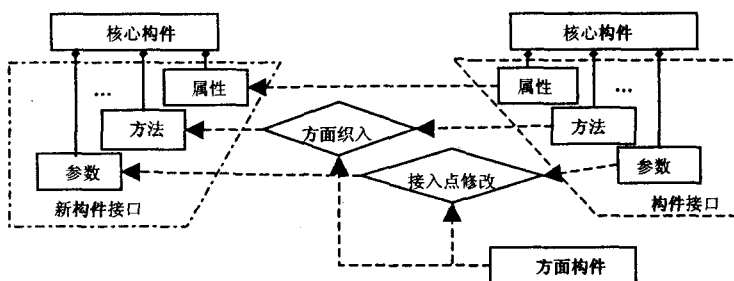


图 2 核心构件与方面构件编织

核心构件与方面构件的编织可用 XML 表示为:

```

<component name = > //新构件。组装后的构件
<weaving> //核心构件与方面构件的编织
<component name> //所用核心构件
<position exp = "before, after, around" > //根据不同接入点属性。其方面加入位置亦不同
<aspect name =, prescribe = exp > </aspect> //根据接入点,织入相应方面
.....
</position>
</component>
.....
</weaving>
.....
</component>

```

(3)方面构件与方面构件的组装。该类组装所得

计算高层态势所需要的数据。

2.2 实验结果

实验结果如图1所示。标题为“2008年8月节点1”是2008年8月节点1的节点级安全态势量化值做出的曲线。纵轴表示态势值,范围0到 10^5 。横轴表示时间,刻度每增加1代表时间增加1小时。

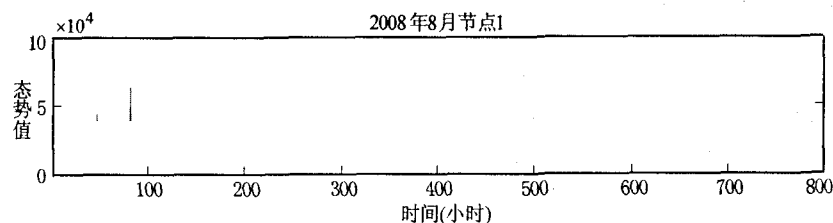


图1 2008年8月态势分析实验结果

从图1可见,出现了一些峰值,这代表此时的网络可能出现异常、遭到攻击,或者网络正常服务能力下降。可以由用户自定义阈值来判断多高的态势值出现时需要关注。

3 结束语

提出一个层次化宏观网络流量数据态势量化分析的方法,基于流量数据依次提取出服务级、节点级和网络级的流量态势量化值。该算法复杂度低,处理效率

较高,可以用于大型网络环境中,进行在线态势感知。

参考文献:

- [1] 陈秀真,郑庆华,管晓宏,等.层次化网络安全威胁态势量化评估方法[J].软件学报,2006,17(4):885-897.
- [2] 潘兆亮,蒋兴浩,陈秀真.广域网络安全态势值统计分析算法[J].信息安全与通信保密,2008,7:67-69.
- [3] 王慧莹,周宁,陈秀真,等.网络态势分析中服务可用性评估方法[J].计算机工程与应用,2008,44(14):130-133.
- [4] Park S T, Khrabrov A, Pennock D M, et al. Static and Dynamic Analysis of the Internet's Susceptibility to Faults and Attacks[C]//Proc. of INFOCOM'03. California, USA: IEEE, 2003:1901-1910.
- [5] Lakkaraju K, Yurcik W, Lee A J. NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness [C]//Proc. of ACM Workshop Visualization and Data Mining for Computer Security (VizSEC/DMSEC). New York, USA: ACM Press, 2004:65-72.
- [6] Yegneswaran V, Barford P, Paxson V. Using Honeynets for Internet Situational Awareness [EB/OL]. 2006. http://www.cs.wisc.edu/~pb/hotnets05_final.pdf.

(上接第162页)

复合构件仍为方面构件,同时采用消息传递来实现不同方面的选择,具体用XML可以表述如下:

```
<aspect name = ""...>
<composition msg = "">//msg描述了接受消息的方面
<aspect name = ""...></aspect>
.....
</composition>
.....
</aspect>
```

4 结束语

采用AOP技术对原有基于构件开发方法进行改进,较好地解决了传统开发中非核心关注点横切多个系统模块的问题。使开发者能够专注于系统核心功能的分析。

面向方面技术经过多年研究,已日趋成熟,但关于方面的形式化语义研究还有待完善。下一步的研究工作主要将进行对基于方面的构件进行形式化的规约定义,实现基于规约的方面构件的自动组装与形式化推导。

参考文献:

- [1] 王忠杰,徐晓飞,战德臣.基于特征的构件模型及其规范化设计过程[J].软件学报,2006,17(1):39-47.
- [2] 任乐敏,钱乐秋.构件组装及其形式化推导[J].软件学报,2003,14(6):1066-1074.
- [3] Kiczales G, Lamping J, Mendhekar A, et al. Aspect-Oriented Programming[C]//The 11th European Conf. on Object-oriented Programming. Finland: Springer-Verlag, 1997: 220-242.
- [4] Sommerville L. 软件工程[M].英文版.第8版.北京:机械工业出版社,2007.
- [5] 盛津芳,王斌,陈松乔.方面化构件模型及其组装方法[J].计算机工程,2006,32(5):39-40.
- [6] 吴穹.构件组装技术研究[D].北京:北京大学,1998.
- [7] 毛鸢池,梁奕,王志坚.异构软件构件组装模型设计与实现[J].计算机工程,2005,31(4):56-57.
- [8] 张世琨,张文娟,常欣,等.基于软件体系结构的可复用构件的制作和组装[J].软件学报,2000,11(9):1351-1359.
- [9] 廖昕,陈松乔.可复用构件组装技术研究[J].计算技术与自动化,2004,23(3):50-52.