一种用于软件运行时分析的行为模式描述语言

李长云，王志兵

（湖南工业大学 计算机与通信学院，湖南 株洲 412008）

摘　要：软件本质上是代替人执行一定行为的，对软件行为的描述与分析一直是软件技术关注的重点。为有效表达软件交互行为，提高分析效率，基于正则表达式，提出了一种用于软件运行时分析的行为模式描述语言BPL。在BPL中，通过对软件运行时可观察行为中反复出现的事件序列的特征抽取和概括，软件行为被描述为一个由小写字母表示的行为踪迹及约束构成。最后使用BPL描述了电子交易过程。

关键词：行为模式；迹模式；约束

中图分类号：TP311　　**文献标志码**：A　　**文章编号**：1673-9833(2010)06-0034-04

A Behavior Pattern Descriptive Language for Software Runtime Behavior Analysis

Li Changyun，Wang Zhibing

（School of Computer and Communication，Hunan University of Technology，Zhuzhou Hunan 412008，China）

Abstract: In essence, software may substitute human to execute some certain actions. The description and analysis for software behavior is always research focus. For the effective expression of the software interactions and the improvement of analysis efficiency, presents a novel descriptive language for software behavior pattern—BPL based on regular expressions, which describes software behavior as a combination of traces and constraints marked with lowercase letters after abstracting and generalizing the features of recurrent event sequences of the observable software behaviors at runtime. Finally, BPL is used to represent the process of electronic transaction.

Keywords：behavior pattern；trace pattern；constraint

0　引言

软件本质上是代替人执行一定行为的[1]，对软件行为的描述一直是软件技术关注的重点，也是软件设计和维护的基础。软件行为描述以形式化或非形式化的语言对软件行为进行描述和文档化。传统的软件行为描述技术着眼于开发阶段的逻辑行为，如VDM [2]、XYZ[3]、演算[4]、Petri网[5]、程序流程图、面向对象语言UML等。

对软件系统在运行阶段的真实行为进行描述和分析，越来越受到重视，已出现一些软件运行时的行为描述方法[6-8]。但是，近年来开放网络环境下由多实体松散聚合而成的软件系统迅速地兴起和发展。开放环境产生了一些新的软件行为描述与分析问题，如组合行为描述与分析、恶意入侵检测等。尤其在开放网络环境下软件在运行阶段发生故障或不可信时，其运行行为往往呈现出一定的特征和规律性。例如在Web Services应用中，当服务不存在时，对它的调用将产生

收稿日期：2010-09-19

基金项目：国家自然科学基金资助项目（60773110），中国博士后科学基金资助项目（20080440216），湖南省自然科学基金资助项目（09JJ6087），湖南省研究生创新基金资助项目（CX2009B200），湖南省教育厅科研基金资助项目（07C234, 09C325）

通信作者：李长云（1971-），男，湖南耒阳人，湖南工业大学教授，博士后，主要研究方向为可信软件和软件动态演化，E-mail：lcy469@163.com

一个异常交互事件。在数字化电子资源的订购方面，校园网用户可以合法下载文章，但如果利用下载工具大量下载文章，其行为可能是不可信的。笔者将这种具备一定特征和规律性的软件运行行为称为行为模式。显然，从软件运行行为所遵循的行为模式中，可诊断软件的故障和推测软件的可信性，行为模式成为软件运行时分析的一个重要对象和实体。行为模式如何构成？如何描述行为模式？这些就成为软件运行时分析首先应解决的问题。

1　行为模式的描述

1.1　行为模式描述框架

软件运行行为分为可观察的行为和不可观察的行为2种。软件系统的边界和外部环境间的交互操作当然是可观察的。现代软件系统一般由多个构件组装而成，对于构件间的交互事件，既使这些构件是第三方的，通过一定的监控技术和监控工具也可能获取到，因此构件间的交互事件也是可观察的。至于构件内部的计算行为很难甚至不能被监测和收集到，视作不可观察的。作为软件运行时分析的对象，关注的是可观察的行为。行为模式是对软件可观察行为中反复出现的事件序列的特征抽取和概括，行为模式的描述着重考虑以下几个方面：

1）抽象性。行为模式是具体软件可观察行为的一种抽象，概括了一类具有共同特点的交互事件序列。

2）片段性。行为模式重点关注软件可观察行为的某些片段，反映可观察行为的一些片段特征。

3）约束性。行为模式不仅是事件序列特征，也包括一些行为约束性质。如事件参数应满足的约束性质，事件间的偏序关系等。

依据上述考虑，行为模式的描述框架如图1所示。

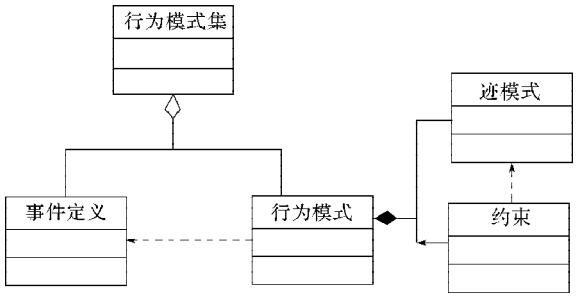

图1　行为模式描述框架

Fig. 1　Description framework of behavior pattern

一个行为模式集（behavior pattern set）包括多个事件定义（event definition）和多个行为模式定义（behavior pattern definition），在一个模式集中定义的事件可由这个模式集中的所有行为模式引用。行为模式由迹模式和约束组成，迹模式是相似事件序列的抽象和概括，而约束依赖于迹模式，反映事件间的约束关系和事件参数应遵守的性质。

行为模式集的BNF（backus-naur form）语法：

<Behavior Pattern Set> ::= “Pattern Set”<IDENTIFIER>
　　{ Event Definition }
　　{ “,” Behavior Pattern
　　Definition }
　　“End”

行为模式定义的BNF语法：

Behavior Pattern Definition ::= “Pattern”
<IDENTIFIER>
　　<Statement>
// 约定模式标识为大写英文字母开始的字符串；
Statement ::= < Trace Pattern> [Constriant]

1.2　事件定义

软件可观察行为由一系列事件组成，事件分为2种：状态变化事件和交互事件。状态变化事件在系统全局状态改变时触发，体现为系统的某些可观察全局变量，如工作流系统中的流程变量++的值的变化。交互事件体现为系统边界和外部环境间的交互操作，以及系统内部构件间的交互动作，无论是边界的交互操作还是内部的交互动作，都可以归结为对构件的方法调用。

对应上述情况，事件定义的BNF语法如下：

Event Definition ::= “Event ”<IDENTIFIER> =
“Update”< Variable > |
“StartM” <Method Definition >|
“EndM” <Method Definition >

Method Definition ::= “Method”< IDENTIFIER > “
(” Parameter List “)”
[“, From” Component][“, To” Component “)”]
[其它相关信息，如执行方法的线程]

一般地，约定事件标识为小写英文字母。变量更新（Update）后接一个对于监测服务是可见的变量，当变量值变化时被触发。方法调用（StartM和EndM）反映构件的交互，StartM开始一个方法调用，EndM结束一个方法调用返回。Method Definition中的方法原型应同应用中构件的方法原型一致，即方法名一样，参数列表也一样。另外，在方法定义中，提供了方法调用源构件（From Component）和方法调用目的构件（To Component）的选项，以及其它相关信息的选项，这就使得从事件定义映射到构件方法上具备不同抽象层次的能力。例如，当应用需要关注多个同名方法调用但源构件不同时，可以在方法定义中填写From Component选项，当无需关注不同源构件时，可以忽略From Component选项。同样地，关注同名构件方法的执行

是否在同一线程中也可在“其它相关信息”的选项中填写方法的线程信息。同时，在如何组织交互事件时，可以根据应用目的的不同灵活处理，例如可以事件发生源构件为线索，也可以事件目标构件为线索，组织形成事件序列。

1.3 迹模式

行为模式主要由迹模式（Trace Pattern）构成。所谓迹指的是事件序列片断，迹模式则是对迹的抽象和概括。为表示相似迹的共性和统计特性，迹模式使用以事件标识为字母的正则表达式来描述。一个正则表达式就是由普通字符（例如字符 a 到 z）以及特殊字符（称为元字符）组成的文字模式。例如，迹模式“$(a*b)$”表示以事件 a 开始，中间接若干任意事件并以事件 b 结束的迹，如“ab”,“$acdeab$”和“$afcb$”等；迹模式“$(ab)\{2,6\}$”表示事件 a 和 b 交替出现2到6次的迹，如“$abab$”,“$abababab$”等。

迹模式的BNF语法：

Trace Pattern ::=“Trace Pattern”<以事件标识为字母的正则表达式>

表1是迹模式所使用的正则表达式语法。

表1 迹模式所使用的正则表达式语法

Table 1 Regular expression grammar for trace pattern

字符	描述
*	匹配前面的子表达式零次或多次。例如，zo* 能匹配 "z" 以及 "zoo"。* 等价于{0,}。
+	匹配前面的子表达式一次或多次。例如，'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,}。
?	匹配前面的子表达式零次或一次。例如，"do(es)?" 可以匹配 "do" 或 "does" 中的"do" 。? 等价于 {0,1}。当该字符紧跟在任何一个其他限制符 (*, +, ?, {n}, {n,}, {n,m}) 后面时，匹配模式是非贪婪的。非贪婪模式尽 可能少的匹配所搜索的字符串，而默认的贪婪模式则尽可能多的匹配所搜索的字符串。例如，对于字符串 oooo"，'o+?' 将匹配单个 "o"，'o+' 将匹配所有 'o'。
{n}	n 是一个非负整数。匹配确定的 n 次。例如，'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o。
{n,}	n 是一个非负整数。至少匹配 n 次。例如，'o{2,}' 不能匹配 "Bob" 中的 'o'，但能匹配 "foooood" 中的所有 o。o{1,}' 等价于 'o+'。'o{0,}' 则等价于 'o*'。
{n,m}	m 和 n 均为非负整数，其中 n<=m。最少匹配 n 次且最多匹配 m 次。例如，"o{1,3}" 将匹配 "fooooood" 中前 3 个 o。'o{0,1}' 等价于 'o?'。请注意在逗号和 2 个数之间不能有空格。
x|y	匹配 x 或 y。例如，'z|food' 能匹配 "z" 或 "food"。'(z|f)ood' 则匹配 "zood" 或 "food"。
[xyz]	字符集合。匹配所包含的任意一个字符。例如， '[abc]' 可以匹配 "plain" 中的 'a'。
[^xyz]	负值字符集合。匹配未包含的任意字符。例如， '[^abc]' 可以匹配 "plain" 中的'p'。

1.4 约束

除迹模式外，行为模式可能还包含约束（Constraint）。约束使用逻辑表达式表示，依赖于迹模式，用于描述迹模式所表达的迹所包含的事件间的约束关系和事件参数应遵守的性质。例如，“valueof(e.arg(1))>6”意味 e 事件第一个参数的值必须大于6。约束的BNF语法如下：

Constraint ::= “Constraint”(<Simple constraint> | <constraint> && <constraint>| <constraint> || <constraint>)

Simple constraint ::= <Boolean Expression>

为方便表达模式约束，预定义了一些函数，见表2。

表2 模式约束函数

Table 2 Pattern-constraint functions

函数名	描述
Event (t,i):e	提取迹 t 中的第 i 个事件
Eventindex(t,e):i	提取迹 t 中第 1 个出现的 e 事件的位置
Eventsum(t,e):i	统计迹中 e 事件出现的次数，e 取值 * 则表示迹中所有事件个数
Valueof(e.arg(i)):v	取得 e 事件第 i 个参数的值
Typeof(e.arg(i)):s	取得 e 事件第 i 个参数的类型
Timeof(e):t	取得 e 事件所发生的时间

注 t 表示迹，i 表示整型，e 表示事件，v 表示类型可变参数，s 表示字符串。

2 应用案例

以一个电子交易系统为例来说明BPL的使用。信购网是一个C2C的电子交易平台，任意人都可以在网上注册并购买商品以及开设店铺。由于交易双方多为陌生个体，没有可靠机构对其身份信息进行可信认证，因此依据交易行为来识别对方的可信性就成为交易方的一个需求。买家需要辨别卖家的销售可信指数，以决定是否订购和汇款；卖家需要辨别买家的消费可信指数，以决定是否交易、发货和折扣。

经过对大量交易过程的观察和分析，可如下所述应用BPL描述4种正常交易行为和6种异常交易行为。

2.1 正常交易过程

1） 买家正常消费模式（按用户id组织）：登录–(login）–搜索商品（search）–下订单（order）–预付款（imprest）–确认收货（accept）–评价（evaluate）–退出（logout），迹模式表示为：l(soiae) {1,}*g；

2） 卖家正常销售模式（按用户id组织）：登录–（login）–查看订单（examine）- 修改价格（modification）–查看付款信息（inquire）–发货（consignment）–退出（logout），迹模式表示为：lem*qcg；

3）买卖双方正常交互模式（按订单号组织）：买家下订单（order）–卖家确认，修改价格（modification）–买家预付款（imprest）–卖家发货（consignment）–买家收货、付款（payment），迹模式表示为：(o*m*i*c*p){1,}；

4）带约束的正常消费模式（买家在下订单前须先登录）：Eventindex(*t*,*o*) > Eventindex(*t*,*l*)。

2.2　异常交易过程

1）买家不诚信交易模式：买家多次（2 次以上）下订单不付款，迹模式表示为：o{2,}*(^p)；

2）卖家不诚信交易模式：买家多次（2 次以上）下订单、预付款而卖家不发货，迹模式表示为：(op){2,}*(^c)；

3）买家恶意推荐模式：同一买家连续多次（5 次以上）以正常消费模式、最小交易额（金额 *money*<10）进行交易，且都给出好评，(oiae){5,}* (Valueof(*.arg(1))<10 && Valueof(*e*.arg(3))=1)。在Method Definition 的Parameter List 中设定：事件“*o*, *i*, *a*”的第一个参数为交易金额，第 2 个参数为卖家 id；事件“*e*”的参数 3 指的是评价值，1 为好评，0 为中评，−1 为差评；

4）买家恶意诋毁模式：同一买家连续多次（3 次以上）以正常消费模式、最小交易额（金额 *money*<10）进行交易，且都给出差评，(soiae) {3,}* (oiae){3,}* (Valueof(*.arg(1))<10&& Valueof(*e*.arg(3))= −1)。在Method Definition 的Parameter List 中设定：事件“*o*, *i*, *a*”的第一个参数为交易金额，第 2 个参数为卖家 id；事件“*e*”的参数 3 指的是评价值，1 为好评，0 为中评，−1 为差评；

5）买家合谋推荐模式：在同一 IP 以不同用户名连续多次（5 次以上）以正常消费模式与同一卖家以最小交易额（金额 *money*<10）进行交易，且都给出好评，(oiae){5,}* (Valueof(*.arg(1))<10 && Valueof(*.arg(2))=id(seller) && Valueof(*e*.arg(3))=1)。在Method Definition 的Parameter List 中设定：事件“*l*”的第一个参数为用户名，第二个参数为登录 IP；事件“*o*, *i*, *a*”的第一个参数为交易金额，第 2 个参数为卖家 id；事件“*e*”的参数 3 指的是评价值，1 为好评，0 为中评，−1 为差评；

6）买家合谋诋毁模式：在同一 IP 以不同用户名连续多次（3 次以上）以正常消费模式与同一卖家以最小交易额（金额 *money*<10）进行交易，且都给出差评，(soiae) {3,}* (oiae){3,}* (Valueof(*.arg(1))<10 && Valueof(*.arg(2))=id(seller) && Valueof(*e*.arg(3))=−1)。在Method Definition 的Parameter List 中设定：事件“*l*”的第一个参数为用户名，第二个参数为登录 IP；事件“*o*, *i*, *a*”的第一个参数为交易金额，第 2 个参数为卖家 id；事件“*e*”的参数 3 指的是评价值，1 为好评，0 为中评，−1 为差评。

3　结语

行为模式描述语言是基于正则表达式的一种描述语言。本文介绍了行为模式描述语言的描述框架、事件定义、迹模式及约束。与其它行为描述方法相比，行为模式描述语言具有如下特点：1）行为模式描述语言把软件之间复杂的交互行为看做是事件序列及行为约束，从而简化了交互行为描述的复杂性，便于后续分析；2）行为描述模式语言基于正则表达式，语法简单，易于理解和使用；3）基于行为模式描述语言的模式匹配算法易实现，时间复杂度较低。

参考文献：

[1] 屈延文. 软件行为学[M]. 北京：电子工业出版社，2004：195−197.
Qu Yanwen. Software Behavior[M]. Beijing: Publishing House of Electonics Industry，2004：195−197.

[2] Jones C B. Systematic Software Development Using VDM [M]. Hertfordshire：Prentice Hall International (UK) Ltd.，1990：1−20.

[3] 虞　凡，覃　征，贾晓琳，等. 基于 XYZ/E 规范的软件测试用例自动生成方法[J]. 计算机工程，2005，31(19)：76−78.
Yu Fan，Qin Zheng，Jia Xiaolin，et al. Automatic Generation Method of Software Test Case Based on XYZ/E Specification [J]. Computer Engineering，2005，31(19)：76−78.

[4] Bodei C，Degano P，Nielson F，et al. Control Flow Analysis for the Calculus[J]. Springer Lecture Notes in Computer Science, 1998，1466：481−488.

[5] Starke P H. Processes in Petri Nets[J]. Springer Lecture Notes in Computer Science，1981，117：350−359.

[6] Kristoffersen1 K J，Pedersena C，Andersena H R. Event-Based Runtime Checking of Timed LTL[J]. Electronic Notes in Theoretical Computer Science，2003，89(2)：210−225.

[7] 宋　巍，马晓星，吕　建. Web 服务组合动态演化的实例可迁移性[J]. 计算机学报，2009，32(9)：1816−1831.
Song Wei，Ma Xiaoxing，Lv Jian. Instance Migration in Dynamic Evolution of Web Service Compositions[J]. Chinese Journal of Computers，2009，32(9)：1816−1831.

[8] 李长云，李赣生，何频捷. 一种形式化的动态体系结构描述语言[J]. 软件学报，2006，17(6)：1349−1359.
Li Changyun，Li Gansheng，He Pinjie. A Formal Dynamic Architecture Description Language[J]. Journal of Software, 2006，17(6)：1349−1359.

（责任编辑：罗立宇）