

# UML 2.0 顺序图的 XYZ/E 时序逻辑语义研究<sup>\*</sup>

黄正宝<sup>1</sup> 张广泉<sup>1,2</sup>

(苏州大学计算机科学与技术学院 苏州 215006)<sup>1</sup>

(中国科学院计算机科学国家重点实验室 北京 100080)<sup>2</sup>

**摘要** UML 2.0 顺序图适合于描述软件体系结构的各个组件之间和复合组件内部各个子组件之间的动态交互行为,但由于 UML 2.0 顺序图的语义不够精确,使得它的描述结果不利于进一步的分析和验证。基于此,本文在定义 UML 2.0 顺序图的语法和语法约束的基础上,给出了 UML 2.0 顺序图的 XYZ/E 时序逻辑语义,为使用 UML 2.0 顺序图与 XYZ/E 相结合的方式描述软件体系结构的动态交互行为奠定了基础。

**关键词** 时序逻辑,形式化语义,UML 2.0,顺序图,XYZ/E

## Semantics of UML 2.0 Sequence Diagrams in XYZ/E

HUANG Zheng-Bao<sup>1</sup> ZHANG Guang-Quan<sup>1,2</sup>

(School of Computer Science and Technology, Soochow University, Suzhou 215006)<sup>1</sup>

(Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100080)<sup>2</sup>

**Abstract** UML 2.0 sequence diagrams are suitable for describing the interaction behavior among components and sub-components of composite component, but the semantics of UML 2.0 sequence diagram is not precise, this makes the description of UML 2.0 sequence diagram can't be used for analysis and validating of software architecture. This article defines the syntax and the constraint of UML 2.0 sequence diagram, then defines the semantics of UML 2.0 sequence diagram in XYZ/E, it offers the foundation for describing the interaction behavior of software architecture with the combination of UML 2.0 sequence diagram and XYZ/E.

**Keywords** Temporal logic, Formal semantics, UML 2.0, Sequence diagram, XYZ/E

UML 是一种语义丰富、通用、可视化的建模语言,它提供了丰富的视图,可以从不同视角描述系统的不同侧面,因此,很多研究人员采用了各种策略把 UML 用于软件体系结构的描述<sup>[1]</sup>。基于 UML 的软件体系结构描述比较直观,便于开发人员之间的理解和交流,但是,由于 UML 缺乏精确形式化语义,使得它不利于对软件体系结构进行进一步的分析和验证工作<sup>[2]</sup>。XYZ/E 以线性时序逻辑为基础,具有精确的形式化语义,并且,它在统一的时序逻辑框架下既能表示系统的动态语义又能表其静态语义,因此,XYZ/E 也可以用于软件体系结构的描述,但是它却不够直观<sup>[3]</sup>。基于此,我们可以通过定义 UML 的 XYZ/E 时序逻辑语义,把 UML 与 XYZ/E 结合起来描述软件体系结构,这样,既能便于开发人员之间的理解和交流,也有利于进一步的分析和验证工作。

UML 2.0 顺序图主要描述各个对象为完成某个行为目标而进行的交互,并且它也提供了对交互的重用<sup>[4]</sup>,这些特征使得 UML 2.0 的顺序图适合用于描述软件体系结构的各个组件之间和复合组件内部各个子组件之间的动态交互行为<sup>[5]</sup>。本文在定义 UML 2.0 顺序图语法和语法约束的基础上,通过定义 UML 2.0 顺序图的 XYZ/E 语义,从而为采用 UML 2.0 顺序图与 XYZ/E 相结合的方式描述软件体系结构的动态交互行为奠定了基础。

## 1 UML 2.0 顺序图的语法及语法约束

UML 2.0 顺序图中对象之间的交互仍然是通过传递消

息进行的,其中的消息既可以是同步消息,也可以是异步消息,在本文中,我们只考虑同步调用消息,并把一个顺序图中所有消息的集合记为 M。

**定义 1** 一条消息表示为四元组  $m = (\text{source}, \text{target}, \text{act}, \text{Event})$ , 其中, source 是消息的源对象; obj 是消息的目标对象; act 是消息对应的行为;

$\text{Event} = \{\text{send}(\text{source}), \text{rev}(\text{target})\}$  是跟消息相关联的事件, send(source) 表示源对象发送消息的发送事件, rev(target) 表示目标对象接收消息的接收事件。

UML 2.0 顺序图除了具有 UML 1.X 顺序图的一些基本特征外,增加了交互片段的概念,为分割和重用交互提供了方便。交互片段可以分为一般交互片段和组合片段两种交互形式<sup>[6]</sup>,其中,一般交互片段表示对象为完成某个任务而相互传递的消息序列;组合片段是对一个或多个交互操作数的引用,因此,每个组合片段由一个或多个交互操作数组成,换句话说,每个组合片段包含一个或多个交互片段。由于每个组合片段拥有一个交互操作符,我们可以把组合片段拥有的交互操作符看作是组合片段的类型。UML 2.0 定义了很多交互操作符,文本主要关注以下几种常用的交互操作符: Alt、Opt、Pal、Strict、Seq、Loop、Ignore 和 Break。

设 Basic 为 UML 2.0 顺序图中所有一般交互片段的集合, CombinedFrame 为所有组合片段的集合,下面给出一般交互片段和组合片段的定义。

<sup>\*</sup> 基金项目:国家自然科学基金(批准号:60073020);中国科学院计算机科学国家重点实验室开放课题(批准号:SYSKF0303);江苏省高校自然科学基金项目(批准号:05KJB520119);重庆市科学技术研究项目(合同号:040803)。黄正宝 硕士研究生,主要研究方向为软件体系结构;张广泉 博士,教授,主要研究方向为软件工程、形式化方法。

**定义 2** 一般交互片断  $basic = \{m_1, m_2, \dots, m_n\}$ , 其中, 对于每个  $i (1 \leq i \leq n), m_i \in M$ .

**定义 3** 组合交互片断是一个二元组  $(type, INTER)$ , 其中,  $type \in CFTYPE$ ,  $CFTYPE$  是组合片断的类型的集合, 表示为  $CFTYPE = \{Alt, Opt, Pal, Strict, Seq, Loop, Break\}$ ;  $INTER = \{inter | inter \text{ 是组合片断中包含的交互片断}\}$ . 对于每个组合片断中的  $INTER$ , 它应该满足  $SINTER$  约束.

$$SINTER \equiv_{\text{def}} \forall cf \in \text{CombineFrame} \cdot ((cf.type \in \{Opt, Loop, Ignore\} \wedge (\text{count}(INTER) = 1)) \vee cf.type \in \{Alt, Pal, Strict, Seq\} \wedge (\text{count}(INTER) = n) \wedge (\forall inter \in INTER \cdot (inter \in \text{Basic} \vee inter \in \text{CombinedFrame})))$$

在约束  $SINTER$  中,  $\text{count}(INTER)$  表示组合片断中包含的交互片段的个数,  $n$  满足  $1 < n < \infty$ .

在 UML 2.0 中, 顺序图是消息和组合片段的集合. 从以上定义可以看出, 组合片断中包含的交互片断, 可以是一般交互片断, 也可以是组合片断, 即组合片段的嵌套. 因此, 组合片断内的每个交互片断都可以看成是一个子顺序图.

**定义 4** 顺序图  $sd = \{s | s \in M \vee \text{CombinedFrame}\}$ , 并且满足以下约束:

(1) 消息和组合片断按照在生命线上从上到下的排列顺序执行.

(2) 每条消息一定包含发送事件和接收事件, 并且发送事件一定在接收事件之前发生.

## 2 UML 2.0 顺序图的 XYZ/E 语义

### 2.1 时序逻辑语言 XYZ/E

XYZ/E<sup>[7]</sup> 是一种以线性时序逻辑为基础的形式化程序设计语言, 所有程序单元 (语句、程序块) 都是合式公式. XYZ/E 的基本语言成分是条件元, 有两种形式:

$$LB = y \wedge R \Rightarrow \$O(v_1, \dots, v_k) = (e_1, \dots, e_k) \wedge \$OLB = z \quad (1)$$

$$LB = y \wedge R \Rightarrow @(Q \wedge LB = z) \quad (2)$$

其中“ $\Rightarrow$ ”表示蕴含;  $R$  和  $Q$  表示一阶逻辑公式, 分别称为条件元的条件部分和动作部分;  $LB$  为标号控制变量,  $y$  和  $z$  为具体的标号, 分别表示条件元的定义标号和转出标号; 符号  $@$  为一阶逻辑算子, 可以是下一时刻算子  $\$O$  或最终时刻算子  $\diamond$ . 两种形式的条件元都是时序逻辑公式, 其语义即为该时序逻辑式的语义模型. 形式(1)条件元定义了程序相邻状态之间的转换关系, 用于描述动态语义; 形式(2)条件元表示程序抽象规范, 用于描述静态语义.

单元 (unit) 是一个条件元序列, 具有如下形式:

$$\square [A_1; \dots; A_n]$$

$$\text{WHERE } B_1 \wedge \dots \wedge B_m$$

其中  $A_1, A_2, \dots, A_n$  是条件元;  $B_1 \wedge \dots \wedge B_m$  是时序逻辑公式, 是对单元的约束或某些特别谓词的定义. 符号“;”和保留字 WHERE 等同于逻辑联结词合取. 单元也是时序逻辑公式, 其语义对应于该时序逻辑式的语义模型. 如果一个单元中的所有  $A_i (i = 1, \dots, n)$  都是式(1)的形式, 并且不包含 WHERE 部分, 那么它就构成一个可执行的程序段; 如果都是式(2)的形式, 那么它就是一个程序的抽象规范. 一个单元也可以同时具有(1)和(2)形式的条件元以表达不同程度的抽象性. 另外, 在本文中还将用到 XYZ/E 的以下时序算子和符号:  $\$U$  是“直到”算子,  $A \$UB$  表示  $A$  为真直到  $B$  为真;  $||$

是并行符号, 表示几个进程的并发执行;  $\gg$  是连续符号, 表示几个动作的连续执行;  $\$A$  是 XYZ/E 中的“所有”量词.

### 2.2 语义描述

根据前面的语法定义, UML 2.0 顺序图的组成元素是消息和组合片断. 下面我们在定义 UML 2.0 顺序图组成元素的 XYZ/E 语义的基础之上, 给出 UML 2.0 顺序图的 XYZ/E 语义.

对于 UML 2.0 顺序图中的每一个组成元素, 都有一个名字, 由于每个不同组成元素的名字都是唯一的, 因此, 可以用作 XYZ/E 条件元中的控制标号.

**定义 5** UML 2.0 顺序图的组成元素到 XYZ/E 条件元转换定义为一个映射

$$Ce: \text{EleSet} \rightarrow \text{XYZces}$$

其中  $\text{EleSet}$  表示组成元素的集合,  $\text{XYZces}$  表示 XYZ/E 的条件元集. 基于映射  $Ce$ , 对于每个组成元素  $element$ , 根据不同那个情况, 它的条件元定义如下所示:

(1) 如果  $element \in M$

$$LB = element \wedge \text{send}(\text{source}) \Rightarrow \$O(\text{rev}(\text{target}) \wedge \text{target.act}) \wedge \$OLB = \text{next}$$

其中,  $\text{target.act}$  表示执行此消息目标对象的某个操作. 对于每个  $act$  具体执行过程, 可以用 UML 2.0 活动图及其相应的 XYZ/E 语义来描述, 本文不予考虑. 条件元中的  $\text{next}$  为 XYZ/E 中的一种转出标号, 代表下一个条件元的定义标号, 如果此条件元是它所在单元的最后一个条件元, 则转出标号用 STOP 表示.

(2) 如果  $element \in \text{CombinedFrame} \wedge element.type = \text{Alt}$

$$LB = element \wedge \text{guard}_1 \Rightarrow \$OP1 \wedge \$OLB = \text{next}$$

.....

$$LB = element \wedge \text{guard}_n \Rightarrow \$OPn \wedge \$OLB = \text{next}$$

其中,  $P1, \dots, Pn$  表示此组合片断内的交互片断,  $\text{guard}_1, \dots, \text{guard}_n$  表示执行某个交互片断应该满足的条件, 并且  $(\text{guard}_1 \vee \dots \vee \text{guard}_n = \$T) \wedge (\text{guard}_i \wedge \text{guard}_j = \$F), i, j = 1 \sim N, i \neq j$ .

(3) 如果  $element \in \text{CombinedFrame} \wedge element.type = \text{Opt}$

$$LB = element \wedge \text{guard} \Rightarrow \$OP \wedge \$OLB = \text{next};$$

$$LB = element \wedge \sim \text{guard} \Rightarrow \$OLB = \text{next};$$

其中,  $P$  表示此组合片断内的交互片断,  $\text{guard}$  表示组合片断要包含  $P$  执行应该满足的条件.

(4) 如果  $element \in \text{CombinedFrame} \wedge element.type = \text{Pal}$

$$LB = element \Rightarrow || [P1, \dots, Pn]$$

其中,  $P1, \dots, Pn$  表示此组合片断内的各个交互片断.

(5) 如果  $element \in \text{CombinedFrame} \wedge element.type = \text{Strict}$

$$LB = element \Rightarrow \gg [P1, \dots, Pn]$$

其中,  $P1, \dots, Pn$  表示此组合片断内的各个交互片断.

(6) 如果  $element \in \text{CombinedFrame} \wedge element.type = \text{Seq}$

$$LB = element \Rightarrow || [P1, \dots, Pn]$$

WHERE  $\$A(m_a \in P_i \wedge m_b \in P_j) ((\text{ma.source} == \text{mb.source}) \rightarrow (\text{finished}(m_a) \wedge \$U\text{finished}(m_b)))$

由于 UML 2.0 规范当中,  $\text{Strict}$  操作符的规定语义仅仅适用于交互片断的第一层, 因此我们在  $\text{Strict}$  类型的组合片

断中,不考虑每个交互片断中的嵌套情况。相应的,本式中的  $P_1, \dots, P_n$  只能是一般交互片断。在本式中,  $P_i, P_j \in \{P_1, \dots, P_n\}$  并且  $P_i \neq P_j$ 。finished( $m$ ) 表示消息  $m$  已经执行完毕,从而 finished( $m_a$ )  $\wedge$  Ufinished( $m_b$ ) 表示消息  $m_a$  一定在消息  $m_b$  之前执行完毕。“WHERE”关键字以及由它引出的公式表示对此条件元的约束。当我们在用一个 XYZ/E 单元表示整个顺序图语义时,可以把对各个条件元的约束放在一起,表示对此单元的约束。

(7) 如果  $element \in CombinedFrame \wedge element.type = Loop$

$LB = element \wedge (\min \leq x \leq \max) \Rightarrow \$O(P \wedge x = x + 1) \wedge \$OLB = element$

$LB = element \wedge \sim(\min \leq x \leq \max) \Rightarrow \$OLB = next$

其中 min 和 max 分别表示执行循环要满足的最小值和最大值,  $P$  表示这个组合片断内要执行的交互片断。

(8) 如果  $element \in CombinedFrame \wedge element.type = Break$

$LB = element \Rightarrow \$OLB = EXIT$

定义 6 UML 2.0 顺序图到 XYZ/E 的语义转换是一个映射

$Semantics; SDs \rightarrow XYZUNITs,$

其中,  $SDs$  是 UML 顺序图集,  $XYZUNITs$  是 XYZ/E 单元集。若顺序图  $sd = \{s_1, \dots, s_N\}$ , 则

$Semantics(ad) = \square[Ce(s_1) \wedge \dots \wedge Ce(s_N)] WHERE A_1 \wedge \dots \wedge A_N$

在这里,  $A_1, \dots, A_N$  分别表示对与  $s_1, \dots, s_N$  相应的条件元的约束。

### 3 实例说明

设在某个权限管理系统中,为新建用户添加组的顺序图 sd 如图 1。

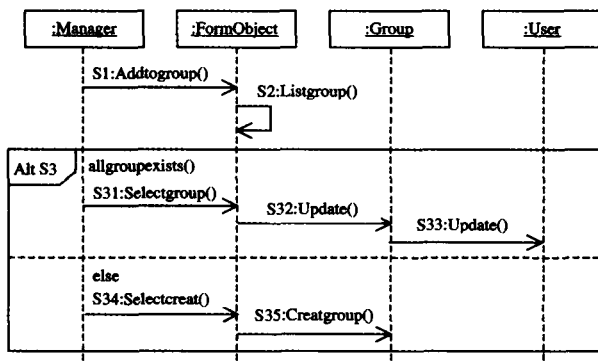


图 1 为新建用户添加组的顺序图

由图可知,  $sd = \{S1, S2, S3\}$

$S1 = (Manager, FormObject, Addtogroup(), \{send(Manager), rev(FormObject)\})$

$S2 = (FormObject, FormObject, Listgroup(), \{send(FormObject), rev(FormObject)\})$

$S3 = (Alt, \{P1, P2\})$

$P1 = \{S31, S32, S33\}$

$P2 = \{S34, S35\}$

$S31 = (Manager, FormObject, Selectgroup(), \{send(Manager), rev(FormObject)\})$

$S32 = (FormObject, Group, Update(), \{send(FormObject), Group(Group)\})$

$S33 = (Group, User, Update(), \{send(Group), rev(User)\})$

$S34 = (Manager, FormObject, Selectcreat(), \{send(Manager), rev(FormObject)\})$

$S35 = (FormObject, Group, Creategroup(), \{send(FormObject), rev(Group)\})$

根据前面的定义,它的 XYZ/E 语义如下:

$Semantics(sd) = \square[LB = START \Rightarrow OLB = S1;$   
 $LB = S1 \wedge send(Manager) \Rightarrow \$O(rev(FormObject) \wedge FormObject.Addtogroup()) \wedge$   
 $\$OLB = S2;$

$LB = S2 \wedge send(FormObject) \Rightarrow \$O(rev(FormObject) \wedge FormObject.Listgroup()) \wedge$   
 $\$OLB = S3;$

$LB = S3 \wedge allgroupexists() \Rightarrow \$OP1 \wedge \$OLB = STOP;$   
 $LB = S3 \wedge \sim allgroupexists() \Rightarrow \$OP2 \wedge \$OLB = STOP;$   
 $] WHERE (P1 = Semantics(P1) \wedge P2 = Semantics(P2))$

$Semantics(P1) = \square[LB = START \Rightarrow OLB = S31;$   
 $LB = S31 \wedge send(Manager) \Rightarrow \$O(rev(FormObject) \wedge FormObject.Selectgroup()) \wedge$   
 $\$OLB = S32;$

$LB = S32 \wedge send(FormObject) \Rightarrow \$O(rev(Group) \wedge Group.Update()) \wedge \$OLB = S33;$   
 $LB = S33 \wedge send(Group) \Rightarrow \$O(rev(User) \wedge User.Update()) \wedge \$OLB = STOP;]$

$Semantics(P2) = \square[LB = START \Rightarrow OLB = S34;$   
 $LB = S34 \wedge send(Manager) \Rightarrow \$O(rev(FormObject) \wedge FormObject.Selectcreat()) \wedge$   
 $\$OLB = S35;$

$LB = S35 \wedge send(FormObject) \Rightarrow \$O((rev(Group) \wedge Group.CreateGroup()) \wedge \$OLB = STOP;]$

结束语 通过定义 UML 的 XYZ/E 时序逻辑语义,把 UML 与形式化方法结合起来描述软件体系结构,可以为软件体系结构描述研究提供一种新的思路。本文基于这种思想,在定义 UML 2.0 顺序图的语法和语法约束的基础上,定义了 UML 2.0 顺序图的 XYZ/E 时序逻辑语义。下一步的工作是定义 UML 2.0 其它视图的 XYZ/E 语义,并在这些工作的基础上,采用 UML 2.0 与 XYZ/E 的这种结合方式描述实例系统的体系结构。

### 参考文献

- 1 Medvidovic N. Modeling Software Architectures in UML. ACM Transactions on Software Engineering and Methodology, 2002, 11(1): 2~57
- 2 戎玫, 张广泉. 形式化与可视化相结合的软件体系结构描述方法研究. 计算机科学, 2005, 3(4): 205~208
- 3 朱雪阳. 软件体系结构形式描述研究: [博士学位论文]. 中国科学院软件研究所, 2004
- 4 Object Management Group. UML 2.0 Superstructure Specification: Final Adopted Specification. <http://www.omg.org/docs/ptc/04-10-02.pdf>
- 5 Ivers J, Clements P, Garlan D. Documenting Component and Connector Views with UML 2.0. Technical Report CMU/SEI-2004-TR-008, Software Engineering Institute, Carnegie Mellon University, 2004
- 6 Storrie H. Semantics of Interactions in UML 2.0. In Proc. IEEE Symp. Visual Languages and Formal Methods (VLFM'03), Auckland, 2003
- 7 唐雅松. 时序逻辑程序设计与软件工程(上、下册). 北京: 科学出版社, 2002