UML 顺序图的形式化描述

Formal Description of UML Sequence Diagram

李景峰[1] 李 琰[2] 陈 平[1]

(西安电子科技大学软件工程研究所 西安710071)[1] (上海贝尔有限公司研发部 上海200092)[2]

Abstract UML, as a standard object modeling language, provides several diagrams to describe the results of system analysis and design. However, its lack of precise semantics makes it difficult to support consistency check and accuracy analysis of the models of large, complex systems. In this paper, a formal description of UML sequence diagram is proposed, which provides the foundation for the rigorous semantic analysis and validation of UML sequence diagram.

Keywords Unified Modeling Language, Sequence diagram, Formalization

1 引言

统一建模语言 UML(Unified Modeling Language)[6]是标准的对象建模语言,它通过定义的多种图和模型元素描述系统分析和设计的结果,主要针对大型、复杂系统的建模。然而,UML 却是半形式化的——其语法结构采用了形式化的规约,但其语义部分则是用自然语言描述的[4,5]。由于复杂系统的建模往往需要进行严格的语义分析,而 UML 却缺乏准确的语义,这使得对模型难以进行一致性检查和正确性分析,进而限制了它的有效性[1,2]。

作为 UML 的一种交互图,顺序图用于描述系统对象之间的交互情况,尽管其表达简单明了,易于理解,但同样存在缺乏准确语义、难以对其进行分析和验证的问题[3]。要解决该问题,首先就要对顺序图进行严格的描述。针对该问题,本文给出了一种 UML 顺序图的形式化描述,它为顺序图的语义分析与验证提供了基础。

2 顺序图的形式化描述

2.1 UML 顺序图简介

顺序图用于描述对象间动态的交互关系,着重体现对象间消息传递的时间顺序。顺序图采用两个轴:水平轴表示不同的实例(其实是类元角色,每个类元角色代表一个特定的对象或一组对象的集合[6],本文中我们称之为实例);垂直轴表示时间。在顺序图中,实例用一个带有垂直虚线的矩形框表示,在矩形框内标有实例名和对应的类名。垂直虚线称为实例的生命线,代表在实例之间的交互作用中该实例的生命期。两个实例生命线间的带有箭头的线表示消息。消息的箭头形状表明了消息的类型是发送还是返回。消息按发生的时间顺序从上到下排列。每个消息旁边标有消息名,也可加上参数。图1是一个指挥员成功登录到指挥控制系统中的顺序图。

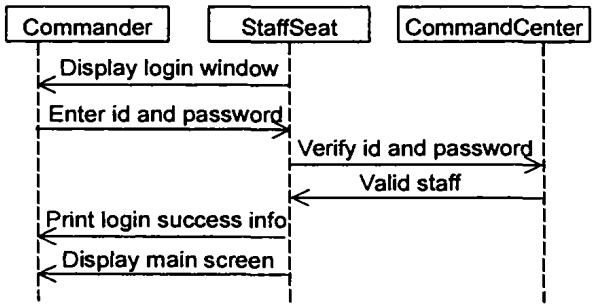

图1 指挥员成功登录到指挥中心的顺序图

2.2 形式化描述

由上所述可以看出一个顺序图是由实例、实例生命线上发送或接受消息的点以及实例间发送的消息构成的。下面,我们分别定义之,并最终给出顺序图的形式化描述。

定义1 设 n 是顺序图 SD 中实例的个数,I_{SD}表示该顺序图中实例的集合,即 $I_{SD}=\{i_1,\cdots,i_n\}$。

定义2 顺序图 SD 中实例 i 的生命线上发送或接收消息的点称为位点,记为$\langle i,l\rangle$。每个实例的生命线上存在着有限个离散的位点与该实例相关联。$L_{SD}(i)$表示实例 i 所有位点的集合,L_{SD}表示顺序图 SD 所有位点的集合:

$L_{SD}(i)=\{0,\cdots,l\_\max_{SD,i}\}$

$L_{SD}=\{\langle i,l\rangle \mid i\in I_{SD}\wedge l\in L_{SD}(i)\}$

在每个位点上可以有一个或多个消息发送给其它实例,或者接收从自身或其它实例发送过来的一条信息。同样也可能向系统环境中发送消息或从其接受消息,如 actor 参与顺序图的情况。

每个消息上标记有消息的名称和对应的参数,其中消息的参数类型应与系统模型中类图给出的操作说明一致。为了区分相同发送实例和接收实例间激活相同操作的两个消息,每个消息还附带有一个唯一的标识号。

定义3 消息标识 $MI=(MN,PL,No)$,其中 MN 是消息名,PL 是消息的参数列表,No 是消息的唯一标识号。

顺序图中可以使用返回标记,它表示从消息处理中返回,而不是一个新消息,所以返回标记与原来的消息具有相同的消息标识。通常情况下它带有一个返回值,并且其类型也与类图中的说明相一致。

定义4 返回标记 $RMI=(MI,V)$,其中 MI 是消息标识,V 是消息的返回值。

定义5 顺序图 SD 中所有消息标识的集合 $\Sigma_{SD}=\{m\_id \mid m\_id=MI\mid RMI\}$。

顺序图中的每个消息与实例生命线的两点相关:消息发送的位点和消息接收的位点。所以我们定义以下函数来描述消息与位点之间的关系。

定义6 函数 λ 描述顺序图中消息与位点之间的对应关系,其中符号>和<分别表示消息的发送和接收:

$\lambda:\Sigma_{SD}\times\{>,<\}\to L_{SD}$。

在某些情况下,实例可以向系统环境发送或从其中接受消息。

定义7 用 E 表示系统环境,函数 δ 和 γ 分别描述消息与

发送或接收该消息的实例或环境之间的对应关系：

$\delta,\gamma:\Sigma_{SD}\rightarrow L_{SD}\cup\{E\}$

其中 $i\in I_{SD},\langle i,l\rangle\in L_{SD},m\_id\in\Sigma_{SD}$

$$\delta:=\begin{cases}i:if\langle i,l\rangle=\lambda(m\_id,>)\\E:else\end{cases}$$

$$\gamma:=\begin{cases}i:if\langle i,l\rangle=\lambda(m\_id,<)\\E:else\end{cases}$$

顺序图中消息是按发生的时间顺序从上到下排列的，所以我们定义这种先后关系如下：

定义8 顺序图 SD 中位点间的有序关系表示为 $\pi,\pi:L_{SD}\times L_{SD}$，其满足：

(1)对于$\forall\ i\in I_{SD},l\in L_{SD}(i)$，有$\langle i,l\rangle\pi\langle i,l+1\rangle$；

(2)对于$\forall\ m_id\in\Sigma_{SD}$，如果$\exists\ \langle i,l\rangle,\langle i',l'\rangle\in L_{SD},\langle i,l\rangle=\lambda(m_id,>)\wedge\langle i',l'\rangle=\lambda(m_id,<)$成立，那么$\langle i,l\rangle\pi\langle i',l'\rangle$，表明消息接收总在发送该消息之后；

(3)对于$\forall\ m_id,m_id'\in\Sigma_{SD}$，如果$\exists\ \langle i,l\rangle,\langle i',l'\rangle,\langle i_1,l_1\rangle,\langle i'_1,l'_1\rangle\in L_{SD},\langle i,l\rangle=\delta(m_id)\wedge\langle i',l'\rangle=\gamma(m_id)\wedge\langle i_1,l_1\rangle=\delta(m_id')\wedge\langle i'_1,l'_1\rangle=\gamma(m_id')\wedge i=i_1\wedge\langle i,l\rangle\pi\langle i_1,l_1\rangle$成立，那么$\langle i',l'\rangle\pi\langle i_1,l_1\rangle$；

(4)对于$\forall\ \langle i,l\rangle,\langle i',l'\rangle,\langle i'',l''\rangle\in L_{SD}$，如果有，$\langle i,l\rangle\pi\langle i',l'\rangle\wedge\langle i',l'\rangle\pi\langle i'',l''\rangle$，那么$\langle i,l\rangle\pi\langle i'',l''\rangle$，即由(1)、(2)、(3)产生的传递闭包。

定义9 函数λ满足以下一致性约束：

(1)每个消息在顺序图中是唯一的

$\forall\ \langle i,l\rangle,\langle i',l'\rangle\in L_{SD},m\_id\in\Sigma_{SD},x\in\{>,<\}:\langle i,l\rangle=\lambda(m\_id,x)\wedge\langle i',l'\rangle=\lambda(m\_id,x)\Rightarrow\langle i,l\rangle=\langle i',l'\rangle$

(2)在每个位点，不允许接收多于一条消息或接收一条消息的同时发送另一条消息

$\forall\ \langle i,l\rangle\in L_{SD},x\in\{>,<\},m\_id,m\_id'\in\Sigma_{SD}:\langle i,l\rangle=\lambda(m\_id,x)\wedge\langle i,l\rangle=\lambda(m\_id',x)\Rightarrow x\Rightarrow$

(3)实例 i 只将消息的返回值 v 发回给发送该消息的实例

$\forall\ \langle i,l\rangle\in L_{SD},m\_id\in\Sigma_{SD},v\in V:\langle i,l\rangle=\lambda((m\_id,v),>)\Rightarrow\exists\ l'\in L_{SD}(i),\langle i,l'\rangle=\lambda(m\_id,<)\wedge\langle i,l'\rangle\pi\langle i,l\rangle\wedge\delta(m\_id)=\gamma((m\_id,v))$

(4)每个消息最多只允许有一个返回值

$\forall\ m\_id\in\Sigma_{SD},v,v'\in V:\delta((m\_id,v))=\delta((m\_id,v'))\wedge\gamma((m\_id,v))=\gamma((m\_id,v'))\Rightarrow v=v'$

定义10 顺序图 SD 可由以下四元组表示：$SD=(I_{SD},L_{SD},\Sigma_{SD},\lambda)$。

3 实例

在此部分，我们给出用上述方法描述图1所示顺序图的结果。为了清楚起见，我们用虚线的小圆圈对图1进行了位点标记，如图2所示。同时为了书写方便，分别用 c、ss、cc 代替 commander、StaffSeat 和 CommandCenter。

$SD=(I_{SD},L_{SD},\Sigma_{SD},\lambda)$

$I_{SD}=\{c,ss,cc\}$

$$L_{SD}=\begin{Bmatrix}\langle c,1\rangle,\langle c,2\rangle,\langle c,3\rangle,\langle c,4\rangle,\langle ss,1\rangle,\langle ss,2\rangle,\\\langle ss,3\rangle,\langle ss,4\rangle,\langle ss,5\rangle,\langle ss,6\rangle,\langle cc,1\rangle,\langle cc,2\rangle\end{Bmatrix}$$

$$\Sigma_{SD}=\begin{Bmatrix}(displayloginwindow,(\ ),0),(enteridpassword,(\ ),1),\\(printlogsuccinfo,(\ ),2),\\(displaymainscreen,(\ ),3),(verifyidpassword,(\ ),4),\\(validstaff,(\ ),5)\end{Bmatrix}$$

$\lambda((displayloginwindow,(\ ),0),>)=\langle ss,1\rangle$

$\lambda((displayloginwindow,(\ ),0),<)=\langle c,1\rangle$

$\lambda((enteridpassword,(\ ),1),>)=\langle c,2\rangle$

$\lambda((enteridpassword,(\ ),1),<)=\langle ss,2\rangle$

$\lambda((printlogsuccinfo,(\ ),2),>)=\langle ss,5\rangle$

$\lambda((printlogsuccinfo,(\ ),2),<)=\langle c,3\rangle$

$\lambda((displaymainscreen,(\ ),3),>)=\langle ss,6\rangle$

$\lambda((displaymainscreen,(\ ),3),<)=\langle c,4\rangle$

$\lambda((verifyidpassword,(\ ),4,,>)=\langle ss,3\rangle$

$\lambda((verifyidpassword,(\ ),4),,<)=\langle cc,1\rangle$

$\lambda((validstaff,(\ ),5),>)=\langle cc,2\rangle$

$\lambda((validstaff,(\ ),5),<)=\langle ss,4\rangle$

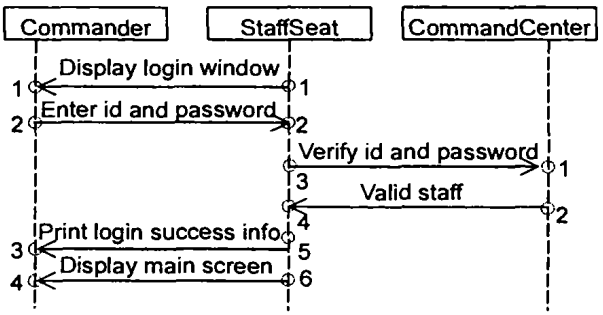

图2 对图1进行位点标记后的顺序图

结束语 作为标准的对象建模语言，UML 提供了多种模型图来描述系统分析和设计的结果。然而，UML 缺乏准确的语义，这种半形式化的特征使其难以支持对复杂的系统模型进行一致性检查和正确性分析。本文给出了 UML 顺序图的一种形式化描述，有助于对顺序图进行严格的语义分析，并为对其进行一致性检查与验证提供了基础。

参 考 文 献

1 Bruel J-M, France R B. Transforming UML Models to Formal Specifications. In: Proc. of the Int. Conf. on Object Oriented Programming Systems Language and Applications (OOPSLA'98) Vancouver, Canada, Oct. 1998

2 Breu R, Hinkel U, et al. Towards a Formalization of the Unified Modeling Language. In: Aksit M, Matsuoka S, eds. Proc. of ECOOP'97-Object Oriented Programming, 11th European Conf. Jyväskylä, Finland, June Springer-Verlag, LNCS 1241, 1997

3 Damm W, Harel D. LSCs: Breathing Life into Message Sequence Charts. In FMOODS'99 IFIP TC6/WG6.1 3rd Intl. Conf. on Formal Methods for Open Object-Based Distributed Systems, 1999

4 邵维忠，梅宏. 统一建模语言 UML 述评. 计算机研究与发展，1999, 36(4): 385～394

5 刘超，张莉. 可视化面向对象建模技术——标准建模语言 UML 教程. 北京航空航天出版社，1999.7

6 Object Management Group(OMG). OMG Unified Modeling Language, Specification - version 1.3, March 2000