

用加标的基本网系统设计群件编辑器¹

王生原 喻坚 袁崇义

Abstract. 本文的内容一方面是介绍 STLEN 设计规范的技术要点, 另一方面是专门针对一个较典型的实例——群件编辑器的设计。前者的基础是一种对 S-元素和 T-元素都加标的基本网系统模型 STLEN (ST-Labeled EN-system), 后者是对象 Petri 网研究领域的一个通用实例。

Key Words: Petri 网; 面向对象; 动态对象模型; 群件

Using Labeled EN-system for the Design of Groupware Editors

Abstract. This paper focuses on two aspects: (1) the essential techniques of the STLEN specification; (2) a design of groupware editors using the STLEN specification. The former is based on a net model STLEN (ST-Labeled EN-system) which is the extension of an EN-system with both S-elements and T-elements labeled, and the later is a typical example in the Object Petri Net literature.

Key Words: Petri Nets; Object-Oriented; Dynamic Object Model; Groupware

1. 引言

近年来, 将 Petri 网与对象技术结合起来已成为广泛关注的研究课题。Petri 网是并行性, 并发性, 同步等方面强有力的建模工具 ([1]), 有许多基于 Petri 网的动态行为分析技术和分析工具, 可以弥补现有对象技术动态建模能力的不足。其次, 二者结合也会提升 Petri 网建模技术。利用封装可改善 Petri 网的组合性, 利用继承和聚合等对象设计技术可改善 Petri 网的可重用性。

尽管人们在这方面已经做了许多很有影响的工作 ([2]-[9]), 但目前的现状是: (1) 过多集中于对象 Petri 网模型本身, 缺乏实用建模技术的研究, 特别是工程化方法的研究; (2) 一些网模型适合于软件开发的较高层次 ([5][7][8]), 而另一些则适合于较低层次 ([3][4][6]), 至今仍缺少能够在一致的概念框架下从一个开发层次演进到另一个开发层次的 (基于网的) 形式化方法。

我们近期的工作是寻求建立一个基于网的、多层次的、适合于分布 (并发) 对象计算的软件开发框架 (参见本文第 5 节)。目前涉及到两个层次的网模型, 本文将介绍的加标的基本网系统 STLEN (ST-Labeled EN-system) 是其中之一。在这个层次上, 我们采用 STLEN 设计规范 (见第 3 节) 作为动态对象模型的设计工具。

本文一方面的内容是介绍 STLEN 设计规范的技术要点, 另一方面是专门针对一个较有挑战性的实例——群件编辑器 (groupware editors) 或协同编辑器 (cooperative editors) 的设计。群件编辑器的例子在 1995 年的一个 workshop ([10]) 上被采纳之后, 对象 Petri 网技术的研究者一直将其作为展现各自研究成果的一个通用实例, 但本文为方便讨论对其作了少量的简化 (见第 2 节)。

¹ Supported by the National Natural Science Foundation of China under grant No. 69973003, and by the China NKBRFSF (973) under grant G1999032706.

第 2 节是对群件编辑器的需求建立静态对象模型，第 3、4 节是本文核心内容——针对群件编辑器的设计问题介绍 STLEN 设计规范的技术要点，第 5 节是我们近期的工作或工作计划的简介。

2. 群件编辑器需求分析

2.1 需求描述

该软件系统允许层次图 (hierarchical diagrams) 的协同编辑。层次图可以是面向对象设计的文档资料，或是硬件逻辑设计图，或是 Petri 网图形，等等。问题的一个关键方面是此编辑器要支持多个用户 (Users)，他们处于不同的工作站，协同设计同一个图形。按照 CSCW (Computer Supported Cooperative Work) 的术语，这样的工具当称之为同步群件 (synchronous groupware)：对于同一图形，每一用户可以有在设计时对其特别定制的视图，看到图形的不同部分，或查看到不同的细节。问题的另一个关键方面是此编辑器要支持层次图，即图形的构件 (components) 能够被点开，展示出其子构件 (subcomponents)。

以下是一个简单的用于用户间交互的协同操作协议：

- ✧ 用户可以任意加入 (join) 或离开 (leave) 一个编辑会晤 (editing session)，并可以不同级别的编辑权限 (editing privilege) 加入 (这里我们不把 privilege 当作优先级对待，因此所讨论的编辑系统是非抢占式的)。例如，某一用户加入时只想看图，而另一用户或许还要对图形进行编辑。当前已加入某个编辑会晤的成员对所有用户是可见的，包括其编辑权限。为简化讨论，我们假定用户不能同时加入两个 (或以上) 的会晤。
- ✧ 图形构件可以被或者不被一个用户拥有。应支持不同级别的拥有权 (ownership)，包括 deletion (删除) 拥有权，encapsulation (封装) 拥有权，modification (修改) 拥有权，以及 inspection (查看) 拥有权。(拥有权须同用户的编辑权限相兼容。)
- ✧ Deletion 拥有权要求别的用户不能对该构件有任何拥有权，即便是 inspection 拥有权。
- ✧ Encapsulation 拥有权可以要求展示只有拥有者可以看到该构件的内部细节，而其他用户只能看到该构件的顶层 (top level) 或接口 (interface)。
- ✧ Modification 拥有权允许用户修改该构件的属性，但不允许删除该构件。
- ✧ Inspection 拥有权只允许用户查看该构件的属性。
- ✧ 所有的拥有权在切换到不同的编辑会晤时必须缴还。
- ✧ 只要简单地选中 (selecting) 一个构件，就获得了对该构件的 inspection 拥有权。欲获得其它的拥有权，首先要选中该构件，然后再辅之以适当的命令。同样为简化讨论，我们也假定用户不能同时取得对两个 (或以上) 构件的拥有权。
- ✧ 构件有什么级别的拥有权及其拥有者的标识对其他用户都是可见的。

2.2 对象模型

完整的系统设计通常应考虑三类模型：对象 (object) 模型、视图 (view) 模型及控制 (control) 模型。限于篇幅，本文仅涉及协同服务协议服务 (service) 的逻辑，因此只讨论群件编辑器的对象模型。此外，图形的编辑问题也不属本文的范围。

关于面向对象分析与设计方法的内容十分丰富，本文不必赘述。对象模型有分静态模型和动态模型之分。我们认为 Petri 网非常适合于动态对象模型的建立，而现有的面向对象开发方法恰好缺乏适合于建立动态模型的形式化方法。另外，本文也不区分分析对象模型 (AOM) 与设计对象模型 (DOM)。

对于群件编辑器的协同服务协议服务，本文用以下的图 2-1 和图 2-2 表示其静态对象模型。图 2-1 表示所涉及的类 (对象)，每一个类 (对象) 包含名字 (见上部)、属性 (见中部)

及方法（见下部），这里需注意子类中隐含的从父类继承的属性和方法没有列出。此外，对每个类（对象）都应包含的方法，如各种（有参或无参的）构造函数、各属性的 set 和 get 方法等均未列出。假定 String、Boolean 等常用类（类型）可直接使用。BBuffer 为一个有界缓冲区的模板，元素类型可自定义。

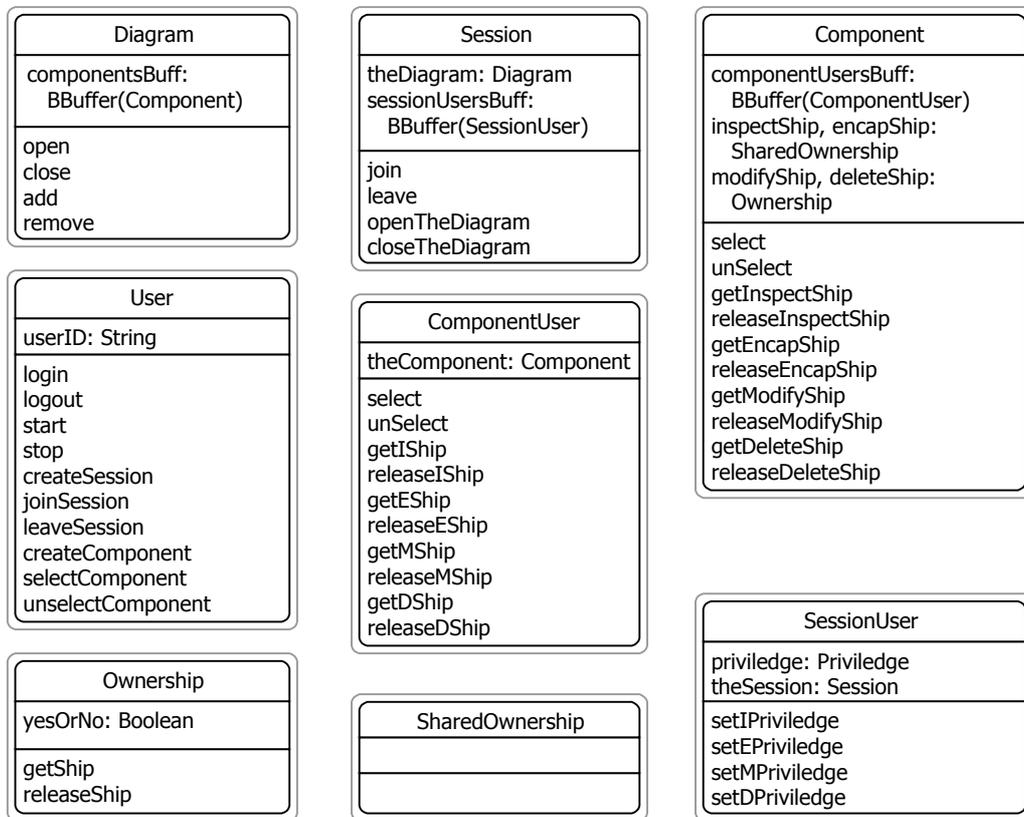


图 2-1

图 2-2 表示类（对象）之间的层次（inheritance, aggregation）及关联（association）关系。对于不熟悉 UML 记号的读者，这里略加说明。Diagram 聚合（aggregation）了零或多个 Component；同时 Component 继承（inheritance）Diagram，也是一种 Diagram。SessionUser 是 User 的子类（inheritance 关系），ComponentUser 又是 SessionUser 的子类。一个 Component

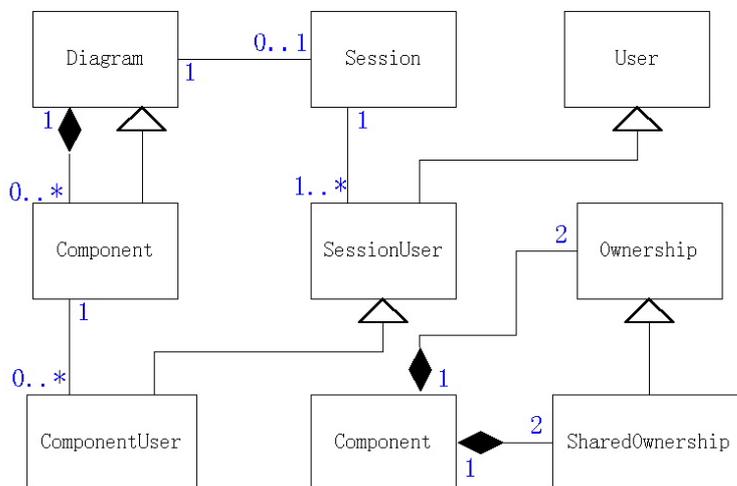


图 2-2

对象聚合了两个 Ownership 对象和两个 SharedOwnership 对象。Diagram 与 Session 之间的 association 关系是一对零或一,即一个 Session 对象对应有一个 Diagram 对象,但一个 Diagram 对象可以有一个也可以没有 Session 对象与之对应。Session 和 SessionUser 之间具有一对多的 association 关系。其余的关系可类推。

3. STLEN 设计规范概要

STLEN 设计规范包括两个部分: 规范的说明以及 STLEN 系统。

3.1 STLEN 系统

STLEN 系统是在 EN-系统(基本网系统)的基础上对 S-元素以及 T-元素加标后重新定义的网系统模型。加标的结果是将元素分成了组(group)。特殊标记“λ”专用于加标外界不可观察的元素。

在 STLEN 设计规范中, STLEN 系统负责刻画类(对象)的动态行为。外界可观察的元素组代表类(对象)的接口(或称 public 的属性和方法)。S-元素的组对应一个或多个 public 的属性(通常是一个,但有时多个属性是相关联的,用来刻画同一类的状态变化); T-元素的组对应一个 public 的方法。

STLEN 系统的变迁规则与 EN-系统相同。但考虑到加标的因素,应将可观察的情态转移以及可观察的变迁序列等新的概念补充进来。附录 A 给出了 STLEN 系统及其相关概念的形式定义。对 STLEN 系统更多的讨论见[11]。

3.2 规范说明部分

STLEN 设计规范的说明部分包括类(对象)名, S-说明和 T-说明。类(对象)名应当与静态对象模型中相应的类(对象)一致。S-说明和 T-说明分别显式地给出本规范 STLEN 系统中外界可观察的 S-元素组以及 T-元素组的说明信息。

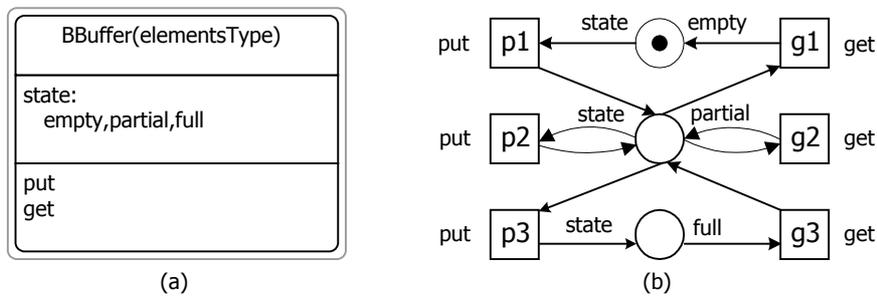


图 3-1

图 3-1 是有界缓冲区类模板 Bbuffer (elementsType) 的一个 STLEN 设计规范, (b) 为其 STLEN 系统, (a)为其说明部分。S-说明中含一个 S-元素组 state, 包含三个 S-元素 empty, partial 和 full。T-说明中含两个 T-元素组 put 和 get, 不必指出包含哪些 T-元素。在 STLEN 系统的图示中, S-元素组的组名(如图 3-1 (b) 中的 state)一般出现在左部(或上部), 而组内的 S-元素名(如图 3-1 (b) 中的 empty)出现在右部(或下部)。T-元素组内的 T-元素名一般出现在变迁符内部(但通常不必标出,因为它们不在体现外界可观察接口的规范说明部分,如图 3-1 (b) 中 p1 等不出现在图 3-1 (a) 的 T-说明中)。对于被加标为“λ”的 S-元素和 T-元素, 其名字都不在图示中标出。

一个类(对象)的 S-说明和 T-说明还包括与其它类(对象)的关系信息。如子类的继承中是否定义 S-元素组以及 T-元素组的别名, 聚合及关联的类(对象)中 S-元素(可以不是整个组)和 T-元素组的提升等问题。这些都是 STLEN 设计规范必要的技术手段, 将在第

4 结中结合实例加以解释。

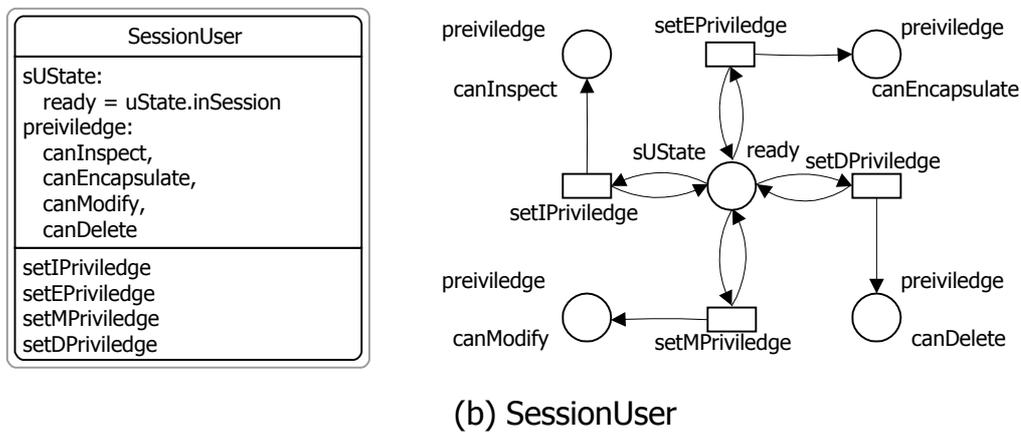
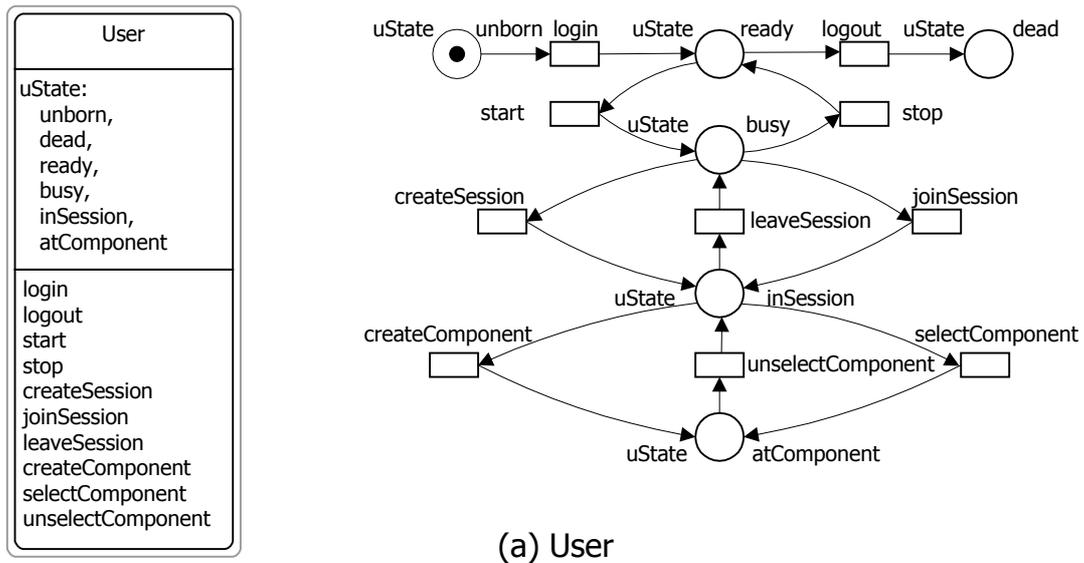


图 4-1

4. 群件编辑器的 STLEN 设计规范

本节讨论群件编辑器的 STLEN 设计规范问题，主要面对第 2.1 节所叙述的协同操作协议服务。由于相关的对象模型已在第 2.2 节论述，读者已有整体印象，所以本节的组织不是按照原来的次序，而是按照面向对象设计的几个主要方面分类进行叙述，这有益于有针对性地解释 STLEN 设计规范的主要技术特征。

4.1 继承

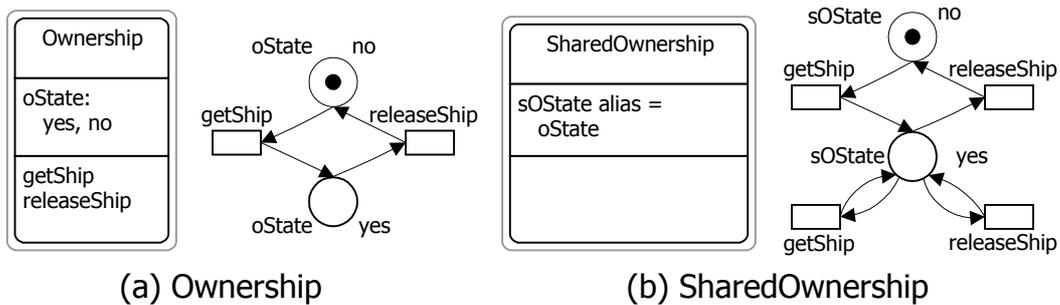


图 4-2

从图 2-2 可知 SessionUser 继承 User 以及 SharedOwnership 继承 Ownership, 图 4-1 及图 4-2 分别为这些类的 STLEN 设计规范。在这里, 以下几点应特别注意:

- (1) 子类的 S-说明, T-说明以及 STLEN 系统继承了其父类相应的部分, 但除非特殊情况, 一般被继承的内容不在子类的规范中重复出现; 如图 4-2 (b) 中, SharedOwnership 中也包含 S-元素组 oState 以及 T-元素组 getShip 和 releaseShip。
- (2) 在子类中, 为了增加规范的可理解性, 可以对被继承的 S-元素组和 T-元素组使用别名 (alias); 如图 4-2 (b) 中, S-元素组 sOState 是 S-元素组 oState 的别名。
- (3) 多个 S-元素组中的 S-元素可以相交, 此时需用 “=” 将它们联系起来 (通常是后定义者显式地表达这种联系); 如图 4-1 (b) 中, S-元素组 sUState 的 S-元素 ready 与从继承父类的 S-元素组 uState 的 S-元素 inSession 是同一个 S-元素。
- (4) 以可读性为原则, 根据不同的情形, 子类的 STLEN 系统的图示可以省略或部分省略从其父类继承的部分; 如图 4-1, 子类的 STLEN 系统的图示中省略了大多数从其父类继承的部分 (除了 S-元素组 sUState 的 ready 与 S-元素组 uState 的 inSession 是相交 S-元素的外); 而图 4-2 中, 子类的 STLEN 系统的图示没有省略从其父类继承的任何部分。

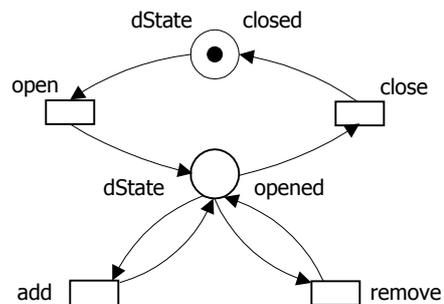
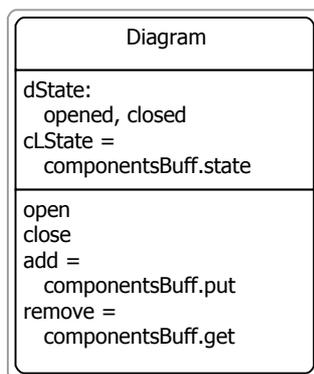


图 4-3

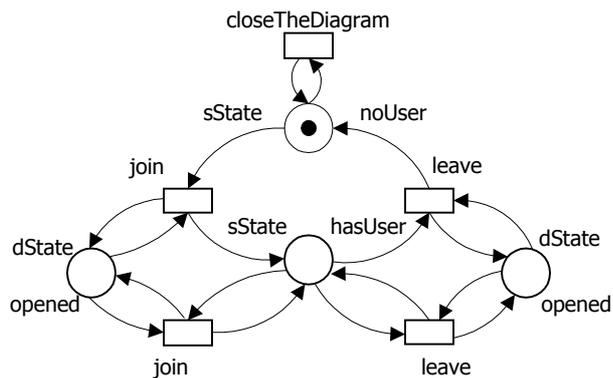
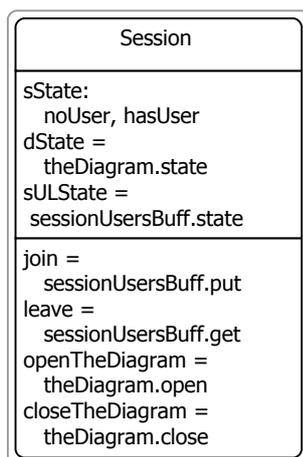


图 4-4

4.2 聚合

聚合关系是一种整体与部分的关系。一个 Diagram 中包含了多个(包括零个)Component, 在 Diagram 的对象模型中用一个 BBuffer (Component) 类型的实例变量 componentBuff 来体现这种聚合 (aggregation) 关系。如图 4-3, 对于 componentBuff 所代表的实例对象, 其行为 (如同模板 Bbuffer 所定义的那样) 在 Diagram 的 STLEN 设计规范中体现为 (1) S-说明

中的 S-元素组 cLState; (2) T-说明中的 T-元素组 add 和 remove; (3) STLEN 系统中 add 与 remove 同其它部分的联系 (注: 如图 4-2 中所述的继承的情形, STLEN 系统的图示中聚合的部分也可大部分省略, 通常只留下同其余部分相连接之处)。

应注意的一点是对于所聚合部分 (子部分) 的一些接口 (外界可观察的元素组) 是否要进行提升, 即是否也作为主体对象的接口。如图 4-3 提升了 componentsBuff 的接口, Diagram 的接口部分包含了 cLState 及 add 和 remove。

4.3 关联

在创建一个 Session 时一定要选择一个 Diagram 与之对应, 因此在类 Session 的设计对象模型 (如图 4-4) 中引入一个 Diagram 类型的实例变量 theDiagram 来体现这种关联 (association) 关系。注意在图 4-4 中, 被关联的 Diagram 的部分接口在 Session 中也进行了提升 (由 dState、openTheDiagram 和 closeTheDiagram 定义)。与第 4.2 节的情形相比, 被关联对象的 STLEN 系统的图示不应在主体 STLEN 系统的图示中出现, 但连接之处例外。

实现关联与实现聚合的方法很相似, 因此在概念上易混淆。聚合的对象是原主体对象的一部分, 而关联的对象相对主体对象而言是独立存在的。设计者对这一点应保持清醒。如 Diagram 中的 componentsBuff 和 Session 中的 sessionUsersBuff 在设计规范中看不出有什么本质的差别, 但前者体现聚合, 而后者体现关联。

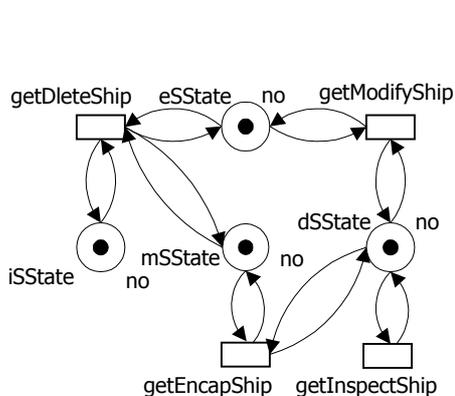
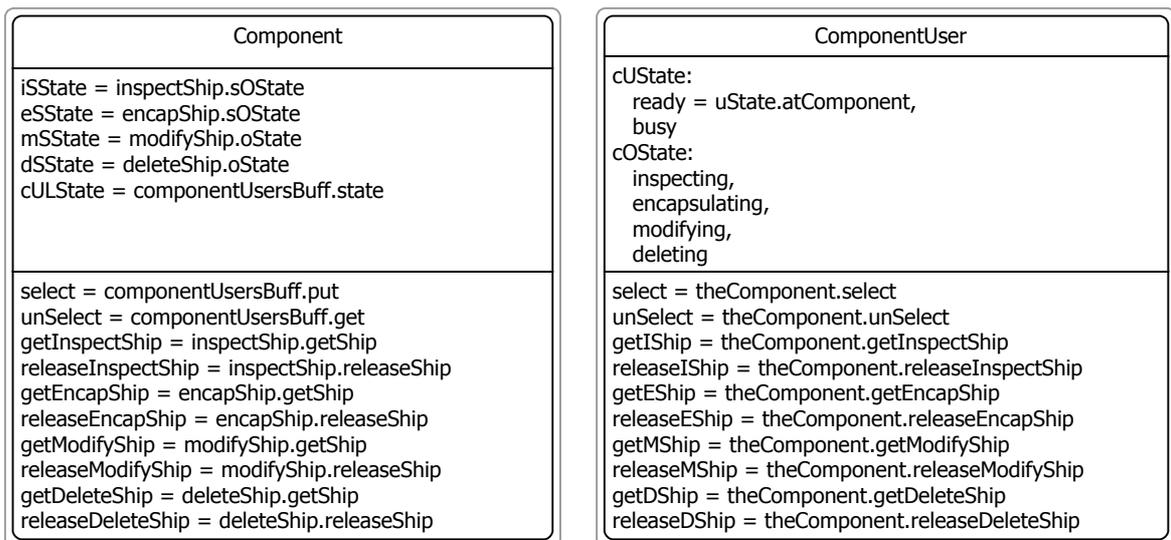


图 4-5

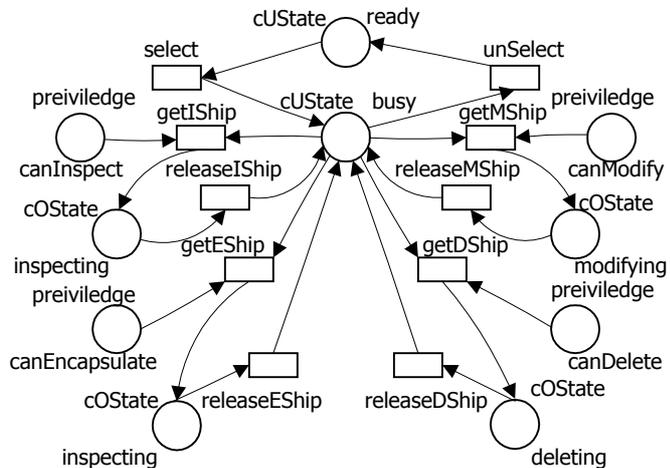


图 4-6

最后, Component 和 ComponentUser 的 STLEN 设计规范分别在图 4-5 和图 4-6 中给出。Component 聚合了两个 Ownership 对象 (由 modifyShip 和 deleteShip 体现) 和两个 SharedOwnership 对象 (由 inspectShip 和 encapShip 体现); 而 Component 中的实例变量 componentUsersBuff 体现从 Component 到 ComponentUser 的一对多的关联关系。ComponentUser 继承 SessionUser, 而 SessionUser 又继承是 User, 因此图 4-6 中的 STLEN 系统的图示只是整个系统的一部分, 由 S-元素组 cUState 的 ready 与其余部分相连。图 4-6 还表明 ComponentUser 的 STLEN 系统与和它关联的 Component 对象(theComponent 所对应的实例) 之间依靠 T-元素组相连, 并提升这些 T-元素组作为接口, 但没有提升其 S-元素组。

4. 4 其它

前面几小节涉及到了在 STLEN 设计规范中, 继承、聚合及关联等几个关键的面向对象设计问题的技术要点。其它方面的设计问题, 如链接 (link)、多对多关联、多继承性、非渐增式继承等在本文的群件编辑器设计中没有体现出来。读者也不难将以上技术加以推广, 解决其它的设计问题。但对于非渐增式继承, 由于牵涉到修改地继承, 因此需对规范的说明部分增加一些关键字, 限于篇幅, 本文不讨论这部分内容。另外, 由于 STLEN 设计规范中既没有针对变量的操作, 也没有显式的消息发送, 所以多态性 (polymorphism) 无从体现, 但其定义的继承关系可以演进到抽象层次较低的规范 (参见下一节), 届时可利用多态性增加可读性和可维护性。

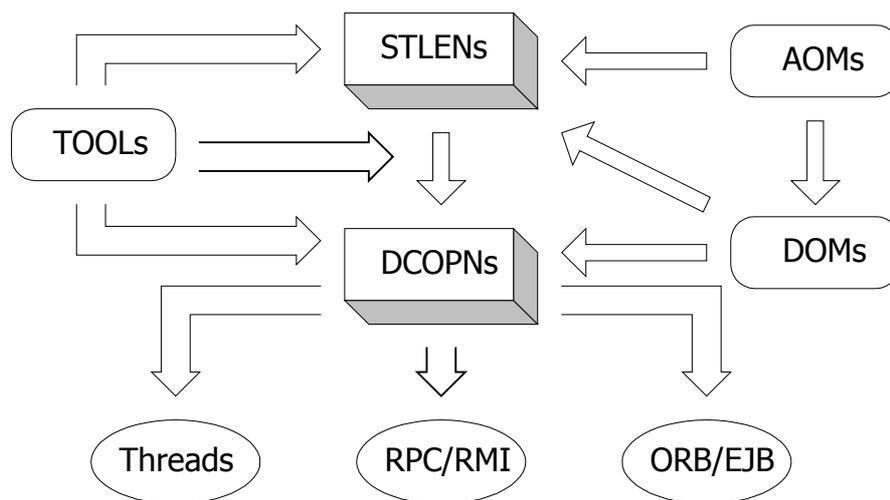


图 5-1

5. 相关的工作

5. 1 演进到 DCOPN 设计规范

STLEN 设计规范的抽象层次较高, 不能刻画详细设计阶段的一些细节, 如数据类型、判定条件以及变量的操作、显式的消息发送等。另一方面, 系统实质上还是静态结构的网系统, 难以描述对象的动态演化行为。

DCOPN (Dynamically Configured Object Petri Nets[DCOPN]) 设计规范是我们已在研究的一种较低层次的对象 Petri 网设计方法。与 STLEN 设计规范相比, DCOPN 设计规范可以弥补前者的不足, 更接近于程序设计语言, 便于同现有的并发 (分布式) 面向对象设计平台 (如 CORBA, COM, J2EE 等) 接轨。但众所周知的是, 此类对象 Petri 网系统的分析技术尚不成熟。

我们目前的一项研究是将 STLEN 设计规范演进到 DCOPN 设计规范。这种演进能使

DCOPN 设计规范保持 STLEN 设计规范的行为语义, 参见[11][12]。

5.2 工作展望

本文结合群件编辑器介绍了 STLEN 设计规范的特征。

图 5-1 描述了我们研究工作的一个较完整的框架。当前的研究主要集中在 STLEN 设计规范层和 DCOPN 设计规范层以及从前者到后者的演进[11]。进一步的工作包括(1)相关工具的设计与开发;(2)与当前流行的分布式(并发)软件系统平台的结合。

参考文献

- [1] 袁崇义, Petri 网原理, 电子工业出版社, 北京, 1997.
- [2] Charles Lakos. From Coloured Petri Nets to Object Petri Nets. Proceedings of 16th Int. Conference on the Application and Theory of Petri Nets, LNCS 935, Turin, Italy, Springer-Verlag, 1995.
- [3] E.Batiston, A.Chizzoni, Fiorella De Cindo. Inheritance and Concurrency in CLOWN. Proceedings of the "Application and Theory of Petri Net 1995" workshop on "object-oriented programs and models concurrency", Torino, Italy 1995.
- [4] C.Sibertin-Blanc. Cooperative Nets. Proceedings of 15th Int. Conference on the Application and Theory of Petri Nets. LNCS 815, Zaragoza, Spain, Springer-Verlag, 1994.
- [5] D.Buchs and N.Guelfi. CO-OPN: A Concurrent Object Oriented Petri Net Approach. Proceedings of 12th Int. Conference on the Application and Theory of Petri Nets, LNCS 524, Gjern, Denmark, 1991.
- [6] R.Valk. Petri Nets as Token Objects — An Introduction to Elementary Object Nets. Proceedings of 19th Int. Conference on the Application and Theory of Petri Nets, LNCS 1420, Springer-Verlag, 1998..
- [7] U.Becker and D.Moldt. Object-oriented Concepts for Coloured Petri Nets. Proceedings of IEEE Int. Conference on System, Man and Cybernetics, vol. 3, 1993, pp 279-286.
- [8] A.Newman, S.M.Shatz, and X.Xie. An Approach to Object System Modeling by State-Based Object Petri Nets. Journal of Circuits, Systems and Computers, Vol.8, No.1(1998) 1-20.
- [9] W.M.P. Vander Aalst and J.Basten. Life-Cycle Inheritance: A Petri-Net-Based Approach. 18th Int. Conference on the Application and Theory of Petri Nets, LNCS1248, Toulouse, France, Springer-Verlag, 1997.
- [10] In Proceedings of the "Application and Theory of Petri Net 1995" workshop on "object-oriented programs and models concurrency", Torino, Italy 1995.
- [11] 王生原, 并发面向对象建模的 Petri 网技术, 北京大学计算机科学技术系博士论文, 2001.
- [12] Sheng-Yuan Wang, Jian Yu and Chong-Yi Yuan. A Net-based Multi-Tier Behaviour Inheritance Modeling Method. To appear in one of the Lecture Notes in Computer Science series, published by Springer-Verlag, 2001.

附录 A: STLEN 系统的形式定义

定义 A.1 网 (Net) 是一个三元组 $N = (S, T; F)$, 这里

(1) S 和 T 是集合, 满足: $S \cap T = \emptyset$ and $S \cup T \neq \emptyset$.

(2) $F \subseteq S \times T \cup T \times S$ 满足: $\text{dom}(F) \cup \text{cod}(F) = S \cup T$, 这里

$\text{dom}(F) = \{x \mid \exists y: (x, y) \in F\}$, and $\text{cod}(F) = \{y \mid \exists x: (x, y) \in F\}$,

S 是 S 元素的集合 (*set of S-elements*). T 是 T 元素的集合 (*set of T-elements*). F 是流关系 (*flow relation*).

对于 $x \in S \cup T$, $\bullet x = \{y \mid (y, x) \in F\}$ 称为 x 的前集 (*the set of pre-elements of x*), $x \bullet = \{y \mid (x, y) \in F\}$ 称为 x 的后集 (*the set of post-elements of x*). 以及对于 $X \subseteq S \cup T$, $\bullet X = \cup \{\bullet a \mid a \in X\}$, $X \bullet = \cup \{a \bullet \mid a \in X\}$.

定义 A.2 EN-系统 (*Elementary Net System*,) 是一个四元组 $N = (B, E; F, c_{in})$, 其中 $(B, E; F)$ 是一个网, $c_{in} \subseteq B$ 称为**初始情态** (*initial case*) .

定义 A.3 设 $N = (B, E; F)$ 是一个网, $u \subseteq E \wedge u \neq \emptyset$, 以及**情态** $c \subseteq B$. 称 u 是一个在情态 c 的**点火步** (*step enabled at c*), 记为 $c[u>$, 当且仅当 $\text{Ind}(u) \wedge \bullet u \subseteq c \wedge u \bullet \cap c = \emptyset$, 这里

$$\text{Ind}(u): \Leftrightarrow \forall e_1, e_2 \in u [e_1 \neq e_2 \Rightarrow (\bullet e_1 \cup e_1 \bullet) \cap (\bullet e_2 \cup e_2 \bullet) = \emptyset].$$

设 $c, c' \subseteq B$, 称 u 为从 c 导致 c' 的**点火步** (*step leading from c to c'*), 记为 $c[u>c'$, 当且仅当 $c[u> \wedge c' = (c - \bullet u) \cup u \bullet$.

定义 A.4 设 $N = (B, E; F, c_{in})$ 是一个 EN-系统. N 的**情态集** (*set of cases*), 记为 C_N , 是满足以下条件的最小集合:

- (1) $c_{in} \in C_N$;
- (2) $\forall c \in C_N \forall c' \subseteq B \forall u \subseteq E (c[u>c' \Rightarrow c' \in C_N)$.

N 的**点火步集** (*set of steps*) 为 $U_N = \{u \mid u \subseteq E \wedge \exists c \in C_N (c[u>)\}$.

设 $c, c' \in C_N$, $\omega = u_1 u_2 \dots u_k$ ($u_1, u_2, \dots, u_k \in U_N$) 称为从 c 导致 c' 的**点火步序列** (*step sequence leading from c to c'*), 记为 $c[\omega>c'$, 当且仅当 $c[u_1>c_1, c_1[u_2>c_2, \dots, c_{k-1}[u_k>c'$.

定义 A.5 ST 加标的 EN-系统 (*ST-Labeled EN system*), 简称 **STLEN 系统**, 是一个二元组 $\Sigma = (N, \beta)$, 其中

- (1) $N = (B, E; F, c_{in})$ 是 EN-系统.
- (2) $\beta: B \cup E \rightarrow L \cup \{\lambda\}$ 是满足以下条件的**加标函数** (*labeling function*):

$$\forall b \in B \forall e \in E [\beta(b) \neq \lambda \vee \beta(e) \neq \lambda \rightarrow \beta(b) \neq \beta(e)],$$

这里 L 代表名字空间的标识符集合, 以及用 $\lambda \notin L$ 标记所有**不可观察的** (*unobservable*) S-元素或 T-元素. 上述条件表明 $(\beta(B) - \{\lambda\}) \cap (\beta(E) - \{\lambda\}) = \emptyset$.

- (3) $\lambda \notin \beta(c_{in})$.

注意 β 通常不是一个内射, 几个可观察的 (*observable*) S-元素 (或 T-元素) 可能用同一个名字来加标. 这里, $x \in B \cup E$ 是**可观察的**, 当且仅当 $\beta(x) \neq \lambda$.

定义 A.6 设 $\Sigma = (N, \beta)$ 为一个 STLEN 系统. 称 $C_\Sigma = \{c \mid c \in C_N \wedge \lambda \notin \beta(c)\}$ 为**可观察情态集** (*set of observable cases*) . 我们假定初始情态 c_{in} 总是一个可观察情态.

称 $U_\Sigma = \{u \mid \exists u' \in U_N (u = \beta^+(u') \wedge u \neq \emptyset)\}$ 为 Σ 的**可观察点火步集合**. 这里 $\beta^+(u) = \{x \mid \exists x' \in u (x = \beta(x') \wedge x \neq \lambda)\}$.

下面的定义中要用到函数 β 的扩展 $\beta^*: U_N^* \rightarrow U_\Sigma^*$. 对 $\omega = u_1 u_2 \dots u_k$ ($u_1, u_2, \dots, u_k \in U_N$), $\beta^*(\omega) = \beta^+(u_1) \beta^+(u_2) \dots \beta^+(u_k)$. 在其中, 若 $\beta^+(u_i) = \emptyset$, 就令 $\beta^+(u_i) = \varepsilon$, 这里 ε 代表 U_N^* 及 U_Σ^* 中的空串. 我们有 $\beta^*(\varepsilon) = \varepsilon$; 对 $\omega \in U_N^*$ 或 $\omega \in U_\Sigma^*$, $\omega \varepsilon = \varepsilon \omega = \omega$ 成立; 且对 $\omega_1, \omega_2 \in U_N^*$, 或 $\omega_1, \omega_2 \in U_\Sigma^*$, $\beta^*(\omega_1 \omega_2) = \beta^*(\omega_1) \beta^*(\omega_2)$ 成立.

定义 A.7 设 $\Sigma = (N, \beta)$ 是一个 STLEN 系统, $c, c' \in C_\Sigma$. 称 $\omega = u_1 u_2 \dots u_k$ 为从 c 导致 c' 的**直接点火步序列** (*immediate step sequence leading from c to c'*), 记为 $c[\omega>>c'$ 或 $c[u_1 u_2 \dots u_k>>c'$, 当且仅当

- (1) $u_1, u_2, \dots, u_k \in U_N$;
- (2) $c[u_1>c_1, c_1[u_2>c_2, \dots, c_{k-1}[u_k>c' \Rightarrow \forall i (1 \leq i < k \rightarrow c_i \notin C_\Sigma)$.

定义 A.8 设 $\Sigma = (N, \beta)$ 是一个 STLEN 系统, $c, c' \in C_\Sigma$. 称 $\omega \in U_\Sigma^*$ 为从 c 导致 c' 的**外部直接点火步序列** (*externally immediate step sequence leading from c to c'*), 记为 $c[(\omega>c'$, 当且仅当 $\exists \omega' \in U_N^* (c[\omega'>>c' \wedge \omega = \beta^*(\omega'))$.