OOPN 集成开发环境

任爱华 牛锦中 孙自安 杜悦冬
北京航空航天大学 计算机系 100083

摘要

本文介绍 OOPN-IDE（Object Oriented Petri nets Integrated Development Environment）集成开发环境的设计与实现原理，全面地介绍了该工具的功能。该集成工具主要包括：并发系统建模，并发系统仿真，并发系统运作，并发系统死锁检测四个子工具。

1 OOPN-IDE 综述

面向对象是一种非常有效的程序设计范型，越来越受到人们的重视和欢迎。这是因为用此种方法开发出来的应用系统能适应不断变化的客观环境；并可在构件级上实现软件重用。但是在并发程序的设计中，运用面向对象方法构造复杂大系统时，往往也要构造大量的对象实体，它们之间的关系也变得越来越复杂，且诸多对象之间隐含有并发性，对象之间相互通讯与制约关系的表达不直观，稍一疏忽便易出错，因而迫切需要有一种更加直观的、形式化的方法来进行辅助设计。

而 Petri 网能简洁地描述系统的动态特性（如：并发、同步、冲突等）和系统中的资源及约束条件，并有相应的分析方法（如：可达图分析、不变量分析以及图论等相关的分析理论），而且是一种图形工具，易于理解和描述，所以被广泛地应用在具有并发、并行、异步和随机性质的系统建模与分析中。然而传统的 Petri 网模型，对大系统来说存在模型的复杂性问题。尽管可通过采用高级 Petri 网和化简方法来降低模型的复杂性，但效果并不十分理想，实现较为困难。如果采用面向对象的 Petri 网模型则比以往传统的 Petri 网模型描述更直观，因为面向对象技术提供了抽象的封装、分类以及继承机制，所以对庞大而复杂的系统描述提供了更有效的简化手段。

若将面向对象技术与 Petri 网方法相结合，充分发挥二者的优势，同时尽量弥补各自的缺点，无疑是一种解决复杂问题的根本方法。因此，我们设计和实现了集成开发环境 OOPN-IDE。本环境将为并发软件的开发提供支持，包括 OOPN 模型的建立、仿真、运作、管理和并发软件的生成。

2 OOPN 的形式化描述和系统建模

面向对象技术对解决复杂性问题以及可重用问题提供了有效的方法。而 Petri 网适合于描述系统的动态特性和系统中的资源约束条件， 既有形式化描述，又可采用图形直观地表示。在面向对象技术中，把对象看为实体，每个对象有自己的数据和任务，根据接收到的通讯信息（消息）来完成相应的任务。为了使每个对象尽可能相互独立，本文采用了将同步制约条件从每个对象内部分离出来的做法，使得对象内部行为描述与外部通讯接口的分析可分开处理，从而提高可重用性。

下面分别来讨论在面向对象 Petri 网技术中对象的描述、网的建立、通讯同步约束的提取以及系统分析的解决办法。

2.1 对象的表示

考虑一个对象 A,它由基本对象 AA 和 AB 复合而成。如图 1 所示，用框子圈住对象的内部，表示封装与抽象，对象的外部接口部分由“消息队列”（用椭圆表示，其作用类似于用圆表示的 state）、“门”（用粗线表示，其作用类似于用方形框表示的 transition）以及它们之间的流关系（用弧线表示）给出。各对象内部分别可以有若干实体（用黑点表示，即 token），各 token 所处的 state 表示各自的当前状态。

本文所介绍的面向对象 Petri 网简称为 OOPN，在 OOPN 中定义了两种对象类型：基本对象与复合对象。在基本对象中，不包含并发部分，基本对象只用于表示顺序行为与静态性质；而在复合对象中则允许并发，因为复合对象由具有独立行为的基本对象构造而成，其控制分布在各基本对象中，并且根据系统要求可以来同步这些基本对象的顺序行为。

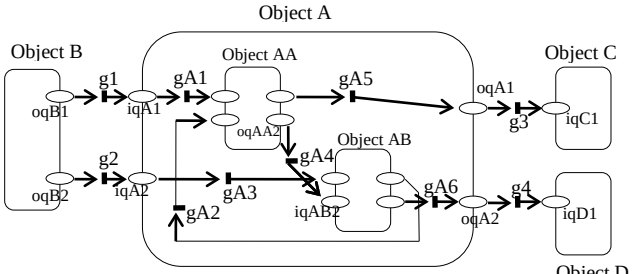


图 1　复合对象 A 及对象 B、C、D 构成的系统

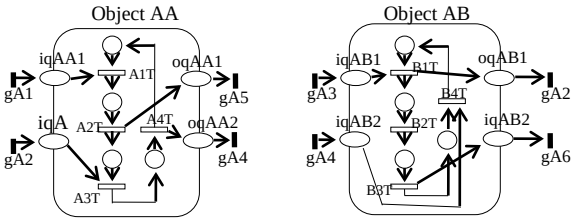


图 2　对象 A 的内部结构：基本对象 AA 和 AB

2.2 OOPN 的形式化描述

1、系统

在面向对象 Petri 网中，系统由多重层次对象及它们之间的消息传递关系组成：

SYSTEM = (O,R)

其中，　O ： 对象集合

R ： 关系集合

例如：图 1 所示系统:

O = {A,B,C,D}

R = {<oqB1,g1,iqA1>,
<oqB2,g2,iqA2>,
<oqA1,g3,iqC1>,
<oqA2,g4,iqD1>}

2、对象的外部结构

对象 $O_i \in O$ 可表示为六元组：

$O_i = (H_i, IG_i, OG_i, IM_i, OM_i, F_i)$

其中，

H_i ： 对象层次，指定父对象

IG_i: 输入门集合

OG_i: 输出门集合

IM_i: 输入消息队集合

OM_i: 输出消息队集合

F_i ： 流关系集合

输入/输出门统称为门，用来连接相应的输出消息队和输入消息队，并进行必要的消息类型转换。

输入/输出消息队是对象的窗口，通过它们可以向外部发送消息要求服务并接收应答或从外部接收消息提供服务并返回应答。这样的消息传递机制有效地将对象的内部与外部相分离，提高了可维护性和可重用性。

流关系集合指示输入门与输入消息队、输出门与输出消息队的连接关系。

3、对象的内部结构

设 PO_i 表示基本对象 i, CO_j 表示复合对象 j, 则整个系统的对象集合为：

$$O = PO \cup CO$$

其中，

$$PO = \bigcup_i PO_i \qquad CO = \bigcup_j CO_j$$

进一步地，我们用 IPO_i 和 ICO_j 分别表示 PO_i 和 CO_j 的内部结构：

关于复合对象，其内部结构定义了该对象所包含的子对象以及它们之间的相互关系：

$$ICO_j = (X, Y, R_j)$$

其中，

$X \in \rho(CO)$, $CO_j \notin X$

$Y \in \rho(PO)$

R_j: 关系集合

这里的$\rho(CO)$、$\rho(PO)$分别表示 CO、PO 的幂集。

关于基本对象，其内部结构明确指定了其静态特性与动态行为。静态特性是用代数方法来描述对象的某些属性或状态；而动态行为则用高级网来表示，因而具有更强的表达分析能力。

基本对象的内部结构可定义为：

$$IPO_i = (D_i, SV_i, S_i, AT_i, LF_i, IN_i, M_。)$$

其中，

D_i : 属性集合

SV_i: 状态变量集合

S_i : 状态集合

AT_i: 活动 transition 集合

LF_i: 局部流关系集合

IN_i: 实例集合

$M_。$: 初始标识

属性与状态变量意思上很相似，都表示基本对象的当前状态，但前者常表示较长久的量（如人的年龄等），而后者则常随实例状态的变化而变化（如机床的忙、闲）。用 state 代表基本对象的当前状态， 所以状态是 state 的非空子集。

每个状态都对应有刻划状态变量的一元状态谓词，定义状态谓词是通过将特定状态映射为基本对象的状态值元组的一个函数来完成。

活动 transition(即 action transition)是 transition 的子集， 起着同步作用，在它的先决条件满足时，就可以引发，去执行预先定义的活动。这些活动代表了顺序程序的执行。活动分为内部活动和外部活动， 主要取决于该活动是否为其他对象提供了服务。

局部流关系表示了基本对象的内部控制流，指示输入/输出消息队（或状态）与活动 transition 之间的连接关系。

模型中有两种 token：一种代表对象的具体实例，只在对象内部流动，不允许在网执行期间被创建或销毁。实例由实例标识符唯一确定和引用。实例的初始状态由初始时存放实例的 state 表示；另一种 token 代表对象之间通讯的信息，可在对象之间流动，允许动态创建或销毁。

2.3 利用 OOPN 建立系统模型

对于一个应用系统，建立面向对象的 Petri 网模型分以下几步：

⑴确定组成系统的各个对象的结构。

①从研究的问题域里出现的名词中选出用来构成系统的对象。

②对每一对象确定其自身行为及与其它对象的联系。

③对较复杂对象，可进一步提取子对象，对每一子对象进行②中的分析；对已比较简单的对象，用简单 Petri 网构造出该对象的内部行为。

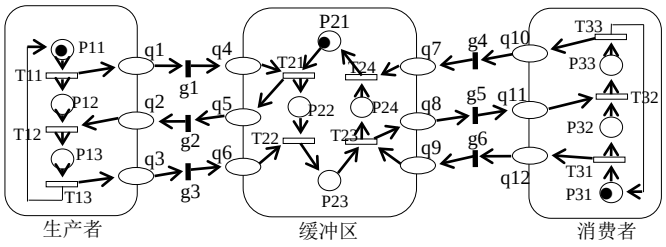
④用消息队表示对象的外部接口。

⑤将内部控制中的 transition 与相应的消息队相连。

⑵构成系统静态结构。

①确定对象间的通讯关系。
②用门将对象间对应的输入消息队与输出消息队相连。

⑶确定系统的初始标识。



P11：生产者空闲。
P12：等待生产许可。
P13：正在生产。
T11：提出生产请求。
T12：获得生产许可。
T13：结束生产。

P21：缓冲区为空。
P22：等待生产结束。
P23：缓冲区为满。.
P24：等待消费结束。

P31：消费者空闲。
P32：等待消费许可。.
P33：正在消费。
T31：提出消费请求。
T32：获得消费许可。
T33：结束消费。

T21：收到生产请求并应答。
T22：缓冲区变满。
T23：收到消费请求并应答。
T24：缓冲区变空。

图 3　　生产者/消费者问题模型

①确定各对象的实例。
②确定各实例的初始状态。
③将代表实例的各个 token 放入与各自初始状态对应的 state 中。

经过以上三步，就可以建立一个面向对象的 Petri 网系统模型。图 3 给出了生产者/消费者问题的面向对象 Petri 网模型, 其中包含一个生产者、一个消费者和一个容量为 1 的缓冲区。
生产者（消费者）在进行生产（消费）之前，须先通过 g1 门（g6 门）向缓冲区提出请求，待应答消息通过 g2 门（g5 门）返回许可之后，才能开始生产（消费）。结束时，还要通过 g3 门（g4 门）告诉缓冲区。很显然，图 3 清楚地表示了系统涉及的生产者、消费者、缓冲区三个对象的结构及相互通讯关系，充分体现了面向对象的思想及 Petri 网图形化的优点。

2.4 同步约束的提取及系统分析

建立了系统的 Petri 网模型之后，必须进行各种性能分析以确定该系统模型是否可靠及符合实际，这其中最重要的是死锁检测。

面向对象 Petri 网的死锁检测过程是：首先根据对象的内部结构，提取出对其输入/输出门发生次序的要求，构造出接口等价网（Interface Equivalent Net, 简称 IE 网），然后将不同对象的 IE 网合并，构成整个系统的 IE 网，通过建立 IE 网的可达树，分析其中是否存在死锁。

具体说分以下几步：

⑴局部分析

①对每一基本对象的内部行为进行可达性分析。
②利用 transition 的发生次序来确定与各 transition 相关联的消息队所连接的输入/输出门的发生次序。
如：与生产者相连接的几个门的引发序列可由如下方法获得：

i ）　　g1:　通过 q1 与 T11 相连
ii）g2：通过 q2 与 T12 相连
iii)g3：通过 q3 与 T13 相连
③根据门的发生次序构造 IE 网。

⑵同步分析

①将各 IE 网中出现的相同的门合并，得到对象间的 IE 网。
②对得到的 IE 网进行可达树分析。

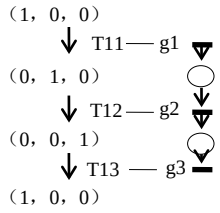


图 4 生产者对象的可达树与 IE 网

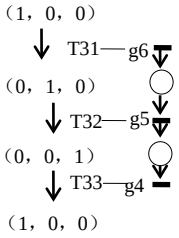


图 5 消费者对象的可达树与 IE 网

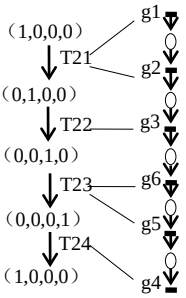


图 6 缓冲区对象的可达树与 IE 网

⑶层次抽象分析

①通过内层的 IE 网确定与内层各门相连的外层复合对象的各门的发生次序，构造出 IE 网。
②对得到的 IE 网进行可达树分析。

在上面三步中，若在某处发现有死锁，则建立的模型需要改进以消除死锁；否则，说明无死锁。以图 3 中的生产者/消费者系统为例进行分析，图 4、5、6 分别描述了该系统中的三个对象—生产者、消费者、缓冲区的 IE 网的构造过程，得到的整个系统的 IE 网与缓冲区的 IE 网恰好相同，显然该网无死锁。

3　OOPN 的具体实现

3.1 用 Java 语言进行语义的定义

要使一模型应用于实际系统的描述和开发，必须使用某种正文语言来进行语义的定义，单靠图形元素是无法有充分描述能力的。

Java 语言可在以下几方面起作用：

- 类的属性的定义，属性可以是 Java 中的简单类型也可以是 Java 类；
- 转移的发生条件和执行动作；
- 消息队列中可存放的消息类型，定义成 Java 类。

将来要把 OOPN 类转化成 Java 类来进行底层的实现，而且 Java 类中仍然保留网结构，即系统的执行仍然按照网的引发规则来进行，而非将网结构转化成语言中的控制结构来实现。

3.2 支持对象实例的动态性

支持两种特殊转移：前驱无后继有唯一位置的转移可在网执行（即系统运行）时动态创建对象实例；而后继无前驱有唯一位置的转移可动态地销毁对象实例。

3.3 对于继承性的处理

OOPN 模型是支持继承的，但没有任何的说明和体现。由于继承性是影响封装不便于分布式处理的，而且超类一般是抽象出来的，没有具体的控制流程相对应。因此决定虽支持继承性，但超类中一般只定义一些仅为了共享而在实际中并无对应的方法，且不能定义内部的控制流。

3.4 对时间的支持

为了表达时间的概念，增加时间到 OOPN 中。具体地说有两种：

- 位置加时间，表示进入其中的 token 要被锁定相应的时间之后才能进一步流动。此举可描述如网络延迟等概念。
- 转移加时间，表示转移的执行所需要的时间。

3.5 引入抑制弧

为使 OOPN 网有“零”测试能力，故引入抑制弧。

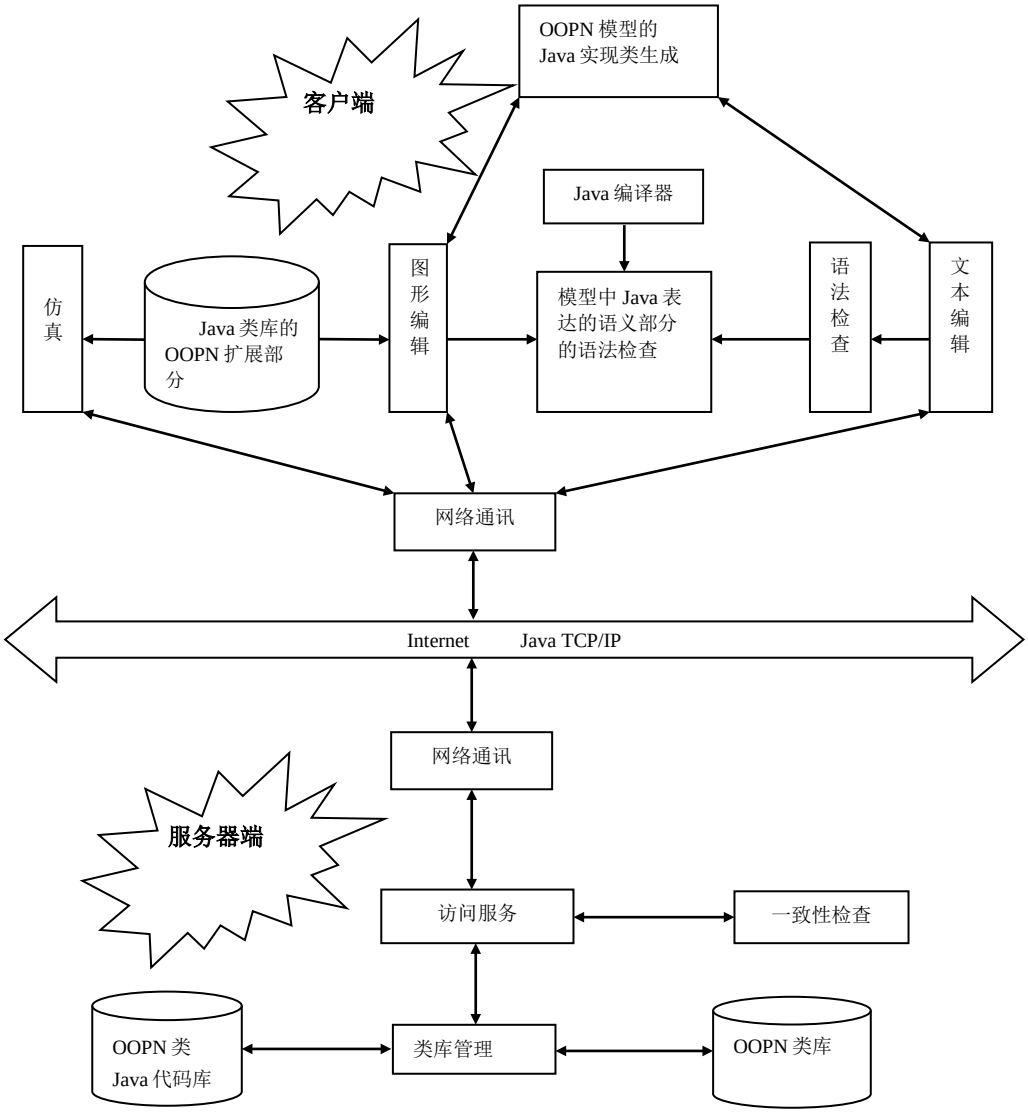


图 7 并发软件开发 OOPN 支持工具

4 OOPN 的体系结构

4.1 系统组成

OOPN-IDE 是在 JDK 运行环境之上开发完成的，可在基于 PC 平台的 Microsoft Window95/NT 的操作系统上运行。整个系统由图形编辑、文本编辑、Java 类的生成、类库管理、多机协同、仿真与运作以及死锁检测等模块组成，其结构设计成了 Client/Server 方式，以支持同一软件的分布式协同建模和管理， Client 端完成与用户的交互，Server 端负责多个 Client 的协同，如图 7 所示。

4.2 图形编辑工具

图形编辑工具支持用户以图形方式建立 OOPN 模型，为此本工具提供了图形编辑工具条供用户选择所需的组件。由于组成 OOPN 简单类和复合类的组件互不相同，并且 OOPN 模型系统还需要添加“token”的操作，所以本工具为简单类、复合类和模型系统分别提供了不同的编辑工具条，如图 8、9、10 所示。

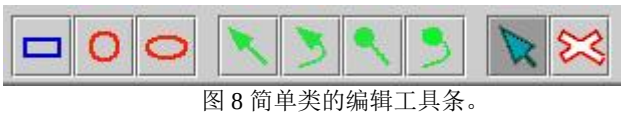


图 8 简单类的编辑工具条。

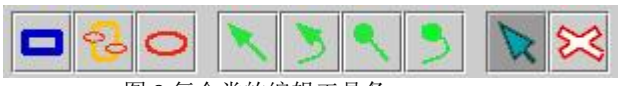


图 9 复合类的编辑工具条。

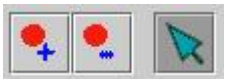


图 10 模型系统的编辑工具条。

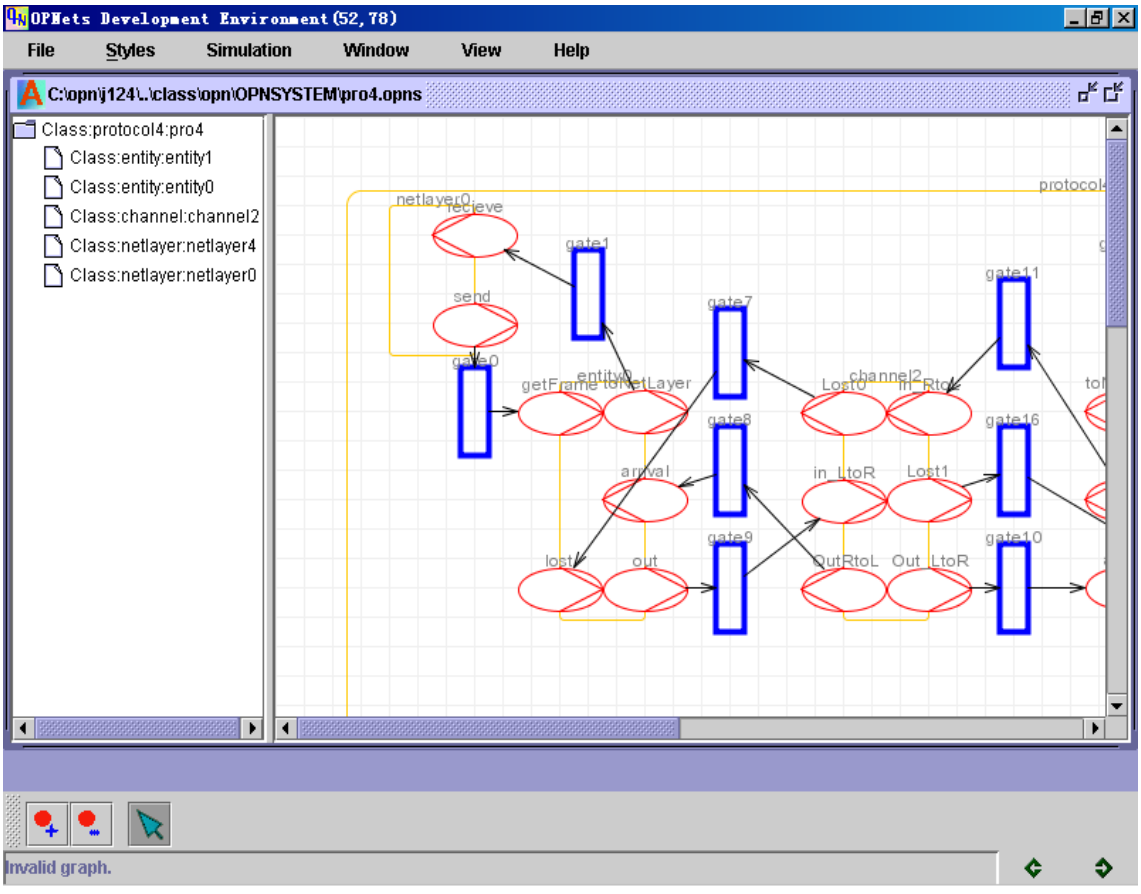


图 11 模型系统的图形编辑窗口。

图形编辑工具的主要功能包括：

- 为 OOPN 模型创建组件。
- 将组件移到合适的位置。
- 删除多余的组件。
- 在组件之间建立连接。
- 设置或浏览组件的属性。
- 放大或缩小组件。
- 对于模型系统，还提供了复合类的层次结构图，以便支持复合类视图和其内部类视图之间的切换，如图 11 所示。

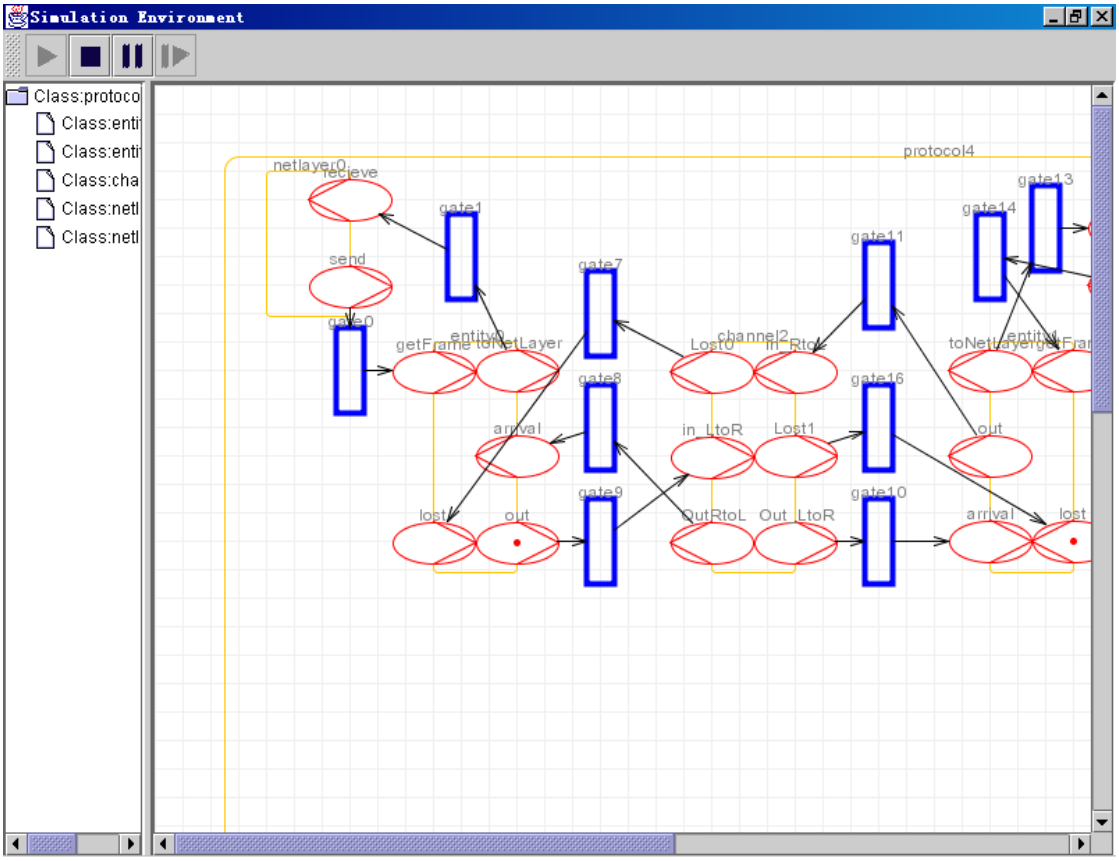


图 12 仿真环境。

4.3 文本编辑工具

文本编辑工具支持用户以文本方式建立 OOPN 模型。文本是有预定义格式的，用户必须按照预定义格式输入文本。文本编辑工具的主要功能就是提供文本编辑窗口，使用户在窗口中按照预定义格式输入 OOPN 模型的文本定义。待用户输入完毕保存文件时，文本编辑工具还要对输入的文本进行语法检查，并反馈检查的结果。

注意：对于消息类，本环境只支持文本编辑功能，并且规定所有定义的消息类都是 Message 类的子类，所以在消息类的初始窗口中已经有预定义的文本，但简单类和复合类的初始窗口都是空白窗口。

4.4 Java 类的生成

对模型中用 Java 语言描述的部分调用 Java 编译器进行语法检查。通过检查后，即可生成 OOPN 实现类。OOPN 类及其实现类均要放到位于 Server 上的相应类库中。

OOPN 网中的各种图形元素都设计成 Java 类，组成了 Java 类库的 OOPN 扩展部分，在图形建模及仿真中调用。

4.5 类库管理

Server 负责存储和管理用户建立的 OOPN 类及相应实现类。

4.6 多机协同机制

本环境支持多个用户同时编辑同一个 OOPN 模型，即支持多机协同机制。本环境实现的多机协同机制允许某个 Client 与任一 Server 建立连接，并打开该 Server 中的 OOPN 模型。当多个用户编辑同一个 OOPN 模型时，任一用户对该模型的修改都会让其它用户立即觉察到。Client 与 Server 的通讯由网络通讯模块完成。Server 对 Client 希望得到的各种信息提供服务，对 Client 反馈的信息进行应答和控制，对多 Client 的协同建模操作进行必要的一致性检查。总之，本环境实现的多机协同机制主要支持以下功能：

- 允许 Client 与任一启动的 Server 建立连接。
- 允许 Client 随时断开已经建立的连接。
- 允许 Client 打开、浏览或编辑 Server 中的模型类。
- 当多个用户同时编辑同一模型类时，每个用户都能觉察到其它用户对该模型类的修改。
- 允许 Client 将当前编辑的模型类存储到远程 Server 中，无论该模型类在打开前是位于本地还是位于远程 Server 中。

4.7 仿真与运作

仿真与运作模块负责启动和管理模型系统的仿真与运作。该模块即可在 OOPN-IDE 中启动运行，也可单独启动运行。主要包括以下功能；

- 启动模型系统的仿真与运行，并打开“Simulation Environment”窗口，以便显示模型系统的仿真状态，如图 12 所示。
- 管理模型系统的仿真与运行，包括重新启动、停止、暂停和继续执行等功能，相应按钮如图 12 所示。
- 提供类视图的切换功能，允许用户查看复合类的任意内部类的仿真与运作状态。
- 提供实例的信息输出窗口，以便显示实例的运作结果。

5 小结

本系统具有如下特点

(1) OOPN 模型利用了面向对象的抽象，有类和对象，能实现一定的共享。
(2) OOPN 模型中的对象有复合对象和简单对象之分，产生了层次和结构化，使系统建模直观，易于表达和理解。
(3) 能动态仿真，可对系统模型加以验证。
(4) OOPN 模型实现了完全的封装，用当事对象可见的消息传递来实现对象的交互。
(5) OOPN 模型的语义和其底层实现使用 Java 语言，从而可以实现平台无关性。
(6) 在模型的实现中保留了 OOPN 网的结构及其图形属性表示，使在系统运行中可自然地进行可视化的监测。
(7) 引入了构件技术和分布式对象技术，使利用本支持工具开发的 OOPN 类能方便地与其他非 Petri 网结构的软件进行集成。

本项目在理论研究方面，总结了目前国内外关于面向对象 Petri 网的研究成果，建立了面向对象 Petri 网的理论体系及形式化描述，并且提出了一种以面向对象 Petri 网为基础的并发软件开发方法。

在实践方面，开发了一套可扩展的基于 OOPN 的并发软件开发的集成化开发环境 OOPN-IDE，其中包括交互式图形建模工具、文本建模工具、多机协同工具、仿真与运作工具、死锁检测工具等等。

进一步的工作，采用构件技术，使 OOPN 模型可调用其它控件共同协作来完成系统的构造实现。要协作，则需要在两部分之间定义简洁、方便的接口。

应用 Petri 网的原因，是它在表达并发上有优势。而分布是与并发紧密相连的，既然把面向对象应用到 Petri 网上形成了一个个的对象，自然也就要考虑如何进行对象的分布与相互通信。现在有的几种分布式对象技术有 CORBA 与 DCOM，新出现的有 Java 的 RMI。由于 RMI 简洁易用，故考虑用 RMI 作为 OOPN 模型系统进行分布式处理的基础来构筑整个的 OOPN 模型的分布式体系。

参考文献

T.Murata,"Petr Nets:Properties,Analysis and Application",Proc. IEEE 77,pp541-580,1989

J.Wngelfriet,G.Leigh,G.Rozenberg,"Net-based description of paralleled object-based systems",LNCS 489,pp229-273

M.Baldassari,G.Bruno,"PROTOB: Object-Oriented graphical modeling and prototyping of real-time system", In Second International Workshop on Computer-Aided Software Engineering, July 1988.

M.Baldassari,G.Bruno,"PROTOB: A hierarchical Object-Oriented CASE tool for distributed system", European Software Engineering Conference, 1989.

M.Baldassari,G.Bruno,"PROTOB:an object oriented methodology for developing discrete event dynamic systems",Computer Language,Vol.16,No.1,1991,pp39-63

S.L.English,"Colored Petri Nets for Object Oriented Modeling",Doctoral Dissertation,June 1993 at the University of Brighton

Y.Deng,S.K.Chang,J.C.A.de.Frgueried,A.Perkusich,"Intergrating Software Engineering Methods and Petri Nets for the Specification and Prototyping of Complex Information System",LNCS 691,pp203-223

D.Buchs,N.Guelfi,"COOPN:A concurrent object-oriented Petri net approach",Proc. of the 12th International Conference on the Application and Theory of Petri Nets,Gjern,Denmark,1991,pp432-454

C.A.Lakos,C.D.Keen,"LOOPN:Language for Object-Oriented Petri Net",Proc. of the SCS Multiconf on Object Oriented Sim.,1991,Vol.123,No.3,pp22-30

L-C.Wang,"The Development of an Object-Oriented Petri Net Cell Control Model",Int. J. Adv. Manuf. Technol.(1996)11:59-69

Yang Kyu Lee,Sung Joo Park,"OOPN:An Object-Oriented High-Level Petri Net Model for Real-Time System Modeling",J. of Systems and Software,Vol.20,No.1,1993

H.Genrich and K.Lautenbach, "System Modeling with High Level PetriNets", Theoret. Comp. Sci. 13 pp109-136

Jensen, "Coloured Petri Nets and the Invariant-Method", Theoret. Comp. Sci. 14, pp317-336, 1981

O.Nierstrasz, "A survey of Object-Oriented Concepts", in Object-Oriented Concepts, Databases and Applications ACM Press, 1989

Masato Notomi, Tadao Murata, "Hierarchical Reachability Graph of Bounded PETRI Nets for Concurrent-software Analysis", IEEE Transactions on Software Engineering, Vol.20, No.5, May 1994

Jeffrey J.P.Tsai, Steve Jennhwa Yang,Yao-hsiung Chang, "Timing Constraint Petri Nets and Their Application to Schedulability Analysis of Real-time System Specifications", IEEE Transactions on Software Engineering, Vol 21, NO.1, January 1995.

胡俊等译，杨文龙校，"Petri 网导论"，1989，北航

杨文龙，"Petri 网：理论与应用"，北航

杨文龙等编著，"基于 Petri 网的并发软件开发方法及支持工具的研究（论文集）"，1993 年 5 月，科技文献出版社

吴芸，杨汉瑜，任爱华，杨文龙，"基于面向对象 Petri 网（OOPN）开发方法的研究"，东南大学学报，Vol.25,No.3A,May 1995OOPN）开发方法的

吴芸，杨汉瑜，任爱华，杨文龙，"Petri 网与面向对象技术结合应用的发展"，计算机世界，1995.9.20, PETRI 专刊。

吴芸，杨汉瑜，任爱华，杨文龙，"基于对象 Petri 网开发方法的建立与应用"，计算机世界，1995.9.20, PETRI 专刊。

23.任爱华 ,牛锦中，张永鸣"一种基于面向对象 PETRI 网的并发程序建模方法"

北京航空航天大学学报，1998 年，vol.24 ，No.4，pp491-494，ISSN 1001-5965

OOPN Integrated Developing Environment for concurrent programs

Aihua Ren, Jinzhong Niu, Zian Sun, Yuedong Du
Dept. of Computer Science, Bijing University of Aeronautics and Astrunautics, 100083

Abstract

The design and implementing principles of OOPN-IDE (Object Oriented Petri nets Integrated Development Environment) are presented in this paper. The integrated tools mainly include the following four tools: the concurrent system modeling tool; the concurrent system simulator; the concurrent system enactor; the concurrent system deadlock detector.