文章编号: 1001-0920(2006)11-1208-06

基于高级Petri网的仿真剧情正规校核方法

刘 飞, 杨 明, 王子才
(哈尔滨工业大学 控制与仿真中心, 哈尔滨 150001)

摘 要: 针对仿真剧情主观校核不理想这一问题, 提出了基于高级Petri网的仿真剧情正规校核方法. 首先给出仿真剧情的形式化定义, 并分析仿真剧情可能存在的错误类型; 其次给出仿真剧情到高级Petri网的映射途径, 并给出基于高级Petri网的仿真剧情校核准则和算法, 此外, 还给出实现仿真剧情动态校核的推理规则和机制; 最后给出了一个正规校核工具框架. 实际应用已经证明了该方法的有效性.

关键词: 仿真剧情; 高级Petri网; 正规校核; 工具框架
中图分类号: TP391.9 **文献标识码**: A

Formal Verification Method of Simulation Scenario Based on High-level Petri Nets

LIU Fei, YANG Ming, WANG Zi-cai
(Control and Simulation Center, Harbin Institute of Technology, Harbin 150001, China. Correspondent: LIU Fei, E-mail: liuf_2001@163.com)

Abstract: To solve the problem of unsatisfactory subjective verification for simulation scenario, a formal verification method of simulation scenario based on high-level Petri nets is presented. A formal definition of simulation scenario is given, and possible errors in simulation scenario are analyzed. Approaches to mapping simulation scenario into high-level Petri nets are proposed. Criteria and algorithms to verify simulation scenario based on high-level Petri nets are presented. Inference rules and mechanisms for dynamic verification of simulation scenario are also given. A formal verification tool framework of simulation scenario is constructed. Practical applications show that the formal verification method is effective.

Key words: Simulation scenario; High-level Petri nets; Formal verification; Tool framework

1 引 言

仿真剧情校核是仿真校核、验证与验收(VV&A)的一项重要内容. 通常, 仿真剧情是以非正规的自然语言形式给出, 对其进行校核时, 一般采用专家审查、走查等主观、定性的评估方法[1]. 对于简单的仿真剧情来说, 定性评估基本满足要求. 然而, 对于复杂仿真系统来说, 其剧情十分复杂且剧情实例数目巨大, 定性评估方法通常会受到专家主观意志的制约, 不能有效、公正地完成复杂仿真剧情的校核任务, 且不能实现自动评估, 从而不能在一定的时间和资源限制内完成规定的任务. 因此, 对于复杂仿真剧情来说, 单靠手工的、定性的主观评估方法已远远不够, 如何使用正规的评估方法以及开发有效的评估工具已成为复杂仿真剧情校核的首要任务.

为了满足当前正在开发的复杂仿真系统的剧情自动校核的需要, 本文提出了基于高级Petri网的仿真剧情正规校核方法, 其基本思想如下: 可以简单认为仿真剧情是时间轴上的任务序列[1], 将任务序列转化为高级Petri网, 利用已存在的Petri网分析技术实现对仿真剧情的正规校核. 本文利用该方法开发了仿真剧情校核工具, 实现了复杂仿真剧情关键内容的自动校核.

收稿日期: 2005-09-30; **修回日期**: 2005-12-27.
基金项目: 国家自然科学基金项目(60434010).
作者简介: 刘飞(1976—), 男, 山东平度人, 博士生, 从事复杂仿真系统建模、VV&A与可信度评估技术等研究;
杨明(1963—), 男, 吉林蛟河人, 教授, 博士, 从事复杂系统仿真理论与方法、分布交互仿真技术等研究.

2　仿真剧情

剧情是仿真系统的关键因素, 任何仿真系统都是在一定的剧情下运行的. 对于仿真剧情来说, 目前还没有一个统一的定义. 一般来说, 仿真剧情是指在某个作战区域, 在特定的时间和背景下发生的一系列作战行动和事件, 构成剧情的基本要素是作战部队及使用的武器装备、战场环境和时间[2]. 简单地说, 剧情可以认为是时间轴上的一个任务序列.

剧情校核是仿真校核、验证与验收(VV&A)的一个关键内容. 为了实现仿真剧情的正规校核, 下面将探讨剧情的形式化定义, 并分析剧情可能存在的错误类型.

2.1　仿真剧情的形式化定义

为了明确仿真剧情包含的内容, 并准确无误地将其转化为高级Petri网, 首先需要对仿真剧情进行形式化的定义.

定义1　仿真剧情是时间轴上的任务序列, 即 $Scenario = \{TA_1, TA_2, \cdots, TA_n; t\}$, 其中 TA_n 为单个任务, t 为时间轴.

定义2　仿真剧情中的任务是一个五元组 $TA = (Action, PreC, PostC, STSeq, [t_a, t_b])$, 其中 Action 为任务执行的活动; $PreC$ 为任务的前置条件, 指明了任务发生的条件; $PostC$ 为任务的后置条件, 指明了任务产生的结果; STSeq 为任务包含的子任务序列, 是可选的, 对于原子任务(不可再分的任务), 此项为空; $[t_a, t_b]$ 为任务 TA 的最早和最晚触发时间, 反映了仿真剧情的时间特性.

定义3　任务的序列定义了任务发生的顺序关系. 通常存在如下几种类型的任务序列:

1) 顺序执行(没有延迟): 可以表示为 (TA_1, TA_2). 这种情况下, 任务 TA_1 执行完毕后, 任务 TA_2 紧跟着执行, 不存在时间延迟.

2) 顺序执行(有延迟): 可以表示为 (TA_1, TA_2, Δ). 这种情况下, 任务 TA_1 执行完毕后, 等待一段时间 Δ 后, 任务 TA_2 才能执行.

3) 选择执行: 可以表示为 $(\alpha TA_1 \vee \beta TA_2)$. 当条件 α 满足时, 执行 TA_1; 当条件 β 满足时, 执行 TA_2.

4) 并行执行: 可以表示为 $(TA_1 \vee TA_2)$. 这种情况下, 任务 TA_1 和 TA_2 并行执行, 根据开始和结束时间的不同, 并行执行又可以进一步分为不同的类型, 例如同时开始和同时结束, 同时开始但不同时结束等.

5) 反复执行: 可以表示为 (TA, r), 这种情况下, 任务 TA 在规定的条件下反复执行, 直到出现结束条件.

为了清晰地说明上面对剧情的形式化定义, 本文针对图1(a)的仿真系统给出一个简单剧情, 如图1(b)所示, 这一剧情可以分解为更细的剧情, 如图1(c)所示.

图1(b)和1(c)中的各个任务说明如下: TA_1 为弹道导弹飞行; TA_2 为 GEO 探测弹道导弹的轨迹; TA_3 为指挥中心下达探测命令; TA_4 为UEWR 探测弹道导弹的轨迹; TA_5 为 XBR 探测弹道导弹的轨迹; TA_6 为指挥中心下达拦截弹发射命令; TA_7 为拦截弹飞行; TA_{21}, TA_{22}, TA_{23} 分别为 TA_2 的子任务.

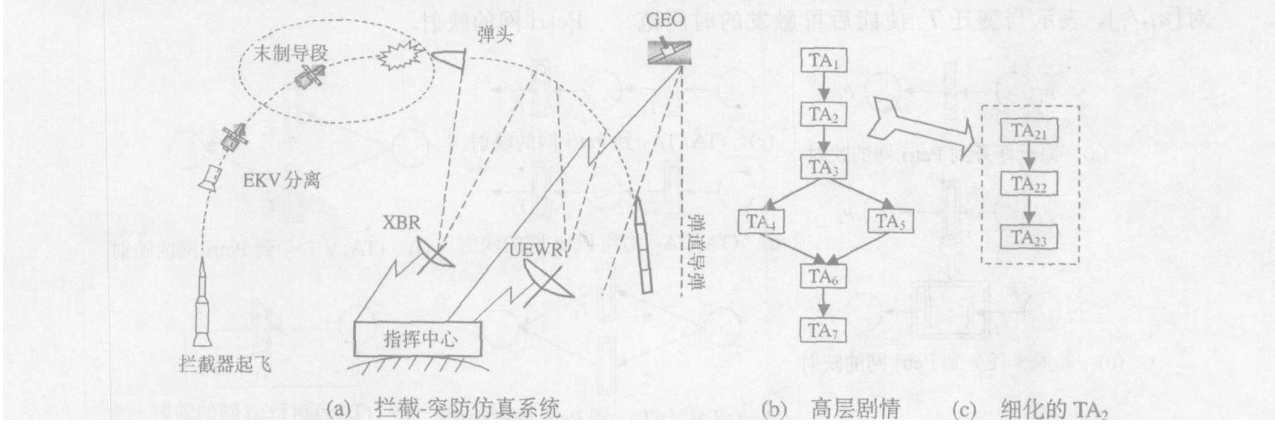

图1　拦截 - 突防仿真系统及其剧情

2.2　仿真剧情可能存在的错误类型

一个复杂仿真系统存在大量的剧情实例, 可能隐藏大量的错误. 例如, 由于剧情编辑人员的粗心导致剧情关键项的描述错误; 由于多个人员的协作编辑导致剧情关键交互的丢失等等[1]. 因此, 为了获得可信的仿真运行结果, 必须对仿真剧情进行认真的校核. 为了取得较好的校核结果, 首先需要分析仿真剧情可能存在的错误的类型. 根据经验, 一般来说, 仿真剧情可能存在以下几种类型的错误:

1) 不完整性错误. 仿真剧情中包含大量的不完整性错误, 可进一步分为以下几类: 丢失关键的任务、丢失关键的交互、丢失任务的前置条件和后置条

件等. 上述任何一种类型错误的发生都不能产生正确的仿真系统运行结果.

2) 不一致性错误. 仿真剧情中同样存在大量的不一致性错误, 例如交互参数描述的不一致, 任务前置条件和后置条件描述的不一致等.

3) 语义错误. 这一错误通常是由剧情编辑人员对用户需求理解的不彻底而造成的, 这类错误是很难校核的, 通常需要领域专家来发现这类错误.

4) 时间错误. 时间是仿真剧情的关键因素, 仿真剧情可以认为是时间轴上的任务序列, 因此各个任务开始和结束时间的正确性也是仿真剧情校核的关键内容.

在本文的后面部分, 将针对上面的错误类型开发适合的校核准则.

3 高级 Petri 网

Petri 网是产生于 60 年代的一种关注系统定性结构性质和各部分行为关系的逻辑层次模型. 几十年来, Petri 网在计算机、通信等多个领域得到了广泛的应用. 高级 Petri 网是原始 Petri 网的扩充形式, 主要包括以下几种: 着色 Petri 网、谓词 Petri 网、时间 Petri 网等. 其中, 时间 Petri 网最适合用于仿真剧情的校核, 下面将对时间 Petri 网作简单讨论.

时间 Petri 网(TPN) 可以使用六元组 $N = (P, T, B, F, M_0, I)$ 来描述[3], 其中 P 是有限非空的库所集合; T 是有限非空的变迁集合; $B: P \times T \to N$ 是从 P 到 T 的连接弧集合; $F: T \times P \to N$ 是从 T 到 P 的连接弧集合; M_0 是初始标识; I 是变迁 T 的延迟函数集合, $I(t_i) = [a_i, b_i]$, 其中 $0 \leq a_i \leq b_i$. 变迁时间对 $[a_i, b_i]$, 表示当变迁 T_i 使能后可触发的时间范围.

Petri 网的最大优点就是拥有坚实的数学基础以及大量的分析方法, 例如可达性分析、不变量分析、状态方程分析等[4~6]. 这些分析方法能够在系统实现前揭示系统存在的错误, 这也是利用 Petri 网实现仿真剧情校核的基础.

4 仿真剧情到高级 Petri 网的映射

为了利用高级 Petri 网校核仿真剧情, 需要将仿真剧情映射为高级 Petri 网, 然后再利用高级 Petri 网的各种分析方法对仿真剧情进行校核. 下面首先考虑第一个问题, 将仿真剧情映射为高级 Petri 网.

4.1 任务到高级 Petri 网的映射

为了将任务表示为高级 Petri 网, 需要做以下映射: 将任务的活动映射到高级 Petri 网的变迁, 将任务的前置条件映射到高级 Petri 网的输入场所, 将任务的后置条件映射到高级 Petri 网的输出场所. 形式化表达如下:

1) Action $\to T$;

2) Pre$C \to P_i$;

3) Post$C \to P_a$.

此外, 考虑到任务是嵌套的, 所以使用如下的非层次 Petri 网来表示原子任务(如图 2(a) 所示) 和层次 Petri 网[7,8] 表示非原子任务(如图 2(b) 所示). 非原子任务可进一步分解为原子任务.

4.2 任务序列到高级 Petri 网的映射

任务序列到高级 Petri 网的映射就是将上面给出的各种类型的任务序列, 合理地映射到高级 Petri 网中. 图 2(c) ~ 2(g) 给出了各种任务序列到高级 Petri 网的映射.

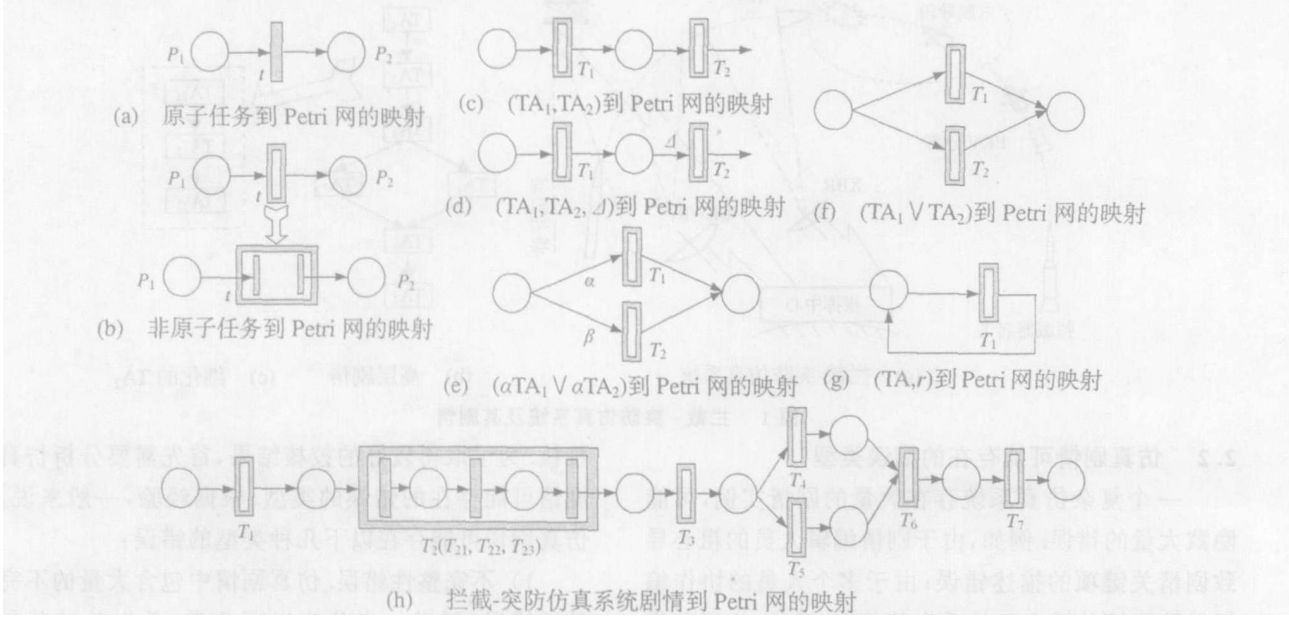

图 2 仿真剧情到高级 Petri 网的映射

4.3 任务的时间约束到高级 Petri 网的映射

任务的时间约束到高级 Petri 网的映射, 就是将任务中的时间约束$[t_a, t_b]$映射到 TPN 中的变迁时间对$[a_i, b_i]$. 这一映射是很容易的, 可以表示为$[t_a, t_b] \rightarrow [a_i, b_i]$.

以上完成了任务、任务序列和时间约束到高级 Petri 网的映射, 这样便可以将整个仿真剧情映射为高级 Petri 网. 图 2(h) 给出了拦截-突防仿真系统剧情的 Petri 网表示.

5 基于高级 Petri 网的仿真剧情正规校核

将仿真剧情转化为高级 Petri 网后, 便可利用 Petri 网的各种分析方法来完成仿真剧情的正规校核. 下面将探讨利用 Petri 网来实现仿真剧情正规校核的途径.

5.1 仿真剧情的完整性和一致性校核

仿真剧情的完整性和一致性校核主要检查以下内容: 检查仿真剧情到高级 Petri 网的转化是否丢失关键项目, 检查场所和变迁是否都有名称, 检查是否存在孤立的 Petri 网等. 完整性和一致性校核能够发现一些表面的错误, 同时也能揭示出现问题较多的地方, 从而为下面其他形式的校核提供指导[9,10]. 完整性和一致性校核可通过自动化的工具来完成, 下面给出其校核算法.

5.1.1 完整性检查算法

开始

生成 Petri 网;

计算变迁的数目 I;

If ($I < K$) // K 是仿真剧情中原子任务的数目

{

报告"丢失活动";

重新生成 Petri 网;

}

For ($i = 0$; $i < I$; $i++$)

{

If (NUMBER (T_i, Input-places) < NUMBER (TA$_i$, Preconditions) // TA$_i$为变迁 T_i 对应的任务

报告"丢失前置条件";

If (NUMBER (T_i, Out-places) < NUMBER (TA$_i$, Postconditions)

报告"丢失后置条件";

If (NULL (T_i, t) // T_i 的时间约束条件 t 为空

报告"丢失时间约束条件";

}

结束

5.1.2 一致性检查算法

开始

生成 Petri 网;

FOR ($i = 0$; $i < I$; $i++$) // I 为变迁总数

{

IF (INCONSISTENCY (仿真剧情中任务的前置条件, 对应的 Petri 网中的变迁的输入场所))

报告"前置条件转化不一致";

IF (INCONSISTENCY (仿真剧情中任务的后置条件, 对应的 Petri 网中的变迁的输出场所))

报告"后置条件转化不一致";

}

结束

5.2 仿真剧情的时间校核

时间是仿真剧情的关键因素, 仿真剧情中各任务的发生和结束时间的设定对错直接关系到仿真剧情的好坏. 当仿真剧情转化为高级 Petri 网后, 对仿真剧情的时间校核就可以转化为对高级 Petri 网中时间约束的校核. 下面将对其作具体讨论.

假设在一个 Petri 网实例中, 变量 x_i 表示 $(i-1)^{th}$ 变迁的激发到 i^{th} 变迁的激发之间的时间间隔[3], 例如在图 3 中, $x_2 = t(T_2) - t(T_1)$.

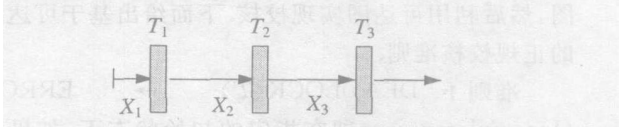

图 3 Petri 网的时间约束关系

接下来给出上述几种类型的任务序列的时间校核准则:

1) (T_1, T_2) 的时间校核准则

$$\begin{cases} a_1 \le x_1 \le b_1, \\ x_1 + a_2 \le x_1 + x_2 \le x_1 + b_2, \\ t(T_1, T_2) = x_1 + x_2. \end{cases}$$

2) (T_1, T_2, Δ) 的时间校核准则

$$\begin{cases} a_1 \le x_1 \le b_1, \\ x_1 + a_2 \le x_1 + x_2 \le x_1 + b_2, \\ 0 < \Delta < x_2, \\ t(T_1, T_2, \Delta) = x_1 + x_2. \end{cases}$$

3) $(\alpha T_1 \vee \beta T_2)$ 的时间校核准则

$$\begin{cases} a_1 \le x_1 \le b_1, \text{ or } a_2 \le x_2 \le b_2, \\ t(\alpha T_1 \vee \beta T_2) = x(\mathrm{TRUE}(T)). \end{cases}$$

4) $(T_1 \vee T_2)$ 的时间校核准则

$$\begin{cases} a_1 \le x_1 \le b_1, \\ a_2 \le x_2 \le b_2, \\ t(T_1 \vee T_2) = \max(x_1, x_2). \end{cases}$$

5) 除了上述每个任务序列单独满足的时间约束条件外, 总的仿真剧情也应该满足一个时间约束条件, 即

$$t_{sl} \leq \sum t(T_1, T_2) + \sum t(T_1 \vee T_2) + \sum t(T, r) + \sum t(T_1, T_2, \Delta) + \sum t(\alpha T_1, \beta T_2) \leq t_{st}.$$

仿真剧情的时间校核准则给出后, 下面将给出时间校核的算法:

开始

初始化 Petri 网, 确定仿真剧情实例;

确定 x_1 和 $[a_i, b_i]$, 其中 $i = (1, 2, \cdots, I)$, I 为此剧情实例的变迁总数;

FOR ($i = 0$; $i < I$; $i++$) // I 为变迁总数

{

IF $(T_1, T_2) / (T_1, T_2, \Delta) / \cdots$)

检查准则(1)/(2)/…;

}

计算此剧情实例总的执行时间;

检查准则 5);

结束

5.3 仿真剧情的语义校核

仿真剧情的语义校核的第 1 步需要生成可达图, 然后利用可达图实现校核. 下面给出基于可达图的正规校核准则.

准则 1 DEADLOCK(T) → ERROR (Incompleteness), 即在指定的初始状态下, 如果存在死锁, 那么仿真剧情存在语义错误.

准则 2 对于任一可达状态标识 $M \in R(N, M_0)$, 和任一位置节点 p_i, $M(P_i) \geq K$ → ERROR (Mis-semantics), K 为有限正整数, 即在指定的初始状态下, 如果该网络无界, 那么仿真剧情存在语义错误.

准则 3 CONFLICT(T_1, T_2) → ERROR (Inconsistency), 在指定的初始状态下, 如果该网络存在冲突现象, 那么仿真剧情存在语义错误.

利用上述 3 条校核准则, 仿真剧情语义校核的算法如下:

开始

生成可达图;

IF (DEADLOCK (变迁))

指出"剧情存在语义错误";

IF ($M(p_i) \geq K$)

指出"剧情存在语义错误";

IF (CONFLICT (T_1, T_2))

指出"剧情存在语义错误";

结束

此外, 在仿真剧情校核中, 应根据 Petri 网的特点灵活选择其他的分析方法. 例如矩阵方程分析方法, 从而有效地完成仿真剧情的语义校核.

5.4 仿真剧情的动态校核

为了实现仿真剧情的动态校核, 提出了如下的校核途径: 首先, 将 Petri 网实例(对应于剧情实例)映射到 XML; 其次, 根据得到的剧情实例运行仿真系统, 记录仿真运行过程的关键事件, 将其写入类似上述格式的 XML 文档; 此外, 还要记录事先规定的性能数据, 将其存入数据库; 最后, 采用推理规则和推理机制, 判断仿真运行后的剧情记录文件与仿真运行前的剧情规划文件的匹配程度. 下面将分别加以讨论.

5.4.1 将高级 Petri 网中的剧情实例映射到剧情 XML 文档

为了能够实现仿真运行前后剧情的校核, 并且考虑到 XML 的各种优点, 本文使用 XML 文档来描述剧情实例. 基于 Petri 网实现仿真剧情静态校核后, 得到了静态校核后的仿真剧情实例, 下面就讨论如何使用 XML 来描述这些剧情实例.

仿真剧情的动态校核内容主要包括事件和各种性能参数, 因此需要从任务中提炼出事件和性能参数. 从前面任务的定义可以知道, 任务是在一定的前置条件下, 任务的执行者执行某个动作产生一定的后置条件. 此外, 还可推出每个原子任务对应一个事件. 因此, 从原子任务中提炼事件的方法如下: 事件可以表示为任务的执行者和动作, 即 Event = Actor + Activity. 此外, 从任务中提炼性能参数需要通过专家来完成. 图 4 给出了仿真剧情的 XML 模式. 根据这一模式, 仿真剧情可以很容易地使用 XML 文档来表示.

```
XML Schema For Simulation Scenario

<?xml version="1.0" encoding="UTF-8">
<schema xmlns="http://www.w3.org/1999/XMLSchema">
  <element name="Task">
    <type>
      <group order="seq">
        <element name="Action" type="string"/>
        <element name="Precondition" type="string"/>
        <element name="Postcondition" type="string"/>
        <element name="Actor" type="string"/>
        <element name="Activity" type="string"/>
        <element name="Event" type="string"/>
        <element name="Time" type="timeInstant">
        <element name="Parameter" type="decimal">
      </group>
    </type>
  <element>
</schema>
```

图 4 仿真剧情的 XML 模式

5.4.2 推理规则和推理机制

判断仿真运行后的剧情记录文件(PostSF) 与

仿真运行前的剧情规划文件(PreSF) 的匹配程度是通过推理机来实现的[11]. 为了实现这一推理, 给出了一套推理规则, 并给出了适合仿真剧情动态校核的推理机制

1) 存在性规则, 表示为 Exist(Event). 存在性规则的推理机制如下: ① 推理机自前向后搜索 PreSF 中的关键事件; ② 利用存在性规则判断 PostSF 中是否存在这一关键事件.

2) 时间校核规则, 表示为 $t_e \in [t_a, t_b]$, t_e 为事件的发生时间, $[t_a, t_b]$ 为任务的最早和最晚触发时间. 时间校核规则的推理机制如下: ① 推理机搜索 PostSF 中的关键事件的发生时间 t_e; ② 推理机搜索 PreSF 中任务的触发时间 $[t_a, t_b]$; ③ 利用时间校核准则完成比较

3) 性能参数校核规则, 表示为 $|P^{pre} - P^{post}| \leq \epsilon$, 其中: P^{pre} 为期望参数值, P^{post} 为仿真运行后的参数值, ϵ 为允许的偏差. 性能参数校核规则的推理机制如下: ① 推理机搜索 PostSF 中的性能参数; ② 推理机根据性能参数搜索对应数据库中的多次运行数据; ③ 利用验证工具中的统计工具计算性能参数的均值或方差; ④ 利用性能参数校核规则完成比较.

6　仿真剧情正规校核工具框架

为了有效地实现本文提出的仿真剧情正规校核方法, 开发了一个正规校核工具框架对其进行支持(如图 5 所示). 该框架的各个组件说明如下:

1) 仿真剧情建模工具: 辅助剧情建模人员使用自然语言对仿真剧情进行建模;

2) Petri 网建模工具: 负责仿真剧情到高级 Petri 网的转化;

3) Petri网分析工具: 利用各种分析方法完成仿真剧情的正规校核;

4) 仿真剧情描述工具: 负责 Petri 网描述的剧情实例到 XML 文档的转化;

5) 推理机: 利用本文给出的推理规则和推理机制实现仿真剧情的动态校核

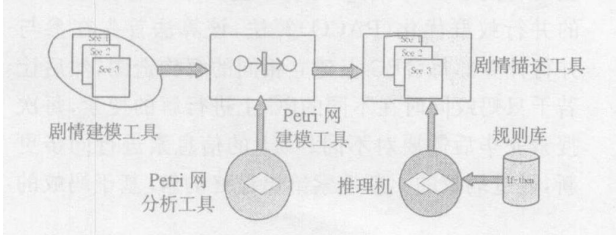

图 5　仿真剧情正规校核工具框架

7　结　　语

针对复杂仿真剧情校核的需求, 本文提出了一种基于高级Petri网的仿真剧情正规校核方法. 目前, 这一方法已经成功地应用到某多武器平台仿真系统的剧情校核中, 其有效性已得到证明. 下一步的研究方向是继续深入研究本文提出的方法并进一步完善根据这一方法设计的仿真剧情校核工具.

参考文献(References)

[1] Defense Modeling, Simulation Office. *Department of Defense Verification, Validation, and Accreditation Recommended Practices Guide* [M]. Alexandria: Department of Defense, 2000.

[2] David C Gross. Report from the Fidelity Implementation Study Group [A]. *Proc of* 1999 *Spring Simulation Interoperability Workshop* [C]. Orlando: SISO, 1999.

[3] Patrice Bonhomme, Gerard Berthelot, Pascal Aygalinc, et al. Verification Technique for Time Petri Nets[A]. 2004 *IEEE Int Conf on Systems, Man and Cybernetics* [C]. Hague, Netherlands: IEEE, 2004: 4278-4283.

[4] Daniel Lamch. Verification and Analysis of Properties of Dynamic Systems Based on Petri Nets[A]. *Proc of the Int Conf on Parallel Computing in Electrical Engineering* [C]. Warsaw, Poland: IEEE, 2002.

[5] Wang J C, Deng Y, Xu G. Reachability Analysis of Real-Time Systems Using Time Petri Nets[J]. *IEEE Trans on Systems, Man, and Cybernetics-Part B: Cybernetics*, 2000, 30(5): 725-736.

[6] Jonathan Billington, Geoffrey R Wheeler, Michael C Wilbur-Ham. Protean: A High-level Petri Net Tool for the Specification and Verification of Communication Protocols[J]. *IEEE Trans on Software Engineering*, 1998, 14(3): 301-316.

[7] Huber P, Jensen K, Shapiro R M. Hierarchies in Coloured Petri Nets[A]. *Advance in Petri Nets* 1990 [C]. Berlin: Springer-Verlag, 1990: 313-341.

[8] Jonathan Lee, Lein F Lai. A High-level Petri Nets Based Approach to Verifying Task Structures [J]. *IEEE Trans on Knowledge and Data Engineering*, 2002, 14(2): 316-335.

[9] Mats P E Heimdahl, Nancy G Leveson. Completeness and Consistency in Hierarchical State-based Requirements [J]. *IEEE Trans on Software Engineering*, 1996, 22(6): 363-377.

[10] Marsha Chechik, John Gannon. Automatic Analysis of Consistency between Requirements and Designs[J]. *IEEE Trans on Software Engineering*, 2001, 27(7): 651-672.

[11] Joseph Giarratano, Gary Riley. *Expert Systems Principles and Programming* [M]. PWS Publishing Company, 1998: 1-20.