摘要：

如何有效地验证实时软件的设计是软件工程领域研究近些年的热点之一。场景被广泛地用于描述实施软件的设计，因为其可以描述具体的交互所以很容易为用户和领域专家使用。在本论文中，我们用统一建模语言（Unified Modeling Language 简称：UML）交互模型作为基于场景的规范，包含UML2.0的交互概览图（Interaction Overview Diagram 简称：IOD）和顺序图。Petri网可以用一种自然的方式为并发约束建模，常被用来为系统规范和设计建模。通常，UML交互图被用来描述直接由用户提供的需求，而Petri网图被用来为由领域和技术专家合成的工作流建模。所以，保持两个模型一致性对保证软件质量是必须和重要的。本论文提出的基于场景的并发模型存在性检验系统，可以解决petri网建模的并发系统和UML交互模型表示的基于场景的规范的存在性­一致性检验问题。

为探究基于场景的并发模型存在性检验工具实现的解决方案，本文首先探讨了相关的概念，工具和技术。本工具采用java开发，采用MVC设计模式，采用java swing技术制作了跨平台的界面。由Visual Paradigm工具生成UML交互概览图，导出xml文件作为工具的输入文件之一。由PIPE工具生成Petri网，导出xml文件作为工具输入文件之二。采用xml文件解析技术解析输入文件将UML交互概览图和Petri网读入内存。验证存在一致性算法采用深度优先遍历搜索IOD图，如果在Petri网中找到一条相应的路径则验证成功，如果没找到则验证失败。通过两个实例验证了本工具的正确性和高效性。

引用：

胡军, 于笑丰, 张岩, 等. 基于场景构件式实时软件设计的一致性检验[J]. 软件学报, 2006, 17(1): 48-58.

Robert Sedgewick，Kevin Wayne，谢路云，算法（第4版），人民邮电出版社，2012

Class diagram：https://www.wikiwand.com/en/Class\_diagram

基于场景的并发模型存在性检验

摘要

场景被广泛得被用于必须技术因为他可以描述具体交互，并因此更容易被用户和领域专家使用。在这篇论文中，我们使用UML（统一建模语言）交互模型作为基于场景的规范，包含了UML2.0的交互概览图（IODs）和顺序图。Petri网图作为一个强有力的形式，可以以一种自然的方式为并发约束建模，并常常被用来为系统规范和设计建模。通常，UML交互图被用来描述直接由用户提供的需求，而Petri网图被用来为由领域和技术专家合成的工作流建模。所以，保持两个模型一致性对保证软件质量是必须和重要的。

在这篇论文中，我们用Petri网图来为并发系统建模，并且为验证UML交互模型表示的基于场景的规格来检查此Petri网图。我们开发一种算法来解决两个验证问题，一是存在性一致性检验，意味着一个IOD图描绘的场景必须在Petri网图上发生，或者任何IOD图不允许的场景一定不允许在Petri网图上发生。二是强制性一致性检验，意味着在Petri网图中描述的相关场景可以立即在IOD中找到对应的场景。

关键词：软件验证，形式化方法，基于场景的规格，交互概览图，Petri网图

1. 引言

场景被广泛得被用于必须技术因为他可以描述具体交互，并因此更容易被用户和领域专家使用。基于场景的规格，比如消息队列图，UML交互概览图，提供了一个描述系统需求的直观可见的方式。他们在描述规范和系统设计方面正在扮演一个越来越重要的角色。这些规格专注于实时和分布式系统中实体之间的信息交换。

由于统一建模语言（UML）在1997年成为OMG的一个标准，UML交互模型，像顺序图和活动图正变成软件开发过程的一个主要工具。作为活动图的一个变体，以顺序图为元素的UML2.0交互概览图专注于流控制的概览和实体之间的信息交互。由于描述多场景的需求很常见，我们可以相信交互概览图会因为其强大而灵活地表达性而变得流行。Petri网图是一个正式并且图形的语言，很适合来为有并发和资源分享的系统建模。有很多Petri网图的应用来为系统规格和设计建模，所以我们通常在系统规格和设计时同时需要UML交互模型和Petri网图。通常，UML交互图和Petri网图被用在不同的软件开发过程。即使用在同一个过程，比如软件需求分析过程，UML交互模型被通常用来描述直接由用户提供的基于场景的需求，而Petri图被用来为领域和技术专家合成的工作流建模。所以，为软件质量保证来保持两种模型的一致性是必要以及重要的。

在这片论文中，我们讨论检查由Petri网图建模的系统设计和由UML交互模型表达的基于场景的规格之间的一致性。我们开发了算法来解决两个验证问题，存在性一致性检验和强制性一致性检验。

存在性一致性检验是用来检查是否一个由给定IOD描述的场景一定会在Petri网图中发生。存在性一致性关乎到规格和验证的安全性。

强制性一致性检验是用来检查一个强制性一致性规格在Petri中是否符合，意味着如果一个给定顺序图描述的相关场景在Petri网图中出现，可以立即在IOD中找到对应的场景。我们来考虑以下三种强制性一致性：

1. 前向强制性一致性： 如果一个给定顺序图D描述的相关场景在Petri网图中发生，那么一个IOD的G一定立即跟着。
2. 后向强制性一致性： 如果
3. 双向强制性一致性：

就像存在性一致性，强制性一致性在规格和验证中也很流行。强制性一致性检验不只是直接由客户和领域专家给出，也可以来自测试和验证过程。比如，假设我们想要找到和一个相关场景有关的一些错误，也就意味着场景在系统中实现得不正确。如果实现得不正确，相应场景不会发生，所以对我们来说很困难来决定什么地方和什么时候一个场景应该发生来进一步找到相关的错误。在这个例子中，我们可以将场景分解为两个部分，包含一个强制性一致性规格，并用一个部分作为参考场景，测试并验证是否在另一个部分存在错误，如图3所示。

这篇论文是这样安排的。在下一节，我们正式介绍以场景为基础的规格。在第三节，我们回顾Petri网图的定义和一些基础性质。存在性一致性检验和强制性一致性检验的解决方案分别在第四第五节给出。第六节给出了一个案例研究，并简单描述了验证工具原型其中本文的算法被实现。相关工作和总结在最后一节。

2 基于场景的规格

2.1 顺序图

一个UML顺序图描述一个交互，是由在一个影响期望操作和结果的协作中的对象间交互的信息组成的集合，其重点在于信息流的时间顺序。在这篇文章中，我们只用一种简单版本的顺序图，描绘一个没有任何选择和循环的场景。这篇文章中考虑的顺序图有两个维度垂直维度表示时间，水平维度表示不同对象。每个对象被分配了一列，信息是水平标注的箭头。

比如，如图4描绘的顺序图，描述了一个著名的铁路十字路系统场景的例子。这个系统操作铁路十字路口的栏杆，在其中有一个铁路十字路口传感器和一个栏杆控制器。当传感器检测到火车正在靠近，它发送一个消息来降低栏杆。在火车驶离这个十字路口后，传感器发送一条消息来升高栏杆。

在顺序图中，消息指的是消息发送和接受。顺序图的语义主要包含消息接受和发送的顺序。在一条路径中事件的顺序（比如消息的接收和发送）是由顺序图中每个对象之间的控制流决定的可见的部分顺序和接受发送消息时间间的松散依赖来推断出得。依照【13】，为了不失一般性，我们认为在下面几种情况下事件e在事件e’之前。

1. 因果关系 发送事件e在相应接受事件e’前
2. 可控性 在一个对象上，e’是发送事件，事件e在e’之上。这个顺序说明了发送事件可以等待其他事件先发生。另一方面，我们有时候没法控制接收事件的顺序。
3. 先进先出顺序 在一个对象上的接收事件e在发送事件e’之上，并且相应地发送事件也是e1在e1’之上

为了给出基于场景规范的正式定义，我们如下规范化顺序图：