【文章编号】 1004-1540(2010)02-0124-06

UML 时间顺序图的自动验证技术

陈 江[1,2],陈建国[1],陆慧娟[1],唐文彬[1]
(1.中国计量学院 信息工程学院,浙江 杭州 310018;
2.浙江网新恒天软件技术有限公司,浙江 杭州 310030)

【摘 要】 UML 顺序图反映对象之间的消息交互顺序,在系统建模中应用十分广泛.对顺序图进行时间扩展得到 UML 时间顺序图,使其具备对实时系统建模的能力.在此基础上研究了 UML 建模工具和模型验证工具 UPPAAL 的接口信息,将 UML 时间顺序图模型转化为时间自动机模型,并对该系统模型进行形式化验证.设计和实现了基于 XML 的 UML 时间顺序图自动验证工具.

【关键词】 UML 时间顺序图;自动验证;系统建模
【中图分类号】 TP18 【文献标识码】 A

The automated verification technology of UML time sequence diagram

CHEN Jiang[1,2], CHEN Jian-guo[1], LU Hui-juan[1], TANG Wen-bin[1]
(1. College of Information Engineering, China Jiliang University, Hangzhou 310018, China;
2. Insigma Hengtian Software Ltd, Hangzhou 310030, China)

Abstract: The UML sequence diagram was widely used in the system modeling reflects the sequence of message interaction in objects. To extend a sequence diagram by the time feature, the UML time sequence diagram was the capacity for real－time system modeling. On the basis of the interface information between the UML modeling tool and the model verification tool UPPAAL, the UML time sequence diagram transferd into timed automata model and performs the formal verification to the system model. Based on the XML, automatic verification tool for UML time sequence diagram are designed and implemented.

Key words: UML time sequence diagram; automatic verification; system modelin

随着计算机技术的飞速发展,实时系统的应用越来越广泛.实时系统中的一点疏漏可能导致灾难性后果,因此确保该类系统正确性和可靠性至关重要.为保证系统设计的准确性,对系统模型进行形式化验证很有必要.模型检验是一种确保设计规范正确性的形式化验证技术,其本质就是建立系统的有穷状态模型,对该模型的状态空间进行搜索,确定系统是否满足预期的目标[1].

【收稿日期】 2010-04-06
【基金项目】 浙江省科技厅重大科技专项(No.2007C13091)
【作者简介】 陈 江(1984－),男,浙江台州人,硕士研究生.主要研究方向为系统建模和模型检测.

自从 OMG(对象管理组织)于 1997 年正式发布 UML 以后,大量商用 UML 建模 CASE 工具出现,其中几种最常用的 UML 建模工具功能见表 1.目前市面上很多建模开发工具能够支持包括建模和代码的自动生成和自动测试,但缺乏对所建模型的形式化验证功能[2,3].在文献[4]中 Zonghua Gu 和 Kang G. Shin 提出用 UPPAAL 工具来对时间 Petri 网建模的系统进行验证,前提是要将 Petri 网转换为时间自动机.为了能够在使用 UML 进行系统设计开发的过程中把形式化验证工作结合进来,文献[5-7]提出对 UML 顺序图对象自动机的构造,通过各个对象的自动机模型来描述整个系统.在文献[8]中提出扩展 UML 顺序图实现对实时系统建模,由于 UML 建模工具缺乏对模型形式化验证功能,因而无法实现对模型的自动验证[9].本文在此基础上研究了 UML 建模工具和UPPAAL模型验证工具的接口信息,提出一种基于 XML 的 UML 时间顺序图模型的自动验证方法.该方法把时间顺序图转化为时间自动机描述的模型,然后启用模型验证工具 UPPAAL 进行验证,再将验证结果反馈回系统设计人员,帮助设计人员尽早发现错误,提高模型的可信度,以减少由于设计错误带来的损失.

表 1　UML 建模工具比较

Table 1　Comparison of UML modeling tools

工具功能	Rational Rose	VisualParadigm for UML	OCL Parse	Argo UML
代码测试	支持	支持	支持	支持
双向功程	支持	支持	支持	支持
OCL 语言	不支持	不支持	支持	支持
UML2.0	支持	支持	支持	支持
形式化验证	不支持	不支持	不支持	不支持

1　UML 简介及形式化定义

UML 作为面向对象的标准建模语言,它统一了 Booch、Rumbaugh 和 Jacobson 的表示方法,UML2.0 中新增加了组合片段等新特性.UML 经过多年的研究、发展并不断完善,已成为诸多领域内建模的首选标准.UML 有用例图、类图、状态图、顺序图、合作图、构件图、配置图、组件图和活动图 9 类图,能够很好地支持系统建模的各个阶段.UML 虽然为软件开发提供了一种标准的建模语言,但 UML 用于实时系统建模还存在时间描述方面的缺陷.UML 提供构造型、标记值和约束三种扩展机制,来增加新构造块、创建新特性和详述新语义.本文用标记值来扩展顺序图的时间信息得到时间顺序图,如图 1.

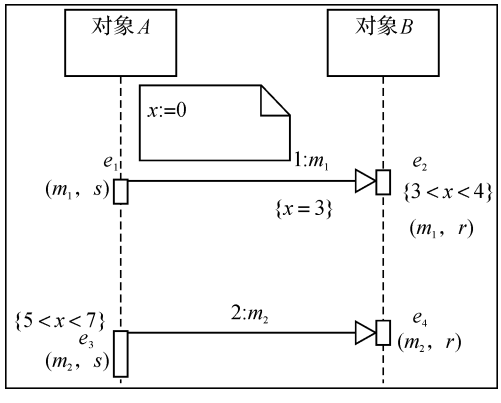

图 1　时间顺序图

Figure 1　Time sequence diagram

本文以图 1 的时间顺序图为例对其进行形式化定义并对时间约束信息进行描述.

本文将顺序图定义为六元组,具体定义如下:

$SD = < O, M, E, T, \prec, \text{obj} >$.

其中:

• $O = \{O_1, O_2 \cdots O_m\}$ 是顺序图中的有穷对象集合.

• $M = \{M_1 M_2 \cdots M_n\}$ 是顺序图上描述的有穷消息的集合.

• $E \subseteq M \times \{s, r\}$ 是发送消息和接收消息的事件的集合.对于消息 m,发送事件用$\{m, s\}$ 表示,接收事件用$\{m, r\}$ 表示.S 表示发送事件的集合,发送事件用 $m!$表示,R 表示接收事件的集合,接收事件用 $m?$表示,$m! \in S, m? \in R, S \cap R = \Phi, S \cup R = E$.

• $T = \{\text{Alt}, \text{Par}, \cdots\}$,$T$ 为顺序图中的组合片段集合,其中 Alt 为选择组合片段,Par 为并发组合片段,T 可以为空集(表示该顺序图不带组合片段).

• $\prec$ 为顺序图中事件之间的可视关系.

• obj 是从 E 到 O 的函数关系,如 $\text{obj}(e) \in O$ 表示事件 e 所对应的对象.对象 O_i 上所有事件的集合用 E_i 来表示 $E_i = \{e \mid e \in E \wedge \text{obj}(e) = O_i\}$.

图 1 的时间顺序图部分形式化表示:

$O = \{$对象 A,对象 $B\}$;

$M=\{m_1,m_2,m_3,m_4,m_5\}$;

$E=\{(m_1,s),(m_1,r),(m_2,s)(m_2,r)\}$;

时间顺序图定义:

$T\_SD=<SD,C>$

其中 SD 为顺序图定义,C 是关于事件的时间不等式:

$$a\leqslant c_0(e_0-e'_0)+c_1(e_1-e'_1)+\cdots+c_n(e_n-e'_n)\leqslant b$$

对于任意事件 e_j 和 e_k,其时间间隔 $\delta(e_j,e_k)$ 形式化描述为:

$$\delta(e_j,e_k)=\begin{cases}t_{k+1}+t_{k+2}+\cdots+t_j(j>k)\\-(t_{j+1}+t_{j+2}+\cdots+t_k)(j<k)\\0(j=k)\end{cases}$$

图 1 的时间顺序图时间约束信息描述:

• $0\leqslant(m_1,r)-(m_1,s)\leqslant 3$,表示消息 m_1 的传递时间在 3 个单位时间之内.

• $1\leqslant(m_2,s)-(m_1,s)\leqslant 4$,表示对象 A 发送消息 m_1 后,必须等待 1 到 4 个单位时间,才能发送消息 m_2.

• $0\leqslant[(m_2,r)-(m_2,s)]-2[(m_1,r)-(m_1,s)]<\infty$

表示对象 A 消息 m_2 的传递时间至少是消息 m_1 传递时间的 2 倍.

2 时间顺序图自动验证的接口分析

2.1 时间顺序图接口分析

OMG 为了能让 UML 模型能在 Internet 上传输并达到共享,将 UML 和 XML 结合,形成 XMI 标准. XMI(XML 元数据交换)是一种基于 MOF(元对象设施)元数据交换格式,用来在不同的 UML 建模工具之间或基于 MOF 的元数据仓库之间进行数据交换,实现共享模型. XML 是一种具有描述数据功能的元标记语言,用于在 web 上组织、发布各种信息,主要具有可扩展性、灵活性和自描述性重要特性[3]. 图 2 显示了三个标准之间的关系.

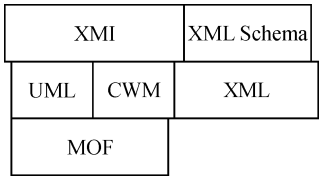

图 2 各标准之间的关系

Figure 2 Relationship between the various standards

本文以铁路控制系统 UML 时间顺序图 XML 模型为例,见图 3.

图 3 铁路控制系统 XML 模型

Figure 3 XML model of railway control system

2.2 UPPAAL 接口分析

UPPAAL 中时间自动机模型[10]以 XML 文件格式进行存储,本文以 UPPAAL 中的自带的火车控制系统为例进行分析. 时间自动机模型的 XML 文件格式分为三个层次,本文以第一层次结构为例进行研究,如图 4.

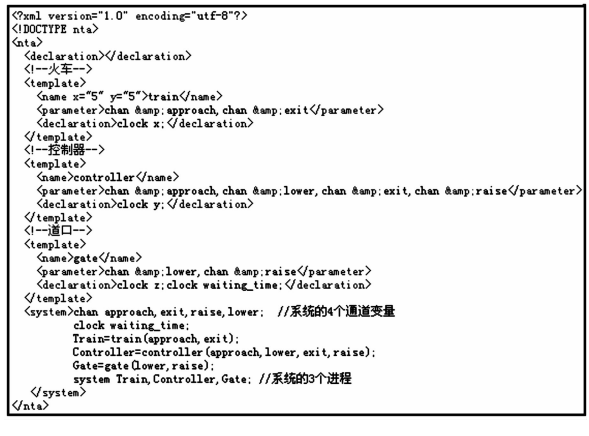

图 4 XML 文档的第一层次结构

Figure 4 First-level structure of XML documents

时间自动机模型的 XML 文件格式第一层次包括<declaration>、<template>、<system>三个子部分. 其中,<declaration>是建模时的全局声明. <system>是系统声明,该系统声明了通道变量 chan approach, exit, raise, lower 和时钟变量 clock

waiting_time. ＜template＞是建立的对象自动机模板，对象自动机个数和＜template＞个数相一致，如图 4 中的模板 train、模板 controller 和模板 gate.

3　时间顺序图的自动验证研究

3.1　模型转化

时间顺序图的自动验证方法通过匹配 UML 时间顺序图接口和 UPPAAL 中时间自动机的接口，将 UML 时间顺序图模型转化为 UPPAAL 中的时间自动机模型，对转化得到的时间自动机模型进行验证. 该方法的具体实施流程如图 5.

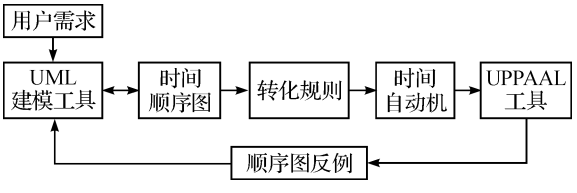

图 5　UML 时间顺序图的自动验证流程

Figure 5　Automatic verification process of UML time sequence diagram

时间顺序图模型到时间自动机模型的转化工作主要分以下两步骤：

1）提取 UML 时间顺序图中的信息

提取 UML 时间顺序图中的对模型验证有用的信息，提取出时间顺序图的对象、消息、系统全局声明、时间约束等信息，具体的算法设计如图 6.

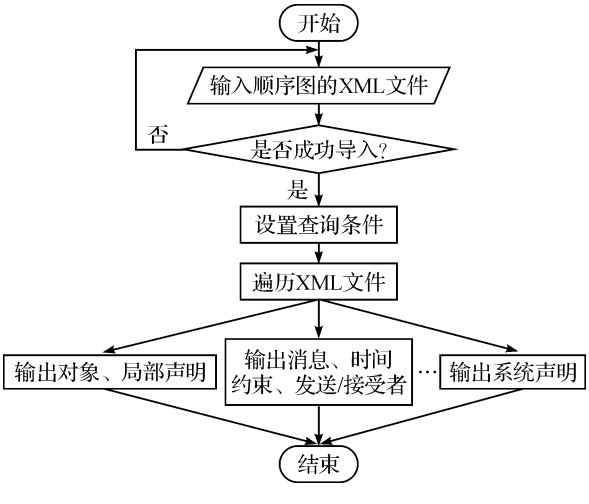

图 6　时间顺序图信息提取算法

Figure 6　Information extraction algorithms of time sequence diagram

2）构建 UPPAAL 中时间自动机验证程序

根据 UPPAAL 中时间自动机的接口信息，构建时间自动机验证程序. 具体的算法框图如图 7.

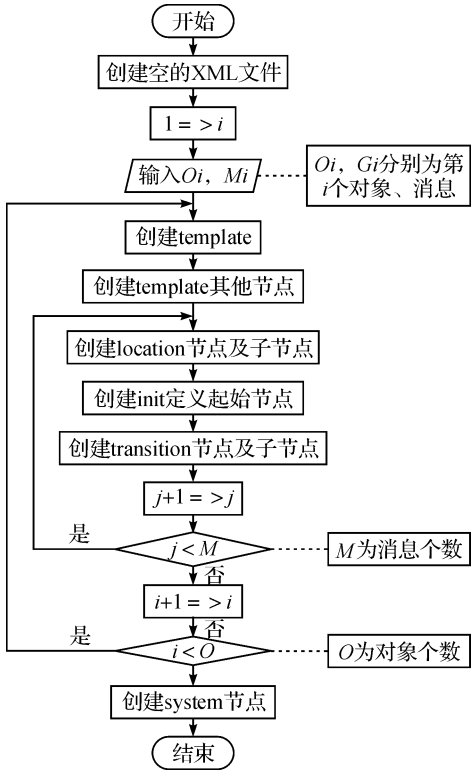

图 7　时间自动机验证程序生成算法

Figure 7　Generating algorithm of timed automata verification program

为规范转化，本文对 UML 时间顺序图中的相关信息和 UPPAAL 中时间自动机模型中的信息的映射关系定义见表 2.

表 2　映射对应关系

Table 2　Mapping correspondence

UML 中的 XML 信息	UPPAAL 中的 XML 信息	注释
project	nta	XML 的根
Model. ModelProperties	system	系统声明
Model. InteractionLifeLine	template	自动机模板
Model. ChildModelProperties	Template. declaration	局部声明
Model. displayModelType. Message	Template. transition	模型的链接
Model. FromEnd. ModelRef. id	Template. source	源节点
Model. ToEnd. ModelRef. id	Template. target	目标节点

3.2　反例故障诊断

模型转化结束后，启用 UPPAAL 工具对转化得到的时间自动机模型进行验证，时间顺序图的模型验证有两个用途：

1)启用 UPPAAL 工具的模拟器,对转化得到的时间自动机模型中的相关执行路径进行模拟,有助于在系统开发的早期过程中结合系统的调试及验证,图 8 为系统执行路径模拟.

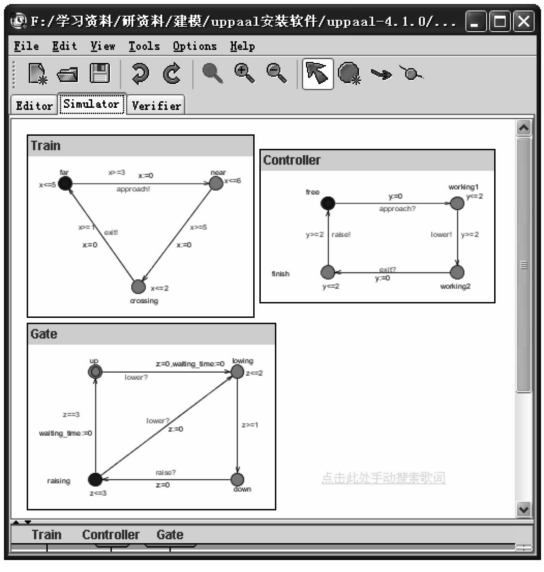

图 8 系统执行路径模拟

Figure 8 Simulation of the system execution path

2)UPPAAL 的查询语言由状态公式和路径公式组成,在 UPPAAL 验证器中,输入查询语言对希望验证的性质进行验证. 如果 UPPAAL 模拟器中检测出否定结果,则表明系统中必定存在产生该否定结果的路径,将错误信息反馈至 UML 时间顺序图. 在 UML 时间顺序图中对产生该错误的执行路径进行调整,重新对模型进行模拟和性质验证,直到模型性质满足用户需求为止.

4 自动验证工具的设计及实现

4.1 自动验证工具体系结构

图 9 显示了时间顺序图自动验证工具的体系结构.将转化得到的模型存到 UPPAAL 工具的安装目录的 demo 文件夹中,使用 LTL 公式验证表达的性质.

4.2 模块实现

本文以出租车调度系统为例进行研究,其时间顺序图如图 10.

根据本文 **3.1** 小节所提到的 UML 事件顺序图模型到事件自动机模型转化的步骤 1 提取出时间顺序图顺序图中对象、消息事件、时间等相关信息,见图 11.

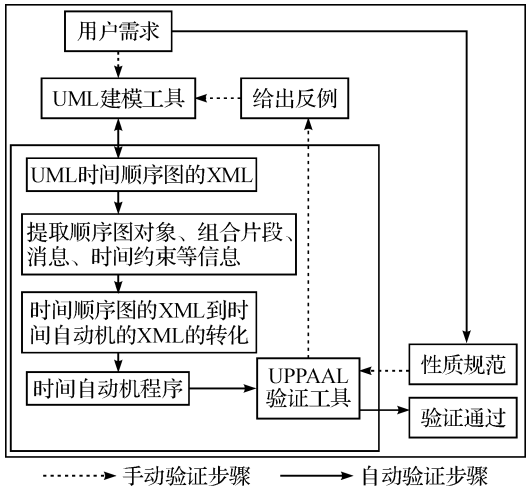

图 9 自动验证工具体系结构

Figure 9 Architecture of automatic verification tool

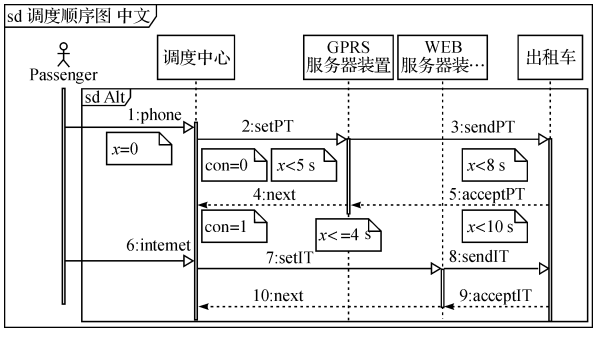

图 10 调度流程 UML 时间顺序图

Figure 10 UML time sequence diagram of scheduling process

图 11 时间顺序图信息提取

Figure 11 Information extraction of time sequence diagram

根据步骤 2 中提到的时间自动机验证程序生成算法,生成的时间自动机验证程序见图 12,得到的时间自动机模型见图 13、14、15、16.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta>
<nta>
  <declaration>// Place global declarations here.</declaration>
  <template>
    <name x="5" y="5">taxi</name>
    <parameter>bool &amp;con, chan &amp;sendPT, chan &amp;acceptPT, chan &amp;sendIT, chan &amp;acceptIT</parameter>
    <declaration>clock x;</declaration>
    <location id="id0" x="-680" y="-376">
      <name x="-696" y="-344">IT_accept</name>
      <urgent/>
    </location>
    <location id="id1" x="-864" y="-368">
      <name x="-904" y="-352">PT_accept</name>
      <urgent/>
    </location>
    <location id="id2" x="-576" y="-488">
      <name x="-586" y="-518">IT_receive</name>
      <label kind="invariant" x="-586" y="-473">x&lt;=14</label>
    </location>
    <location id="id3" x="-936" y="-488">
      <name x="-1016" y="-512">PT_receive</name>
      <label kind="invariant" x="-946" y="-473">x&lt;=13</label>
    </location>
    <location id="id4" x="-760" y="-528">
      <name x="-792" y="-568">Str_T</name>
    </location>
    <init ref="id4"/>
```

图 12　时间自动机验证程序

Figure 12　Verification program of timed automata

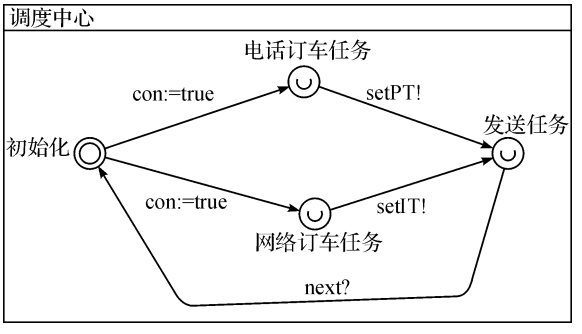

图 13　对象 Center 的时间自动机模型

Figure 13　Timed automata model of center

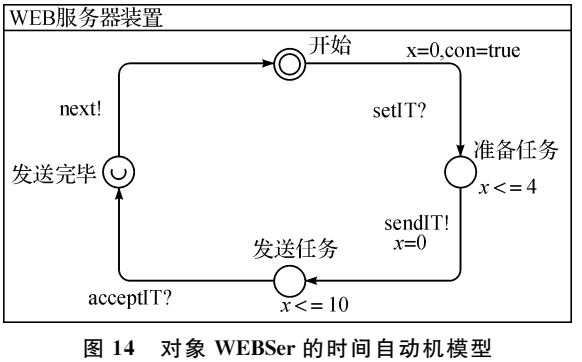

图 14　对象 WEBSer 的时间自动机模型

Figure 14　Timed automata model of WEBSer

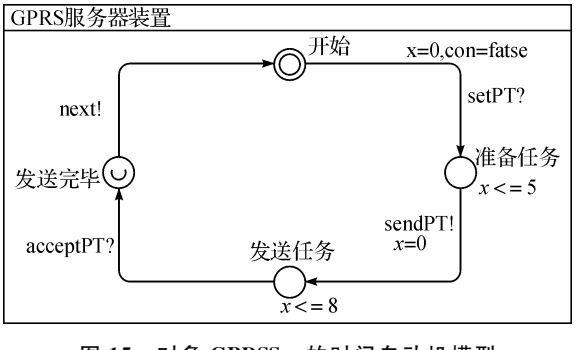

图 15　对象 GPRSSer 的时间自动机模型

Figure 15　Timed automata model of GPRSSer

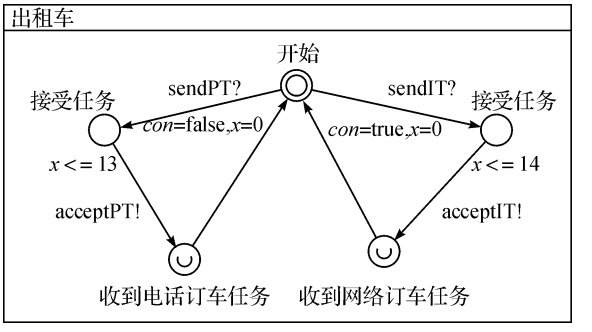

图 16　对象 Taxi 的时间自动机模型

Figure 16　Timed automata model of taxi

5　结　语

本文在对 UML 建模工具充分调研的基础上，提出了基于 XML 的 UML 时间顺序图的自动验证方法，实现了将 UML 时间顺序图模型自动转化为UPPAAL中的时间自动机模型，然后启用 UPPAAL 工具对转化得到的时间自动机模型进行验证，并在此研究的基础上，设计和实现了基于 XML 的 UML 时间顺序图自动验证工具，从而为解决目前市面上 UML 建模工具缺乏形式化验证功能提供了良好的解决方案.

【参　考　文　献】

[1] CLARKE E M, WING J M. Formal methods: State of the art and future directions[J]. ACM Computing Surveys, 1996,28(4):626-643.

[2] ODONI A R. The flow management problem in air traffic control[C]// Proceedings of Flow Control of Congested Networks. Berlin: Springer, 1987:269-298.

[3] CLARK T. UML-the unified modeling language lecture notes in computer science[M]. [S. L.]: Springer, 2002:503-517.

[4] GU Z H, SHIN K G. An integrated approach to modeling and analysis of embedded real-time systems based on timed petri nets[C]// Proceeding of 23rd International Conference on Distributed Computing Systems (ICDCS'03). California: Springer, 2003:1063-6927.

[5] 郭　华，庄　雷，张习勇. UPPAAL-一种适合自动验证实时系统的工具[J]. 微计算机信息，2006，22(5)：52-54.

[6] 刘传会，戎　玫，张广泉. UML2.0 顺序图的一种有穷自动机模型[J]. 计算机工程与科学，2008，30(12)：118-121.

[7] 王建光，段　富. 一种 UML 模型到 XML 模型的转换方法[J]. 计算机技术与发展，2007，17(7)：123-126.

[8] 陈　江，陈建国，陆慧娟，等. UML 时间顺序图的实时系统建模及验证[J]. 中国计量学院学报，2010，21(1)：46-51.

[9] 龚嘉宇，李宣东，郑国梁. UML 时间顺序图的可达性分析[J]. 计算机科学，2005，32(6)：169-175.

[10] 周　娟，蒋登峰. 基于 Matlab 的 ADC 自动测试系统开发[J]. 中国计量学院学报，2008，19(3)：220-224.