基于通信的列车控制系统可信构造：形式化方法研究

鲍勇翔[1], 陈铭松[1], 孙海英[1], 繆炜恺[1] ,陈小红[1] ,周庭梁[2]

[1] *华东师范大学 计算机科学与软件工程学院*

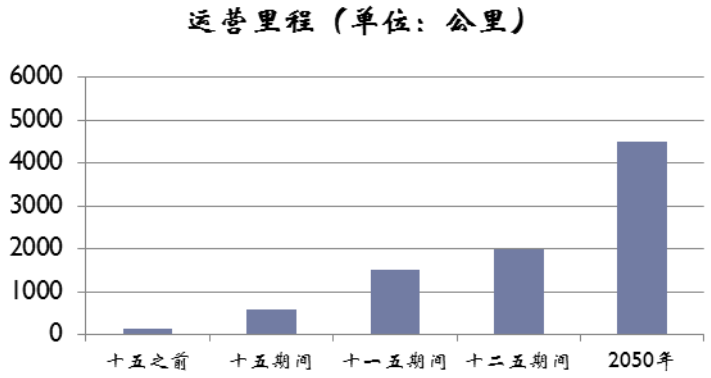
[2] *卡斯柯信号有限公司（上海）*

内容提要

- 背景介绍
- CBTC需求规约可信建模和验证
- CBTC设计形式化建模及验证
- CBTC实现的测试
- CBTC多层次可信构造的方法与工具平台
- 总结和展望

背景介绍

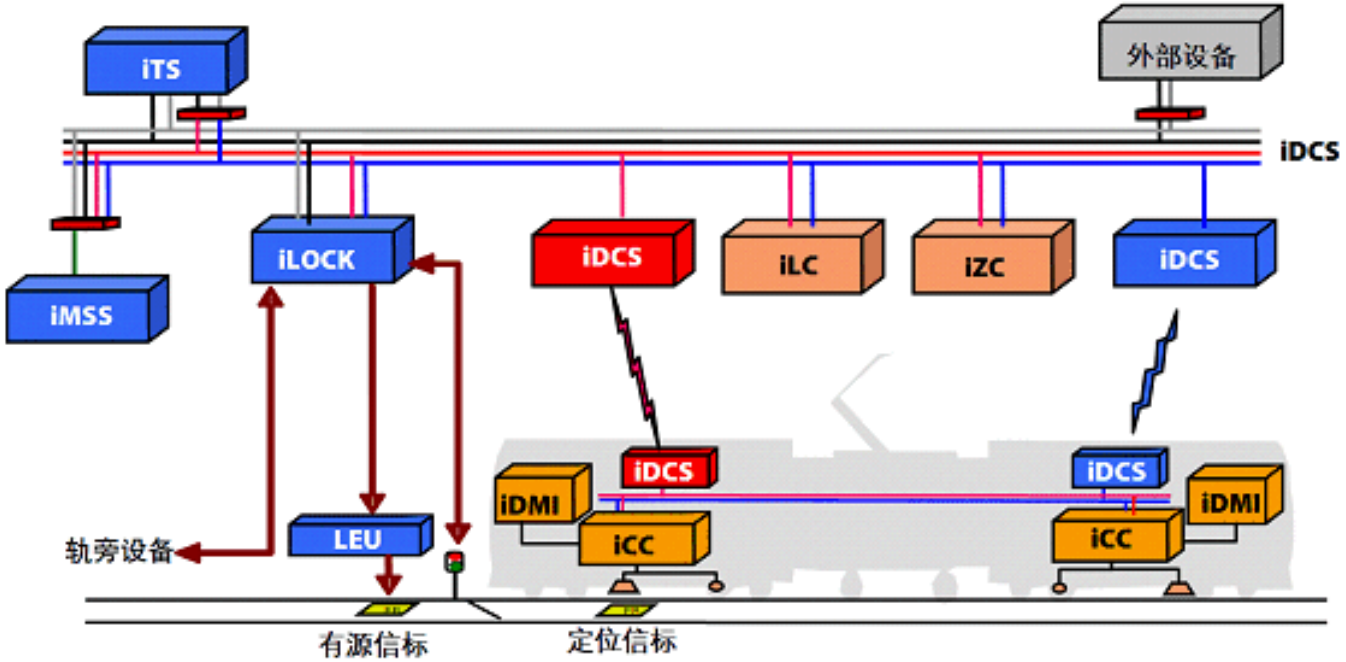
- 城市轨道交通正进入发展高峰期
  - 战略意义重大，推动城市化进程
  - 经济效益巨大，节能环保，绿色出行
  - 社会效益显著，有效降低城市交通拥塞



- 基于无线通信的列车自动控制系统CBTC
  - 世界范围内轨道交通信号的主流标准制式
  - 实现移动闭塞，减小列车运行间隔，提高系统运能
  - 控制更加灵活与精确，提高了系统运行的效率与安全性
  - 自主可控率低，售后服务环节存在缺陷，核心技术受制于人
- 可信的CBTC技术成为了轨道交通行业健康发展与降低运营成本的主要瓶颈！

CBTC简介

- 典型的CBTC系统由联锁系统CBI，列车自动控制系统ATC，数据通信系统DCS，列车自动监控系统ATS组成。



智能型基于无线通信和移动闭塞的列车控制系统iCMTC框架图
（来源：卡斯柯信号有限公司）

背景介绍

- CBTC涉及到计算、通信与控制三个方面的高频实时协作,设计异常复杂,可信难以保证。



2009年12月22日，上海地铁1号线因信号系统发出错误控制指令导致制动距离不足，发生列车侧撞事故。

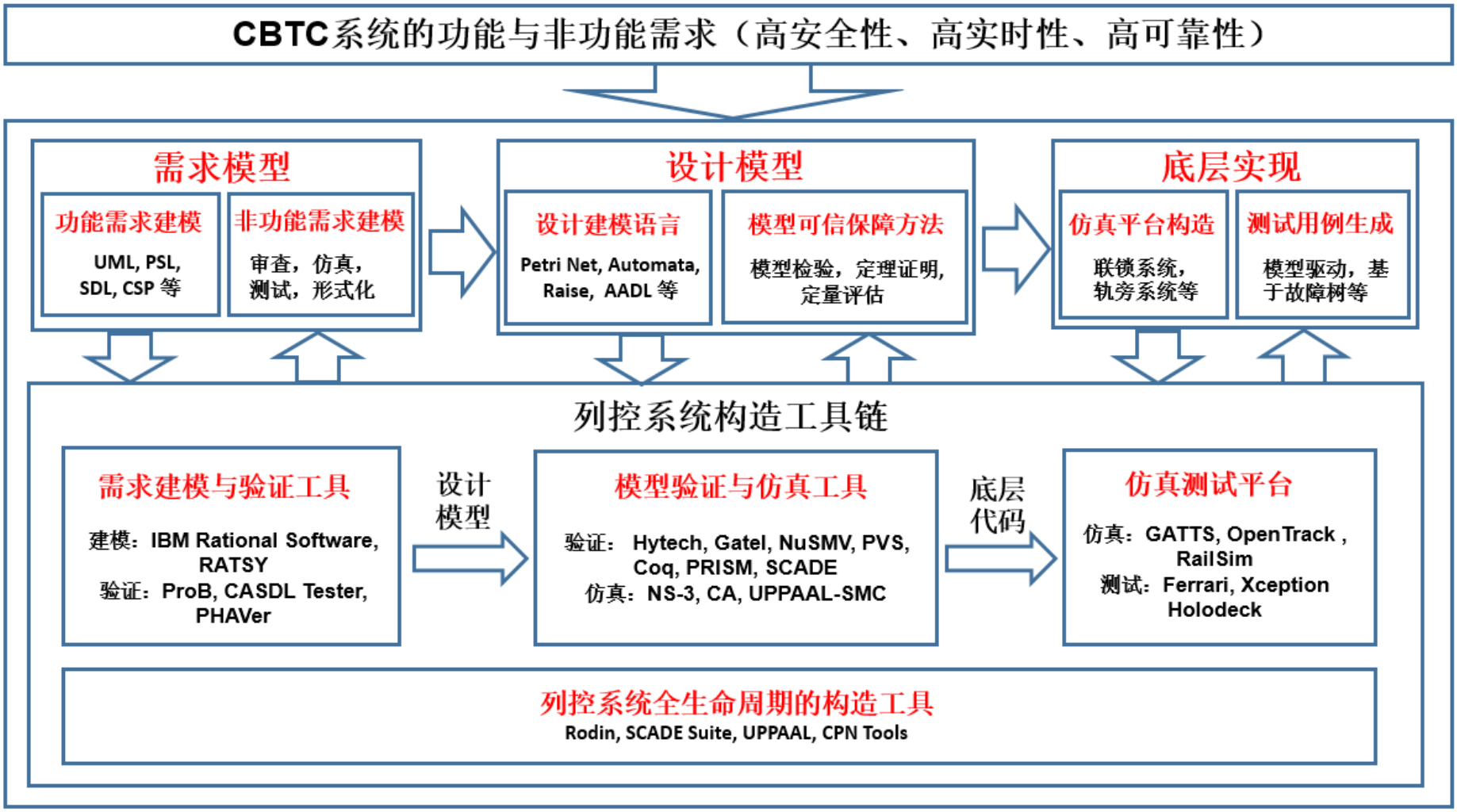
2011年9月27日，上海地铁10号线因信号故障，导致上海豫园路站两辆列车追尾。

2012年11月1日，由于乘客随身携带的Wi-Fi干扰了地铁信号的车-地通信，导致深圳地铁列车急停。

2014年5月2日，韩国首尔地铁2号线由于自动安全距离保持装置出现故障，导致170余名乘客受伤。

EN50128标准明确表示强烈推荐以形式化方法来引导CBTC系统的可信构建

基于形式化方法的CBTC通用可信构造框架



内容提要

- 背景介绍
- CBTC需求规约可信建模和验证
- CBTC设计形式化建模及验证
- CBTC实现的测试
- CBTC多层次可信构造的方法与工具平台
- 总结和展望

CBTC功能需求建模方法

- 基于统一建模语言(UML)的方法
  - 顺序图：轨旁子系统和车载子系统消息交换建模
  - 状态图：列控系统的模式转化建模，移动需求授权管理

- 基于属性描述语言(PSL)的方法
  - 形式化的需求模型：描述列控系统行为和环境行为
  - 断言属性集：描述的列控系统行为必须满足的属性
  - 允许属性集：描述列控系统行为中允许的属性
  - 可以直接运用RATSY(Requirements Analysis Tool with Synthesis)工具仿真

CBTC功能需求建模方法

- 基于规约描述语言(SDL)的方法
  - 支持自顶向下的系统功能行为建模
  - 从系统级定义了车载设备的功能接口和数据交换格式
  - 从模块级描述了模块功能的交互场景和数据

- 基于逻辑的方法
  - B方法对轨旁子系统的数据进行确认的过程进行了建模
  - Z方法对欧洲列控系统的无线闭塞中心的描述

- 基于进程代数的方法
  - 采用混成CSP对CTCS-3每一个基本操作场景进行建模

CBTC非功能需求建模方法

- 安全性需求
  - 安全性需求一般采用基于时序逻辑的方法描述
  - CNL语言集成了LTL、正则表达式、一阶逻辑和混成等建模元素，支持CBTC系统进行安全建模

例如，“任意两辆列车不能同时出现在同一列轨位置”，可以表示为：

```
“for all Train t1, t2,
such that t1 != t2 then never(t1.position = t2.position)”
```

- 实时性需求
  - 定性约束,如列车门必须在列车开启之前关闭
  - 定量约束,如CBTC中列车与RBC间的通信都必须要在5毫秒内完成

需求建模方法比较及支持工具

需求建模方法	功能需求	安全性需求	实时性需求	优点	建模工具
基于UML的方法	√	×	√	图形化，直观	Rational Rose, PowerDesigner
基于PSL的方法	√	√	×	可直接被仿真，使用方便	RATSY
基于SDL的方法	√	×	×	支持自顶向下的系统行为建模	Pragma Dev Studio, Rational SDL Suite
基于逻辑的方法	√	√	√	通过逻辑推理实现系统的功能精化与正确性证明	OVADO
基于进程代数的方法	√	√	√	支持描述并发行为	PAT

需求可信保障方法比较

需求可信保障方法	优点	适用系统规模	缺点	对应建模方法	工具	可信保障程度
审查	易于执行	各种规模均适用	依赖于审查者经验,自动化程度低	自然语言等	IBM Requisite Pro	低
仿真	可以直观地观察系统运行过程	各种规模均适用	仿真耗时长	基于属性描述语言等	ProB, UPPAAL	中
测试	工程化程度高	各种规模均适用	无法发现所有错误	基于UML等	CASDL Tester	中
形式化方法	自动化程度高,验证结果完备	小规模	需要专业知识, 状态空间爆炸	基于逻辑公式、自动机等	PHAVer, ARMC	高

内容提要

- 背景介绍
- CBTC需求规约可信建模和验证
- CBTC设计形式化建模及验证
- CBTC实现的测试
- CBTC多层次可信构造的方法与工具平台
- 总结和展望

CBTC设计形式化建模

设计形式化建模包括结构与行为建模两部分

结构模型为静态模型，主要描述系统的模块组成关系

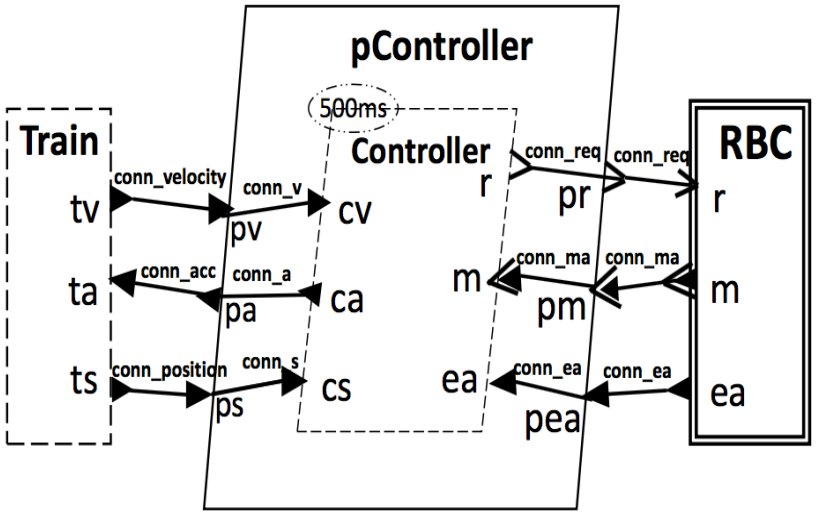
行为模型为动态模型，主要描述系统的运行行为

基于AADL架构的结构建模

- AADL（Architecture Analysis & Design Language）是一种体系结构描述语言,可用于列控系统的架构建模

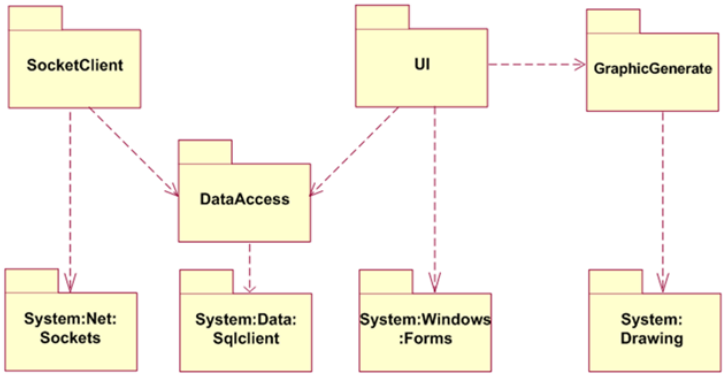
- 提供附件(Annex)建模扩展机制
  - 混成附件 - 列车
  - 行为附件 - 列控中的控制器

- 开源工具平台OSATE
  - http://osate.github.io

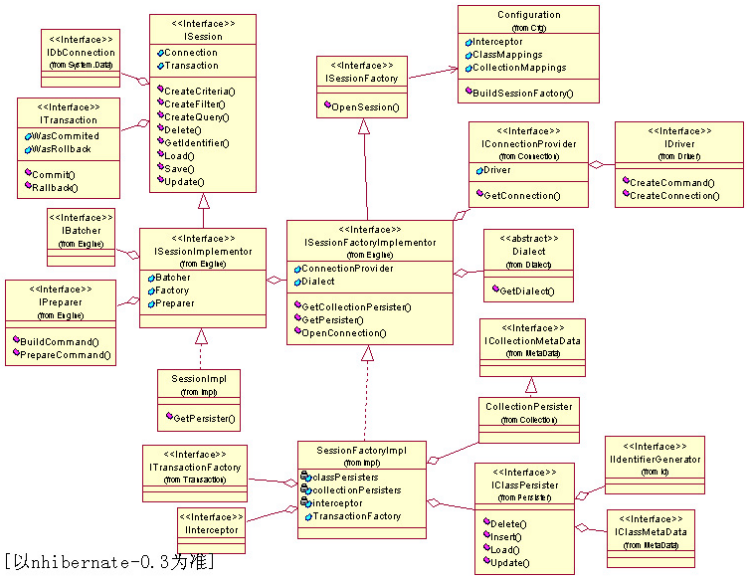


基于UML的结构建模

包图用于组织层次结构，可用于CBTC架构设计



类图用于定义组成实体，从静态角度对 CBTC 进行结构描述



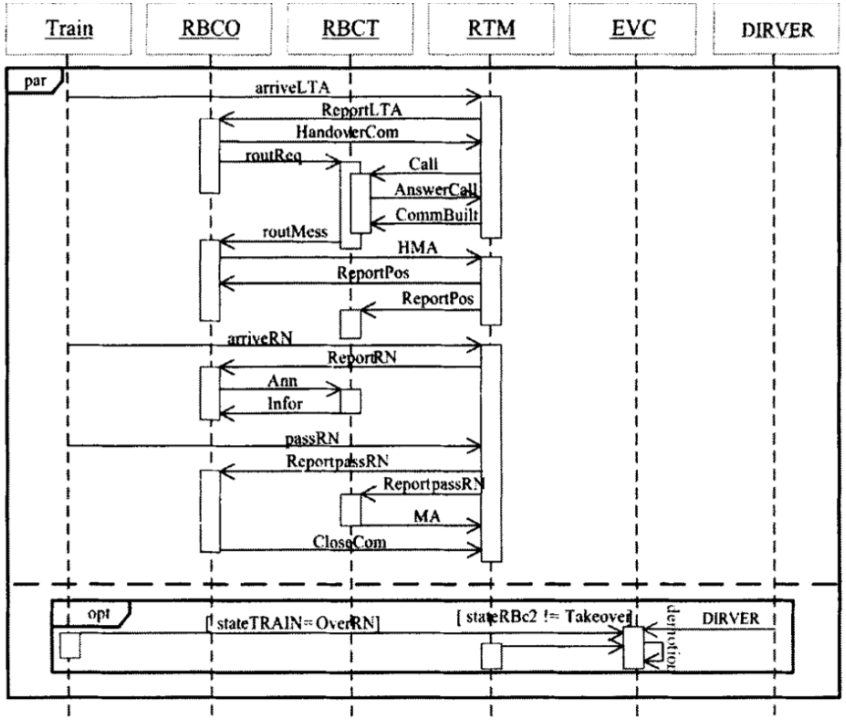
[以nhibernate-0.3为准]

CBTC系统设计的行为建模

- 行为建模用于对CBTC组件的交互和协同行为建模，目前主流的方式有
  - 基于UML的方法
  - 基于自动机的方法
  - 基于Petri 网的建模方法
  - 基于马尔科夫决策过程的建模方法
  - 基于RAISE的建模方法

基于UML的行为建模方法

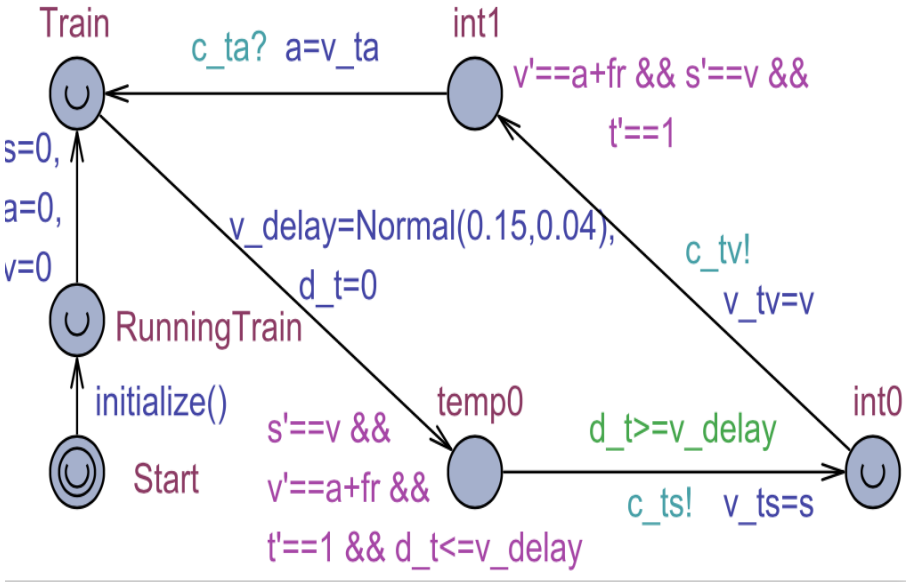
- 交互图（包括顺序图与协作图）
  - 例如，利用UML顺序图描述了RBC切换场景中各个组件间的交互过程
- 状态图
- 活动图



基于自动机的行为建模方法

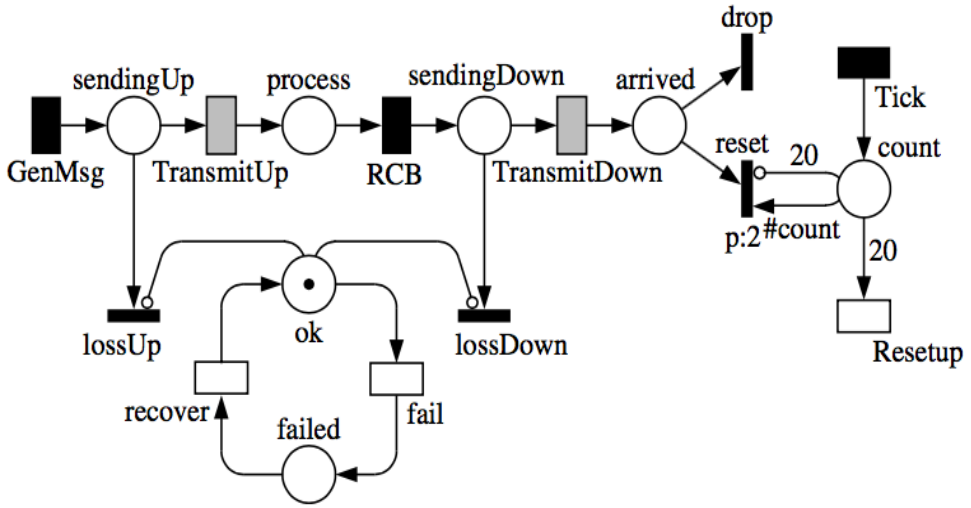
- 为了应对CBTC的某些特征,常常使用扩展的自动机对CBTC建模
  - 混成性 => 混成自动机
  - 高实时性 => 时间自动机

建模工具SCADE、Simulink、UPPAAL其底层核心计算模型采用的都是自动机或其变种



基于Petri 网的行为建模方法

- Petri 网常被用作对列控系统中的并发行为进行建模，例如
  - 列车调度操作建模
  - 联锁和信号系统行为建模
  - 相邻的RBC 区域边间切换，如下图所示



基于扩展的Petri网的行为建模

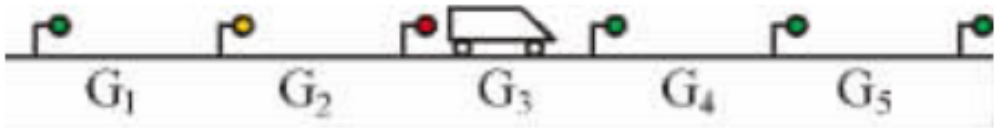
- 为了提高对系统的并发性建模能力，丹麦Aarhus大学的Kurt Jensen提出了着色Petri网（Colored Petri Net, CPN）
  - 实现了欧洲列车控制系统功能方面的形式化描述
  - 支持从顶层需求到底层实现的逐步精化

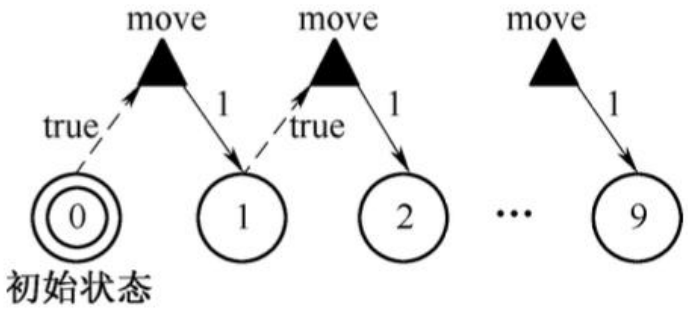
- 应用随机Petri网（SPN）支持对系统中的不确定性进行建模
  - 火车和无线闭塞中心传输的位置和移动授权数据丢失情况

基于马尔科夫决策过程的建模方法

由于列控系统某一时刻系统行为概率分布并不依赖于之前的历史状态而只与当前状态有关,这种行为符合马尔科夫的**无记忆性**,因此非常适合采用MDP进行建模.

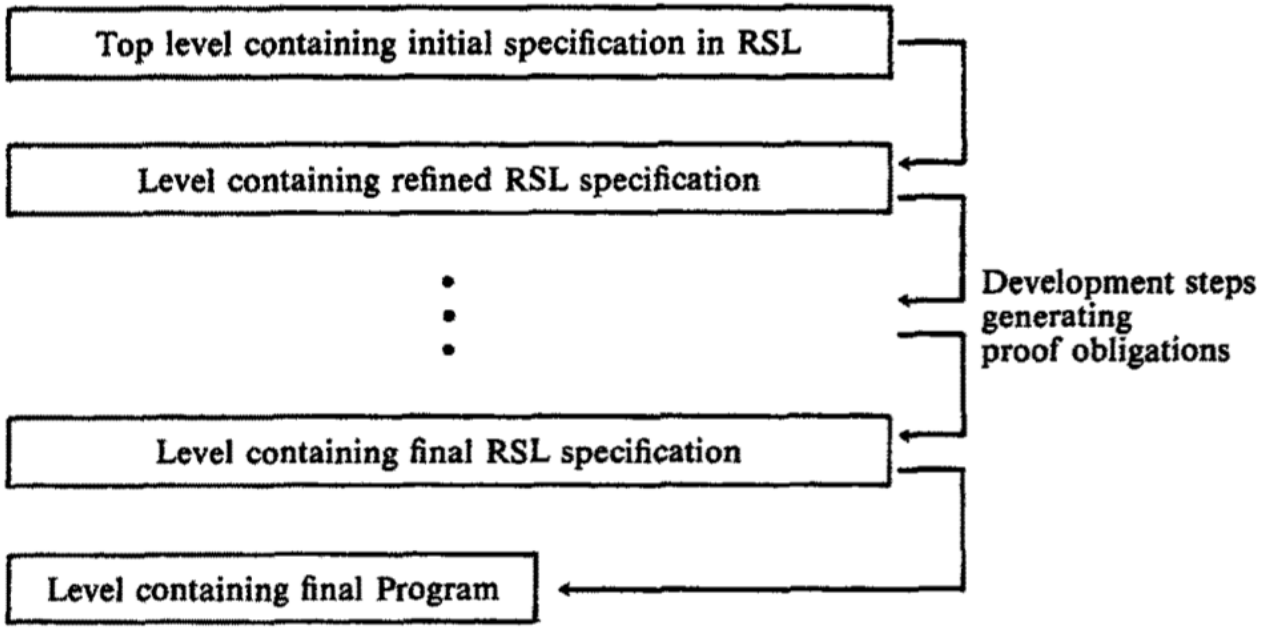
例如列车通过区间场景如图所示. 区间轨道区段共5段，从左至右依次为G1到G5. 列车物理运动过程是一个 MDP.





基于RAISE的建模方法

- 支持从最初的抽象规约描述到设计层次建模
- 能够对正在运行的多个任务进行描述，很好的支持并发和分布式的特性的建模



建模方法比较与工具支持

模型/语言/方法	实时性建模	并发性建模	不确定环境建模	混成性建模	优点	建模工具
基于自动机的方法	√	√	√	√	图形化，支持仿真	UPPAAL, Simulink
基于Petri Net的方法	√	√	√	×	很强的并发建模能力	CPN Tools
基于AADL的方法	×	×	×	√	支持层次化建模	OSATE
马尔科夫过程	×	√	√	×	支持并发建模	Matlab, PLASMA
基于RAISE方法	√	√	×	×	支持从需求到设计的精化	Eden, rsltc
基于UML的方法	√	√	×	√	图形化,应用广泛	Rational Software

设计层次的可信保障方法

- 基于仿真确认的方法
  - 基于事件的仿真（例如，基于CPN的连锁系统仿真）
  - 基于时间的仿真（例如，基于TA的RBC子系统仿真）
- 基于形式化验证的方法
  - 模型检验（例如，RBC验证，SCADE数据流模型验证）
  - 定理证明（例如，基于混成霍尔逻辑的CTCS-3验证）
- 基于定量评估的方法
  - 贝叶斯网络（例如，列控系统进行了量化分析）
  - 统计模型检验（例如，RBC的优化控制策略评估）
  - 不确定时间活动图（例如，CBTC自动控制子系统建模）

设计模型的可信保障方法比较

类别	可验证性质类型	手段	形式化程度	保障手段	模型表达能力	自动化程度	状态空间爆炸	常用工具
仿真确认	可靠性，安全性，实时性	仿真	低	建模+执行	强	高	无	UPPAAL, Simulink, Modelica, SCADE
形式化验证	安全性，实时性	模型检验	高	建模+验证	弱	高	存在	CPN, UPPAAL, CHARON, NuSMV, SCADE
		定理证明	高	建模+证明	强	低	存在	Coq, Rodin
定量分析	可靠性，实时性	概率模型检验	高	建模+验证	弱	高	存在	PRISM
		统计模型检验	中	建模+执行	强	高	无	UPPAAL-SMC, PLASMA

内容提要

- 背景介绍
- CBTC需求规约可信建模和验证
- CBTC设计形式化建模及验证
- CBTC实现的测试
- 可信构造的方法与工具平台
- 总结和展望

CBTC实现的测试

- 面对规模庞大、逻辑复杂的CBTC系统，形式化验证难以在有限的时间内保证系统的正确性
- 测试是面向系统上下文运行环境的动态验证方法，其核心问题是高效的测试生成
- 通常采用形式化程度较高的描述语言对测试对象进行建模并构建定制化的测试方法
  - 基于标号迁移系统的测试
  - 基于时间自动机的测试
  - 基于UML模型的测试
  - 基于故障模式的测试
  - ……

基于标号迁移系统的测试生成

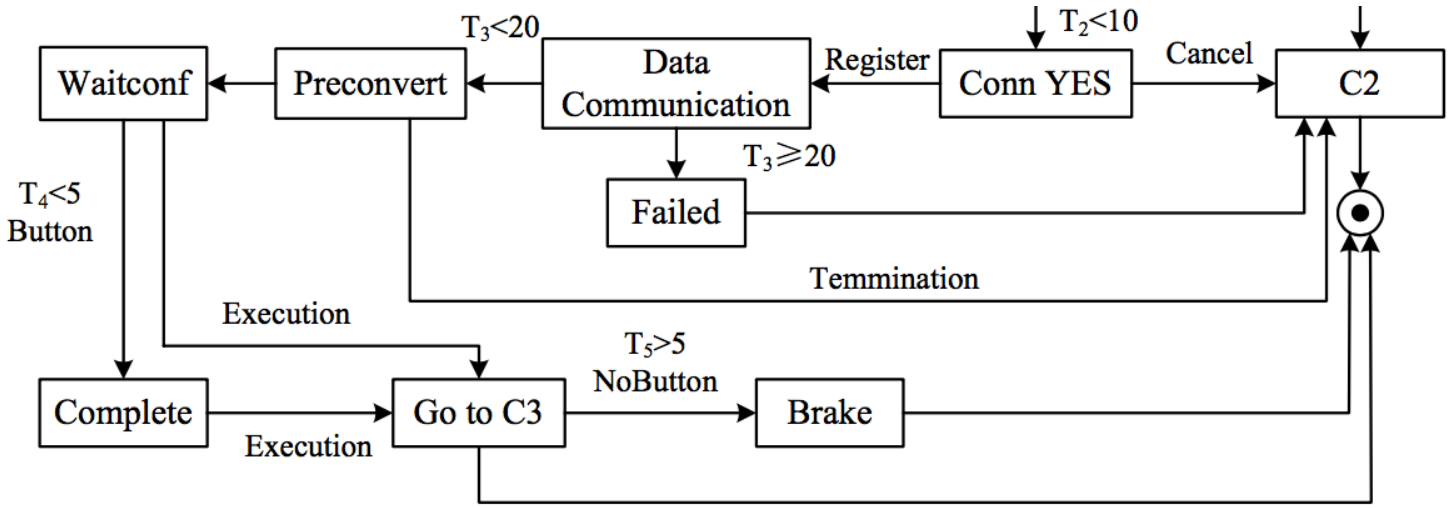
- 标号迁移系统（Labeled Transition System）是EN50128推荐使用的各形式化语言的通用语义模型
- LTS通过系统状态集合和状态之间带有标签的迁移关系定义系统运行时行为
- 输入输出标号迁移系统IOLTS (Input Output Labeled Transition System)是LTS在测试领域的变体，它能区分测试输入和输出
- 基于标号迁移系统测试生成的理论基础是定义系统规约与被测系统之间的一致性关系

基于时间自动机的测试生成

- 作为典型的硬实时系统，验证系统功能**是否能够在期望的时限内正确地输出**是CBTC系统测试的核心需求
- 基于时间自动机的测试生成通常会采用时间自动机的测试变体－时间输入输出自动机TIOA (Timed Input Output Automata)
  - 将经典TA的动作集合划分为不相交的输入和输出动作集合
  - 基于环境的时间受限输入输出一致性关系是测试生成的理论基础
- UPPAAL-TRON和UPPAAL Cover是具有代表性的基于TIOA实现的实时一致性测试工具时间受限输入输出一致性关系

基于UML模型的测试生成

- 作为工业建模标准，半形式化的UML模型以其直观易学受到了CBTC产业界的青睐



- 利用简路径覆盖思想从UML活动图中提取测试路径
- 利用分类划分方法对功能输入域数据进行分割并生成测试用例

基于故障模式的测试生成

- 故障注入测试在轨道交通信号各系统的测试过程中有着广泛的应用
- 故障注入测试采用人为故意的方式产生可以加速系统失效发生的故障并作用于被测系统以验证系统对所注入故障的响应情况是否符合预期
- 基于变异测试可以有效融合基于规约模型的测试生成与采用故障注入的测试执行方法以提高CBTC系统实现的覆盖度

CBTC系统实现测试方法比较

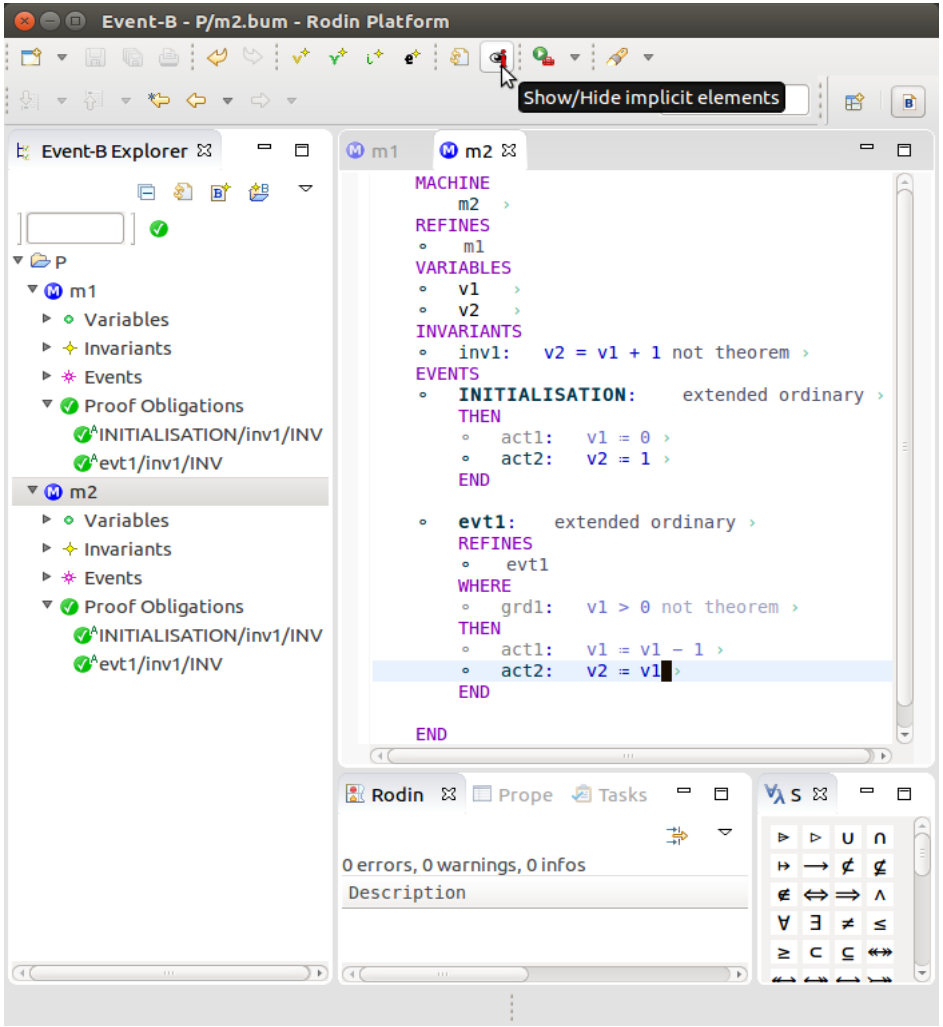
方法名称	针对系统行为特征	形式化程度	典型应用场景
基于标号迁移系统的测试生成	安全性,并发性	高	列车自动防护系统
基于时间自动机的测试生成	实时性	高	联锁软件、车载控制系统
基于UML模型的测试生成	功能与场景正确性	中	列车自动控制系统
基于故障模式的测试生成	故障行为	低	列车定位系统、区域控制器

内容提要

- 背景介绍
- CBTC需求规约可信建模和验证
- CBTC设计形式化建模及验证
- CBTC实现的测试
- CBTC多层次可信构造的方法与工具平台
- 总结和展望

Rodin

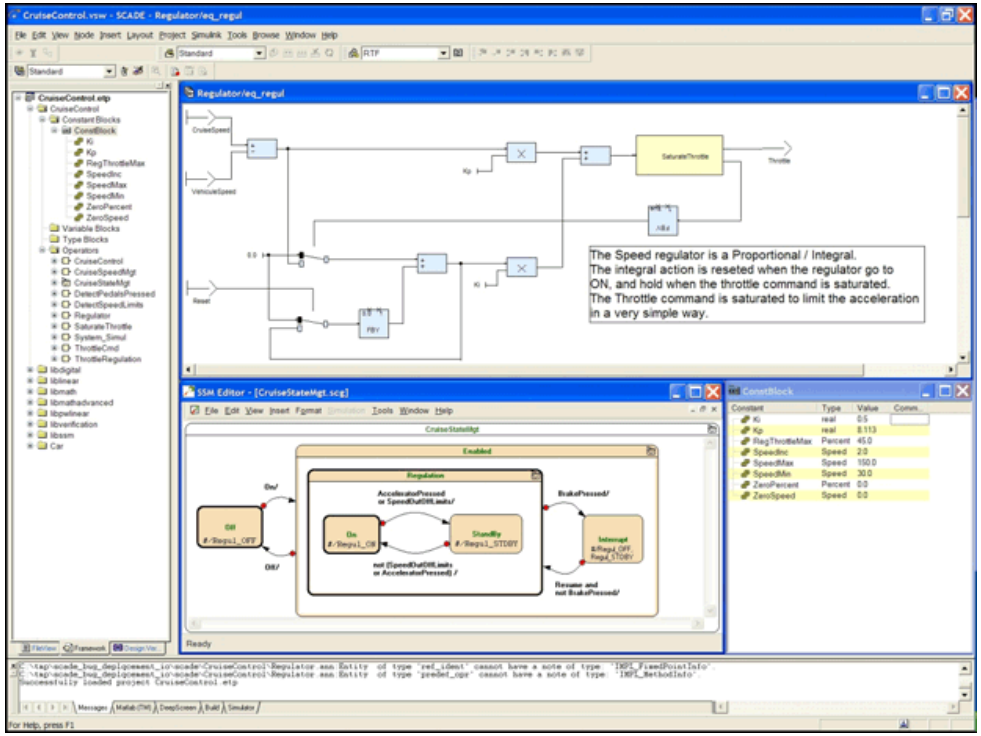
- Rodin是支持Event-B的建模与验证平台.
- 通过逐步引入状态变量和约束对模型进行精化，同时在精化过程中通过证明义务（proof obligation）保证各层次之间的一致性
- 轨道交通领域中被使用得最为广泛的形式化建模方法之一（Thales的联锁系统、Siemens公的区域控制系统等）



SCADE Suite

- SCADE套件 提供了包括图形化建模、形式化验证、代码自动生成、测试、仿真等全流程工具的支持

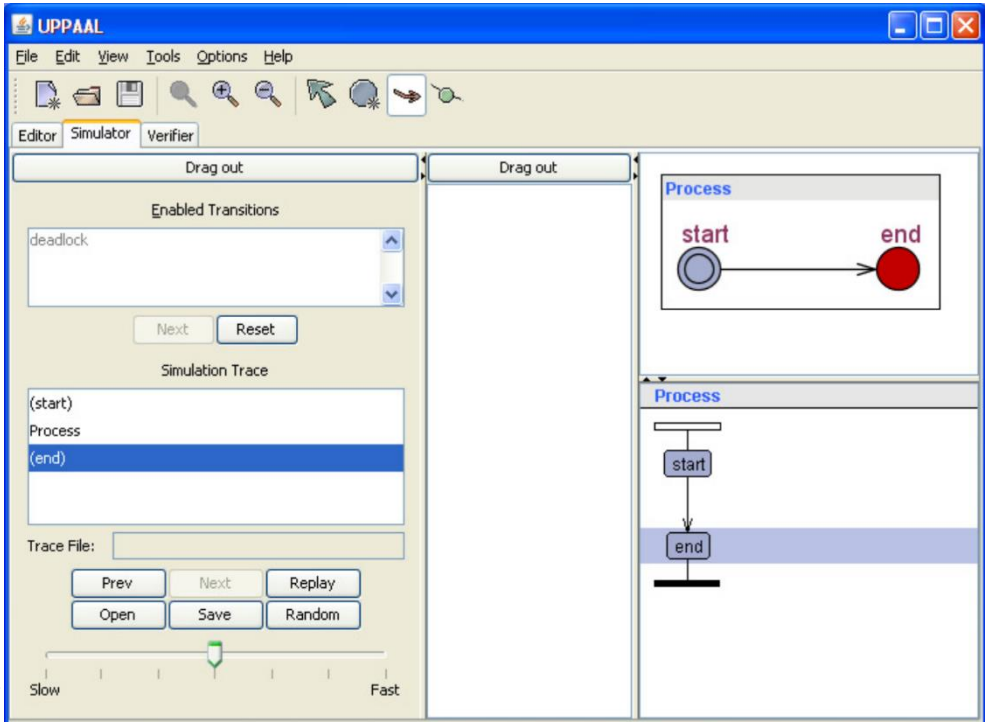
- 支核心是同步语言Lustre
- SCADE可以自动生成满足EN50128规范的代码
- 对目前CBTC的开发流程支持最好的工具



UPPAAL

- 以时间自动机为基础的针对实时系统的形式化集成开发环境
- 包括图形化建模、模型仿真、验证器

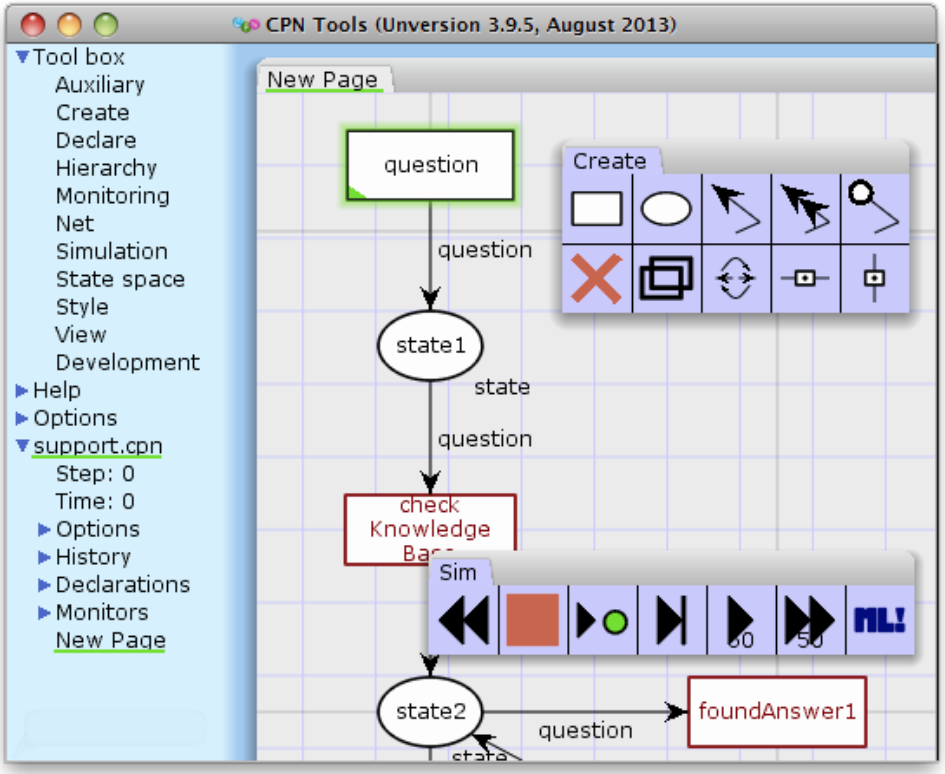
- 支持时间属性建模
- 具有图形化的系统建模仿真用户界面
- 支持自动化模型检查



CPN 工具

- 用于编辑，模拟和分析着色Petri网的工具
- 它包含模拟器和状态可达分析工具

- 支持真正的并发而非交错式并发的建模
- 支持层次化的并行行为建模
- 支持系统级模型的图形化建模、仿真、分析和验证



可信构造的方法与工具平台

工具名称	主要用途	可信验证类型	形式化程度	典型应用场景
Rodin	需求，设计阶段	安全性验证	高	联锁系统、车载系统、区域控制系统
SCADE 套件	设计，实现阶段	安全性验证	低	联锁软件、列车自动控制和防护软件
UPPAAL	需求，设计阶段	安全性，实时性验证	高	区域控制器、列车自动控制系统
CPN 工具	需求，设计阶段	安全性验证	中	列车车载软件、区域控制器

内容提要

- 背景介绍
- CBTC需求规约可信建模和验证
- CBTC设计形式化建模及验证
- CBTC实现的测试
- CBTC多层次可信构造的方法与工具平台
- 总结和展望

总结与展望

- 工业界花费在CBTC设计、开发与验证上的代价巨大. 在竞争激烈的CBTC行业,如何在保证所构造的CBTC的可靠性的同时又能行之有效地减少可信构造的代价,已经成为CBTC研发的严峻挑战
- 由于系统自身的复杂性以及所处环境的复杂,CBTC在开发过程中依然还存在如下问题:
  - 缺少不确定环境下需求的建模与评估方法，需求模型与规约的正确性与合理性难以得到保证
  - 在设计阶段，缺乏通用模型转换语言及支持工具，不同模型之间难于集成，形式化验证极易引起“状态空间爆炸”
  - 自动化综合程度低,模型很难自动化生成底层代码
  - 缺乏不同抽象模型之间协同验证的方法，验证结果无法重用,功能的一致性与正确性难以维护

总结与展望

- 随着列车的提速以及特有的复杂运行环境，CBTC可信保障变得愈加困难
- 虽然形式化方法已成为CBTC领域最有前景的系统可信性保障手段并取得了令人鼓舞的成果，但是目前还远未在实际的工业界得到全面应用
- 学术界和工业界还需在理论方法，工具平台以及工程实践等多方面进一步投入，研究如何在保障CBTC可信构造的同时，缓解或避免使用形式化方法所带来的诸多限制

谢谢各位专家，

恳请指正！