软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(6):1638−1653 [doi: 10.13328/j.cnki.jos.005997]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

轨道交通联锁领域特定语言的形式化*

赵梦瑶[1], 陈小红[1], 孙海英[1], 刘 静[1], 陈良育[1], 周庭梁[2]

[1](上海市高可信计算重点实验室(华东师范大学),上海 200062)

[2](卡斯柯信号有限公司,上海 200071)

通讯作者: 陈小红, E-mail: xhchen@sei.ecnu.edu.cn

摘 要: 作为轨道交通系统的核心子系统之一,对联锁系统进行形式化建模与分析,是保证其安全性的重要手段.形式化建模需要领域知识和形式化知识的结合,由于形式化知识难以掌握,领域专家在建模整个过程中都需要形式化专家的帮助.为了解决这个问题,针对联锁系统的故障随机性、行为实时性、构件可重用的特点,提出设计联锁领域特定语言 IS-DSL 描述具体的联锁系统的参数,并基于随机混成自动机模板自动生成联锁系统的形式化模型,以进一步在此基础上进行安全分析.首先对联锁系统模型进行分析,根据不同案例设计其领域特定语言;其次,确定联锁系统的系统模型模板,包括环境构件模板和控制器模板,并举例抽取其随机混成自动机模板;在模板基础上定义系统模型生成过程,让领域专家可以通过领域特定语言,输入参数自动生成具体的随机混成自动机系统模型;最后以某站联锁系统为例,展示了基于模板的具体系统模型的生成过程,并通过基于系统模型的事故预测分析,证明了该方法的可行性与有效性.

关键词: 联锁系统;模板重用;形式化建模;随机混成自动机;领域特定语言

中图法分类号: TP311

中文引用格式: 赵梦瑶,陈小红,孙海英,刘静,陈良育,周庭梁.轨道交通联锁领域特定语言的形式化.软件学报,2020,31(6):1638−1653. http://www.jos.org.cn/1000-9825/5997.htm

英文引用格式: Zhao MY, Chen XH, Sun HY, Liu J, Chen LY, Zhou TL. Formalizing railway interlocking domain specific language. Ruan Jian Xue Bao/Journal of Software, 2020,31(6):1638−1653 (in Chinese). http://www.jos.org.cn/1000-9825/5997.htm

Formalizing Railway Interlocking Domain Specific Language

ZHAO Meng-Yao[1], CHEN Xiao-Hong[1], SUN Hai-Ying[1], LIU Jing[1], CHEN Liang-Yu[1], ZHOU Ting-Liang[2]

[1](Shanghai Key Laboratory of Trustworthy Computing (East China Normal University), Shanghai 200062, China)

[2](Casco Signal Co. Ltd., Shanghai 200071, China)

Abstract: As a core subsystem of the rail transit systems, the formal modeling and analysis of the interlocking system is an important means to ensure its safety. Formalization requires both domain knowledge and formal knowledge. Since formal knowledge is difficult to master, domain experts need the help of formal experts throughout the modeling process. To solve this problem, aiming at the characteristics of fault randomness, real-time behavior, and reusability of components in railway interlocking systems, a specific language IS-DSL is proposed to describe the parameters of specific interlocking system. A formal model of interlocking system is generated automatically based on the stochastic hybrid automata (SHA) templates, to carry out further safety analysis. In this study, the model of

* 基金项目: 国家重点研发计划(2018YFB2101300); 国家自然科学基金(61332008, 91418203, 61672230, 61572195, 11471209, 61802251); 上海市经济和信息化委员会专项资金(160306)

Foundation item: National Key R&D Program of China (2018YFB2101300); National Natural Science Foundation of China (61332008, 91418203, 61672230, 61572195, 11471209, 61802251); Specific Foundation of Shanghai Municipal Commission of Economy and Informatization (160306)

本文由“信息物理系统软件设计自动化”专题特约编辑卜磊教授、陈铭松教授、朱祺教授、刘超教授推荐.

收稿时间: 2019-08-20; 修改时间: 2019-10-23; 采用时间: 2020-01-13

interlocking system is analyzed firstly, and the domain specific language is designed according to different cases. Secondly, the templates of the interlocking system model, including environment component templates and controller template are established, and the SHA templates are extracted as examples. Based on these templates, the system model generation process is defined, so that the domain experts can automatically generate the specific SHA model by inputting parameters through the IS-DSL. Finally, the interlocking system of a station is taken as an example to show the generation process. The following accident prediction analysis based on this system model proves the feasibility and effectiveness of the proposed approach.

Key words: interlocking system; template reuse; formal modeling; stochastic hybrid automata (SHA); domain specific language

轨道交通联锁系统是以计算机、现代多媒体和通信网络技术等技术为手段,以道岔机、信号机和轨道电路作为基础设备,主要负责处理进路内的信号机、道岔、轨道电路之间的安全联锁关系[1].它包括实现联锁关系、建立进路、控制道岔的转换和信号灯的开放以及进路解锁,是轨道交通信号系统中不可或缺的核心部分,是保证列车行车安全的重要子系统.它有欧盟安全完整性等级最高的 SIL4 级安全需求[2]、复杂的逻辑和很高的实时性能,若发生故障,则能危害整列车的安全.因此,必须确保联锁系统的高安全性.

为了保证这样的安全性,目前很多欧洲铁路控制与防护标准,如 EN50128[2],EN50129[3]等强烈推荐在开发之前,使用形式化方法进行建模和分析.形式化方法以严密的数学理论和相关推理为基础,有严格的语法规范和确定的语义定义,并且是自证正确的,因此非常适用于开发联锁系统这样安全攸关的系统.但是,形式化方法又有着固有的学习成本高、不易使用的缺点.一般来说,领域专家的形式化建模离不开形式化专家的帮助,建模的主体依然是形式化的专家.在联锁系统这类领域知识特别复杂的系统中,领域专家需要快速构建模型并分析复杂业务逻辑,根据分析结果快速进行错误定位与错误修正,建模与分析的主体应该是领域专家.如何让领域专家能自主构建形式化模型,是一个需要解决的问题.

现有的轨道交通领域的形式化建模工作主要分为 3 类:基于自然语言的形式化方法、基于半形式化的方法和基于领域特定语言(DSL)的方法.基于自然语言的形式化建模[4−6]都是形式化专家根据领域专家的自然语言描述直接建立形式化模型,并采用相应的形式化验证工具进行性质验证,这类方法需要领域专家和形式化专家的紧密合作,对于不具有形式化知识的领域专家来讲很难使用.基于半形式化的形式化建模与验证中[7−11],先由领域专家使用半形式化语言(例如 UML)描述系统,然后形式化专家再把半形式化模型转换为形式化模型并进行形式化分析.但这些半形式化语言都是软件专业建模语言,不是领域专家所擅长的语言.基于领域特定语言的形式化建模与验证中[12−15],首先由领域专家使用领域特定语言描述系统,然后形式化专家再把领域特定语言模型转换为形式化模型并进行形式化分析.例如,Idani 等人[12]将图形领域定制语言的模型转换为形式化 B 模型.这类方法允许领域专家只要根据领域特定语言给出领域模型,就可以享受形式化方法带来的好处,便于形式化专家和领域专家的合作,也减少了人工建模出错误的可能性.

基于以上分析,我们认为,基于领域特定语言的建模方法更方便领域专家使用.但现有的基于特定语言的方法并不能直接应用于联锁系统,原因如下:

(1) 使用的形式化模型,如文献[12]使用的 B 方法和文献[13]使用的时间弧 Petri 网等,并不能应对联锁系统的故障随机性和行为实时性的特点.联锁系统中,事故通常是由设备故障造成的,而设备故障通常具有随机性的特点,例如由雷击等意外事件导致轨道电路出现短路等故障.同时,联锁系统属于典型的实时系统,逻辑复杂且有较高的实时性要求[16];

(2) 可重用构件多,需要多次重用列车、轨道等构件.联锁系统在各个站的组成都类似,每个站的联锁系统都有着相同的构件,如列车、轨道、道岔、信号灯、联锁表、控制器,它们的主要区别在于车站布局(车站站场图)不同导致的实例个数不同.

因此,在前期工作基础上[14,15],本文提出了基于模板的联锁系统模型生成方法,建立联锁系统的特定领域语言,通过模板重用,允许领域专家根据实际需要自行构建系统模型,而不必接受形式化方法的培训.这种模板可以用能建模故障随机性和实时性的随机混成自动机[17]或价格时间自动机[18]进行构建.

本文首先由领域专家对联锁系统模型进行分析,根据不同案例设计其领域特定语言;其次,由领域专家确定

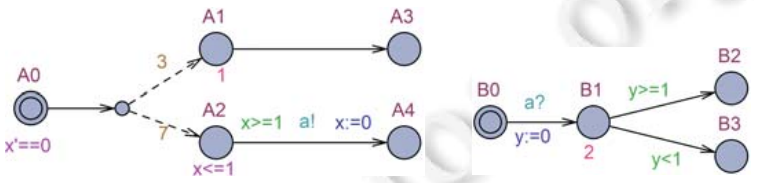
联锁系统模型的模板组成,并由形式化专家抽取系统模型的模板,举例构建其随机混成自动机模板;然后,由领域专家通过领域特定语言输入参数自动生成系统模型.本文以某站联锁系统为例,自动构建了其系统的混成自动机模型,并在验证平台 UPPAAL-SMC 上进行事故预测分析,证明了本文方法的可行性和有效性.

本文第 1 节介绍随机混成自动机概念和 UPPAAL-SMC.第 2 节给出自动生成联锁系统模型的方法框架.第 3 节定义联锁领域特定语言 IS-DSL.第 4 节确定联锁系统的模板组成,并举例用随机混成自动机构建模型模板.第 5 节在模板基础上,根据领域特定语言输入参数自动生成具体的系统模型.第 6 节针对某站的实际情况进行系统的生成和事故预测分析,验证了本文方法的可行性和有效性.第 7 节介绍相关工作.最后,第 8 节总结全文并给出进一步的研究方向.

1 预备知识

本节主要介绍一些概念和知识,主要包括随机混成自动机和 UPPAAL-SMC[19]平台.随机混成自动机是一种被广泛接受的混合系统模型,也是建模时间和随机性的重要模型,已经被广泛应用于计算机仿真、自动机及其他领域[20].随机混成自动机包含了随机特性,在引入随机事件后,更适合于以故障诊断为目的的建模.与传统时间自动机(time automata,简称 TA)[21,22]不同,随机混成自动机提供对离散行为、连续行为和随机行为的描述,对存在随机行为的混成系统能够很好的建模.随机混成自动机的定义是一个七元组 $H=(L,l_0,X,\Sigma,E,F,I)$,其中,L 是有限位置集合;$l_0\in L$ 是开始位置集合;X 是连续变量的有限集;Σ是动作的有限集;E 是迁移的有限集,其中每条迁移边可表示为(l,g,a,φ,l'),其中,l 和 l'表示位置,g 为定义在 R^x 上的谓词,动作标签 $a\in\Sigma,\varphi$是定义在 R^x 上的二元关系;对于每一个位置 $l,F(l)$是其时间延迟函数,$I(l)$是其不变式.

通过广播信道和共享变量,不同的随机混成自动机之间相互通信,组成随机混成自动机网络(network of stochastic hybrid automata,简称 NSHA).图 1 展示了一个由随机混成自动机 A 和 B 组成的随机混成自动机网络.A 有 5 个状态:A0~A4,有一个时钟变量 x.B 有 4 个状态:B0~B3,有一个时钟变量 y.在状态 A0 中,x'==0 表示 x 的值在状态 A0 处没有改变.在状态 A1 中,标志 1 表示 A 在状态 A1 处的延迟遵循λ=1 的指数分布.在状态 A2 中,$x\leqslant 1$ 为不变式,在该状态,时钟 x 始终满足该不变式.实线箭头表示状态之间的转换,在迁移上有“guard”,“update”和“sync”,分别表示迁移条件、变量更新和同步.在状态 A2 到 A4 的迁移上,“guard”为 $x\geqslant 1$,表示当 $x\geqslant 1$ 时迁移才会发生,“update”为 x:=0,表示更新 x 的值为 0.



(a) A (b) B

Fig.1 An example of NSHA

图 1 随机混成自动机网络样例

随机混成自动机支持不确定性操作,例如 A 离开状态 A0 时有两个目标状态,A1 和 A2.随机混成自动机用带有权重信息的虚线箭头表示进入目标状态的概率.例如,从状态 A0 到状态 A1 的概率为 3/(3+7),从状态 A0 到状态 A2 的概率为 7/(3+7).为了使在一个随机混成自动机网络中的两个随机混成自动机进行同步,各随机混成自动机通过广播信道和共享变量相互进行通信.例如在图 1 中,两个随机混成自动机通过广播信道 a 进行通信,a?表示广播信道 a 接收消息,a!表示广播信道 a 发送信息.在进行发送和接收操作的同步之后,两个随机混成自动机同时进入各自的下一状态,如图 1 中所示,状态 A2 转换到状态 A4,状态 B0 转换到状态 B1.

UPPAAL-SMC 是一个工具环境,支持随机混成自动机的建模、仿真与验证,它是 UPPAAL[23]工具链中对统计模型检验(statistical model checking,简称 SMC)[24]支持的扩展.目前,UPPAAL-SMC 在各种研究领域中被广泛

接受,例如轨道交通领域和软件工程领域等[25,26].UPPAAL-SMC 提供了许多时间自动机的随机解释的相关查询,同时允许用户可视化地模拟运行表达式的值.本文在 UPPAAL-SMC 中用到查询语法如下:

$$simulate\ N[\leqslant bound]\ \{E1,\ldots,Ek\}.$$

其中,N 为自然数,表示要进行仿真的次数;$bound$ 为仿真的时间限制;$E1,\ldots,Ek$ 是 k 个表达式,可被监控和可视化.

2 方法框架

为实现列车行车的安全性,联锁系统的首要目的是在允许列车移动的同时,控制道岔和信号灯以防止列车发生相撞或脱轨[27].其主要功能是通过轨道电路监视相关轨道段的占用情况、控制道岔位置、发送信号给列车司机,告知其是否能进入轨道.其通常的工作流程如下:(1) 列车(train)向控制器(controller)发出请求申请进路并等候回馈;(2) 控制器收到列车请求后,查询联锁表(interlock table),联锁表将进路结果返回给控制器;(3) 控制器检查该进路轨道占用情况,轨道(track)收到控制器指令后,检查占用情况,并向控制器返回结果;(4) 控制器对轨道占用情况检测完毕后锁闭道岔(point)并命令道岔移动到相应位置,道岔移动完成后向控制器反馈;(5) 控制器完成对道岔的控制后向信号灯(signal light)发送变绿信号,控制信号灯变绿;(6) 控制器在列车驶出轨道后解锁相应的道岔、控制相应的信号灯变红,整个流程结束.

从上述流程中可知:联锁系统的运行不仅仅只涉及控制器,还涉及列车、轨道、道岔、信号灯和联锁表.其中,列车包含其车载系统,轨道分为多个区段,每个区段与一个轨道电路相关联,用以检测轨道是否被列车占用.道岔是轨道的连接处,根据道岔的不同位置,列车可以驶入不同的进路.道岔可以处于解锁与锁闭两种状态,且若其处于解锁状态,则表明该道岔未与岔口相连.当道岔处于锁闭状态时,列车才能通过该道岔.信号灯位于不同的轨道区段之间,其状态为红灯或者绿灯,表明列车应该停止或允许前进.进路由联锁表定义,一般在铁路站场设计时被创建,每个进路由按拓扑结构顺序连接的轨道段组成,只有在进路建立后,列车才能获准进入该进路.所有的这些一般都可以从一个站场图中获得,例如从图 2 可以看出,该站场包含 2 个道岔($SW1$,$SW2$)、9 个信号灯($S1$~$S9$)与 7 段轨道($T1$~$T7$).

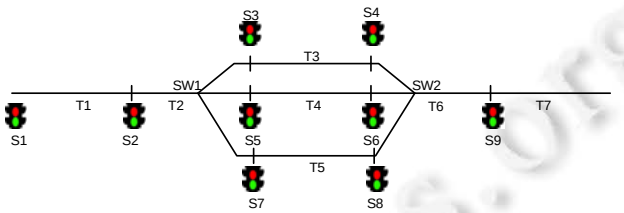


Fig.2 Track layout of a station

图 2 某站的站场图

要建立一个联锁系统的模型,不仅仅是要建模控制器,还需要建模与其交互的环境,这些环境包括多辆列车、多段轨道、多个道岔、多盏信号灯和一个联锁表.在不同的车站,其站场图不同,系统模型也会不同.也就是说,联锁系统在各个站的组成都类似,它们的主要区别在于车站站场图不同导致的实例个数不同.从中也不难发现,虽然构件数量不同,但构件类型是固定的,也就是说,每个联锁系统都会涉及控制器、列车、轨道、道岔、信号灯和联锁表.我们可以将每种类型的构件抽取出来做成模板,进行形式化建模.在模板的形式化建模中,适用的语言要可以应对设备故障的随机性以及较高的实时性要求,可以采用随机混成自动机和价格时间自动机进行建模.

各个联锁系统模型的不同,主要是构件数量的不同,但后面我们也发现有其他的不同.比如,为了安全的目的,需要建模异常,包括设备故障的概率等等.可以将这些不同抽取出来,定义为领域特定语言,由领域专家进行设定,然后可以跟构件模板结合起来形成完整的系统模型,便于后续的安全分析.基于上述分析,我们设计了基于模板的联锁系统生成和应用框架,如图 3 所示.

- 首先,由领域专家根据领域特定语言提供模型参数列表;
- 然后,结合由形式化专家建立的系统模板进行特定系统模型生成.系统模板由两部分组成:环境模板和控制器模板.其中:环境模板包括了联锁系统的每个环境构件的模型,包括火车、道岔、联锁表、信号灯和轨道;控制器模板定义了控制器的行为模式.在结合模板过程中,可以实例化系统中的各个模板,以便生成特定的系统模型;
- 最后在 UPPAAL-SMC 平台上模拟生成的系统模型,获取仿真数据,结合安全性质进行安全分析,以验证系统的安全性.

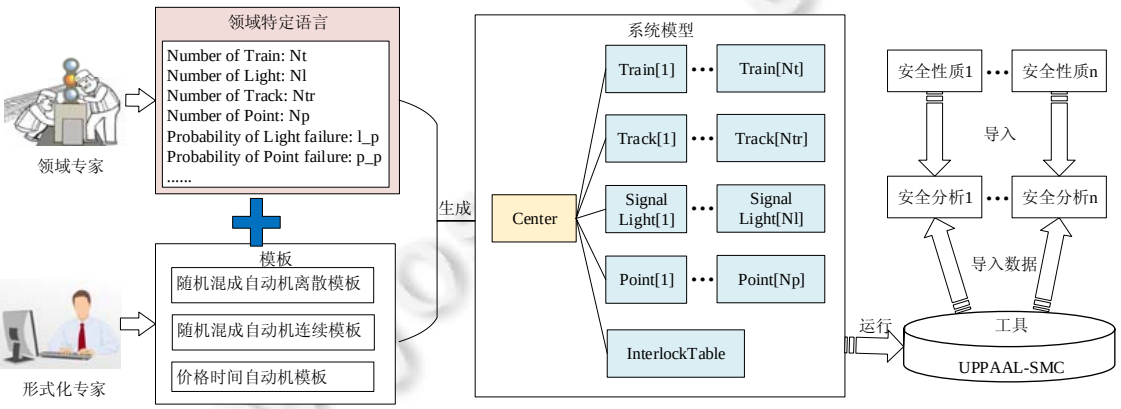


Fig.3 Framework of our approach

图 3 方法框架

3 联锁领域特定语言 IS-DSL

根据上述分析,联锁系统的模型首先在构件的数量上是不同的,也就是说,列车、道岔、信号灯和轨道实例化的个数随案例变化.这些构件的数量要成为语言的一部分.不仅如此,由于这些实体数量的变化,导致其进路表是不同的.假设图 2 这样的站场中包含 3 条进路,*R*1~*R*3,*R*1 由轨道段 *T*1~*T*3,*T*6 和 *T*7 组成,需要信号灯 *S*1~*S*4 和 *S*9 为绿灯,道岔 *SW*1 和 *SW*2 向上打开;*R*2 由轨道段 *T*1,*T*2,*T*4,*T*6 和 *T*7 组成,需要信号灯 *S*1,*S*2,*S*5,*S*6 和 *S*9 为绿灯,道岔 *SW*1 和 *SW*2 向前打开;*R*3 由轨道段 *T*1,*T*2,*T*5~*T*7 组成,需要信号灯 *S*1,*S*2,*S*7~*S*9 为绿灯,道岔 *SW*1 和 *SW*2 向下打开.其进路表见表 1.

Table 1 Interlocking table of a station

表 1 某站的联锁表

Route			Signal light		Point			Track
ID	From	To	Green	Red	Up	Front	Down	
*R*1	*S*1	*S*9	*S*1~*S*4,*S*9	*S*5~*S*8	*SW*1,*SW*2			*T*1~*T*3,*T*6,*T*7
*R*2	*S*1	*S*9	*S*1,*S*2,*S*5,*S*6,*S*9	*S*3,*S*4,*S*7,*S*8		*SW*1,*SW*2		*T*1,*T*2,*T*4,*T*6,*T*7
*R*3	*S*1	*S*9	*S*1,*S*2,*S*7~*S*9	*S*3~*S*6			*SW*1,*SW*2	*T*1,*T*2,*T*5~*T*7

这样的一个联锁表的表示也应该成为领域特定语言的一部分.我们发现:一个联锁表是由进路组成,每个进路又包含一系列的轨道,轨道边上又有信号灯和道岔,因此,可以将联锁表定义为一组进路,每个进路有自己的标识(rid)和轨道序列,每段轨道使用轨道的标识符(*tr_id*)、进入该轨道所需的信号灯的标识符(*li_id*)和道岔标识符(*p_id*)及道岔方向组成,那么每个进路可以表示为

$$rid{:}tr\_id1(li\_id1;p\_id1{:}direction)\rightarrow tr\_id2(li\_id2;p\_id2{:}direction)\rightarrow\ldots,$$

以表 1 中的 *R*1 为例,表示为:*R*1:*T*1(*S*1)→*T*2(*S*2)→*T*3(*S*3;*SW*1:*UP*)→*T*6(*S*4;*SW*2:*UP*)→*T*7(*S*9).*R*1 包括 *T*1~*T*3,*T*6,*T*7 这 5 段轨道,其中,*T*3(*S*3;*SW*1:*UP*)表示进入轨道 *T*3 需要信号灯 *S*3 为绿灯状态,道岔 *SW*1 的方向向上.

除了数量之外,我们发现还有两种类型的不同.一种是构件发生故障的概率.这个是从安全角度考虑,不同的构件由于购买时间、使用时间、放置的场景不同,其发生故障的概率是不同的.这个概率可能是一个具体的数字,比如超过 5 年之后,信号灯坏的概率为 30%,还有可能这个概率只是符合某一种分布,比如正态分布、均匀分布或者指数分布.故障发生概率这种类型跟安全关系很大,可以通过统计规律得到,属于领域专家的技术范畴,因此我们也将它设计在联锁特定语言中.另一种是构件的物理特性的不同.这些物理特性包括列车的最大允许速度、列车的最大加速度、轨道的长度、列车的发车间隔时间等.这些物理性质因车而异、因轨道而异,但是跟安全又有着莫大的关系,是领域专家熟悉的,因此也将它们设计在语言中.

总之,本文总结出 3 种类型的不同案例的联锁系统模型的不同,包括构件数量的不同、构件发生故障的概率不同以及构件物理特性的不同.分别将这 3 种不同定义到联锁特定语言中,由此得到了联锁领域特定语言 IS-DSL,如图 4 所示.

```
//构件数量及进路信息
Number of Train: Nt
Number of Light: Nl
Number of Track: Nr
Number of Point: Np
rid: tr_id1(l_id1;p_id1:direction)->tr_id2(l_id2;p_id2:direction)...
//构件发生故障的概率
Probability of Light failure: l_p
Probability of Track failure: tr_p
Probability of Point failure: p_p
//构件物理特性
Maximum Speed of Train: v_max
Maximum Acceleration of Train: a_max
Track Length: tr_len
Departure Interval Time: intv_time
```

Fig.4 IS-DSL

图 4 联锁系统领域特定语言

4 系统模型的模板

4.1 模板的基本组成

针对联锁系统的特性,结合我们之前的工作[14,15],本节设计模板的基本组成.首先,系统模板可以分为控制器模板和环境模板,其中,环境模板包含列车、轨道、道岔、信号灯和联锁表.根据联锁系统的处理流程,它们之间的交互如下:列车进入轨道前向控制器发出请求进路信号“request”,控制器收到请求后,向联锁表发送查询信号“checkTable”,联锁表将结果“result”返回给控制器.控制器根据联锁表返回信息,向相应轨道发送检测占用状态命令“checkOccupied”,轨道收到控制器命令后,向控制器返回占用状态“occupied”或未占用状态“allUnoccupied”.若轨道状态为未被占用,则控制器向道岔发出道岔锁闭信号“doLock”,收到道岔发出的锁闭状态信号“turnLock”后,向信号灯发出变绿信号“doGreen”,列车收到信号灯发出的绿灯信号“green”后,进入轨道,发出驶入信号“trainEnter”,轨道设置状态为占用状态.列车驶出轨道后,发出信号“trainLeave”,控制中心收到该信号后解锁相应的道岔、控制相应的信号灯变红,轨道收到该信号后设置状态为未占用状态.

在这样的交互过程中,由于轨道、道岔、信号灯以及列车的数量都不止一个,控制器作为处理所有的控制逻辑的单元,其控制逻辑非常复杂,不方便复用.因此,本文将控制器中将对列车的调度、轨道的检测、道岔的处理以及信号灯的处理逻辑都分别拆分出来,将控制器分为 5 部分:控制中心(center)、轨道检测子模块(CTrack)、道岔处理子模块(CPoint)、信号灯处理子模块(CSignalLight)以及列车调度器(dispatcher),即:

$$Controller=Center||CTrack||CPoint||CSignalLight||Dispatcher.$$

其中,控制中心负责控制所有的轨道、道岔、信号灯、列车、联锁表;列车调度器负责向不同的列车发送调度信号“send”,用以控制不同列车在不同的时刻进入同一轨道;轨道检测子模块负责向每一条轨道发送

“checkOccupied”消息,用以检查每条轨道的占用情况;道岔处理子模块向每个道岔发送“doLock”“doUnlock”消息,用以控制道岔状态变化;信号灯处理子模块向每个信号灯发送“doGreen”“doRed”消息,用以控制信号灯状态变化.

在上述的所有模块中,道岔处理子模块和信号灯处理子模块都涉及对多个道岔和信号灯的多种控制,包括“doLock”与“doUnlock”“doGreen”与“doRed”,为了方便,我们将其分解为对多个道岔和信号灯的单信号控制.以道岔处理子模型为例,将其分为两部分:道岔锁闭子模块(ControlPointLock)负责控制每个道岔锁闭,道岔解锁子模块(ControlPointUnlock)负责控制每个道岔解锁.类似地,可以将信号灯处理子模块(CSignalLight)分解为两部分:绿灯控制子模块(ControlLightGreen)负责控制每个信号灯变绿,红灯控制子模块(ControlLightRed)负责控制每个信号灯变红.

站在环境的立场,Track 和 SingalLight 两种构件均涉及设置状态和查询状态两种操作,为保证设置构件状态期间能正常查询构件的状态,我们将其拆分为设置子模块和查询子模块.以轨道为例,我们将轨道涉及的操作分为两部分:一部分为轨道设置(STrack),负责对轨道的状态进行设置;另一部分为轨道查询(RTrack),负责对轨道的状态进行查询,即 *Track=STrack||RTrack*.类似地,我们将信号灯涉及的操作分为两部分:一部分为信号灯设置(SSignalLight),负责对信号灯的状态进行设置;另一部分为信号灯查询(RSignalLight),负责对信号灯的状态进行查询,即 *SignalLight=SSignalLight||RSignalLight*.

综上所述,将系统模型的模板分为两大部分,控制器与环境,而控制器又可以分为 7 个子模块,环境也分为 7 个子模块,这些模块之间相互发送消息,通过协作,共同完成控制器的功能,具体的分解和协作关系如图 5 所示.

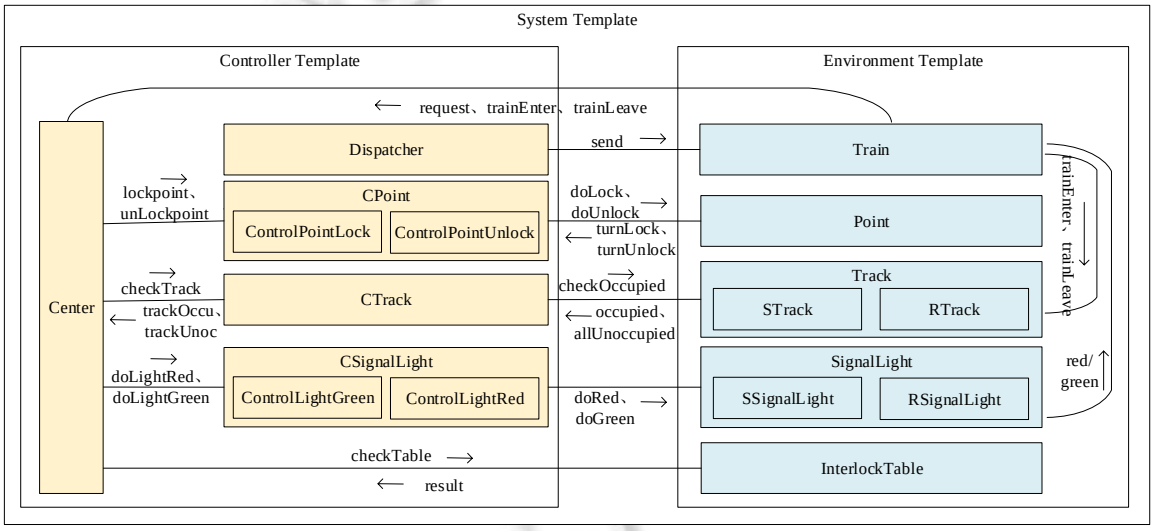


Fig.5 Interlocking system template

图 5 联锁系统模板

4.2 模板抽取举例

对于图 5 中的每个模块的模板,都需要构建其形式化模型.本节以 SHA 为例,说明如何构建其模板.在所有的模板构造中,我们发现有两种类型的模板.一种是抽取后固定,形状不再发生变化,要变化的只是参数,这类模板包括列车模板、信号灯设置模板、轨道设置模板、道岔模板和控制中心模板;另一种是需要针对一组构件进行建模,其模板分为两部分:先对一个构件进行建模,然后根据领域特定语言的参数进行动态生成,我们给出具体的生成算法.这类模板叫组合模板,图 5 中的联锁表模板、信号灯查询模板、轨道查询模板、轨道检测子模块模板、道岔处理子模块模板、信号灯处理子模块模板与调度模板都是这种类型.

由于模板很多,我们针对每种类型各举一例,说明如何建立模板的模型.其他模板的抽取过程类似,可以参

阅 https://github.com/zmy3036190149/SHA.

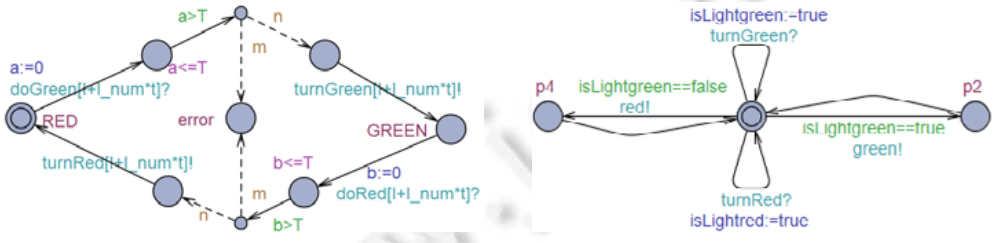
4.2.1 固定模板

这种类型的模板抽取主要是首先根据系统的处理流程找到与构件相关的过程描述,包括与构件相关的所有行为,构建其基本时间自动机.每次构件发送或接收消息(动作)时,构件的自动机从一个状态移动到另一个状态.每个构件的动作被转换为基本自动机中的状态和转换.在此基础上进行故障建模,异常事件可用概率进行表示,故在上一步生成的时间自动机上增加随机概率事件,用于表示构件发生的异常事件,用不同概率模拟构件发生不同事件的情况.最后增加时间约束,可以采用随机混成自动机的状态上的跳转延迟来表示时间约束.若状态跳转延迟服从指数分布,在该状态设置指数分布的参数;若服从均匀分布,则定义相应的时钟变量 x 和常量 T,表示一条迁移上的消息到另一条迁移上的消息之间的时间.在时间自动机中前一条迁移的“update”内赋时钟变量初值为 0,在中间状态将时钟限制为 $x\leqslant T$,在后一条迁移的“guard”内定义时钟变量的不等式 $x>T$.针对连续变量的时间约束,首先定义相应的连续变量,然后在各迁移上添加连续变量的更新表达式及判断条件,在各个状态上添加连续变量的变化函数.

下面选取信号灯设置模型 SSignalLight 的模板抽取作为例子进行说明.

在构造基本时间自动机时,SSignalLight 接收变绿信号“doGreen”后发送绿灯信号“turnGreen”;接收变红信号“doRed”后发送红灯信号“turnRed”.根据该流程,获得了 SSignalLight 的基本自动机.在此基础上进行故障建模,信号灯在由红灯状态到绿灯状态和由绿灯状态到红灯状态的变化过程中可能发生故障,即在收到“doGreen”消息到发出“turnGreen”消息之间和收到“doRed”消息到发出“turnRed”消息之间发生故障.故在时间自动机中增加一个错误状态(error).消息为“doGreen”的迁移到错误状态间有一条概率为 m%的迁移,到消息为“turnGreen”的迁移间有一条概率为 n%的迁移,其中,$m+n=100$.类似可以建模从绿灯状态到红灯状态变化过程中信号灯出现故障的情况.

增加时间约束时,从领域知识中可获知 SSignalLight 的时间约束,即信号灯状态变化有延迟.根据时间约束定义时钟变量,定义局部时钟 a 表示信号灯由红灯状态变为绿灯状态的变化延迟时间,局部时钟 b 表示信号灯由绿灯状态变为红灯状态的变化延迟时间.时钟 a 为信号灯由红灯状态变为绿灯状态的延迟时间,即从 RED 状态收到“doGreen”消息到下一状态 GREEN 的时间.故在迁移“dogreen”的“update”上将 a 赋值为 0,并在中间状态将 a 限制为“$a\leqslant T$”,在下一条迁移的“guard”上定义 a 的不等式“$a>T$”.类似在时间自动机中定义时钟变量 b,由此得到 SSignalLight 的随机混成自动机如图 6(a)所示.



(a) 信号灯设置 SSignalLight (b) 信号灯查询 RSignalLight-Single

Fig.6 SHA template of signal light

图 6 信号灯随机混成自动机模板

4.2.2 组合模板

这类模板主要是针对一组构件进行建模.首先对一个构件进行建模.具体如何建模取决于这组构件要做什么事情,这来自于系统的处理流程中与构件相关的过程描述.在这个描述中,依然可以遵循“每次构件发送或接收消息(动作)时,构件的自动机从一个状态移动到另一个状态”的规律转换.与构建固定模板步骤相同,在构建其基本自动机后,需进行故障建模和增加时间约束.接下来,在一个构件模型的基础上对一组构件进行建模,假设该组构件数量为 n.遍历一个构件模型的所有迁移,若迁移的同步信息针对单个环境构件分两种情况处理:

(1) 若迁移起始和终点是同一状态,则根据环境构件数量 n,增加 n–1 条迁移,并修改迁移上的“guard”“sync”和“update”信息;(2) 若迁移起始和终点不是同一状态,则增加 n–1 条迁移和 n–1 个状态,并修改迁移和状态相关信息.若迁移的判断条件是针对单个环境构件,则修改迁移条件为 n 个条件的组合.

下面我们以信号灯查询模型 RSignalLight 的模板抽取为例,说明这种模板抽取的过程.为了得到一组信号灯的情况,先对一个信号灯进行建模.RSignalLight 接收到“turnGreen?”消息时,对绿灯标识符(isLightgreen)赋值为真,当 isLightgreen 为 true 时,发送消息”green!”,当 isLightred 为 true 时,发送消息“red!”.根据该流程,获得了 RSignalLight-Single 的随机混成自动机模板,如图 6(b)所示.

一组信号灯建模过程在一个信号灯的基础上进行,增加相应的不同信号灯的绿灯标识符和红灯标识符.对于 n 个信号灯,则应有 n 个红灯标识符和 n 个绿灯标识符,定义数组 *isLightgreen*[n],*isLightred*[n].从初始状态到初始状态,增加 n 条消息为“*turnGreen*[i]?”的迁移,对 *isLightgreen*[i]赋值为真.消息为“green!”的迁移判断条件通过函数 *isLightgreen*(*rid*)确定,rid 为进路 ID,若进路 rid 上的信号灯均为绿灯状态,则返回 true.类似得到消息为“red!”的迁移判断条件.具体的生成算法见算法 1.

算法 1. RSignalLight 模板生成算法.

```
1.  Input: RSingleSignalLight=(L,l0, X,Σ,E,F,I), SignalLight Number Nl;
2.  Output: RsignalLight.
3.  for e in E    //traverse all transition
4.    if isrelated(e.a,singleLight)    //the action of transition e belong to a single light
5.      if e.l ==e.l'    //the start and end points of the migration are the same location
6.        for i:=1 to Nl do
7.          new ei, E.add(ei)
8.        end for
9.      else    //the start and end points of the migration are different locations
10.       for i:=1 to Nl do
11.         new ei, E.add(ei)
12.         new li, L.add(li)
13.       end for
14.     end if
15.   else if isrelate(e.guard,singleLight)    //The variables of the migration condition
                                                  are related to a single light
16.     change(e.guard)    //Modify the migration condition to a combination condition
17.   end if
18. end for
```

5 系统模型的自动生成

根据 UPPAAL-SMC 平台中模型的运行,系统模型应该包含模型声明、环境模型和控制器模型以及环境与控制器的交互.全局变量以及交互经常定义在声明中.以图 5 中的环境模板、控制器模板和领域特定语言表示的案例为输入,进行系统模型的生成,需要理清楚这些声明从哪里来,如何对环境模板进行实例化,对控制器进行实例化以及对交互进行实例化.

在环境模板进行实例化时,根据图 5 中模板的参数信息,基本上需要如下 3 方面的信息:构件的数量、构件发生故障的概率以及构件的物理特性,这些都可以从领域特定语言描述中得到.在环境模板中,对列车、信号灯、轨道、道岔的实例化个数进行赋值(句式详见表 2);对列车模板中的时钟变量 x 的边界值常量赋值;分别对信号灯模板、轨道模板、道岔模板中的随机概率的值 m 和 n 进行赋值;对联锁表模板中查询结果标号 $r1$,$r2$ 和 $r3$

进行赋值等等.对于固定的模板,这些参数的值输进去以后就直接实例化了;对于组合模板,则需要根据相应的生成算法生成相应的实例.例如,RSignalLight 的实例可以在 RSignalLight-Single 的基础上,根据输入的信号灯的数量 *Nl* 和算法 1 进行生成,在一个信号灯的 RSignalLight 基础上增加 *Nl* 个绿灯状态标识符,*Nl* 个红灯状态标识符和 *Nl* 条消息为"*turnGreen*[*i*]?"的迁移,并修改消息为"green!"的迁移判断函数 *isLightgreen*(*rid*),当进路 rid 上的信号灯均为绿灯状态时返回 true,红灯情况类似.

根据轨道的数量 *Nr* 修改 RTrack 模型,修改方法与修改 RSignalLight 的方法类似.

在进行控制器的实例化时,图 5 中的控制中心(center)、轨道检测子模块(CTrack)、道岔处理子模块(CPoint)、绿灯控制子模块(ControlLightGreen)、红灯控制子模块(ControlLightRed)、列车调度器(dispatcher)模板、道岔锁闭子模型(ControlPointLock)、道岔解锁子模型(ControlPointUnlock)和列车、信号灯、轨道、道岔的实例化个数都是有关系的,可以根据领域语言给出的列车、信号灯、轨道、道岔的实例化个数修改控制器子模块的模板,并进行物理特性参数的赋值.

例如:在轨道控制子模块 CTrack 中增加 *Nr* 个状态和 *Nr* 条消息为"*checkoccupied*[*i*]!"的迁移,各迁移上的消息按顺序排列.道岔处理子模块、信号灯处理子模块和列车调度器的修改方法类似.

接下来需要定义系统中的交互.系统交互由系统上下文图中各构件之间的通信实现,因此定义系统交互即是定义构件间通信涉及的所有消息.通信在 UPPAAL-SMC 中使用广播信道,一般在全局声明中定义所有消息.例如 broadcast chan *green*[*Nt***Nl*],假设根据联锁表可知,共有 *Nl* 个信号灯.每辆列车都需要记录其请求的进路的信号灯状态,因此对于 *Nt* 辆列车需定义 *Nt***Nl* 个绿灯信号信道.类似可定义其他信道及广播信道,消息具体声明见表 2.

最后,定义系统模型的声明及全局变量.在模型声明中声明系统中的所有模型.由图 5 可以确定系统是由 14 个模型组成,其声明详见表 2.系统所需的全局变量为模型之间的共享信息,应该包括轨道占用标识符和在全局声明中每个构件的实例化个数.此外,控制中心所需函数的变量也需在全局声明中声明.

Table 2 Partial results generated by the system

表 2 系统生成部分结果

模型声明	System Train,Point,STrack,RTrack,SSignalLight,RSignalLight,InterlockTable,Center, Dispatcher,ControlPointLock,ControlPointUnlock,CTrack,ControlLightGreen,ControlLightRed
模型实例化	const int *TRAINS*=*Nt*; typedef *int*[0,*TRAINS*-1] *train_t*; const int *LIGHTS*=*Nl*; typedef *int*[0,*LIGHTS*-1] *light_l*; const int *TRACKS*=*Nr*; typedef *int*[0,*TRACKS*-1] *track_t*; const int *POINTS*=*Np*; typedef *int*[0,*POINTS*-1] *point_p*

交互信道	broadcast chan *request*[*Nt*], *send*[*train_t*], *trainEnter*[*train_t*], *trainLeave*[*train_t*]; //列车相关信道
	broadcast chan *checkTable*[*Nt*], *result*[*Nt*]; //联锁表相关信道
	broadcast chan *lockpoint*[*Nt*], *doLock*[*Nt***Np*], *turnLock*[*Nt***Np*], *unlockpoint*[*Nt*], *doUnlock*[Nt**Np*],
	turnUnlock[*Nt***Np*], *up*[*Nt***Np*], *front*[*Nt***Np*], *down*[*Nt***Np*]; //道岔相关信道
	broadcast chan *checkTrack*[*Nt*], *trackOccu*[*Nt*], *trackUnoc*[*Nt*], *checkOccupied*[*Nt***Nr*], *occupied*[*Nt*],
	allUnoccupied[*Nt*], *uncu* [*Nt***Nr*], *uncu*[*Nt***Nr*]; //轨道相关信道
	broadcast chan *green*[*Nt***Nl*], *doGreen*[*Nt***Nl*], *turnGreen*[*Nt***Nl*], *doLightRed*[*train_t*], *red*[*Nt***Nl*], *doRed*[*Nt***Nl*],
	turnRed[*Nt***Nl*] ,*doLightGreen*[*train_t*], *allGreen*[*train_t*]; //信号灯相关信道

全局变量	*int y*[*Nt***Nr*]={0,0,...,0} // 轨道占用标识符
	const *int t_num=Nt, l_num=Nl, p_num=Np, tr_num=Nr*; //每个构件的实例化个数
	int route_id, trackID[*Nr*], *PointInfo*[*Np*][*Np*], *lightID*[*Nl*]; //控制中心所需函数的变量

6 案例研究

本文以某站的联锁系统为案例,阐述如何按照本文方法根据具体的情况生成联锁系统的模型,并在此模型基础上给出了事故的预测分析,以此证明生成模型的有效性.图 2 为其站场图,包含 7 条轨道段、9 个信号灯、2 个道岔和 3 条进路,每条进路包含 5 条轨道段、5 个信号灯以及 2 个道岔,具体进路信息见表 1.

6.1 某站的联锁系统描述及其系统模型的生成

根据站场图及联锁表分析,请领域专家填写,可以得到领域特定语言的系统描述如下.

Number of Train: 2 Number of Light: 9 Number of Track: 7 Number of Point: 2 *R*1: *T*1(*S*1)→*T*2(*S*2)→*T*3(*S*3;*SW*1:*Up*)→*T*6(*S*4;*SW*2:*Up*)→*T*7(*S*9) *R*2: *T*1(*S*1)→*T*2(*S*2)→*T*4(*S*5;*SW*1:*Front*)→*T*6(*S*6;*SW*2:*Front*)→*T*7(*S*9) *R*3: *T*1(*S*1)→*T*2(*S*2)→*T*5(*S*7;*SW*1:*Down*)→*T*6(*S*8;*SW*2:*Down*)→*T*7(*S*9)	Probability of Light Failure: 1% Probability of Track Failure: 1% Probability of Point Failure: 1% Initial Speed of Train: 80m/s Maximum Speed of Train:97m/s Maximum Acceleration of Train: 1m/s^2 Track Length: 1000m Departure Interval Time: 180s

将这样的描述导入,做系统模型生成.系统模型中的系统声明、全局变量声明以及交互信道直接带入表 2 就可以直接生成,最重要的是环境模板的实例化和控制器模板的实例化.在环境模板进行实例化时,对于列车模板、轨道设置模板、信号灯设置模板、道岔模板这些固定的模板,将参数的值输进去以后就是直接实例化了.例如列车模板中(图 7),要求初始速度 *V*0、列车允许最大速度 *V*max、列车允许最大加速度 *A*max,轨道长度 *S*max 的参数值,从领域描述中获取这些值以后,列车模型就直接实例化了.

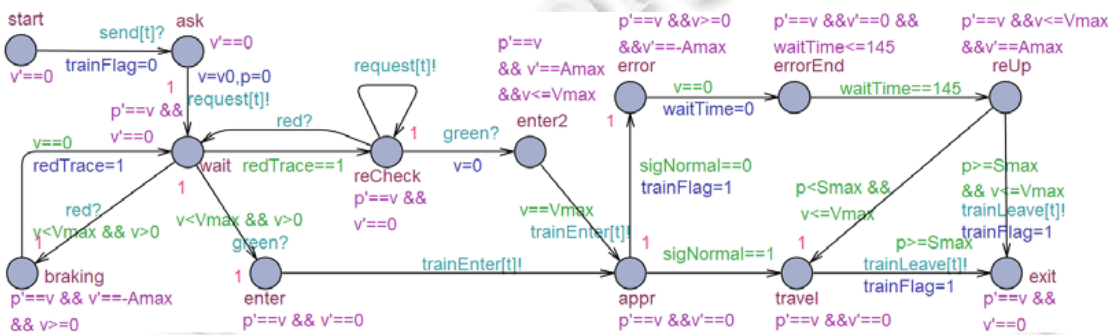


Fig.7 SHA template of train in a station

图 7 某站联锁系统的列车随机混成自动机模型

对于信号灯查询模板、轨道查询模板、联锁表模板这些组合模板,则需要根据相应的生成算法生成相应的实例.例如,根据信号灯个数和轨道个数修改 RSignalLight 模板和 RTrack 模板的迁移数量和判断条件.由于有 7 条轨道段,故有 7 个轨道占用标识符 isTrackoc;有 7 条消息为“occu?”和“unoc?”的迁移,在迁移上对相应的标识符赋值为真.消息“allunoccupied!”的迁移判断条件为函数 *isTrackun*(*rid*).该函数判断进路 rid 上的轨道是否均未被占用,轨道占用迁移情况类似,故 RTrack 模型如图 8 所示.其他控制器子模块的构建方法与 RTrack 模型的构建方法类似,限于篇幅,具体模型见 https://github.com/zmy3036190149/SHA.

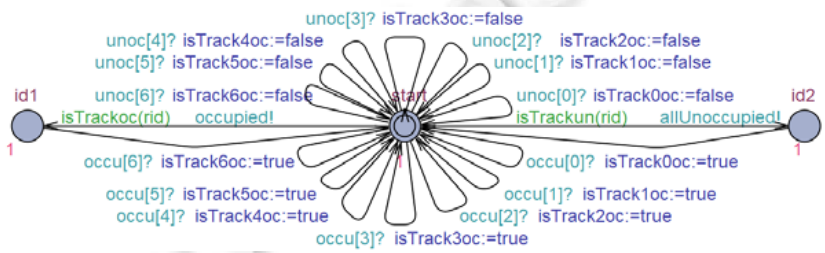


Fig.8 SHA template of RTrack in a station

图 8 某站联锁系统的轨道设置随机混成自动机模型

得到环境构件模型后,对控制器模板进行实例化.根据构件个数修改子模板,构建轨道检测子模型、绿灯控制子模型、红灯控制子模型、道岔处理子模型及列车调度器模型.以轨道检测子模型为例,由于每个进路有 5 条轨道,故有一条以初始状态为始端的迁移,迁移消息为“checkTrack?”,有 5 条消息为“checkOccupied!”的迁移,依次相接.最后一条带“checkOccupied!”消息的迁移发出后,若收到轨道查询子模块发送的未占用信号

“allUnoccupied?”,则向控制中心发送轨道未占用信号 “trackUnoc!”; 若收到轨道占用信号“occupied?”,则向控制中心发送轨道占用信号“trackOccu!”.由此得到轨道检测子模型如图 9 所示.其他子模型的构建方法与轨道检测子模型的构建方法一致,最后重用控制中心模板,控制中心模板为固定模板,输入参数后即可实例化,具体模型见 https://github.com/zmy3036190149/SHA.

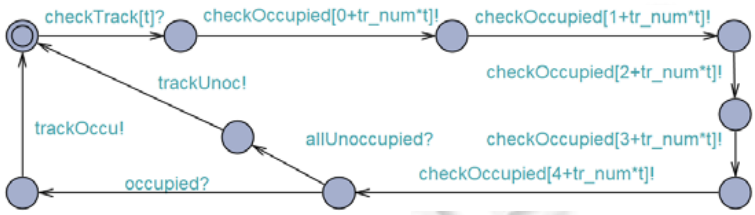


Fig.9 SHA template of CTrack in a station

图 9 某站联锁系统的轨道检测子模块的随机混成自动机模型

将这些所有的实例化了之后,就可以得到可以在 UPPAAL-SMC 上运行的系统模型.由于模型为连续模型,可以对列车的位置、速度等进行仿真,获得仿真数据.仿真环境如下:处理器:Intel(R)Core(TM) i5-4460 CPU@3.20GHz;内存:8G;操作系统:64 位 Windows 10 家庭中文版;UPPAAL 版本:4.1.19.

输入性质验证表达式 simulate 1[≤500]{*Train*(0).*p*,*Train*(1).*p*},表示对两列车的位置变量在 500 个时间单位内进行 1 次仿真.经过仿真,得到两列车的位置数据结果.

6.2 基于联锁系统模型的分析:事故预测

在联锁系统基础上可以进行各种安全分析,本节研究如何进行事故的预测.由于联锁系统涉及大量变量,对该系统的验证通常会引发状态空间爆炸问题[28],本文采用基于仿真的方式分析.在分析时,首先要定义事故模型.本文仅考虑轨道交通系统风险中的碰撞情况,基本想法是:当列车进入同一路线后,若列车之间的距离为 0m,则两车相撞.具体描述为:第 1 列车驶入某一进路,行驶一段时间后,第 2 列车进入该进路.在该过程中,第 1 列车和第 2 列车按照各自的速度及加速度行驶.若在某一时刻两列列车的相对距离为零,则两列列车发生碰撞.根据第 3 节中 Train 的定义,每个 Train 的路程为 *p*(这个 *p* 是离出发站的距离,不是位移),则两列列车 $train_1$ 和 $train_2$ 之间的距离为:$Distance=train_1.p-train_2.p$.由此,我们定义一个事故模型:

定义 1(基于距离的事故模型). 假设有两列列车 $train_1$ 和 $train_2$,两列车进入同一进路后列车之间的距离为 *Distance*,其中 $Distance=train_1.p-train_2.p$.若 *Distance*=0,则两列车相撞.

基于第 6.1 节中仿真得到的两列车的位置数据,可以定义当其位置曲线相交时,两辆列车即存在相撞的情况.图 10 中,*train*[0]和 *train*[1]的位置曲线相交了,就可预测出两列车存在相撞.

为了验证本文方法生成模型的有效性,本文与文献[14]中的工作进行了对比实验.文献[14]中使用的随机混成自动机离散模型,没有速度、位置等连续概念.在同上的参数设置下,仅修改两列车进入轨道的时间,得到多条预测结果,见表 3.

由表中可以看出:针对同一案例,尽管本文方法更耗时,在本文的预测模型预测出事故的情况下,文献[14]的预测模型不一定能预测出事故;在文献[14]预测出事故的情况下,本文的预测模型均能预测出事故.相较于文献[14]中的离散模型,本文生成的混成模型更加精确.其实,第 2 节中提出的框架也适用于文献[14].

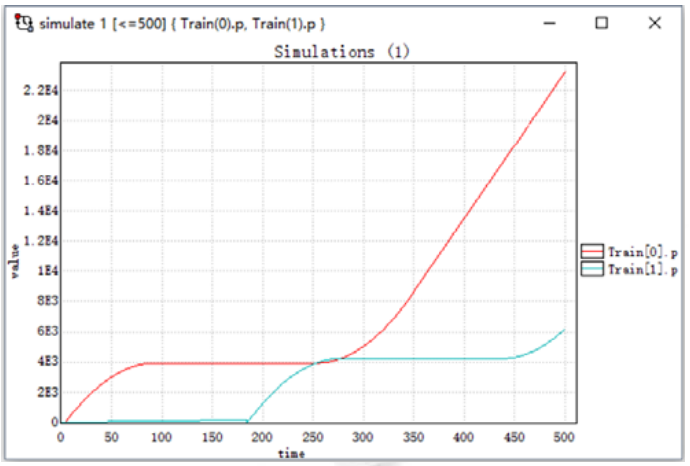


Fig.10 Verification results of expression simulate 1[≤500]{*Train*(0).*p*,*Train*(1).*p*}

图 10 表达式 simulate 1[≤500]{*Train*(0).*p*,*Train*(1).*p*}的验证结果

Table 3 Comparation between Ref.[14] and this paper

表 3 本文方法与文献[14]中预测结果对比

ID	第 1 辆列车进入时间(s)	第 2 辆列车进入时间(s)	文献[14]预测		本文预测		正确结果
			结果	耗时(s)	结果	耗时(s)	
1	137.485	293.536	不碰撞	0.125	碰撞	0.563	碰撞
2	287.845	443.896	不碰撞	0.11	不碰撞	0.609	不碰撞
3	394.716	597.481	不碰撞	0.125	碰撞	0.61	碰撞
4	481.955	676.260	不碰撞	0.14	碰撞	0. 623	碰撞
5	579.237	936.344	碰撞	0.109	碰撞	0.645	碰撞
6	742.268	1036.728	碰撞	0.141	碰撞	0.656	碰撞
7	1377.25	1829.354	碰撞	0.156	碰撞	0.665	碰撞

7 相关工作

本文研究轨道交通联锁系统的形式化模型的自动生成,相关的工作主要包括轨道交通系统的形式化建模与验证.关于这方面的研究有很多.根据初步建模使用的语言,我们将这方面的工作分为 3 类:基于自然语言的形式化方法、基于半形式化的形式化方法以及基于领域特定语言(DSL)的形式化方法.

在基于自然语言的形式化建模与验证中,形式化专家根据领域专家的自然语言描述,直接建立形式化模型,并采用相应的形式化验证工具进行性质验证.例如,Hartig 等人[4]在对用自然语言描述的铁道车辆的安全需求分析之后,使用 ACSL 来将其形式化,然后通过 FRAMA-C[4]对形式化模型进行演绎验证.Zafar 等人[5]对自然语言描述的移动块联锁的安全性进行了形式化分析,提出了一个使用图论和 Z 符号分析降低系统复杂性的安全性过程.Russo 等人[6]提出了基于形式化方法的轨道拓扑和列车运行条件验证工具.并使用 Event-B[29]对生成的属性进行形式化、证明和验证.上述建模与验证方法对于不具有形式化知识的领域专家来讲很难使用,而本文提出的方法是使用随机混成自动机模板进行联锁系统的建模,建立模板后,领域专家只要输入参数即可重用模板,相比来说会方便使用很多.

基于半形式化的形式化建模与验证中,首先由领域专家使用半形式化语言(例如 UML)描述联锁系统,然后,形式化专家把半形式化模型转换为形式化模型并进行形式化分析.例如,Xu 等人[7]将使用 UML 等建模语言指定的铁路联锁模型转换为形式化模型,并进行数学分析,弥补半形式化语言和形式化语言之间的差距,从而促进模型驱动工程(MDE)方面的安全关键系统的开发.Hansen 等人[8]描述了 xUML 子集到流程代数规范语言 mCRL2 的转换.黄友能等人[9]采用扩展后的 UML 对 ZC 子系统的系统功能进行半形式化建模,然后根据转换规

则将 UML 模型转换为线性混成自动机的形式化模型,使用 BACH 软件进行验证.Fotso 等人[10]使用 SysML/KAOS[30]对系统需求、领域特性以及与混合 ERTMS/ETCS 3 级标准[31]相关的安全不变量进行半形式化建模,然后自动转换为 B 系统规范,以获得形式化规范的体系结构,最后用 Rodin 工具对其进行验证.杨璐等人[11]采用消息顺序图(message sequence chart,简称 MSC)[32]对 ZC 切换场景功能和受限活性,然后将 MSC 模型转换为形式化的时间自动机,通过 UPPAA 对其进行建模和验证.这些半形式化的方法,无论是 UML 还是 SysML,都其实还是软件专业建模语言,领域专家应用起来还是比较吃力.而我们的方法是专门为领域专家设计.

基于 DSL 的形式化建模与验证中,首先由领域专家使用领域特定语言描述联锁系统,然后,形式化专家再把 DSL 模型转换为形式化模型并进行形式化分析.这类做的比较好的工作是 Idani 等人[12]在 2019 年发表的工作,他们将模型驱动工程(MDE)范式与形式化方法相结合,首先使用 MDE 工具定义图形 DSL 的一个示例,然后使用形式化 B 方法定义其底层操作语义,并确保模型行为相对于其安全属性的正确性.他们利用 Meedus 工具对域模型的执行场景进行动画和可视化.从 DSL 工具中设计的给定模型开始,Meedus 要求 prob 对 B 操作进行动画处理,并通过 B 变量估值获得达到的状态.然后,它将这些估值转换回初始 DSL,从而自动修改域模型.这种方法允许比当前的视觉动画技术更务实的、以领域为中心的动画,因为最终的 DSL 工具允许领域专家自行设计和验证各种领域模型,而这些专家不必接受形式化方法的培训.Kaymakçı 等人[13]的工作其实也跟这个类似,他们采用时间弧 Petri 网的构建联锁系统形式化模型的模板,开发了一个可视化的工具来自动生成联锁系统的形式化 TAPN 模型,然后将安全需求描述为 CTL 公式,在 TAPAAL 上进行验证.这个可视化的工具实际上就是用来给出站场图,这跟本文的思路其实很相似,只要领域专家给出相应的参数,形式化模型就构建好了.从另一个方面讲,也减少了人工建模出错误的可能性.我们的方法其实也属于这一类,与它们相比,我们的方法以随机混成自动机作为形式化手段,既考虑了联锁系统的实时性,又能兼顾故障随机性,更加适合作为联锁系统的建模语言.不仅如此,本文还充分利用了联锁系统中构件可重用的特点,利用模板进行形式化方法的重用.另外,我们之前的工作[14]使用的是随机混成自动机模型,但是文献[14]中的模型是离散模型,没有速度、位置等连续概念.从这个角度来说,本文的模型比文献[14]中的模型更适合联锁系统.

8 总结与展望

联锁系统的安全使得形式化的建模非常必要.本文提出了基于模板的联锁系统自动生成方法,在分析联锁系统的前提下,定义了领域特定语言,确定了系统的模板以便重用,使得领域专家可以输入参数就可以生成基于随机混成自动机等的系统模型,并在此基础上进行分析.本文的主要工作包括:

(1) 定义了联锁领域特定语言,针对构件数量、构件发生故障的概率及构件的物理特性分别进行了设计;
(2) 确定了联锁领域的模板,包括环境构件模板和控制器模板,对模板进行了固定模板和组合模板的分类,根据分类,制定了模板的随机混成自动机模型;
(3) 定义了基于系统模板的自动生成方法,在环境构件模板和控制器模板的基础上,通过让领域专家输入领域特定语言确定参数,生成具体的系统随机混成自动机模型,能建模故障随机性、行为实时性,其仿真验证数据能用于后续的安全分析.

在案例研究中,以一个具体车站的联锁系统为例,根据系统模型自动生成方法生成了具体模型,还基于此模型进行了事故预测分析,与现有方法的事故预测相比具有好的效果,也验证了本文方法模型生成的有效性.

本文工作尚有不足之处:文中的模板只考虑了列车行驶故障、信号灯故障、道岔故障、轨道故障等硬件故障,还未考虑联锁系统中软件故障以及更多的复杂的现实因素.未来工作中,将在模板中注入这些故障,让故障出现的时机等更接近现实,使得生成的形式化模型更真实,能验证出更多的现实错误.另外,还需要设计算法能快速地在仿真验证中发现错误.

References:

[1] Zhao ZX. Computer Interlocking System Technology. Beijing: China Railway Publishing House Co., Ltd., 1999 (in Chinese).

[2] Railway applications—Communication, signaling processing systems—Software for railway control protection systems. Document EN50128. CENELEC, 2011.

[3] Railway application—Communications, signaling and processing systems—Safety related electronic systems for signaling. Document BS EN 50129. 2018.

[4] Hartig K, Gerlach J, Soto J, *et al*. Formal specification and automated verification of safety-critical requirements of a railway vehicle with frama-c/jessie. In: Proc. of the FORMS/FORMAT 2010. Springer-Verlag, 2011. 145−153.

[5] Zafar NA, Khan SA, Araki K. Towards the safety properties of moving block railway interlocking system. Int'l Journal of Innovative Computing Information & Control, 2012,8(7):5677−5690.

[6] Russo AG, Ladenberger L. A formal approach to safety verification of railway signaling systems. In: Proc. of the 2012 Annual Reliability and Maintainability Symp. IEEE, 2012. 1−4.

[7] Xu T, Santos OM, Ge X, *et al*. Use of model transformation for the formal analysis of railway interlocking models. WIT Trans. on the Built Environment, 2010,114:815−826.

[8] Hansen HH, Ketema J, Luttik B, *et al*. Towards model checking executable UML specifications in mCRL2. Innovations in Systems Software Engineering, 2010,6(1-2):83−90.

[9] Huang YN, Zhang PJ, Hou XP, *et al*. Modeling and Verification Method of ZC Subsystem in Urban Rail Transit Based on Hybrid Automata. China Railway Science, 2016,37(2):114−121 (in Chinese with English abstract).

[10] Fotso SJT, Frappier M, Laleau R, *et al*. Modeling the hybrid ERTMS/ETCS level 3 standard using a formal requirements engineering approach. In: Proc. of the Int'l Conf. on Abstract State Machines, Alloy, B, TLA, VDM, and Z. Southampton: Springer- Verlag, 2018. 262−276.

[11] Yang L, Chen YG. Modeling and Verification of switch scene of zone controller based on MSC and UPPAAL. Railway Standard Design, 2018,62(5):171−174+179 (in Chinese with English abstract).

[12] Idani A, Ledru Y, Wakrime AA, *et al*. Towards a tool-based domain specific approach for railway systems modeling and validation. In: Proc. of the Int'l Conf. on Reliability, Safety, and Security of Railway Systems. Cham: Springer-Verlag, 2019. 23−40.

[13] Kaymakçı ÖT, Oz MAJGUJoS. An automatic formal model generation and verification method for railway interlocking systems. 2017,30(2):133−147.

[14] Wang Y, Zhong W, Chen XH, *et al*. Predicting accidents in interlocking systems: An SHA model-based approach. Int'l Journal of Performability Engineering, 2017,13(6):897−912.

[15] Wang Y. Accident prediction of interlocking system based on stochastic hybrid automata [MS Thesis]. Shanghai: East China Normal University, 2018 (in Chinese with English abstract).

[16] Svendsen A, Olsen GK, Endresen J, *et al*. The Future of Train Signaling. Springer, Berlin, Heidelberg, 2008. 128−142.

[17] Bortolussi L, Policriti A. Stochastic programs and hybrid automata for (Biological) modeling. In: Proc. of the Conf. on Computability in Europe: Mathematical Theory and Computational Practice. Springer-Verlag, 2009. 37−48.

[18] Chen MS, Gu F, Xu SY, *et al*. Formal Evaluation of Scheduling Strategies for Smart Building Air-Conditioning Systems under Uncertain Environment. Ruan Jian Xue Bao/Journal of Software, 2016,27(3):655−669 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4987.htm [doi: 10.13328/j.cnki.jos.004987]

[19] Bulychev P, David A, Larsen KG, *et al*. UPPAAL-SMC: Statistical model checking for priced timed automata. In: Massink M, Wiklicky H, eds. Proc. of the Workshop on Quantitative Aspects of Programming Languages and Systems (QAPL 2012). 2012. 1−16.

[20] Bemporad A, Di Cairano S. Optimal control of discrete hybrid stochastic automata. In: Proc. of the Int'l Workshop on Hybrid Systems: Computation and Control. Springer-Verlag, 2005. 151−167.

[21] Alur R, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183−235.

[22] Bengtsson J, Yi W. Timed automata: Semantics, algorithms and tools. In: Proc. of the Advanced Course on Petri Nets. Springer-Verlag, 2003. 87−124.

[23] Bengtsson J, Larsen K, Larsson F, *et al*. UPPAAL—A tool suite for automatic verification of real-time systems. In: Proc. of the Int'l Hybrid Systems Workshop. Berlin, Heidelberg: Springer-Verlag, 1995. 232−243.

[24] Legay A, Delahaye B, Bensalem S. Statistical model checking: An overview. In: Proc. of the Int'l Conf. on Runtime Verification. Springer-Verlag, 2010. 122−135.

[25] Fang H, Shi J, Zhu H, *et al*. Formal verification and simulation for platform screen doors and collision avoidance in subway control systems. STTT, 2014,16(4):339−361.

[26] Wang YY, Chen XH, Chen MS, *et al*. Choosing the best strategy for energy aware building system: An SVM-based approach. In: Proc. of the 28th Int'l Conf. on Software Engineering and Knowledge Engineering (SEKE). 2016. 547−550.

[27] Hartonas-Garmhausen V, Campos S, Cimatti A, *et al*. Verification of a safety-critical railway interlocking system with real-time constraints. Science of Computer Programming, 2000,36(1):53−64.

[28] Fantechi A. Distributing the challenge of model checking interlocking control tables. In: Proc. of the Int'l Symp. on Leveraging Applications of Formal Methods, Verification and Validation. Springer-Verlag, 2012. 276−289.

[29] Abrial JR. Modeling in Event-B: System and Software Engineering. Cambridge University Press, 2010.

[30] Gnaho C, Semmak F. Une extension SysML pour l'ingénierie des exigences dirigée par les buts. In: Proc. of the INFORSID. 2010. 277−292.

[31] EEIG ERTMS Users Group. Hybrid ERTMS/ETCS Level 3. Principles Ref: 16E042, Version 1A, 2017.

[32] Rudolph E, Graubmann P, Grabowski JJCN, *et al*. Tutorial on message sequence charts. Computer Networks and ISDN Systems, 1996,28(12):1629−1641.

附中文参考文献:

[1] 赵志熙.计算机联锁系统技术.北京:中国铁道出版社,1999.

[9] 黄友能,张鹏基,侯晓鹏,等.基于混成自动机的城市轨道交通 ZC 子系统建模与验证方法.中国铁道科学,2016,37(2):114−121.

[11] 杨璐,陈永刚.基于 MSC 与 UPPAAL 的区域控制器切换场景建模与验证.铁道标准设计,2018,62(5):171−174+179.

[15] 王焱,基于随机混成自动机的联锁系统事故预测[硕士学位论文].上海:华东师范大学,2018.

[18] 陈铭松,顾璠,徐思远,陈小红.不确定环境下智能大厦空调系统调度策略评估.软件学报,2016,27(3):655−669. http://www.jos.org.cn/1000-9825/4987.htm [doi: 10.13328/j.cnki.jos.004987]

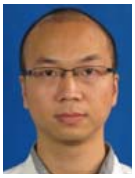
赵梦瑶(1995－),女,河北邢台人,硕士生,CCF 学生会员,主要研究领域为需求工程,形式化方法.

刘静(1964－),女,博士,教授,博士生导师,CCF 专业会员,主要研究领域为可信软件,模型驱动式软件开发方法,面向服务的软件架构.

陈小红(1982－),女,博士,副教授,CCF 专业会员,主要研究领域为需求工程,形式化方法,安全攸关系统.

陈良育(1980－),男,博士,副教授,主要研究领域为程序分析和验证.

孙海英(1976－),女,讲师,CCF 专业会员,主要研究领域为形式化建模,形式化测试,时空逻辑.



周庭梁(1980－),男,博士,高工,主要研究领域为安全苛求系统,可信测评,形式化方法.