基于场景的联锁软件形式化模型生成方法

董 昱 高雪娟
(兰州交通大学自动化与电气工程学院 兰州 730070)

摘 要 为保证列车运行安全和旅客生命财产安全,对车站联锁控制系统进行有效的分析、验证和测试是必不可少的,而形式化模型是联锁系统分析、验证和测试的基础。以计算机联锁软件的UML半形式化模型为基础,以事件确定有限自动机模型作为描述系统的形式化模型,研究UML2.0顺序图转换为事件确定有限自动机模型的方法。首先选取一组与交互行为相关的全局变量作为状态向量来分析和消解顺序图各个场景的消息以及不同场景间的同一消息的前后置状态向量值是否存在矛盾,从而得到一致性的需求场景;然后提取各对象的事件序列生成对应的事件确定有限自动机;最后通过组合系统中对象的自动机模型得到系统的事件确定有限自动机模型。该方法改善了安全苛求软件的设计与开发,为软件质量评估提供了技术支撑。

关键词 计算机联锁软件,事件确定有限自动机,顺序图,场景分析
中图法分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.043

Method for Generating Formal Interlocking Software Model Based on Scenario

DONG Yu GAO Xue-juan
(School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract It is necessary to do analysis, verification and test of station interlocking system to ensure the safety of train running and people lives. Especially, the formalized model is the foundation of these works. This paper studied the method transforming the sequence diagram(SD) into the event deterministic finite automata(ETDFA) model which is based on UML semi-formal model of computer interlocking software, and used ETDFA as the formal model to describe the system. Firstly, a set of global variables associated with sequence diagram's interaction were selected as the sate vector to analyze and eliminate the contradiction in pre-value and post-value of message in every scenario and the same message among multiple scenarios, and then based on the sequence event of each object, corresponding ETDFA model was generated. At last, by composing all objects' ETDFA model, system's ETDFA formal model was got. The method improves safety software's designing and development and provides technical support of software quality.

Keywords Computer interlocking software, ETDFA, Sequence diagram, Scenario analysis

计算机联锁系统是保证列车在站内运行安全的重要技术,通过进路控制功能实现旅客安全上下车及货物的调车作业,是在既有继电联锁系统的基础上,用联锁软件取代传统控制台实现联锁逻辑运算。相比继电联锁系统,计算机联锁系统为铁路信号向智能化和网络化方向发展提供了条件,可减少继电器的维修工作量,便于软件的升级及站场的改造,方便增加新功能。

作为安全相关的计算机联锁系统[1,2],其错误的输出可能会造成生命财产的重大损失,而作为实现联锁逻辑运算功能的计算机联锁软件(SIL4)的质量直接决定计算机联锁系统的安全性和可靠性,因此,对计算机联锁软件在设计、实现、运营和维护的各阶段进行分析、验证和测试是保证软件高安全和高可靠必不可少的技术方法。

根据EN40128[3],安全完善度等级[4]为SIL4的安全系统的软件需求规格说明书和软件设计推荐使用形式化方法;IEC61508[5,6]提出为保证安全相关系统的安全充分性,系统的需求说明和设计必须使用形式化方法。建立有效的计算机联锁软件的形式化模型是实现软件分析、验证和测试的基础,可以在设计阶段验证方案,实现阶段生成有效的测试用例,运营维护阶段分析和诊断系统。文献[7]对安全苛求系统采用故障树分析结合标签转移系统模型检测的方法分析和验证其安全性;文献[8]运用RAISE验证分布式铁路控制系统的功能及安全需求;文献[9]利用Verus验证工具,采用结合CTL时序逻辑的符号模型检测技术来验证Ansaldo联锁系统的安全逻辑。上述形式化方法及基于模型的分析方法[10-12]对安全分析人员掌握形式化方法、理论水平要求较高,不能得到广泛应用。UML模型以定义良好、功能强大、普遍适用的优点,为联锁软件生成形式化模型提供了非常好的条件,但是,UML是半形式化的,在描述系统行为时缺乏严格的语义规范和数学基础,因此,本文以计算机联锁软件中基本进路建立为例,提出基于场景的联锁软件形式化模型生成方法。该方法基于UML顺序图描述系统需求场景,通过分析顺序图中

到稿日期:2014-02-19 返修日期:2014-04-19 本文受基于受控拉格朗日函数的多欠驱动度力学系统控制器设计(61164010)资助。
董 昱(1962-),男,教授,主要研究方向为交通信息工程及控制,E-mail:dongyu@mail.lzjtu.cn;高雪娟(1990-),女,硕士生,主要研究方向为交通信息工程及控制、软件测试。

消息的前后置状态向量值得到一致的需求场景；构造单个对象的确定有限自动机，组合各对象自动机模型得到系统的确定有限自动机形式化模型。该模型可作为系统分析、验证和测试的基础。

1 计算机联锁软件

计算机联锁系统是以联锁软件完成联锁逻辑运算驱动室外设备来实现以进路控制为主要联锁功能的系统[13]。联锁软件完成联锁系统的核心功能，即联锁逻辑运算。计算机联锁软件需包含以下 3 部分信息：

(1)车站站场拓扑图。描述车站中信号机、道岔和轨道电路等室外设备之间的物理连接方式，用来确定唯一的站场形状。

(2)联锁软件中的静态模型。定义联锁软件中主要对象的类型和各对象之间的静态关系。包括轨道区段、信号机、道岔、按钮和进路等各对象的属性和操作，如区段的空闲/占用、区段类型(股道、无岔区段、道岔区段)、道岔区段所包含的道岔、区段锁闭状态等，进路数据以及各对象之间的关系，如一条基本进路由一架信号机防护，包含 1 个或多个道岔与轨道区段。

(3)联锁软件的动态行为。为保证安全，信号机、进路和道岔三者之间彼此存在复杂的制约关系，一个实体的简单操作会影响其他实体的状态属性。描述联锁软件是如何实现道岔转换、进路办理、进路锁闭、信号开放、进路解锁、调车中途折返解锁等联锁功能的。

2 UML2.0 顺序图

顺序图[14]是交互图的一种，强调交互对象之间的信息的时间顺序。UML2.0 顺序图添加了 alt、opt、loop、break、par、neg、ref 等多种组合片段[15]，增强了面向对象系统交互行为分析与设计中的建模能力。办理进路用例所对应的顺序图如图 1 所示。

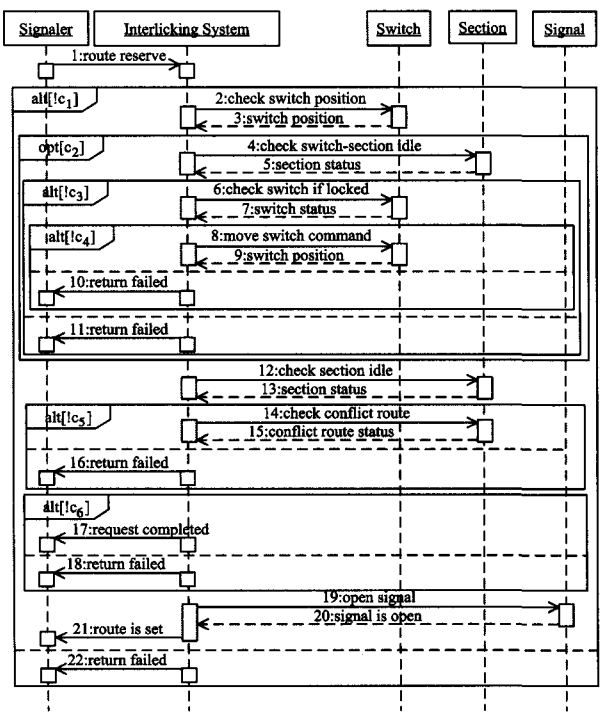

图 1 办理进路顺序图

各监护条件 $c_i(i=1,\cdots,5)$表示的含义如表 1 所列。

表 1 各监护条件表示的含义

监护条件	表示意义
c_1	操作不规范无对应进路
c_2	进路中道岔位置不正确
c_3	道岔所在区段有车占用
c_4	道岔已锁闭
c_5	进路中区段不空闲
c_6	敌对进路已建立

2.1 UML2.0 顺序图的语法

为准确描述顺序图中对象之间的动态交互过程，提取出顺序图中的有效事件序列，对 UML2.0 顺序图进行形式化定义，包括事件之间的偏序关系及新增加的组合片段特征。

定义 1 顺序图 SD 用一个十二元组表示，$SD=(O,M,S,R,C,C_c,C_r,f_s,f_r,f_{mc},\rightarrow,<)$。

其中，O 是对象集合；M 是顺序图中的消息集合；S 是消息对应的发送事件集合；R 是消息对应的接受事件集合，$S(m)\cap R(m)=\emptyset$；C 是组合片段集合；C_c 是组合片段执行条件的集合；C_r 是组合片段操作范围的集合；f_s 是从发送事件到对象的一个函数关系；f_r 是从接收事件到对象的一个函数关系；f_{mc} 表示消息到组合片段的函数关系；$\rightarrow$表示消息 M 的全序关系；$<$表示顺序图中的事件之间的先后关系，分为可视顺序和强制顺序。

可视顺序是从顺序图中可观察到的事件之间的先后发生关系，即任何消息的发送事件先于接收事件发生以及顺序图生命线的时间轴上方的事件先于下方的事件发生，用$<$表示。由于顺序图表示的局限性，一个对象只能控制自身的发送时间的消息先后，对于接收事件的时间是不确定的，因此，为了使系统设计实现过程不造成歧义，一些学者定义强制顺序关系，用$\ll$表示。对于事件 e_1、e_2，如果 $e_1=s(m)$，$e_2=r(m)$，则 $e_1\ll e_2$；如果$(O(e_1)==O(e_2))\wedge(e_1<e_2)\wedge(e_2\in S)$，则 $e_1\ll e_2$。

2.2 顺序图中单个对象的形式化定义

顺序图描述了多个对象之间的信息交互，对于单个对象的形式化定义如下。

定义 2 顺序图中对象 O_i 由一个六元组表示，$O_i=(M,C,C_c,C_r,n,f_{mc})$。

其中，n 是事件发生的顺序标记。

3 基于场景的系统形式化模型生成方法

3.1 联锁软件顺序图场景一致性分析子算法

由于 UML2.0 新增组合片段，一个顺序图可以表示多个场景。通过对顺序图中消息的前后置条件进行矛盾检测，来进行场景的一致性分析。图 1 所示的办理进路顺序图对应的 9 个场景如图 2 所示。

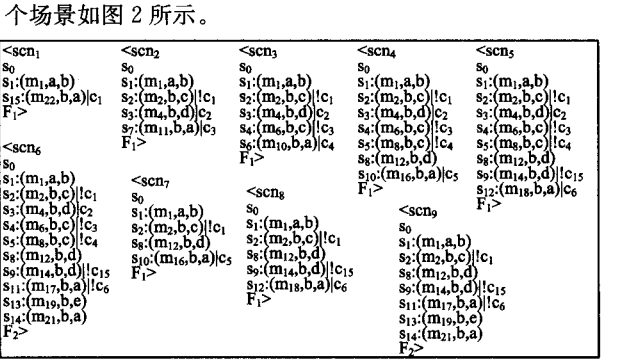

图 2 办理进路顺序图对应的 9 个场景

图 2 中，a 表示 Signaler 对象，b 表示 Interlocking System

对象，c 表示 Switch 对象，d 表示 Section 对象，e 表示 Signal 对象，F_1 表示进路建立成功的场景终点，F_2 表示进路建立失败的场景终点。

为验证联锁软件顺序图场景的一致性，需提取与交互行为相关的全局变量组成一个状态向量，分析消息的前后置状态向量值，分以下两种情况进行顺序图矛盾检测：

(1)对于单个场景，若前一消息的后置状态向量值与后一消息的前置状态向量值不相等，则这两个消息之间存在矛盾。

(2)对于多个场景，若它们之间存在同一消息，检查该消息在各场景的状态向量值是否符合联锁逻辑，若不符合，则场景之间存在矛盾。

根据需求场景，选取 7 个布尔型的车站联锁系统的全局变量构成的状态向量为：SwiPosCorrect(道岔位置是否在规定位置)，SwiSecIdle(道岔所在区段是否空闲)，SwiLock(道岔是否已经锁闭)，SectIdle(进路中的区段是否空闲)，ConflictRouteBuilt(敌对进路是否建立)，SigOpen(信号机是否开放)，RouteBuilt(进路是否建立)。

场景 6 对应的消息的前/后置状态向量值如表 2 所列，场景 9 对应的消息的前/后置状态向量值如表 3 所列。

表 2　场景 6 中消息的前/后置状态向量值

M	Pre-state vector value	Post-state vector value
m_1	⟨null,null,null,null,null,f,f⟩	⟨null,null,null,null,null,f,f⟩
m_2	⟨null,null,null,null,null,f,f⟩	⟨f,null,null,null,null,f,f⟩
m_4	⟨f,null,null,null,null,f,f⟩	⟨f,t,null,null,null,f,f⟩
m_6	⟨f,t,null,null,null,f,f⟩	⟨f,t,null,null,null,f,f⟩
m_8	⟨f,t,t,null,null,f,f⟩	⟨t,t,t,null,null,f,f⟩
m_{12}	⟨t,t,t,null,null,f,f⟩	⟨t,t,t,t,null,f,f⟩
m_{14}	⟨t,t,t,t,null,f,f⟩	⟨t,t,t,t,t,f,f⟩
m_{17}	⟨t,t,t,t,t,f,f⟩	⟨t,t,t,t,t,f,f⟩
m_{19}	⟨t,t,t,t,t,f,f⟩	⟨t,t,t,t,t,t,f⟩
m_{21}	⟨t,t,t,t,t,t,f⟩	⟨t,t,t,t,t,t,t⟩

表 3　场景 9 中消息的前/后置状态向量值

M	Pre-state vector value	Post-state vector value
m_1	⟨null,null,null,null,null,f,f⟩	⟨null,null,null,null,null,f,f⟩
m_2	⟨null,null,null,null,null,f,f⟩	⟨t,null,null,null,null,f,f⟩
m_{12}	⟨t,null,null,null,null,f,f⟩	⟨t,null,null,t,null,f,f⟩
m_{14}	⟨t,null,null,t,null,f,f⟩	⟨t,null,null,t,t,f,f⟩
m_{17}	⟨t,null,null,t,t,f,f⟩	⟨t,null,null,t,t,f,f⟩
m_{19}	⟨t,null,null,t,t,f,f⟩	⟨t,null,null,t,t,t,f⟩
m_{21}	⟨t,null,null,t,t,t,f⟩	⟨t,null,null,t,t,t,t⟩

根据消息之间的状态向量值，表 2 和表 3 中无消息矛盾，分析其他 7 个场景消息的前/后置条件，无消息矛盾，因此，建立的办理进路顺序图模型满足场景一致性。

3.2　系统形式化模型生成子算法

根据联锁软件的场景生成顺序图的自动机模型分为 2 步：首先根据顺序图中每个对象的事件序列生成对应的子自动机模型；然后合成不同场景的子自动机模型生成整个系统的组员自动机模型。

联锁系统的输入输出为离散事件，具有典型的离散事件特性，系统可处于有限状态中的任何一个，其模型可抽象为事件确定有限自动机。事件确定有限自动机 ETDFA 是有限自动机 DFA 的扩展，一个状态转换表示一次消息交互。ETDFA 描述顺序图中交互对象间的消息序列，通过多个对象的积自动机描述系统行为的交互过程。

定义 3　事件确定有限自动机[16]可以定义为一个七元组 $ETDFA=(Q,C_c,E(M),\Sigma,\delta,q_0,F_s)$，$\Sigma=\{(c,e)\mid c\in C_c, e\in E(M)\}$，$E(M)$为消息对应的事件，其中，$Q$ 为有限状态集合；M 为输入字母表的有限集合，其中一个字母代表一个输入事件；δ 为状态转移函数，$\delta: Q\times\Sigma\to\Sigma$；$q_0$ 为 ETDFA 的初始状态，$q_0\in Q$；F_s 为 ETDFA 的终止状态集合，$F_s\subseteq Q$。

对象 $O_i=(M,C,C_c,C_r,n,f_{mc})$对应的 $EFDFA_i=(Q,C_c,E(M),\Sigma,\delta,q_0,F_s)$，其中，$q_0$ 为初始状态，未记录事件；输入字母表 Σ 是 C_c、$E(M)$组成的序偶对，每次迁移转换可表示为执行条件/消息事件($C_c/E(M)$)；F_s 是 ETDFA 的终止状态集合，一个场景对应一个终止状态。非初始状态 q_i($i\geqslant 1$)是事件序列到对象 O_i 的映射。

算法的具体步骤为：①创建初始状态 q_0。②根据事件依次创建状态 $q_1,q_2,\cdots,q_n$，其中，n 表示事件个数。③根据事件 e_i 与 e_j 之间的关系确定 q_i 与 q_j 之间的迁移。如果 e_i 与 e_{i+1}都不在顺序图的任何组合片段内，那么 $\delta(q_i,q_{i+1})=e_{i+1}$；如果 e_i 与 e_{i+1}在顺序图中的同一组合片段的相同操作域内，那么 $\delta(q_i,q_{i+1})=e_{i+1}$；如果 e_i 与 e_{i+1}一个在组合片段内，一个在组合片段外，则根据事件所在组合片段类型判断 q_i 与 q_{i+1}之间的迁移关系。④确定终止状态。

Signaler 的 ETDFA、Switch 的 ETDFA、Signal 的 ETDFA、InterlockingSystem 的 ETDFA、Section 的 ETDFA 分别如图 3—图 7 所示。

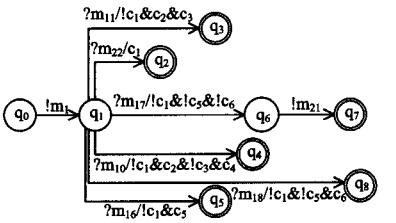

图 3　Signaler 对应的 ETDFA

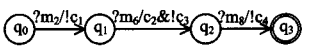

图 4　Switch 对应的 ETDFA

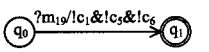

图 5　Signal 对应的 ETDFA

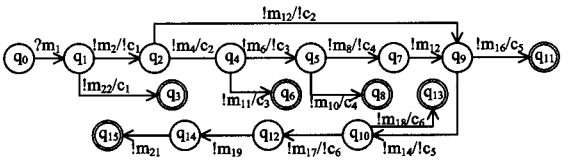

图 6　InterlockingSystem 对应的 ETDFA

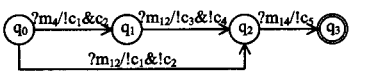

图 7　Section 对应的 ETDFA

对各对象生成的 ETDFA 作组合运算得到系统的自动机模型，对于系统的终点状态，限定有进路办理成功和进路办理失败两种，图 2 对应的系统 ETDFA 如图 8 所示。

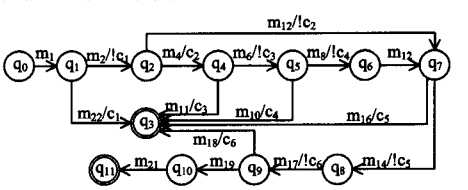

图 8　系统对应的 ETDFA 模型

结束语　本文以办理进路顺序图为基础研究计算机联锁

(下转第 226 页)

部分进行简化，同时通过状态约束表达动作的间接影响可以减少动作规则的数量；3)将动作规则拆分成因果规则声明和执行条件声明；4)处理噪音和观察不完全情况，包含噪音和观察不完全的数据集更加接近实际情况。

参 考 文 献

[1] Certicky M. Action Learning with Reactive Answer Set Programming: Preliminary Report[C]// The Eighth International Conference on Autonomic and Autonomous Systems (ICAS 2012). 2012: 107-111

[2] Estlin T, Gaines D, Chouinard C, et al. Increased Mars rover autonomy using AI planning, scheduling and execution[C]// IEEE International Conference on Robotics and Automation, 2007. IEEE, 2007: 4911-4918

[3] Yang Q, Wu K, Jiang Y. Learning action models from plan examples using weighted MAX-SAT[J]. Artificial Intelligence, 2007, 171(2): 107-143

[4] McCarthy J. Elaboration tolerance[OL]. 1997-09-09. http://www-formal.stanford.edu/juc./elaboration.html

[5] Gelfond M, Kahl Y. Knowledge Representation, Reasoning, and the Design of Intelligent Agents[OL]. http://redwood.cs.ttu.edu/~mgelfond/FALL-2012/book.pdf

[6] IPC(2003) [OL]. http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume20/long03a-html/node37.html

[7] Balduccini M. Learning Action Descriptions with A-Prolog: Action Language C[C]// AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning. 2007: 13-18

[8] Gebser M, Grote T, Kaminski R, et al. Reactive answer set programming[M]. Logic Programming and Nonmonotonic Reasoning. Springer Berlin Heidelberg, 2011: 54-66

[9] Wang X. Learning by observation and practice: An incremental approach for planning operator acquisition[C]// ICML. 1995: 549-557

[10] Sil A, Yates A. Extracting STRIPS Representations of Actions and Events[C]//RANLP. 2011: 1-8

[11] Boose J H, Gaines B R. Knowledge acquisition for knowledge-based systems: Notes on the state-of-the-art[J]. Machine Learning, 1989, 4(3/4): 377-394

[12] Benson S. Learning action models for reactive autonomous Agents[D]. Stanford university, 1996

[13] 谢颖. 归纳逻辑程序设计初探[D]. 北京：北京师范大学哲学系，2008

[14] Stuart R, Peter N. 人工智能——一种现代方法(第2版)[M]. 姜哲，金奕江，等译. 北京：人民邮电出版社，2010

[15] Lorenzo D. Learning non-monotonic causal theories from narratives of actions[C]// NMR. 2002: 349-355

[16] Pasula H, Zettlemoyer L S, Kaelbling L P. Learning Probabilistic Relational Planning Rules[C]// ICAPS. 2004: 73-82

[17] Yang Q, Wu K, Jiang Y. Learning Actions Models from Plan Examples with Incomplete Knowledge[C]// ICAPS. 2005: 241-250

(上接第 195 页)

软件的事件确定有限自动机模型的生成方法。该方法通过UML顺序图来描述系统对象之间的交互行为，选取一组相关的全局变量构成状态向量对顺序图中各场景的交互信息的前后置状态向量值进行分析，检测场景中存在的矛盾，得到一致的需求场景；然后从顺序图中提取每个对象的事件序列集合，构造各个对象的时间确定有限自动机模型，最后组合各对象的自动机模型得到系统的有限自动机模型。该方法可以根据系统顺序图建立具有严格数学基础的系统确定有限自动机模型。生成的系统有限自动机模型可作为系统测试用例生成的基础模型，也可用于对系统行为进行诊断分析，这是以后工作的重点。

参 考 文 献

[1] 张曙光. 高速铁路系统生命周期安全评估体系的研究[J]. 铁道学报，2007，29(2)：20-26

[2] The European Committee for Electro-technical Standardization. EN 50126 Railway Applications-the Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)[S]. UK: BSI, 2002

[3] CENELEC. EN 50128 Railway Applications: Communications, signaling and processing systems-Software for railway control and protection systems[S]. UK: CENELEC, 2001

[4] CENELEC. EN 50126 Railway Applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety(RAMS) [S]. UK: CENELEC, 1999

[5] IEC. IEC61508 Functional Safety of electrical/ electronic/programmable electronic safety- related systems-part3: software requirements[S]. UK: IEC, 2000

[6] IEC. IEC61508 Functional Safety of electrical/ electronic/programmable electronic safety- related systems-part7: Overview of techniques and measures[S]. UK: IEC, 2006

[7] 王曦，徐中伟，梅萌. 基于模型检测的软件安全性验证方法[J]. 武汉大学学报，2010，56(2)：156-160

[8] Haxthausen A E, Peleska J. Formal Development and Verification of a Distributed Railway Control System[J]. IEEE Transactions on Software Engineering, 2000, 26(8): 1546-1563

[9] Garmhausen V H, Campos S, Cimatti A. Verification of a safety-critical railway interlocking system with real-time constraints [J]. Elsevier Science of Computer Programming, 2000(36): 53-64

[10] Yang Shuang-hua, Yang Li-li. Automatic safety analysis of control systems[J]. Journal of Loss Prevention in the Process Industries, 2005, 18(3): 178-185

[11] Bozzano M, Villa-orita A. The FSAP/NuSMV-SA Safety Analysis Platform[J]. Software Tools for Technology Transfer, 2007, 9(1): 5-24

[12] Koh K Y, Seong P H. SMV model-based safety analysis of software requirements[J]. Reliability Engineering & System Safety, 2009, 94(2): 320-331

[13] 赵志熙. 计算机联锁系统技术[M]. 北京：中国铁道出版社，1999：20

[14] Booch G, Rumbaugh J, Jacobsn I. UML 用户指南[M]. 邵维忠，译. 北京：机械工业出版社，2001：59-78

[15] Kim H, Russell M. Learning UML 2.0[M]. California: O' Reilly, 2006: 1-286

[16] 张琛，段振华. 应用 UML2.0 模型的测试用例生成方法[J]. 西安交通大学学报，2011，45(8)：18-23