文章编号：0253-374X(2021)03-0391-09　　**DOI**：10.11908/j.issn.0253-374x.20317

基于通信的列车控制系统数据安全影响分析

陈宇佳，曾小清，袁腾飞

(同济大学 道路与交通工程教育部重点实验室，上海 201804)

摘要：为在设计阶段验证交互数据在系统中的安全特性，基于有色Petri网与失效传播模型提出了一种对系统模型自动分析得到输入数据失效最小割集的安全影响分析方法。首先，建立系统的有色Petri网模型，通过对库所数值离散化处理得到用例，对变迁进行单元测试得出失效行为函数，获得具有失效行为变迁的失效传播有色Petri网；然后基于有色Petri网模型的状态空间，通过失效空间生成算法与变迁回溯法，实现由模型状态空间到失效状态空间，再到失效状态树的转换，并通过失效状态树的合并与计算，获得系统失效最小割集。最后通过实验，比较不同数据安全保障措施下系统的失效最小割集变化，验证了安全保障措施对系统安全的作用，实验结果证明了该分析方法的正确性与有效性。

关键词：系统工程；安全评价；失效传播；信号系统；数据失效

中图分类号：U282^{+}.2　　**文献标志码**：A

Analysis of Safety Impact of Data in Communication-Based Train Control System

CHEN Yujia, *ZENG Xiaoqing*, *YUAN Tengfei*

(Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract: In order to verify the safety characteristics of interactive data in the system in the design stage, a safety impact analysis method based on colored Petri nets and the failure propagation model is proposed to automatically analyze the system model and obtain the minimum cut set of input data failure. First, the colored Petri net model of the system is established, and the case is obtained by numerical discretization of the place. The failure behavior function is obtained by unit test of the transition using the use case, and the failure propagation colored Petri net with failure behavior transition is obtained. Then, based on the state space of the colored Petri net model, the failure space generation algorithm and the transition backtracking method are used to realize the transformation from the model state space to the failure state space, and then to failure state tree. The minimum cut set of system failure is obtained by merging and calculating the failure state tree. Finally, through the experiment, the change of the minimum cut set of system failure under different data safety measures is compared, and the effect of security measures on system security is verified. The experimental results prove the correctness and effectiveness of the analysis method.

Key words: system engineering; safety evaluation; failure propagation; signal system; data failure

基于通信的列车控制系统(communication based train control, CBTC)通过列车精确定位与车地通信保证保障列车安全高效运行。由于外部环境或人为等因素的存在，尽管大部分系统内部具备安全保障措施，模块与子系统之间的数据传输仍存在出错的风险[1-3]。2017年11月15日新加坡的裕群地铁站追尾事故，就是信号系统版本过渡的不完整引起的新系统的输入数据错误，进而导致了安全距离计算错误而引发的。在系统设计或改造初期较难兼顾细节，容易对数据设计考虑不足，可能造成容错需求和执行阶段容错处理过程之间的脱节，进而引发较高的风险或返工成本。如上述事故，若能提前重点关注该输入数据，就有机会减轻或避免上游数据引发的一系列影响，避免事故发生。

近年来，数据在系统中的安全影响在越来越多的研究中得到重视，很多针对数据的系统安全分析与保障方法随之涌现。Wang等[4]采用区分于传统

收稿日期：2020-08-07
基金项目：上海市科学技术委员会项目(20DZ1202900, 19DZ1204200)，上海市住房和城乡建设管理委员会项目(JS-KY18R022-7)
第一作者：陈宇佳(1989—)，男，博士生，主要研究方向为交通信息工程。E-mail：chenyujia2@126.com
通信作者：曾小清(1969—)，女，教授，博士生导师，工学博士，主要研究方向为交通控制与安全等方向。E-mail：zengxq@tongji.edu.cn

论文拓展介绍

的自然语言描述的布尔逻辑表达数据约束,然后转换成简化有序二元决策图进行数据安全性判断。Song等[2]提出了新的通讯单元结构进行数据传输安全的保障。Sokołowska等[5]从通讯角度进行数据安全失效现象分析与保障措施设计。Iliasov等[6]采用目标结构表示法构建安全案例,针对系统结构的底层数据进行形式化分析,最终进行案例的分析与验证。周果等[7]从数据安全在系统安全中的角色出发进行数据结构建模,并采用图搜索算法和内积法进行安全验证。Wang等[8]基于仿真实验平台,采用蒙特卡洛仿真分析配置数据的错误对系统运行情况的影响,并得出了仿真系统的测试报告。上述研究成果中,文献[2, 4-6]采用了有效的措施,提高了数据的安全性,文献[6-8]提出了面向系统数据的安全影响分析方法,这些研究对通过提高数据正确性来增强系统可用性与可靠性具有重要意义。然而这些方法仍然存在一些不足,首先,这些研究仅分析了错误输入对系统输出结果的表现,较难反过来用于方案设计的论证;其次,部分研究仅关注数据的正确性验证与保障,对数据的安全关键点缺乏针对性。

CBTC是复杂度高的并发系统且具有明显的数据驱动特性,明确系统输入数据对系统的安全影响至关重要。在系统设计阶段,采用形式化方法对系统功能建模分析通常严谨有效。有色Petri网(colored Petri network,CPN)[9]具有强大的分析能力和高级编程语言的优势,适用于复杂动态并发模型的分析,然而面对大规模的系统仍然面临状态空间爆炸问题[10]。近年来,针对模块化系统的基于失效传播与转换表示(failure propagation and transformation notation, FPTN)[11]得到了一定程度的发展[12]。失效传播转换计算(failure propagation and transformation calculus, FPTC)[13]是在FPTN基础上扩展的失效评估方法,对系统复杂度的容忍能力较强,且可在系统原设计结构基础上转换为失效传播与转换图(failure propagation and transformation graph, FPTG)进行分析,并支持一定程度的自动化。FPTC关注网络上的令牌传递,不需要遍历生成状态空间,因此不存在状态空间爆炸问题,但缺乏系统性分析能力。CPN与FPTG具有结构上的相似性,若将二者结合,既可支持在系统层面进行数据的影响的论证,又可实现失效关键点定位的功能。

综上所述,本文结合CPN与失效传播机制,提出了一种可以在设计阶段进行的对数据在系统中的重要性与影响情况进行安全影响分析的方法。该方法分为失效传播CPN模型(failure propagation and transformation CPN,FPT-CPN)的构建与分析两部分。

1 数据驱动FPT-CPN构建

根据CBTC国际标准IEEE1474的描述,一般情况下,CBTC系统由列车自动监控系统(automatic train supervision, ATS)、区域控制器(zone controller, ZC)、计算机联锁(computer interlocking, CI)、车载控制器(vehicle onboard control, VOBC)等部分组成。CBTC子系统之间通过数据传递实现协作,对各子系统,其功能的实现依赖于传入数据的上游子系统,各子系统具有明显的数据驱动特性。

系统设计中涉及数据的工作包括数据内容设计与数据容错方案设计。数据内容设计为数据的参数化设计,如列车运行信息是ZC的输入数据,将列车运行信息细化为车头位置、车尾位置、列车时速等具体参数的过程;数据容错方案设计是确定数据安全保障措施的过程,如对列车时速采用三取二还是逻辑容错。为针对不同数据设计方案进行安全影响的分析,需构建数据驱动的FPT-CPN模型。数据驱动FPT-CPN的建模流程为:①根据系统需求,构建数据驱动的CPN模型;②对CPN模型的结构进行扩展,转换为FPT-CPN。

1.1 FPT-CPN构建

有色Petri网(colored Petri net, CPN)是由矩形节点和圆形节点组成的有向图,可通过一个9元组描述:

$$\mathrm{CPN}=(P,T,A,\Sigma,V,C,G,E,I)$$

其中:P为一个有限集,表示库所;T为一个有限集,表示变迁;A为一个有限集,$A\subseteq P\times T\cup T\times P$,表示弧;$\Sigma$为一个非空有限集,表示颜色集;$V$为一个有限集,表示变量类型;$C$等一个颜色集函数,为每一个库所分配一个颜色集;$G$为一个谓词函数,为每一个变迁分配一个谓词;$E$为一个弧表达式函数,为每一个弧分配一个表达式;$I$为一个初始化函数,为每一个库所分配一个表达式。CPN的结构如图1所示,其中p1、p2和p3表示库所,add表示变迁,INT为库所数据类型,p1的初始化令牌为2,i_1、i_2表示通过弧传递的变量。在CPN中,通常用库所表示实体。考虑数据驱动特性,系统数据均采用库所表示。

在CPN的基础上进行扩展,引入失效库所P_F和失效变迁T_F,定义FPT-CPN如下:

$$\text{FPT-CPN}=(\mathrm{CPN},D_F,P_F,T_F,G_F)$$

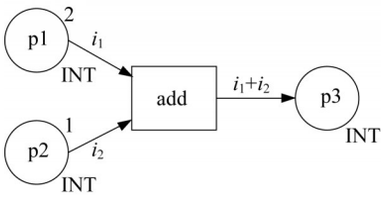

图 1　CPN 结构

Fig. 1　Structure of CPN

其中：P_F 为一个映射函数，为每个库所分配一个失效库所；T_F 为一个映射函数，为每个变迁分配一个失效变迁；D_F 为一个有限集，表示数据失效的集合；G_F 为一个谓词函数，为每个 $t\in T_F$ 分配一个失效行为函数。FPT-CPN 的结构如图 2 所示，其中 f_p1、f_p2 和 f_p3 表示失效库所，h_add 表示失效变迁，h_add$\in T_F$，F_INT 为 INT 类型数值对应的失效类型。

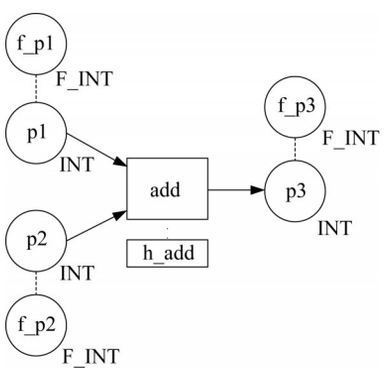

图 2　FPT-CPN 结构

Fig. 2　Structure of FPT-CPN

1.2　FPT-CPN 模型转换

首先，根据数据类型定义数据失效集 D_F，数据失效定义如表 1 所示。失效事件的定义为＜库所名称：失效模式＞。假设在 main 页有数值型库所 main′place，其失效模式为 gt，则该事件表达为 main′place：gt。

对每个 $t_i\in T_F$ 进行单元测试，得到失效传递函数 $f_i\in G_F$。令 $O(t_i)$ 表示 $t_j\in T$ 对应的输出库所集合，$I(t_j)$ 表示 $t_j\in T$ 对应的输入库所集合。用例生成算法如下。

算法 1　数值用例生成算法

（1）　初　始　化：CPN＝$(P, T, A, \Sigma, V, C, G, E, I)$，令库所数量为 n，变迁数量为 m，令 $Q=\{S_1, S_2, \cdots, S_n\}$，其中 S_i 表示库所 $p_i\in P$ 的取值集合。

表 1　数据失效定义

Tab. 1　Definition of data failure

数据类型	失效模式	符号	说明
数值型(num)	Greater	gt	大于正常值
	Smaller	lt	小于正常值
	Value	v	值错误(不确定与正常值的大小关系)
布尔型(bool)	True to False	tf	将 true 错误输出为 false
	False to True	ft	将 false 错误输出为 true
字符型(str)	Wrong String	wstr	错误字符内容
枚举型(enum)	Others	ot	错为其他枚举内容
不明确	Undecided	Und	复合类型数值，待细分为上述类型
All	Normal	*	无失效
	omission	om	输出丢失
	Customized	ctm	自定义失效

（2）对每个 $p_k\in P$，如果 $C(p_k)=$num，即库所 p_k 的类型为 num：则：

$$S_k\leftarrow\{S_k\}\bigcup_{i=0}^{\mathrm{ceil}\left(\frac{e_k-s_k}{j_k}\right)}(s_i+j_k\cdot i)\bigcup\{e_k\}$$

其中：s_k 为 p_k 的数值下界；e_k 为 p_k 的数值上界；j_k 为 p_k 的数值步长。

（3）Repeat

给定起始用例集合 $M\leftarrow Q$

Repeat（对每个 $t_j\in T$）

对每个 $p_i\in O(t_j)$，

$$Q(p_i)\leftarrow Q(p_i)\bigcup G(t_j)$$

$$M\leftarrow Q$$

Until $M=Q$

（4）输出最终用例集：Q

基于 D_F 进行失效行为 G_F 的生成。首先，对应表 1 逻辑构建失效判别函数，令 D 表示一个失效判别函数，为每个颜色集分配一个失效判别函数。令 v_t 表示真值，v_f 表示失效值，有失效类型 $t=D_c(v_t, v_f)$。G_F 生成算法如下所示。

算法 2　G_F 生成算法

（1）　初　始　化：CPN＝$(P, T, A, \Sigma, V, C, G, E, I)$，$G_F=\varnothing$

（2）对每个 $t_i\in T$，失效输入集 $F_{1i}=\{Q(p_1)\times Q(p_2)\times\cdots\times Q(p_n)|p_k\in I(t_i)\}$。

对 $(v_t, v_f)\in F_{1i}\times F_{1i}$，对 $p_k\in O(t_i)$，有：

$$f_{ik}(I(t_i), D_{C(p_k)}(v_t, v_f))\leftarrow\sum_{(x_{n1}, x_{n2})}^{F_{1i}\times F_{1i}} D_{C(p_k)}(G(t_i)(x_{n1}), G(t_i)(x_{n2}))$$

$$G_F(i,k) \leftarrow f_{ik}$$

2 FPT-CPN分析方法

基于FPT-CPN的分析过程如下：首先，对每个用例，分别生成该FPT-CPN对应的失效状态空间；应用状态空间转换算法，将失效状态空间转换为失效布尔表达式；合并失效布尔表达式，计算得出最小割集。

2.1 失效状态空间生成

以图3所示系统为例，在三取二结构组件cmp外部添加main页，构建一个虚拟系统，其中两个子页的变量是统一的，库所data表示原始输入数据；库所d1、d2和d3表示输入数据；库所s1、s2和s3表示判断结果；库所output表示输出；变迁gen_data为流程控制；变迁cmp表示判断功能；变迁gen_by_s1、gen_by_s2和gen_by_s3表示输出功能的实现；变迁add1、add2、add3、add4表示功能的控制；P_HIGH和P_LOW为优先级常量；true、false是CPN中的布尔常量；变量a、b、c、ba、bb、bc、ab为CPN中的不定变量，起传递值的作用。该系统的状态空间如图4所示，其中圆角矩形内上面数字表示状态空间标识，下面的冒号前后数字分别表示网络结构上上游节点和下游节点的个数。

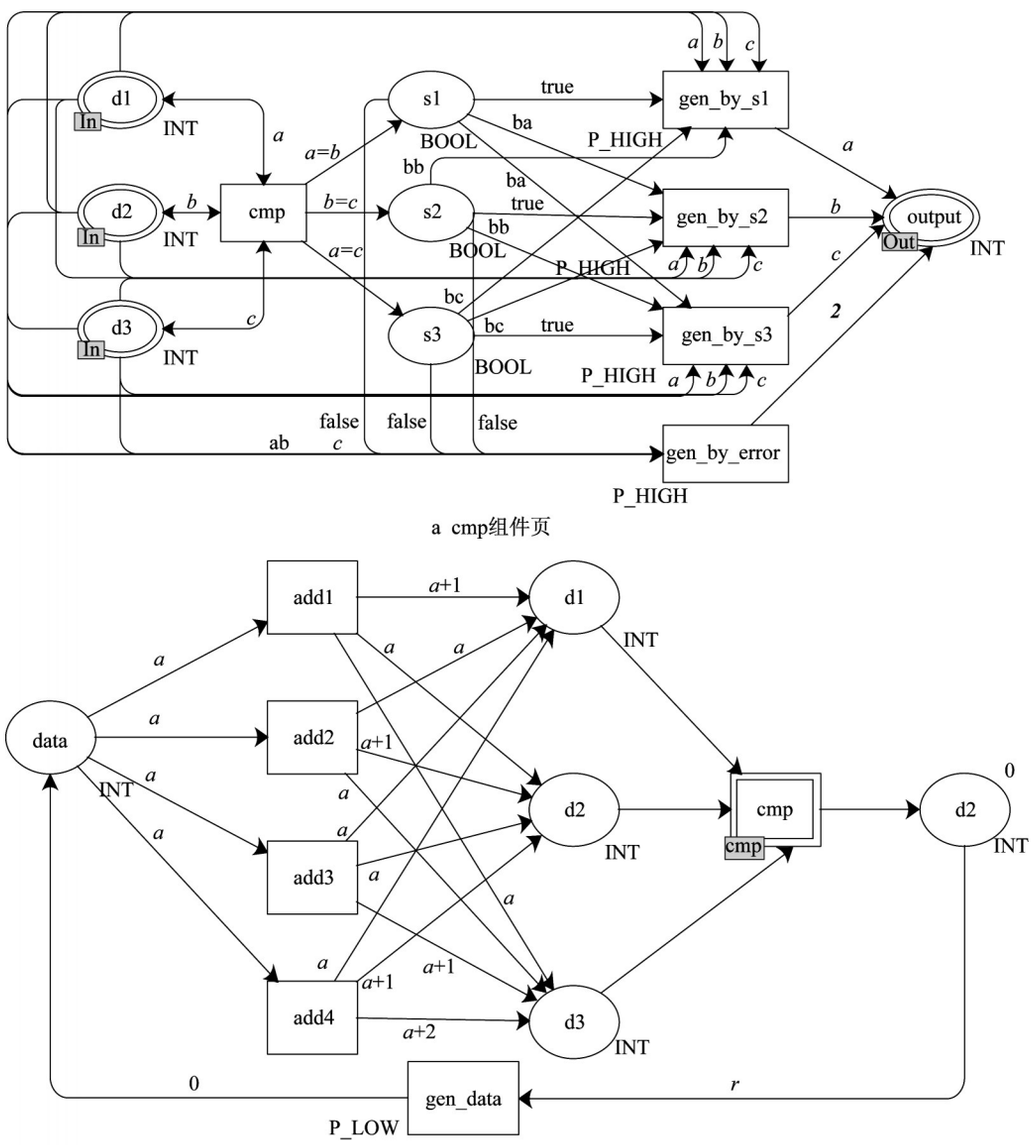

图3 三取二结构的CPN模型

Fig. 3 CPN model of two out of three structure

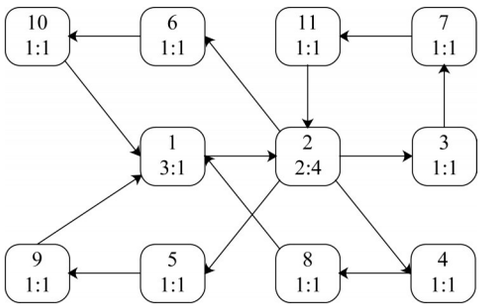

图 4　状态空间图（三取二结构）

Fig. 4　Graph of state space (two out of three structure)

失效状态空间生成算法如下所示。其中状态空间生成算法在本文中不进行进一步阐述。

算法 3　失效状态空间生成算法

（1）定义：令 S_n 表示全局计算的第 n 个状态空间，对 $\exists s \in S_n$，s 为系统状态节点。令 $T_{\text{pout}}^{(i)}$ 表示库所 $p_i \in P$ 输出的变迁的集合。对状态节点 s，令 $T_{\text{sout}}^{(j)}$ 表示状态节点 $s_j \in S_n$ 对应输出的变迁的集合。

（2）初始化：CPN $=(P, T, A, \Sigma, V, C, G, E, I)$，对CPN生成初始状态空间 S_0，令失效状态空间集为 $S=\{S_0\}$，获得目标输入库所集合 P_{in}，$P_{\text{in}} \subseteq P$。

（3）备选触发变迁集合 $T_{\text{cand}} \leftarrow \bigcup T_{\text{pout}}^{(i)} | p_i \in P_{\text{in}}$，令状态空间变量 $s_s \leftarrow S_0$，$p \leftarrow P_{\text{in}}$

（4）对 s_s，p

Repeat

故障注入的目标状态节点集合 $S_N \leftarrow \{s_i | s_i \in s, T_{\text{sout}}^{(i)} \cap T_{\text{cand}} \neq \varnothing\}$

对每个 $s_i \in S_N$ 中的 $t \in T_{\text{sout}}^{(i)}$

测试用例集 $U_P \leftarrow p(P_{\text{in}})$，其中，$p$ 表示返回所有非空子集的函数

对每个 $P_U \in U_P$，

生成新的 S'，$s_s \leftarrow S'$

$p \leftarrow \text{diff}(p, P_U)$，其中diff为差集算子

对 s_s，p 执行步骤4

Until $S_N = \varnothing$

$S \leftarrow S \cup \{s_s\}$

假设库所d1、d2、d3的正确输入值为(0,0,0)，对其分别由失效值域 $\{-1,0,1,2\}$ 赋值，采用算法3生成失效状态空间，得到一组状态空间 S，其中一组状态空间示例如图5所示。对每个 $S_i \in S$，定义 $S_i = (P, K_i, F_i)$，其中 K_i 表示所有库所取值的集合，F_i 表示所有库所失效值的集合。建立映射函数 $C: S \rightarrow S'$，对 $S_i = (P, K_i, F_i)$，有 $C(S_i) = (P, F_i)$。由此，获得新的状态空间 $S' = \bigcup_s^S \{C(s)\}$。图5所示的状态空间转换后的结果见图6。

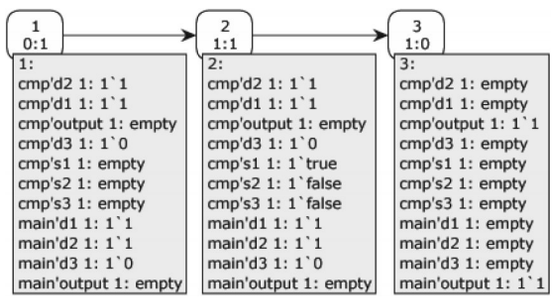

图 5　原始状态空间示例

Fig. 5　Example of original state space

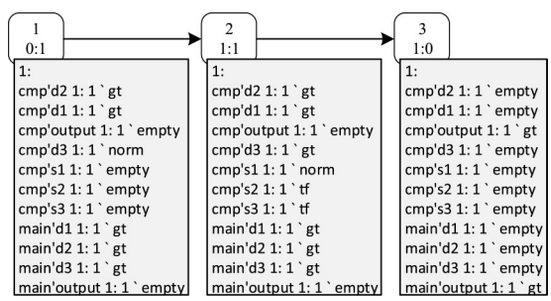

图 6　失效状态空间

Fig. 6　Failure state spaces

2.2　状态空间到失效状态树的转换

对 $S_i \in S'$，分别采用变迁回溯法确立失效状态树。失效状态树的定义与故障树一致，表达失效逐步传递，最终影响顶事件的树结构。区别在于，失效状态树描述的是某特定系统状态空间下的行为。变迁回溯法包括如下三个步骤：

（1）根节点与子节点确定。选择目标输出库所，确立目标失效的“顶事件”。根据分析目标，确立所有的输入库所和分析的“底事件”。

（2）路径抽取。选取所有以目标输出库所为输出的变迁，沿网络的方向逆向搜索直到找到某些变迁的输入库所全部为输入库所，得到失效传递的路径。

（3）布尔逻辑关系表达。对某个状态空间，其存在多个输入路径，则路径之间关系为“或”，若只存在一个输入路径，则该路径触发变迁的多个库所失效条件为“与”。

以上述三取二组件的cmp部分为例，令page′item：failure表示页面page下的对象item上的失效模式failure，选择分析顶事件为main′output：gt。对应图6的状态空间，其转换的对应失效状态树如图7所示，其中 x_1 表示cmp′d1：gt，x_2 表示cmp′d2：

gt。

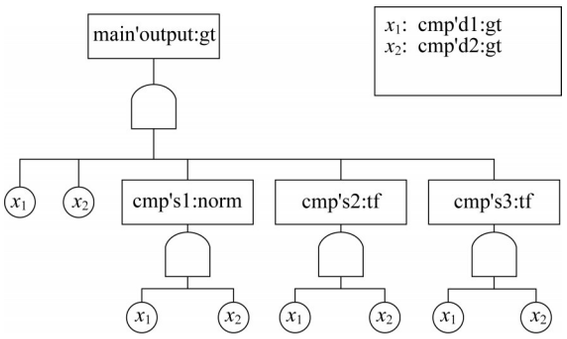

图7 失效状态树

Fig. 7 Failure state tree

2.3 失效状态树合并化简与最小割集计算

对每个失效状态$S_i \in S'$对应的失效状态树T_i分别求解最小割集,再求并集得到目标失效的最小割集。本文采用布尔代数法对每个失效状态树进行求解。以图7为例,该失效状态树的布尔表达式为

$$T = x_1 \bigcap x_2 \bigcap (x_1 \bigcap x_2) \bigcap (x_1 \bigcap x_2) \bigcap (x_1 \bigcap x_2)$$

显然,经化简,得$T = x_1 \bigcap x_2$,对应最小割集为{{cmp′d1:gt,cmp′d2:gt}}。合并20个相似的失效状态树,得该组件的失效main′output:gt对应的失效最小割集{{cmp′d1:gt,cmp′d2:gt},{cmp′d1:gt,cmp′d3:gt},{cmp′d2:gt,cmp′d3:gt}}。符合该组件的冗余特性。

3 实验分析

3.1 区域控制器切换场景建模

基于某实际线路设计,以区域控制器(zone controller, ZC)切换场景为例,构建系统模型。ZC切换是区域控制系统的特有场景,描述的是列车由一个ZC的管辖区域行驶到相邻ZC管辖区域的过程。相邻的ZC之间存在一个重叠区域,列车行驶进入重叠区域后,开始切换的流程。原ZC会向接管ZC发送移交申请,接管ZC收到申请后,开始通过原ZC发送移动授权(movement authority, MA)等控制命令,此时列车同时收二者的控制;在列车车尾越过切换应答器后,原ZC发出切断请求,列车切断与原ZC的通信,完成ZC切换。

首先,定义系统关键状态。假定ZC1为移交ZC,ZC2为接管ZC,根据列车运行的不同阶段i,定义列车状态s_i,各状态如表2所示。

ZC的切换过程中,列车状态从s_0逐步过渡到s_4,根据该状态变化构建模型。模型由顶层main(图8)和三个子层ZC1(图9),ZC2(图10),Train组成,

表2 列车状态定义

Tab. 2 Definition of train state

列车状态(i)	说明
0	列车在ZC1中运行
1	移动授权到达ZC分界点时
2	列车前端越过ZC分界点时
3	列车尾端越过ZC分界点时
4	列车最小安全末端越过ZC分界点后

main中的三个替代变迁分别对应三个子层。模型各层的库所、变迁与变量定义是统一的。在图8~图10中,库所TrainState表示列车的状态;以req和resp为前缀的库所,分别表示请求数据和响应数据;MA、MA1、MA2、MA_final和MA2_Stored表示移动授权数据;库所ZC_Connected、Conn_Train_ZC2、Conn_Train和conn_zc2表示ZC1、ZC2和Train之间的连接状态;库所Ctrl表示控制流程;库所malimit表示MA发送次数的控制;DB_ZC2和DB_Version为传递参数,表示数据库版本;变迁reqMA、joinMA、genMA、sendMA、send_ctrl_MA、send_MA和gen_MA分别表示对MA的操作;变迁req、resp、conn_resp和conn_train_resp表示请求与响应操作;变迁control为控制操作。各变量定义如图11所示,其中STRING和BOOL为基本数据类型,STRING表示字符串型,BOOL表示布尔型。

采用CPN Tools计算状态空间,得到该模型存在5个终止结点,图12为其中终止结点之一的部分库所的状态,经检查,5个终止结点的Train'MA_count的参数与图12相同,表示该切换过程列车均成功接收到各阶段应接收到的MA,符合系统需求。由模型可以看出,在ZC切换过程中,最重要的功能在于保证输出的MA的完整,验证了ZC的核心功能[14]。

3.2 多种安全保障措施下的对比实验

本文在ZC1的输入环节中加入安全保障措施,通过验证其效果证明本文分析方法的有效性。在安全保障措施中选用三取二结构组件和数据一致性验证、逻辑容错。实验设计分组见表3。

其中,三取二的方法采用的是如图3a所示的结构。一致性验证,是通过多个数据之间的逻辑关系判断,来实现逻辑冗余,在实验中将车头位置表示为列车车尾位置与车长之和,对此进行车头位置的数据校验。逻辑容错,是对数据内在逻辑关系进行判断,采取最安全的方案,因此将二者接收的列车车速进行比较,取较小值作为最终取值。

为了更直观比较不同保障措施的效果,对每个数据分别取20个离散值然后组合注入错误值加入

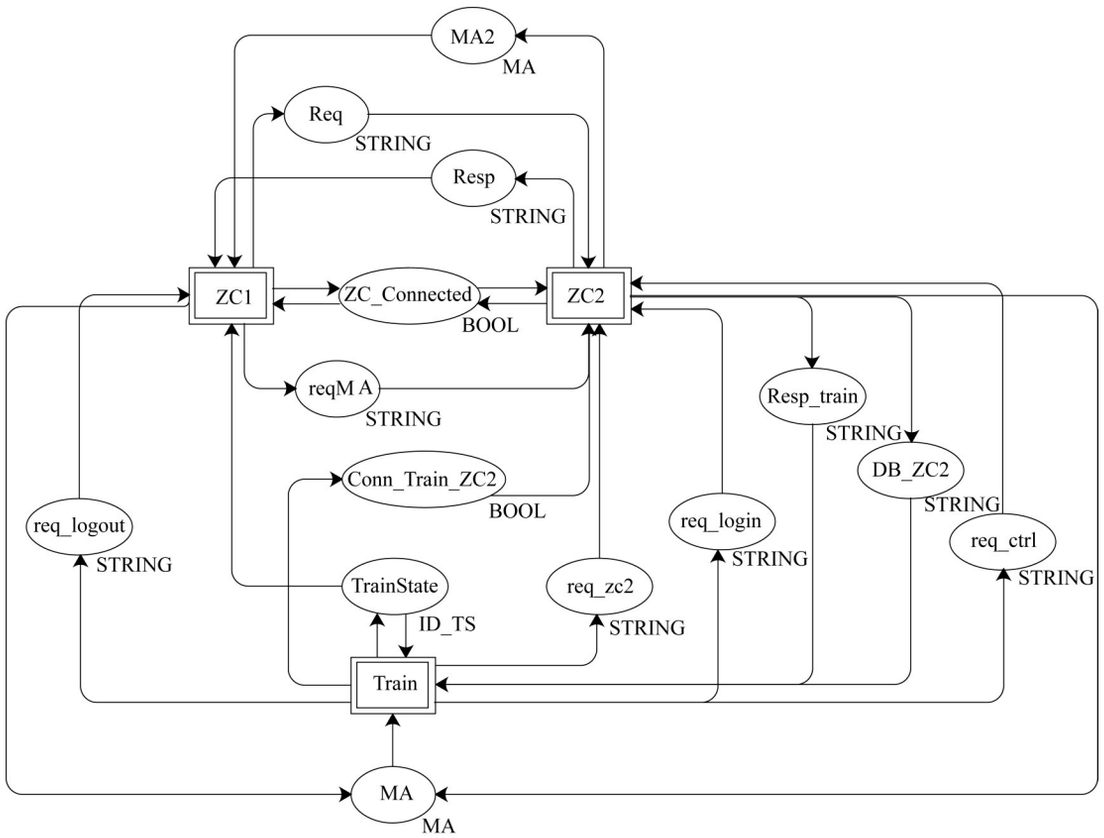

图 8　ZC切换模型(main)

Fig. 8　ZC transfer model(main)

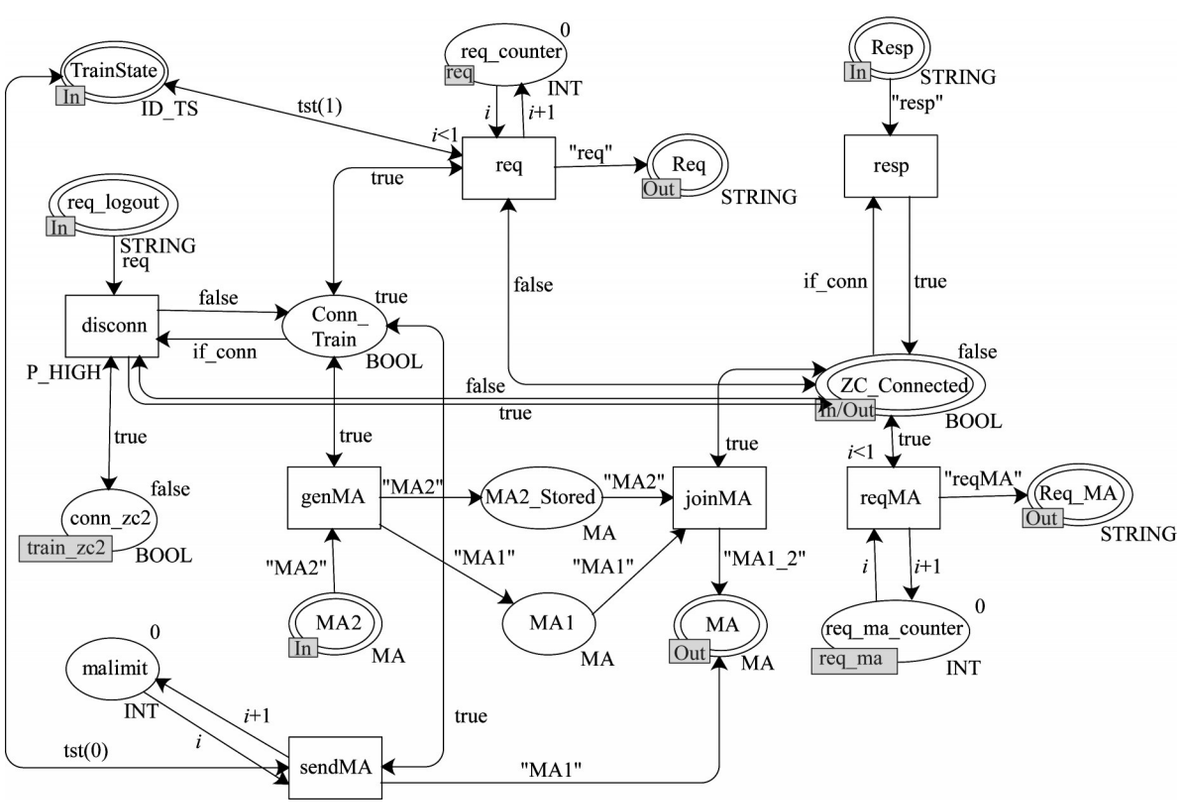

图 9　ZC切换模型(ZC1)

Fig. 9　ZC transfer model(ZC1)

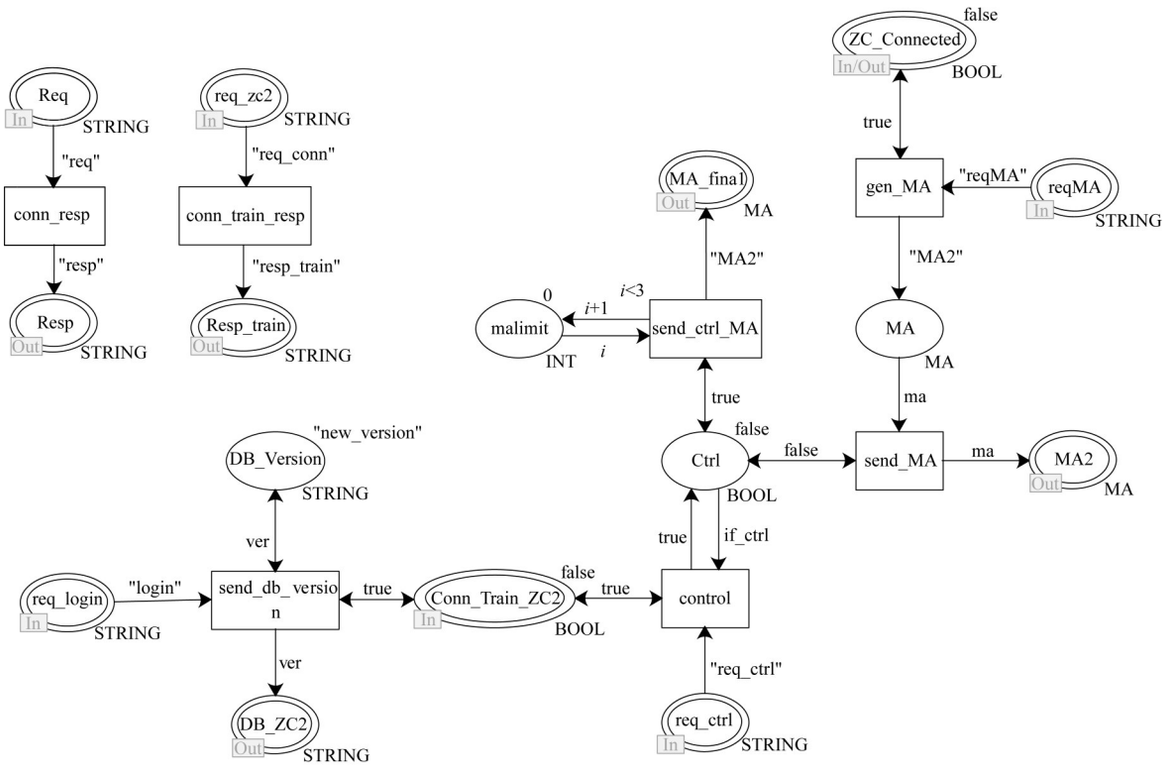

图 10 ZC切换模型(ZC2)

Fig. 10 ZC transfer mosdel(ZC2)

```
colset T_STATE = with p0|p1|p2|p3|p4;
colset ID_TS = index tst with 0..4;
colset MA=STRING;
colset MA_INFO=product ID_TS*MA;
var ts: ID_TS;
var ts1: ID_TS;
var i:INT;
var s:STRING;
var if_conn:BOOL;
var ma: MA;
var rstate:INT;
var n_req:INT;
var n_req_ma:INT;
var ver:STRING;
var ver_zc2:STRING;
var ver_new:STRING;
var login:BOOL;
var if_ctrl:BOOL;
var req:STRING;
```

图 11 变量说明

Fig. 11 Description of variables

```
146:
Train'ts_index 1: 1`5
Train'TrainState 1: 1`tst(4)
Train'MA 1: empty
Train'MA_count 1: 1`(tst(0),"MA1")++
1`(tst(1),"MA1_2")++
1`(tst(2),"MA1_2")++
1`(tst(3),"MA1_2")++
3`(tst(4),"MA2")
Train'req_counter 1: 1`0
Train'req_ma_counter 1: 1`0
Train'req_zc2 1: empty
Train'Resp_ZC2 1: empty
Train'req_zc2_counter 1: 1`1
Train'Conn_Train_ZC2 1: 1`true
Train'req_login 1: empty
Train'DB_ZC2 1: empty
Train'DB_Version 1: 1`"new_version"
```

图 12 ZC切换场景终止节点状态

Fig. 12 State of termination node in ZC switch scenario

表 3 实验方案

Tab. 3 Experimental scheme

序号	数据	库所	保障方法	备注
1	All	All	无	
2	车头位置	THeadPosition	三取二	
3	车速	TrainVelocity	三取二	
4	车头位置	THeadPosition	一致性验证	车尾位置+列车长度
5	车速	TrainVelocity	逻辑容错	ZC2获得的车速与ZC1获得的车速,取较大值

实验,主要库所的正确值取值见表4。其中,ZC1与ZC2的边界位置设为2 500,列车长度设为144。对正确值v_t,步长s,采用失效值空间$F_v=\bigcup_{i=0}^{n=2}\{v_t+i, v_t-i\}$。

表 4　输入库所取值范围

Tab. 4　Value range of input place

库所	最小值	最大值	步长
THeadPosition	2 000	2 950	50
TrainVelocity	0	19	1
TrainVelocity_ZC1	0	19	1
TrainVelocity_ZC2	0	19	1
TTailPosition	1 856	2 806	50

表 5　实验的最小割集

Tab. 5　Minimum cut sets of experiments

序号	最小割集
1	{{THeadPosition:gt},{TrainVelocity:gt}}
2	{{THeadPosition1:gt,THeadPosition2:gt},{THeadPosition1:gt,THeadPosition3:gt},{THeadPosition3:gt, THeadPosition2:gt},{TrainVelocity:gt}
3	{{THeadPosition:gt},{TrainVelocity1:gt,TrainVelocity2:gt},{TrainVelocity1:gt,TrainVelocity3:gt},{TrainVelocity2:gt,TrainVelocity3:gt}}
4	{{THeadPosition:gt,TTailPosition:gt},{TrainVelocity:gt}}
5	{{THeadPosition:gt}}

3.3　结果分析

以输出到列车的ZC1′MA:gt为目标失效,分别得到每组实验的最小割集,结果见表 5。

由实验1可见,在未进行容错与处理的条件下,最小割集很小,数据输入的失效可以直接引发整个系统的危险;对实验2、实验3和实验3,可得出可预期的结论,冗余通道对输出的保护是明显的,车头位置数据的割集为二元组,有效提高了失效发生条件的复杂度;实验4中,加入一致性检验后,车尾位置一并加入了最小割集,提高了失效发生的复杂度。同时也可发现,该割集是顶事件的必要不充分条件,具有更苛刻的触发条件。在实验5中,经过逻辑容错后,{TrainVelocity:lt}对结果的影响已经完全得到避免。实验结果证明,冗余结构具有通用性,且可以有效提高失效的触发难度,另外,逻辑冗余具有更强的容错能力,条件允许的情况下采用逻辑冗余,对系统容错能力有较大提升。

4　结论

本文结合失效传播方法与有色Petri网,建立了一套由系统原理模型到失效分析的自动化分析方法。该方法的实验结论表明,通过目标失效的最小割集,可以明确系统方案中数据的薄弱环节与重要危险侧输入;通过不同方案对比,可有效比对安全保障方案,进行系统安全设计。该方法对设计过程中进行数据对系统的安全影响提供了依据,为安全保障措施设计提供了技术支撑。

作者贡献申明:

陈宇佳:模型设计,仿真实验设计与实现;
曾小清:方案分析;
袁腾飞:数据处理。

参考文献:

[1] AZIMINEJAD A, LEE A W, EPELBAUM G. Underground communication radio propagation prediction for CBTC data Communication subsystem design [J]. IEEE Vehicular Technology Magazine, 2015, 10(3): 71.

[2] SONG H F, WU W, DONG H R, *et al*. Propagation and safety analysis of the train-to-train communication system [J]. IET Microwaves, Antennas & Propagation, 2019, 13(13): 2324.

[3] HEINRICH M , VATEVAGUROVA T , Arul T , *et al*. Security requirements engineering in safety-critical railway signalling networks[J].Security and Communication Networks, 2019,2019:1.

[4] WANG T D, ZHAO H B, ZHU L F, *et al*. Satisfiability Verification of engineering data safety rules of balise based on ROBDD [M]. New York: IEEE, 2016.

[5] SOKOLOWSKA L, TORU A. Safety method for wireless data transmission in control command systems[C]//proceedings of the 22nd International Scientific on Conference Transport Means 2018. [S.l.]: Kaunas University of Technology, 2018:49-56.

[6] ILIASOV A, ROMANOVSKY A. Formal analysis of railway signalling data [M]//2016 IEEE 17th International Symposium on High Assurance Systems Engineering. New York: IEEE, 2016: 70-77.

[7] ZHOU G, ZHAO H. Data safety verification of computer interlocking in urban railway signaling [J]. Journal of the China Railway Society, 2016, 38(8): 63.

(下转第 466 页)