基于时序关系的系统失效可达图生成方法

范亚琼 陈海燕
(南京航空航天大学计算机科学与技术学院 南京 211106)

摘 要 针对状态事件故障树生成系统可达图过程中存在的状态空间爆炸问题,提出了一种基于时序关系的系统失效可达图生成方法。通过分析触发和被触发类型事件的时序关系,对存在时序关系的事件进行排序,根据时序关系获得系统构件间的所有不可同时到达状态对,对构件间的可同时到达状态建立笛卡尔积,获得系统的所有可同时到达状态对,根据连接表和最小割集获得系统失效的状态可达图,从而有效解决系统失效可达图生成过程中存在的状态空间爆炸问题。应用基于时序关系的系统失效可达图方法生成鱼攻系统失效可达图,实验结果验证了该方法的可行性与稳定性;同时也为表明其能有效地缓解状态空间爆炸问题,为状态事件故障树生成系统可达图提供了一种新的方法。

关键词 状态事件故障树,时序关系,系统失效可达图,状态空间爆炸
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.12.032

System Failure Reachability Graph Generation Method Based on Temporal Relation

FAN Ya-qiong CHEN Hai-yan
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract In view of the state space explosion problem in the process of system reachability diagram for state/event fault tree, a method of system failure reachability diagram based on temporal relation was proposed in this paper. By analyzing the relationship between the triggering and the triggered event, the sequence of events are sorted. According to the temporal relation, all the pairs of the unreachable states of the system components can be obtained. Through establishing the Cartesian product of the reachable state of the components, all the reachable states of the system can be obtained. According to the connection table and the minimum cut set, the system can obtain the state reachable graph of the system failure, which effectively solves the problem of state space explosion in the generation process of the system failure map. The system failure reachability graph method based on sequence relation is used to generate the reachability graph of the torpedo attack system. The experiment verfied the feasibility and stability of the method. And the experiment shows that the method can alleviate the problem of state space explosion effectively, and provide a new method for the system to generate the system reachable graphs.

Keywords State/Event fault tree, Temporal relation, System failure reachability graph, State space explosion

1 引言

作战系统具有的一些特殊属性,如时序关系、功能依赖等,使得普通故障树很难应用到作战系统中。状态事件故障树[1](State/Event Fault Tree,SEFT)既可以表述构件内部的时序行为活动,又可以表述系统的因果失效关系,适用于描述作战系统的失效因果链。由于状态事件故障树缺乏严格的语义,分析过程大部分是基于模型转换转换成确定随机 Petri 网(Deterministic Stochastic Petri Nets,DSPN)模型[2]或 Markov 模型[3,4]的,但是在进行模型的等价转换时存在状态空间爆炸问题。

目前,国内外的研究人员在 SEFT 方面进行的研究很少,而对于 SEFT 在模型等价转换过程中存在的状态空间爆炸问题,其研究成果更少。已有的研究成果主要分为两个方面:基于 DSPN 的状态空间缩减方法[5]的研究,以及基于交互 Markov 的状态空间缩减方法[6]的研究。基于 DSPN 的状态空间缩减方法主要采用转移优先权方法、顶层事件状态终止方法、最小化图方法等,但是这些过程的主要缺陷是在进行状

到稿日期:2016-11-08 返修日期:2017-03-06 本文受十三五重点基础科研项目(JCKY2016206B001),江苏省六大人才高峰项目(XXRJ-004),软件新技术与产业化协同创新中心资助。
范亚琼(1990-),女,硕士生,主要研究方向为软件工程、系统建模与仿真,E-mail:804353898@qq.com;陈海燕(1979-),女,讲师,主要研究方向为数据挖掘、民航信息化等。

态空间缩减之前，需要先生成一个完整的系统可达状态图，这个巨大的任务通常会伴随着状态空间爆炸问题；交互 Markov 的状态空间缩减方法是基于弱互模拟技术的，该缩减方法依赖于构件合并顺序，当构件间顺序推进不当，或构件内部状态较多时仍会出现状态空间爆炸问题。Roth 博士[8]提出了在状态事件故障树的基础上直接生成系统失效可达图的方法，但是直接生成图方法依赖构件间的直接触发关系获得系统的不可达对，无法直接识别无直接触发关系的构件间的状态不可达对，依然会造成阶段性的空间爆炸。

本文针对 SEFT 直接生成系统失效可达图过程中存在的状态空间爆炸问题，提出了一种基于状态事件故障树构件内部时序关系的系统可达图生成方法。该方法以作战系统可靠性评估为背景，通过分析时序关系获得系统内部构件间的所有不可达状态对，排除构件间不可同时到达状态组，对构件间的可同时到达状态建立笛卡尔积，避免了已有的直接生成图方法中不可达状态对删除不完全的情形，有效缓解了系统失效可达图生成过程中存在的状态空间爆炸问题，缩短了系统失效图的生成时间。

2 问题描述

在状态事件故障树可靠性评估模型中，将状态事件故障树转换到系统失效状态可达图是最为关键的环节之一，而状态空间爆炸问题[9]是转换过程的关键所在。状态事件故障树中构件间的交互分为直接触发关系和间接触发关系两类。触发关系既可以是事件交互，也可以是信息传递。如图 1 所示，构件 C_1 通过触发事件 $C_1.T_1$ 触发构件 C_2 产生事件 $C_2.T_1$，通过 $C_1.T_2$ 触发构件 C_3 产生事件 $C_3.T_1$，其中构件 C_1 与构件 C_2、构件 C_1 与构件 C_3 分别存在直接触发关系，构件 C_2 和构件 C_3 存在间接触发关系。

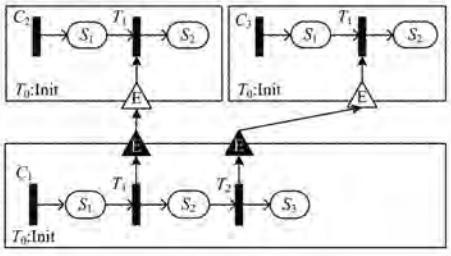

图 1 示例图

现有的状态事件故障树直接生成图方法可以通过直接触发关系确定两构件间的不可同时到达状态对，但是对于具有间接触发关系的不相邻的两构件间，无法确定构件内部的状态关系，这会导致获得的不可达状态组是不完全的，只能先建立整体系统的笛卡尔积，再删除其中具有不可达状态组的元组，未能有效解决状态空间爆炸问题。

3 基于时序关系的系统生成图方法

针对 SEFT 生成系统失效状态图过程中存在的状态空间爆炸问题，本文提出了基于时序关系的系统失效生成图方法，即根据时序关系[10-11]获得构件间不可同时到达状态对，通过笛卡尔积组合计算获得系统可同时到达状态对，进而通过连接表和最小割序集[12]获得系统失效图。该方法的成立基于以下前提条件：

(1)SEFT 的动态部分仅允许事件端口交互。一个状态事件故障树的动态部分由构件内部状态机以及状态依赖组成。仅允许触发事件塑造状态机之间的依赖关系。

(2)因果失效关系模型仅能通过纯状态门。该规则使得 SEFT 逻辑门语义与标准故障树的逻辑门一致，简化了 SEFT 失效模型的识别。

(3)禁止使用时序逻辑门，如延迟门、概率延迟门等。在动态 SEFT 模型中的时序行为通过时间事件塑造。

(4)失效场景通过状态表示。

以上 4 个前提条件简化了系统失效生成图方法。基于时序关系的系统失效生成图方法如图 2 所示。

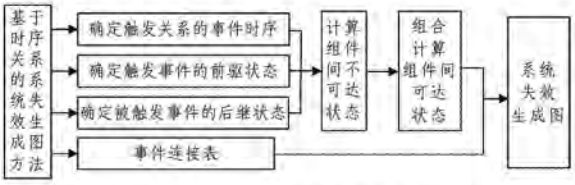

图 2 基于时序关系的系统失效生成图方法

系统失效生成图算法主要分为 6 步。

Step1 根据构件内部时序行为活动和构件间的触发关系，对系统内部触发和被触发类型事件进行排序。根据事件时序关系建立以下 3 条规则。

规则 1 对于构件 C_1 内部包含的两个触发或被触发类型事件 E_1，E_2，若事件发生时间 E_1 先于 E_2，则存在时序关系：$C_1.T_{E_1}<C_1.T_{E_2}$。

规则 2 对于存在触发关系构件 C_1 和 C_2，即构件 C_1 中的事件 E_1 与构件 C_2 中的事件 E_2 存在触发关系，则存在时序关系：$C_1.T_{E_1}=C_2.T_{E_2}$。

规则 3 对于构件 C_1，C_2，C_3，若构件 C_1 中的事件 E_1 与构件 C_2 中的事件 E_1 存在触发关系，构件 C_1 中的事件 E_2 与构件 C_3 中的事件 E_1 存在触发关系，且 $C_1.T_{E_1}<C_1.T_{E_2}$，则存在时序关系 $C_2.T_{E_1}<C_3.T_{E_1}$。

Step2 计算触发关系的前驱状态和后继状态，事件的前驱状态是指同一构件内部，仅能在事件发生前被激活，事件发生后不能被激活的状态。事件的直接前驱状态是指唯一与事件直接相邻接的前驱状态。事件的后继状态是指同一构件内部，仅能在事件发生后被激活，事件发生前不能被激活的状态。

(1)确定触发事件的前驱状态集。触发事件的直接前驱状态使用 Dijkstra 算法，对于不可达状态，距离为无穷大。根据构件可达图识别触发事件的所有不可达状态，所有不可达状态构成了触发事件的前驱状态集。

(2)确定被触发事件的后继状态集。从初始状态开始，根

据构件可达图执行 Dijkstra 算法，识别所有不可达状态，距离为无穷大。若这些不可达状态仅可通过触发关系被激活，则这些不可达状态构成了被触发事件的后继状态集。

Step3　根据时序关系识别不可达状态组，不可达状态组是 SEFT 模型中不同构件的两个状态的组合，由于模型的时序行为，其不可能被同时激活。在该步骤中，根据 Step1 中获得的触发或被触发事件的时序关系，得到两两构件间的不可同时到达状态组，其中时序关系分为两种：1）由直接触发关系获得的相等时序关系；2）由间接触发关系获得的先后时序关系。

（1）根据相等时序关系识别存在直接触发关系的构件间的不可达状态组。

对于存在相等时序关系的两个事件，通过直接触发关系判断其构件间的状态不可达对，那么触发事件的前驱状态和被触发事件的后继状态不可能同时到达，建立前驱状态和后继状态的笛卡尔积，获得不可达状态组。同时，若被触发事件与初始状态唯一连接，则为被触发事件的前驱状态和触发事件的后继状态建立不可达状态对。

（2）根据先后时序关系识别构件间存在间接触发关系的不可达状态组。

对于通过中间构件相关联的两构件，根据 Step1 中获得的时序关系 $T_{E_1}<T_{E_2}$，事件 E_1 的前驱状态与事件 E_2 的后继状态构成不可达状态组。

Step4　排除不可达状态组，获得整个系统的可达状态组。对于系统 $S=\{C_1,C_2,\cdots,C_k\}$，其中 $C_i=\{s_1,s_2,\cdots,s_m\}$ 表示第 i 个构件包含 m 个状态，通过笛卡尔积组合计算所有构件间的可同时到达状态。设参与笛卡尔积组合计算的构件集为 D（D 初始为空），可达状态组为 R（R 初始为空）。

（1）初始时，从构件集 S 中取出 C_i 放入集合 D 中，$D=\{C_i\}$，$S=S-D$，则 $R=\{R_1,R_2,\cdots,R_m\}=\{\{s_{i1}\},\{s_{i2}\},\cdots,\{s_{im}\}\}$，$i=1,\cdots,k$，$\{s_{ij}\}$ 表示构件 C_i 中的第 j 个状态构成的集合。

（2）从 S 中选择任意构件 C_j，$D=\{C_i,C_j\}$，对于所有的 $R_i\in R$，$i=1,2,\cdots,m$，获得 R_i 的不可达状态集 $\overline{R_i}$，对 C_j 的子状态 $\{s_{j1},s_{j2},\cdots,s_{jm}\}$ 进行组合计算：$R_i\times(C_j-\overline{R_i})$。

（3）重复第（2）步，直到所有的构件都包含在 D 中，S 为空，R 中包含所有的可达状态组。

Step5　建立连接表，获得系统可达图。源状态事件故障树的每一个事件都存在一种连接模式。该模式描述了其前驱状态到后继状态的转移。如表 1 所列，第一列是唯一被允许的两状态之间的转换。

表 1　连接表

事件/序列	状态转移	类型
E	$S_0\rightarrow S_1$	确定/随机/触发

从图的初始状态，从而通过 Dijkstra 算法实现系统可达图，无效状态是不可达的，去掉所有的不可达状态，从而得到一个最小化可达图。

Step6　根据系统的最小割集，获得系统失效可达图。由于前提条件仅允许纯状态逻辑门，因此可将 SEFT 转换成构件故障树，利用构件故障树的最小割集生成算法，若可达状态中包含的失效状态满足最小割集，则将该状态作为系统的失效状态。

4　模型的算例验证与分析

本节给出作战系统中鱼攻系统执行作战任务时任务失败的 SEFT 失效模型实例，如图 3 所示。该例中的失效结果为鱼攻系统未能成功执行指控系统发出的攻击命令，导致一次作战任务失败。作战任务失败的可能原因包括以下几个方面：1）通信故障；2）指控系统中软件部分信息解算错误，未能向鱼雷发出命令；3）鱼雷系统失效。

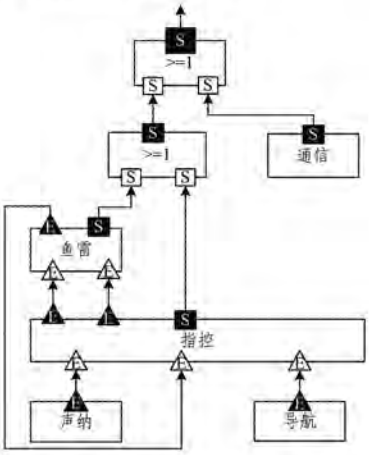

图 3　鱼攻系统 SEFT 失效模型

该系统主要由鱼雷系统、指控系统、声纳、导航系统、通信系统 5 个构件组成，分别如图 4－图 8 所示。鱼雷系统的主要功能是接收指控命令，与指控系统进行遥测信息交互，进行目标攻击；指控系统的主要功能是接收导航信息和声纳系统发出的目标信息，进行情报综合处理，组织鱼雷攻击；声纳系统的主要功能是采集目标信息；导航系统的主要功能是向指控系统发送导航信息；通信系统的主要功能是作战系统需要协同作战，任一网段发生故障均会引起系统失效，该系统监测鱼雷、指控、声纳、导航系统的通信是否正常，若通信失败，如发生断电，则系统失效。SEFT 模型通过两个 OR 门，以自底向上的方式连接 5 个构件，描述了该失效结果发生的因果关系链，即通信发生故障、鱼雷系统初始化失败或指控系统软件失效均会导致系统失效。

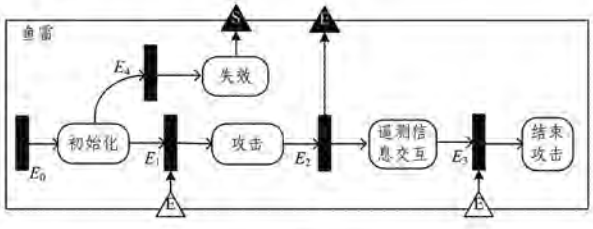

图 4　鱼雷构件

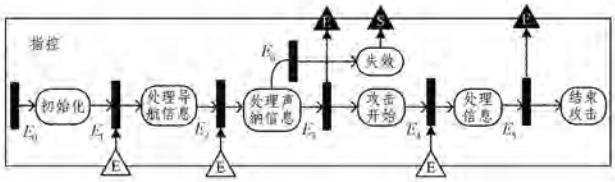

图 5 指控构件

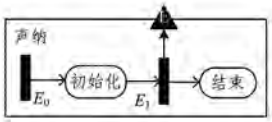

图 6 声纳构件

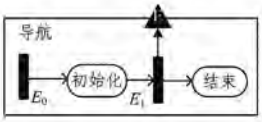

图 7 导航构件

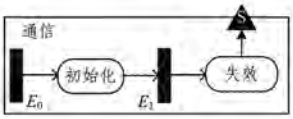

图 8 通信构件

为方便表示，将鱼雷、指控、声纳、导航、通信分别用构件 C_1，C_2，C_3，C_4，C_5 表示。声纳构件状态 C_3 从左到右依次用 S_0，S_1 表示。导航构件状态 C_4 从左到右依次用 S_0，S_1 表示。鱼雷构件状态 C_1 从左到右依次用 S_0，S_1，S_2，S_3，S_4 表示，其中 S_4 表示失效。指控构件状态 C_2 从左到右依次用 S_0，S_1，S_2，S_3，S_4，S_5，S_6 表示，其中 S_6 表示失效状态。通信构件状态 C_5 从左到右依次用 S_0，S_1 表示。

Step1 确定构件内部与构件间各触发关系的时序关系。

$(C_4.T_{E_1}=C_2.T_{E_1})<(C_3.T_{E_1}=C_2.T_{E_2})<(C_1.T_{E_1}=C_2.T_{E_3})<(C_1.T_{E_2}=C_2.T_{E_4})<(C_1.T_{E_3}=C_2.T_{E_5})$

Step2 获得构件间的不可达状态组。

首先获得具有相等时序关系的不可达状态对。

声纳与指控存在时序关系 $C_3.T_{E_1}=C_2.T_{E_1}$，则不可达状态对为：

$\{C_3.S_0,C_2.S_2\}\{C_3.S_0,C_2.S_3\}\{C_3.S_0,C_2.S_4\}$

$\{C_3.S_0,C_2.S_5\}\{C_3.S_0,C_2.S_6\}\{C_3.S_1,C_2.S_0\}$

$\{C_3.S_1,C_2.S_1\}$

同理，依次分析时序关系 $C_4.T_{E_1}=C_2.T_{E_2}$，$C_1.T_{E_1}=C_2.T_{E_3}$，$C_1.T_{E_2}=C_2.T_{E_4}$，$C_1.T_{E_3}=C_2.T_{E_5}$，获得两构件间的不可达状态对。

其次，通过具有间接触发关系的构件获得具有间接时序关系的不可达状态对。经过分析可得导航与鱼雷之间存在的间接时序关系 $C_4.T_{E_1}<C_1.T_{E_1}$，则存在不可达状态对$\{C_4.S_0,C_1.S_1\}\{C_4.S_0,C_1.S_2\}\{C_4.S_0,C_1.S_3\}$。同理，由于声纳与鱼雷之间存在间接时序关系 $C_3.T_{E_1}<C_1.T_{E_1}$，导航与声纳之间存在间接时序关系 $C_4.T_{E_1}<C_3.T_{E_1}$，分别计算出其不可达状态对。

Step3 获得系统内可达状态对。按照 $C_4\times C_2\times C_3\times C_1\times C_5$ 的组合顺序进行构件状态组合，从而获得系统内所有构件的可达状态对。

$\{C_4.S_0,C_2.S_0,C_3.S_0,C_1.S_0,C_5.S_0\}\{C_4.S_0,C_2.S_0,C_3.S_0,C_1.S_0,C_5.S_1\}$

$\{C_4.S_1,C_2.S_1,C_3.S_0,C_1.S_0,C_5.S_0\}\{C_4.S_1,C_2.S_1,C_3.S_0,C_1.S_0,C_5.S_1\}$

$\{C_4.S_1,C_2.S_2,C_3.S_1,C_1.S_0,C_5.S_0\}\{C_4.S_1,C_2.S_2,C_3.S_1,C_1.S_0,C_5.S_1\}$

$\{C_4.S_1,C_2.S_3,C_3.S_1,C_1.S_1,C_5.S_0\}\{C_4.S_1,C_2.S_3,C_3.S_1,C_1.S_1,C_5.S_1\}$

$\{C_4.S_1,C_2.S_3,C_3.S_1,C_1.S_2,C_5.S_0\}\{C_4.S_1,C_2.S_3,C_3.S_1,C_1.S_2,C_5.S_1\}$

$\{C_4.S_1,C_2.S_4,C_3.S_1,C_1.S_2,C_5.S_0\}\{C_4.S_1,C_2.S_4,C_3.S_1,C_1.S_2,C_5.S_1\}$

$\{C_4.S_1,C_2.S_5,C_3.S_1,C_1.S_3,C_5.S_0\}\{C_4.S_1,C_2.S_5,C_3.S_1,C_1.S_3,C_5.S_1\}$

$\{C_4.S_1,C_2.S_6,C_3.S_1,C_1.S_0,C_5.S_0\}\{C_4.S_1,C_2.S_6,C_3.S_1,C_1.S_0,C_5.S_1\}$

$\{C_4.S_0,C_2.S_0,C_3.S_0,C_1.S_4,C_5.S_0\}\{C_4.S_0,C_2.S_0,C_3.S_0,C_1.S_4,C_5.S_1\}$

$\{C_4.S_1,C_2.S_1,C_3.S_0,C_1.S_4,C_5.S_0\}\{C_4.S_1,C_2.S_1,C_3.S_0,C_1.S_4,C_5.S_1\}$

$\{C_4.S_1,C_2.S_1,C_3.S_1,C_1.S_4,C_5.S_0\}\{C_4.S_1,C_2.S_1,C_3.S_1,C_1.S_4,C_5.S_1\}$

$\{C_4.S_1,C_2.S_2,C_3.S_1,C_1.S_4,C_5.S_0\}\{C_4.S_1,C_2.S_2,C_3.S_1,C_1.S_4,C_5.S_1\}$

$\{C_4.S_1,C_2.S_6,C_3.S_1,C_1.S_4,C_5.S_0\}\{C_4.S_1,C_2.S_6,C_3.S_1,C_1.S_4,C_5.S_1\}$

Step4 建立连接表，如表 2 所列。

表 2 连接表

序号	事件/序列	转换	类型
(1)	$C_4.E_1\rightarrow C_2.E_1$	$C_4.S_0\rightarrow C_4.S_1$ $C_2.S_0\rightarrow C_2.S_1$	触发
(2)	$C_3.E_1\rightarrow C_2.E_2$	$C_3.S_0\rightarrow C_3.S_1$ $C_2.S_1\rightarrow C_2.S_2$	触发
(3)	$C_2.E_3\rightarrow C_1.E_1$	$C_2.S_2\rightarrow C_2.S_3$ $C_1.S_0\rightarrow C_1.S_1$	触发
(4)	$C_1.E_2\rightarrow C_2.E_4$	$C_1.S_1\rightarrow C_1.S_2$ $C_2.S_3\rightarrow C_2.S_4$	触发
(5)	$C_2.E_5\rightarrow C_1.E_3$	$C_2.S_4\rightarrow C_2.S_5$ $C_1.S_2\rightarrow C_1.S_3$	触发
(6)	$C_2.E_6$	$C_2.S_2\rightarrow C_2.S_6$	随机事件
(7)	$C_5.E_1$	$C_5.S_0\rightarrow C_5.S_1$	随机事件
(8)	$C_1.E_4$	$C_1.S_0\rightarrow C_1.S_4$	随机事件

Step5 获得系统的最小割集，根据连接表生成系统的失效状态可达图。该 SEFT 模型中仅含有状态逻辑门，采用经典的故障树定性分析方法获得系统的最小割集，即$\{C_2.S_6\}$，

$\{C_5, S_1\}$，$\{C_1, S_4\}$。将失效状态进行合并，生成系统的失效可达状态图，如图 9 所示，其中 S_6 表示包含所有失效状态。

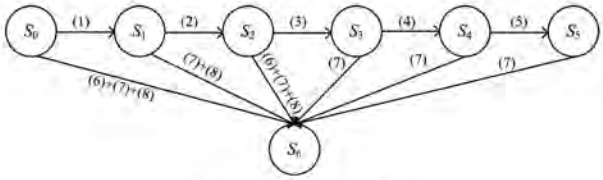

图 9　系统失效生成图

5　评价结果及分析

本文将常规的状态事件故障树直接生成图方法应用到鱼攻系统中，从稳定性、空间占用情况以及时间复杂度 3 个方面来说明基于时序关系的系统失效可达图生成方法，并分析该系统失效可达图生成方法的可行性及优缺点。图 10 和图 13 分别按照不同的顺序对构件进行合并。图 10－图 13 中的转折点为当前组合的状态空间情况，连线部分不具有意义，仅是为了做对比，以表示状态空间的变化情况。

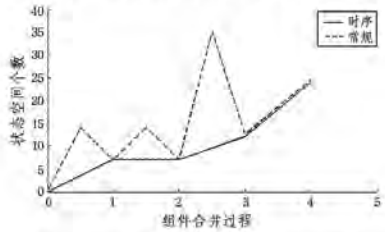

图 10　状态空间变化情况随构件组合计算的对比图

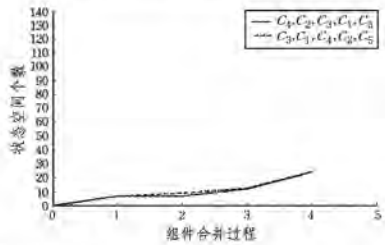

图 11　基于时序关系的状态空间随构件组合顺序的变化情况

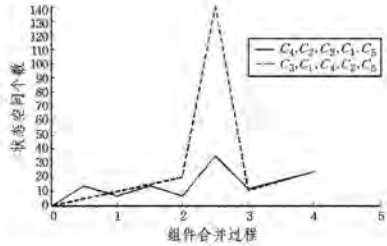

图 12　基于常规方法的状态空间随构件组合顺序的变化情况

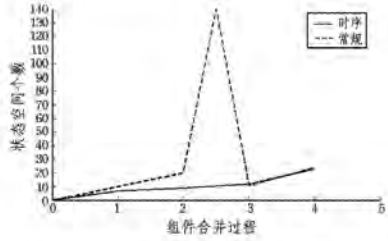

图 13　时序与常规方法的状态空间对比图

(1)图 10 为状态空间变化情况随构件组合计算的对比图，其中横坐标 1 表示导航与指控组合计算 $C_4 \times C_2$，横坐标 2 表示导航、指控、声纳组合计算 $C_4 \times C_3 \times C_2$，横坐标 3 表示导航、指控、声纳、鱼雷组合计算 $C_4 \times C_2 \times C_3 \times C_1$，横坐标 4 表示导航、指控、声纳、鱼雷、通信组合计算 $C_4 \times C_2 \times C_3 \times C_1 \times C_5$。

从图 10 中可以看出，在生成图的过程中，基于时序关系的 SEFT 状态空间增长缓慢，且增长过程中不会出现暴增现象；而常规的直接生成图在增长过程中会出现状态空间激增现象。

(2)图 11 给出应用基于时序关系的状态空间缩减方法，状态空间随不同构件组合顺序的变化情况；图 12 给出应用常规的直接生成图状态空间缩减方法，状态空间随不同构件组合顺序的变化情况，其中横坐标表示组合合并过程；图 13 给出两种方式的状态空间对比情况。图 13 中横坐标 1 表示声纳与鱼雷的组合计算 $C_3 \times C_1$，横坐标 2 表示声纳、鱼雷、导航的组合计算 $C_3 \times C_1 \times C_4$，横坐标 3 表示声纳、鱼雷、导航、指控的组合计算 $C_3 \times C_1 \times C_4 \times C_2$，横坐标 4 表示声纳、鱼雷、导航、指控、通信的组合计算 $C_3 \times C_1 \times C_4 \times C_2 \times C_5$。

从图 11－图 13 可以看出，基于时序关系的状态空间缩减方法不受组合计算先后顺序的影响，比较稳定，而常规的直接生成图方法对组合顺序敏感，不同的构件组合计算顺序会直接影响空间缩减方法的性能。

(3)现引用咖啡机案例[2]和虚构复杂系统案例[9]来说明基于时序关系的系统失效生成图方法能有效地缩短系统失效图的生成时间。如表 3 所列，不同的系统包含数量不同的组件、状态及事件。鉴于篇幅，本文不做详细描述，仅给出简单数量关系，详细的系统内部状态事件连接及组件间触发关系连接情况可见文献[2,9]。

表 3　时间对比

		鱼攻系统	单台咖啡机	三台咖啡机	虚构复杂系统
	组件	5	4	12	4
	状态	18	13	39	9
	事件	13	19	57	9
时序方法生成	时间	9ms	<5ms	10min	<5ms
常规方法生成	时间	25ms	8ms	14min	7ms

由上述对比可知，在评价方法上，常规直接生成图方法未识别构件间的间接触发关系，组合计算时具有间接触发关系的构件之间不能有效地删除不可达状态对。基于时序关系的系统失效生成图方法通过构件触发关系的时序活动，建立构件间存在触发关系的各触发事件的时序关系，该时序关系包含了存在间接触发关系的事件时序，可以直接获得系统所有的不可达状态对，直接建立所有的可达状态集。因此，状态空间缩减不受组合顺序的影响，缩减情况比较稳定，且有效地减少了系统失效生成图时间，基于时序的系统失效生成图方法对缩减空间和时间更有效。

结束语　本文建立了基于时序关系的系统失效生成图方法，通过分析构件内部触发类型事件的时序活动，建立了构件

间的触发类型事件的时序关系，获得系统内部构件间的所有不可达状态对，删除构件间不可同时到达状态组，对构件间的可同时到达状态建立笛卡尔积，通过连接表和最小割序集建立系统失效生成图。实验结果验证了方法的可行性与稳定性。所提方法有效解决了作战系统进行可靠性[13-16]分析时存在的状态空间爆炸问题，同时也为状态事件故障树生成系统可达图提供了一种新的方法。

参 考 文 献

[1] KAISER B. State Event trees: A safety and reliability analysis techniqure for software controlled systems[D]. Kaiser-slautern: University Kaiserslautern, 2007.

[2] KAISER B, GRAMLICH C. State-Event-Fault-Trees-A Safety Analysis Model for Software Controlled Systems[J]. Reliability Engineering and System Safety, 2007, 92(11): 1521-1537.

[3] GUCK D, HAN T, KATOEN J P, et al. Quantitative timed analysis of interactive markov chains [M]// NASA Formal Methods. Springer Berlin Heidelberg, 2012: 8-23.

[4] XU B F. Research on security analysis method of component based embedded software [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2014. (in Chinese)
徐丙凤. 构件化嵌入式软件安全性分析方法研究[D]. 南京：南京航空航天大学，2014.

[5] ROTH M, LIGGESMEYER P. Qualitative analysis of state/event fault trees for supporting the certification process of software-intensive systems[C]// IEEE International Symposium on Software Reliability Engineering Workshops. 2013: 353-358.

[6] XU B F, HUANG Z Q, HU J, et al. A state event fault tree quantitative analysis method [J]. Chinese Journal of Electronics, 2013, 41(8): 1480-1486. (in Chinese)
徐丙凤，黄志球，胡军，等. 一种状态事件故障树的定量分析方法[J]. 电子学报，2013，41(8)：1480-1486.

[7] LIU W B. Study on the dynamic fault tree analysis method based on modular idea [D]. Nanjing: Nanjing University of Science and Technology, 2009. (in Chinese)
刘文彬. 基于模块化思想的动态故障树分析方法研究[D]. 南京：南京理工大学，2009.

[8] ROTH M, HARTOYO A, LIGGESMEYER P. Efficient reachability graph development for qualitive analysis of state/event fault trees[C]// IEEE International Symposium on Software Reliability Engineering Workshops. 2015: 144-151.

[9] ROTH M, LIGGESMEYER P. Sequential Logic for State/Event Fault Trees: A Methodology to Support the Failure Modeling of Cyber Physical Systems[M]// Computer Safety, Reliability, and Security. 2015: 121-132.

[10] ROTH M, LIGGESMEYER P. Modeling and Analysis of Safety-Critical Cyber Physical Systems using State/Event Fault Trees[OL]. http://hal.archives-ouvertes.fr/hal-00848640.

[11] WALKER M D. Pandora: A Logic for the Qualitative Analysis of Temporal Fault Trees[D]. University of Hall, UK, 2009.

[12] TANG Z, DUGAN J B. Minimal cut set/sequence generation for dynamic fault trees[C]// Proccedings of the Annual Reliability and Maintainability Symposium (RAMS). Charlottesville, USA, 2004: 207-213.

[13] LIU D. Key technology research on reliability design and analysis of spatial information processing system [D]. Changsha: National Defense Science and Technology University, 2008. (in Chinese)
刘东. 空间信息处理系统可靠性设计与分析关键技术研究[D]. 长沙：国防科学技术大学，2008.

[14] LI Y F. New method of dynamic fault tree analysis of complex system and its application [D]. Chengdu: Electronic Science and Technology University, 2013. (in Chinese)
李彦锋. 复杂系统动态故障树分析的新方法及其应用研究[D]. 成都：电子科技大学，2013.

[15] QIN Q N. The complex system reliability modeling, analysis and comprehensive evaluation method of [D]. Beijing: Beijing Jiaotong University, 2012. (in Chinese)
覃庆努. 复杂系统可靠性建模、分析和综合评价方法研究[D]. 北京：北京交通大学，2012.

[16] GUO Y. Research on reliability evaluation method of software system based on component [D]. Harbin: Harbin Institute of Technology, 2013. (in Chinese)
郭勇. 基于构件的软件系统的可靠性评估方法研究[D]. 哈尔滨：哈尔滨工业大学，2013.

(上接第 162 页)

[20] ABID N, ZILLIO S D, BOTLAN D L. Real-time specification patterns and tools[C]// 17th International Workshop on Formal Methods for Industrial Critical Systems (FMICS 2012). Paris, 2012: 1-15.

[21] CHEN Z Y, HUANG S B, BAI Y, et al. CTL formalized specification templates in model checking[J]. Journal of Harbin Engineering University, 2013, 34(4): 483-487. (in Chinese)
陈志远，黄少滨，白玉，等. 模型检测中的 CTL 形式化描述模板[J]. 哈尔滨工程大学学报，2013，34(4)：483-487.

[22] BEHRMANN G, DAVID A, LARSEN K. A tutorial on UPPAAL[J]. Formal Methods For The Design of Real-time Systems, 2004, 4(12): 200-236.

[23] DAI S X, HONG M, GUO B, et al. Schedulability Analysis Model for Multiprocessor Real-Time Systems Using UPPAAL[J]. Journal of Software, 2015(2): 279-296. (in Chinese)
代声馨，洪玫，郭兵，等. 多处理器实时系统可调度性分析的 UPPAAL 模型 [J]. 软件学报，2015(2)：279-296.