第 34 卷 第 5 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.5 Computer Engineering March 2008
·博士论文· 文章编号：1000—3428(2008)05—0042—03 文献标识码：A 中图分类号：X928

基于操作流程图的人为差错辨识方法

孙志强，李欣欣，谢红卫，李政仪
(国防科技大学自动控制系，长沙 410073)

摘 要：提出操作流程图的概念，利用操作流程图来辨识人为差错。提供了操作流程图的 3 种基本模块，分析了如何建立操作流程图，给出利用操作流程图进行人为差错路径辨识算法，算法所得到的结果与事件树的结果是完全一致的。讨论了操作流程图针对事件树的优势及其发展趋势。

关键词：人为差错；人为差错辨识；事件树；操作流程图

Method for Human Error Identification Based on Operation Process Diagram

SUN Zhi-qiang, LI Xin-xin, XIE Hong-wei, LI Zheng-yi
(Department of Automatic Control, National University of Defense Technology, Changsha 410073)

【Abstract】A concept of Operation Process Diagram(OPD) is presented and the means of how to identify human error using OPD is discussed. The three basic modules of OPD are illustrated. And, the means of OPD construction and human error identification are presented. The error identification results are identical to the results from even tree. The OPD's advantage over event tree and further research are discussed.

【Key words】human error; human error identification; event tree; Operation Process Diagram(OPD)

1 概述

人因可靠性分析(Human Reliability Analysis, HRA)是概率风险评估(Probabilistic Risk Assessment, PRA)中必不可少的环节之一，而人为差错辨识 (Human Error Identification, HEI)则是HRA的关键步骤和难点之一[1]。只有辨识出人为差错，才能有效地进行人因可靠度评估和风险分析。

目前，大多数HRA方法都包含人为差错辨识阶段，其中以THERP(Technique for Error Rate Prediction)的HRA事件 树[2]方法最为典型，它通过将任务分解为比较细致的人为操作动作，然后确定各动作可能的差错，并采用事件树的形式来确定人为差错路径，最终获得任务对应的人为差错概率。这种方法的应用非常广泛，并被其他多种HRA方法借鉴，被适当修改直接用于辨识人为差错，如CREAM(Cognitive Reliability and Error Analysis Method)方法[3]。与传统的事件树类似，HRA事件树也具有建树过程直观、计算过程比较方便等优点；同样，HRA事件树也存在以下一些缺陷：

(1)对于简单系统来说，HRA 事件树的规模或许还可以接受；而对于大型复杂系统来说，HRA 事件树的分叉数量将很大，最终可能导致“组合爆炸”问题。HRA 事件树本身是为了简化差错辨识过程而设计的，很明显，这种庞大的规模有悖于其主旨。

(2)传统的 HRA 事件树本质上是一种二叉树，它将人的操作行为简单地分为“成功/失败”两种模式，显然，用这种方式来描述人为操作是不够的。为此，文献[4]提出了一种多态事件树的概念，它将人的操作差错分为多种不同的模式，这一方面细化了对人为操作的描述，但另一方面又使事件树的分支数量剧增，再一次增加了事件树的规模。

(3)HRA 事件树的建立需要充分把握系统的操作流程，因此，可能需要领域专家的参与，至少也需要与领域专家或操作人员进行充分交流。

文献[5]提出了一种基于状态转移的人为差错定性辨识方法(Task Analysis For Error Identification, TAFEI)，该方法通过对任务进行自上而下的目标分解，确定某个目标下的系统状态，然后分析状态之间可能存在的转移关系，那些可能存在但不被允许的状态过程即被确认为人为差错。TAFEI 能够比较全面地辨识人为差错，但其目标是针对独立的操作行为确定操作差错，并没有确定导致系统故障的人为差错路径。同样，它也需要充分分析系统操作流程，并全面分析系统状态，然后调用领域知识来分析状态之间的关系。

本文的目标在于通过融合 TAFEI 中的目标分解技术和 HRA 事件树中操作流程分解技术来建立一种人为差错辨识方法，即操作流程图方法。这种方法能够较为简单直观搜索导致任务失败的人为差错路径，与事件树和 TAFEI 相比，该方法有着自己独特的优势：

(1)操作流程图旨在寻找人为差错路径，而 TAFEI 的目标则在于寻找单个操作差错，从这个角度上说，TAFEI 可在操作流程图之前使用，找出可能的单个差错之后，再利用操作流程图方法建立相关的差错路径。

(2)在系统过于复杂时，对应的事件树规模将过于庞大，导致分析过程相对繁琐，对于多态事件树来说尤其如此。此外，事件树越庞大，分析结果也就越难以理解。而操作流程图描述了操作的全过程，这对于理解分析结果是非常有帮助的。

基金项目：国家部委基金资助项目
作者简介：孙志强(1978－)，男，博士研究生，主研方向：人因工程与安全性分析；李欣欣，高级工程师；谢红卫，教授、博士生导师；李政仪，讲师
收稿日期：2007-03-10 **E-mail：**sunzq@nudt.edu.cn

(3)在建立事件树之前，必须对任务进行较为详细的分析，获得任务的操作过程。因此，从某种意义上讲，操作流程图是任务分析的直接结果，而事件树则是由操作流程图衍生出来的，如图 1 所示。

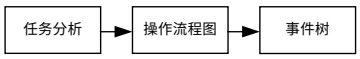


图 1　操作流程图与事件树的时序关系

2　操作流程图的基本概念

流程图是软件工程的概念，它利用图示化的方式来描述程序的执行过程。操作流程图借用流程图中的概念来描述人为操作过程。与流程图类似，操作流程图包含 3 种不同的图示化模块，分别为任务启动/终止模块、操作行为模块和条件判别模块。

2.1　任务启动/终止模块

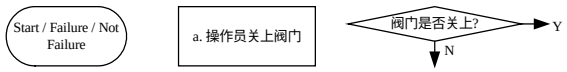
该模块用于描述系统的启动或者终止状态，共有 3 种不同类型，分别为 Start/Failure/Not Failure，其中，Start 表示任务流程启动；Failure 表示任务流程失败；Not Failure 表示该任务并没有完全失败，处于成功或者半成功的状态。人为差错路径指的就是从 Start 到 Failure 的链路。其图示化形式如图 2(a)所示。

2.2　操作行为模块

该模块用于描述任务流程中的单个操作，其中的文字就是对操作的描述。一般情况下，操作人员是按照一定次序或者根据相关条件有前有后的来执行相关任务的，因此，在建立操作流程图时，需要对操作行为模块采用字母或者数字的形式进行编号，其图示化形式如图 2(b)所示。

2.3　条件判别模块

在某些任务流程中，要根据上一步操作行为的结果来决定接下来的动作，这就需要用到图 2(c)所示的条件判断模块。条件判别模块总是与操作行为模块结合在一起的，其条件直接对应着操作行为模块的操作结果。



(a)任务启动/终止模块　(b)操作行为模块　　(c)条件判别模块

图 2　操作流程模块

3　操作流程图的分析过程

操作任务的流程可以分为 4 种类型[6]：顺序固定，顺序随条件变化，并行顺序和自由顺序。对于一个设计良好的人机交互界面来说，为了减少人为差错的概率，应该尽可能地人避免后两种类型的任务。因此，本文着眼于前两种类型的操作任务，分析如何建立流程图，并设计相应的人为差错路径搜索算法。对于一个具体任务而言，基于操作流程图来搜索其人为差错路径，基本上可以分为 3 步：目标树的建立，操作流程图的建立，人为差错路径的辨识。

3.1　目标树的建立

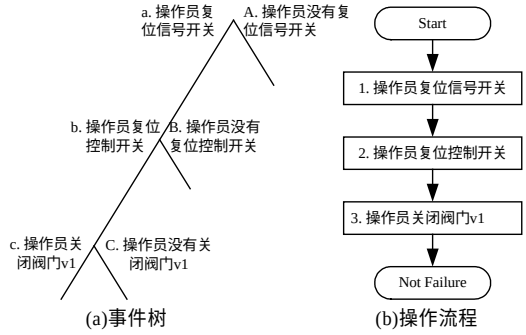
对于较为复杂的系统来说，其总体操作目标往往是比较抽象的，按照文献[6]的观点，人在选择具体的操作动作时更习惯于受到任务目标的直接驱动。如果目标比较抽象，操作人员一般会首先将其分解为一些具体可控的子目标，然后针对子目标确定相应的操作行为。因此，在分析系统可能发生的人为差错时，也可以利用层次任务分析技术[6]对任务进行分析，从而建立系统的目标树。目标树这一概念是Mohammad提出的[7]，其目的在于通过一系列功能原语来描述系统的功能架构，并已经在故障诊断、系统设计改进等领域中得到了应用。本文借用目标树这一概念，偏重于对系统中与人为操作相关联的目标进行描述与分解。目标树的建立过程总体上类似于故障树，此处不再赘述。需要指出的是，目标分解工作在哪一层次上终止，至今尚未有令人完全信服的答案，一般来说，当子目标已经比较具体时，目标分解就可以停止了。

3.2　操作流程图的建立

这一步的目的是在第 1 步的基础上，针对不同的子目标，建立相对应的操作流程图。对于顺序固定和顺序随条件变化这两种不同类型的操作任务来说，其操作流程图的建立过程是不一样的。

(1)顺序固定的任务

对于这类操作任务来说，操作人员按照预先规定好的顺序进行操作。操作过程中出现某个差错后，如果不能及时纠正，那么整个操作流程也就失败了。图 3(a)所示的事件树描述的操作任务就是典型的固定顺序的任务(小写字母表示正常的操作行为，而大写字母则用于描述操作差错)[8]。该任务所对应的操作流程图如图 3(b)所示。该任务包含了 3 个前后相连顺序固定的操作动作。图 3(b)中任何操作行为模块之间的连接断开之后，Start和Not Failure之间将不存在任何通路，也就是说，在这种情况下，操作任务将必然失败。

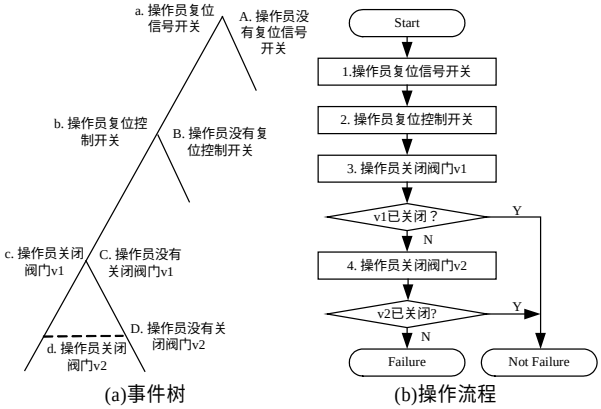


(a)事件树　　(b)操作流程

图 3　事件树与对应的操作流程(顺序固定)

(2)顺序随条件变化的任务

这类操作任务一般包括两种情况：一种是系统中包含备用部件，另一种是多人同时操作同一个任务。图 4(a)描述的操作流程属于第 1 种情况[8]，其流程图如图 4(b)所示。在这个操作流程中，如果操作人员忘记关闭阀门v1，但能够及时发现并关闭阀门v2，那么操作流程仍然能够免于失败。在这种情况下，v2 就可以看作v1 的备用部件。

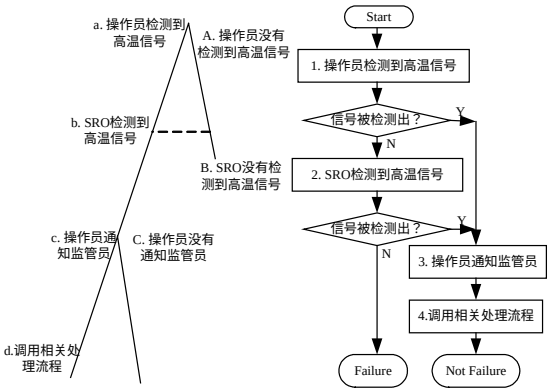


(a)事件树　　(b)操作流程

图 4　事件树与对应的操作流程(顺序随条件变化)1

图 5(a)描述的操作任务属于第 2 种情况[8]，在该流程中，当系统处于高温状态时，首先由普通操作人员检测高温信号，如果检测到则报告值班人员调用紧急处理方案；上级操作人员(Senior Reactor Operator, SRO)会检查普通操作人员是否检测出高温信号，如果发现普通操作人员没有检测到高温信号，则报告值班人员调用紧急处理方案；只有当普通操作人员和SRO都没有检测到高温信号时，整个任务才告失败。在这种情况下，属于同一个任务由多人合作(SRO与普通操作人员)。

图 5(a)所示的操作任务其流程图如图 5(b)所示。可以看出，顺序随条件变化的操作任务包含了判别模块，这说明操作流程存在不同的分支。



(a)事件树　　(b)操作流程

图 5　事件树与对应的操作流程(顺序随条件变化)2

3.3　人为差错路径的辨识

建立操作流程图的最终目的是为了辨识导致操作失败的人为差错路径。相对而言，对于顺序固定的操作任务来说，其人为差错的分析要比顺序随条件变化的任务简单一些。前已提及，在这种情况下，当 Start 模块与 Not Failure 模块之间的链路断开后，可认为任务失败。接下来讨论如何按照这一原则来搜索人为差错路径。

(1)顺序固定的任务

令K表示操作行为模块的数量，F表示人为差错路径集合，A_i表示第i个操作行为，$\overline{A_i}$表示第i个操作行为对应的所有可能的差错模式(可能仅仅包含一种模式，也可能包含多种差错模式，取决于分析的需要以及领域专家的分析能力)。该类型的任务其人为差错路径搜索算法如下：

Step 1　令 i=1。

Step 2　获取一条人为差错路径：$f_i = A_1 \to A_2 \to A_3 \to \cdots \to A_{i-1} \to \overline{A_i}$，令 $f_i \in F$。

Step 3 如果 $i<K$，转向 Step 4；否则，转向 Step 5。

Step 4 将 i 加 1，即 $i=i+1$，转向 Step 2。

Step 5　算法结束。

由该算法可以看出，K 个操作行为模块对应着 K 条人为差错路径。基于该算法，图 3 的操作流程对应的人为差错路径为 $\overline{A_1}, A_1 \to \overline{A_2}$和$A_1 \to A_2 \to \overline{A_3}$。这与事件树得到的结果{A, aB, abC}是一致的。

(2)顺序按条件变化的任务

这类任务的人为差错路径搜索算法较为复杂。实际上，一方面，人为差错路径包含Start模块和Failure模块之间的通路，这些通路将构成差错路径集合F的一个子集F_1；另一方面，当Start模块和Not Failure模块之间的通路断开后，也将形成人为差错路径，这些路径则构成另外一个子集F_2。这两个子集的并集即构成整个集合F。因此，可以将搜索算法归纳为

Step1　如果流程图中包含Failure模块，遍历Start模块与Failure模块之间的所有通路，将搜索得到的通路表示为一个集合F_1。

Step2　找出 Start 模块与 Not Failure 模块之间的所有通路，将其表示为一个集合 S。

Step3　以S为基础搜索人为差错路径。得到的人为差错路径表示一个集合F_2。

Step4　$F = F_1 \cup F_2$。

Step5　算法结束。

以上算法中，Step 3 是关键步骤之一，同时也是最复杂的。接下来对 Step 3 进行扩展分析。假定集合 S 中存在一条通路为 $A_1 \to A_2 \to \cdots \to A_m \to$ Not Failure，那么针对该通路可以按照如下算法来寻找相关的人为差错路径：

Step3.1　令 i=1。

Step3.2　断开A_i与A_{i+1}之间的连接关系，其含义为A_i没有成功完成，用 $A_1 \to A_2 \to \cdots \to \overline{A_i}$ 表示。

Step3.3　判断 S 中是否存在 $A_1 \to A_2 \to \cdots \to \overline{A_i}$ 开头的通路，如果存在，$A_1 \to A_2 \to \cdots \to \overline{A_i} \notin F_2$；否则，$A_1 \to A_2 \to \cdots \to \overline{A_i} \in F_2$。

Step3.4　如果 $i<m$-1，则将 i 加 1，转向 Step 3.2；否则，算法结束

对集合S中的所有通路都执行一遍上述算法，即可获得完整的集合F_2。

以图 4 中的操作流程图为例，首先分析得到Start模块和Failure模块之间的通路集合为F_1={ $A_1 \to A_2 \to \overline{A_3} \to \overline{A_4}$ }；然后分析得到Start模块和Not Failure模块之间的通路集合为S={$A_1 \to A_2 \to A_3$, $A_1 \to A_2 \to \overline{A_3} \to A_4$ }。在下一个层面上针对通路$A_1 \to A_2 \to A_3$进行分析，由于集合S中不存在以$\overline{A_1}, A_1 \to \overline{A_2}$开头的元素，可得 $\overline{A_1} \in F_2$，$A_1 \to \overline{A_2} \in F_2$；由于S中存在以 $A_1 \to A_2 \to \overline{A_3}$ 开头的元素，因此 $A_1 \to A_2 \to \overline{A_3} \notin F_2$。类似地，针对 $A_1 \to A_2 \to \overline{A_3} \to A_4$，可得 $\overline{A_1} \in F_2$，$A_1 \to \overline{A_2} \in F_2$ 和 $A_1 \to A_2 \to \overline{A_3} \to \overline{A_4} \in F_2$。消除冗余元素，可得 $F_2 = \left\{\overline{A_1}, A_1 \to \overline{A_2}, A_1 \to A_2 \to \overline{A_3} \to \overline{A_4}\right\}$。最后，$F = F_1 \cup F_2 = \left\{\overline{A_1}, A_1 \to \overline{A_2}, A_1 \to A_2 \to \overline{A_3} \to \overline{A_4}\right\}$。这与事件树得到的结果{A, aB, abCD}是一致的。同样，针对图 5(b)可得 $F = \left\{A_1 \to \overline{A_3}, \overline{A_1} \to \overline{A_2}, \overline{A_1} \to A_2 \to \overline{A_3}\right\}$，这与事件树得到的结果{aC, AB, AbC}是一致的。

4　结束语

本文提出了一种基于操作流程图的人为差错辨识与分析方法，操作流程图包含 3 种基本模块，能够描述顺序固定和顺序随条件变化的这两种类型的任务，利用操作流程图能直观有效地搜索得到导致操作失败的人为差错路径。操作流程图具有以下优点：

(1)建立过程比较简单直观。操作流程树可以通过分析系统操作说明书、与领域专家和操作人员交流，建树过程只需要考虑如何描述操作行为序列，无须一步到位的具体考虑操

（下转第 65 页）