

离散事件系统部分可诊断性分析

陆伟^{1,3} 张龙妹² 朱怡安³

(西安财经学院信息学院 西安 710100)¹ (西安科技大学通信与信息工程学院 西安 710054)²

(西北工业大学软件与微电子学院 西安 710072)³

摘要 针对离散事件系统部分可诊断性问题,提出一种量化评价与分析方法。该方法以树状结构的故障模型为基础,引入可诊断度与可诊断深度指标,能够从可诊断故障覆盖程度与精确程度两个方面对系统可诊断性进行评价,其优点是评价结果量化表示,能为部分可诊断系统的进一步评价、分析与对比提供参考。此外,还讨论了故障模型对系统可诊断度与可诊断深度两个评价指标的影响,并给出了故障模型构造的一般原则。实例分析与讨论结果表明,所提出的可诊断度与可诊断深度指标能够准确反映系统在特定故障模型下的部分可诊断状态。所提出的部分可诊断性评价方法能为基于离散事件模型的复杂系统设计与评价提供依据,并能够进一步为智能、自适应和自愈系统的设计提供参考。

关键词 离散事件系统,可诊断性,故障模型,系统评价

中图法分类号 TP306 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.2.038

Partial Diagnosability Analysis of Discrete-event Systems

LU Wei¹ ZHANG Long-mei² ZHU Yi-an³

(School of Information, Xi'an University of Finance and Economics, Xi'an 710100, China)¹

(Communication and Information Engineering College, Xi'an University of Science and Technology, Xi'an 710054, China)²

(School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710072, China)³

Abstract A quantitative evaluation and analysis method was proposed for partial diagnosable discrete event systems. The proposed method is based on the fault model which is depicted as a tree structure in this paper. Two indicators, diagnosable degree and diagnosable depth, were introduced in the method which can evaluate the diagnosability of systems in cover range and precision respectively. The advantage of the method is that the quantitative values of the evaluating result can be used to analyze and compare different systems which are all partial diagnosable. Furthermore, the impact of different structures of fault model on diagnosable degree and diagnosable depth was discussed and some general principles for constructing fault model were given. The results of analysis and discussion on an example show that the diagnosable degree and diagnosable depth indicators can reflect the diagnosable status of the system accurately when the system is partial diagnosable. The proposed method is useful for designing and analyzing complex systems based on discrete event model and can be helpful for designing and analyzing intelligent systems, self-adaptive systems and self-healing systems.

Keywords Discrete event systems, Diagnosability, Fault model, System evaluation

1 引言

离散事件系统(Discrete Event Systems, DESs)^[1]是工程技术、经济、军事、社会等领域一种常见的系统描述方法,其最大优点在于不需要深入系统内部结构。它是描述难以精确建模的复杂系统的理想方法。基于离散事件系统模型的系统诊断近年来得到越来越多研究人员的关注^[2]。

一般来说,可诊断性分析是离散事件系统诊断之前的一个重要步骤。M. Sampath 等人最早于1995年提出了离散事件系统的可诊断性定义,在系统所有行为均已知的条件下,讨

论了系统的可诊断性判定与诊断方法^[3]。他们在系统可诊断性的讨论中仅仅讨论了系统是否满足所有故障均为可诊断的条件,即完全可诊断的情况。

对于复杂系统而言,一般情况下很难满足完全可诊断的条件,因此,随后的研究人员又分别给出了全局可诊断性、模块化可诊断性以及分散式可诊断性等定义^[4,5],用于区分系统在不同情况下的可诊断性判定。大部分实际系统最终设计结果均为部分可诊断情况,以上研究均未能给出系统部分可诊断情况下的可诊断性量化评价与分析方法。本文针对这一问题展开研究,提出一种系统部分可诊断性的量化评价方法,

到稿日期:2014-04-14 返修日期:2014-06-25 本文受航天支撑技术基金(2013-HT-XGD),国家自然科学基金(61103003)资助。

陆伟(1975-),男,博士,讲师,主要研究方向为自主计算、软件工程, E-mail: luweinpu@nwpu.edu.cn; 张龙妹(1977-),女,博士,讲师,主要研究方向为信号与信息处理、无线传感器网络; 朱怡安(1961-),男,博士,教授,博士生导师,主要研究方向为并行计算、自主计算、软件工程。

并通过实例分析说明所提出方法的意义以及在实际系统设计与分析中的作用。

2 离散事件系统模型描述

故障模型或者故障假设一直是系统可靠性和容错研究领域的基础和前提,当前研究人员从故障原因、故障定位、故障对系统的影响、属性描述、可能的检测手段以及解决策略等不同角度对故障展开了初步研究^[7,8]。虽然不同研究对故障描述的视角不同,但大部分研究都需要利用不同方法对故障集合加以划分,从而降低故障问题的空间。本文对故障集合划分后的结果进行统一描述,从而为后续讨论提供基础。

定义 1(基本故障) 表示使计算系统状态发生变化,并导致系统进入异常状态的最小单位事件,记为 f 。计算系统所有可能的基本故障构成了系统基本故障集合,记为 Σ_f ,即 $\Sigma_f = \{f_1, f_2, \dots, f_n\}$ 。

在系统的离散事件系统模型中, $\Sigma_f \subseteq \Sigma$, 为不失一般性, 设 $\Sigma_f \subseteq \Sigma_{mo}$ (Σ_{mo} 表示不可观测事件集合)。

定义 2(宏故障) 在系统基本故障事件集合 Σ_f 中, 根据故障的属性或者类型不同(基于不同的划分方法), 可以得到 m 个两两不相交的基本故障事件集合, 记为 Σ_F ($\Sigma_F \subseteq 2^{\Sigma_f}$), 其中每个基本故障事件集合称为一个宏故障, 记为 F_j ($j=1, 2, \dots, m$)。

定义 3(基本故障的覆盖集) 设系统宏故障集合 $\Sigma_F = \{F_1, F_2, \dots, F_m\}$, 当且仅当 $\forall f_i \in \Sigma_f, \exists F_j \in \Sigma_F$ 使得 $f_i \in F_j$ 时, 称 Σ_F 为 Σ_f 的覆盖集, 记为 $\Sigma_F \in C(\Sigma_f)$ 。

定义 4(故障路径) 表示在语言 L 中出现故障 F_i 的任意一条可能的路径, 记作 $t_L(F_i)$, F_i 的所有可能路径构成了该故障的路径集合, 记作 $T_L(F_i)$ 。即

$$T_L(F_i) = \{t_L(F_i)\} = \{st \in L | s \in \Psi(F_i), t \in L/s\}$$

定义 5(故障签名) 表示故障 F_i 在语言 L 中任意一条路径的投影, 记作 $s_L(F_i)$, 故障 F_i 在 L 中所有签名的集合记作 $S_L(F_i)$ 。即

$$S_L(F_i) = \{s_L(F_i)\} = \{P(t) | t \in T_L(F_i)\}$$

定义 6(故障特征签名) 设 $s_L(F_i) \in S_L(F_i)$ 表示故障 F_i 在 L 中的任一签名, 若该签名 $s_L(F_i)$ 在 L 下的所有逆中均出现 F_i , 则签名 $s_L(F_i)$ 称为故障 F_i 在 L 下的特征签名, 记作 $s_L^c(F_i)$, 故障 F_i 在 L 下的特征签名集合标记为 $S_L^c(F_i)$ 。即

$$S_L^c(F_i) = \{s_L(F_i) | s_L(F_i) \in S_L(F_i) \wedge \forall t \in s_L^{-1}(F_i) \Rightarrow F_i \in t\}$$

其中, $F_i \in t$ 表示 $\exists f \in F_i \Rightarrow f \in t$ 。

故障特征签名表示在语言 L 中能够确定某故障 F_i 发生的特殊签名。

定义 7(可诊断故障) 设 F 为 Σ_F 中任一宏故障, 若 F 在系统模型 G 中的所有签名均为特征签名, 则称 F 在系统模型 G 下为可诊断故障, 或称 F 是可诊断的。

定义 8(故障模型) 故障模型 M_F 是一棵带权的无序平衡树, 表示为 $T(\text{root}, (T_i, w_i))$, 树中的叶子节点代表所有基本故障 f (即 $M_F \in C(\Sigma_f)$), 非叶子节点代表宏故障 F , 节点的权值代表了该故障节点在树中同层次故障节点中的重要性 (可以根据应用需求通过不同算法评估)。

3 系统部分可诊断性问题

M. Sampath 等人在没有讨论故障模型情况下, 最早给出

了系统可诊断性定义, 在以上可诊断故障定义基础上, 该定义可表述为定义 9。

定义 9(系统可诊断性) 设离散事件系统的 FSM 模型为 G , 其基本故障集合为 Σ_f , Σ_F 为 Σ_f 的一个基本划分, 若 Σ_F 中所有故障均为可诊断的, 则系统 G 具备可诊断性。

定义 9 要求系统基本故障集合一个划分中的所有故障均为可诊断, 这一要求是十分严格的, 尤其在不对系统基本故障集合作任何划分的情况下。因此, 定义 9 所给出的系统可诊断性条件是必然的、完全的可诊断条件, 我们将满足这一可诊断性条件的系统称为强可诊断性系统。

一般情况下, 实际系统很难满足强可诊断性条件, 只能满足非必然(弱可诊断)或部分故障可诊断的条件。对于弱可诊断情况下的系统诊断方法, 一些研究成果陆续出现^[9-12]; 而在部分可诊断情况下, 如何进一步评价系统可诊断性仍然是一个需要进一步解决的问题。

4 系统部分可诊断性量化分析方法

4.1 可诊断度与可诊断深度

在系统故障模型基础上, 离散事件系统的可诊断性可以进一步表示为可诊断故障对故障模型的覆盖程度与诊断精确程度。

定义 10(系统可诊断度) 设离散事件系统 FSM 模型为 G , 故障模型为 M_F , M_F 中可诊断故障与全部故障的加权比称为系统在 M_F 下的可诊断度, 标记为 $D_{M_F}^o(G)$ 。

故障模型 M_F 中的叶子节点权值由用户自定义, 用户可以根据故障发生频度以及故障对系统可能造成的影响等方面通过不同算法进行评估。在基本故障层次化划分后通过计算可得非叶子节点权值。

根据定义可得, $D_{M_F}^o(G) = D^o(F) = \sum_i \lambda_{F_i} \times D^o(F_i)$, 其中, F 为 M_F 根节点, λ_{F_i} 为 F 子树 F_i 的权值。

可诊断度的定义量化表示了系统故障模型 M_F 中可诊断节点对 M_F 的加权覆盖程度, 可诊断度的值越大说明故障模型 M_F 中加权可诊断故障对 M_F 的覆盖程度越高。

设某系统故障模型为 M_F , 如图 1 所示(图中故障节点权值经过归一化处理)。系统设计结果使得 M_F 中可诊断故障情况如图 1 中阴影节点所示。

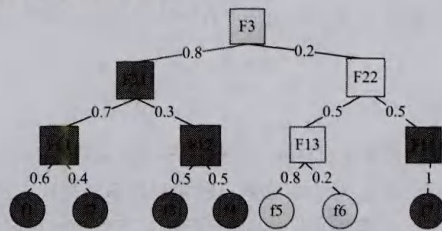


图 1 故障模型 M_F 的可诊断情况

根据图 1 中 M_F 可诊断故障情况可得系统可诊断度:

$$D_{M_F}^o(G) = D^o(F_3) = \sum \lambda_{F_i} \times D^o(F_i) = 0.9$$

可诊断度在量化表示系统可诊断程度的同时, 却不能很好地反映故障可诊断的精确程度。例如, 设与图 1 同一个系统的故障模型可诊断情况如图 2 所示, 则该系统的可诊断度同样为 0.9, 这说明系统两种设计结果对故障模型的覆盖程度相同。但根据故障模型中可诊断故障节点情况可以直观发现, 图 1 的可诊断精确程度要优于图 2。

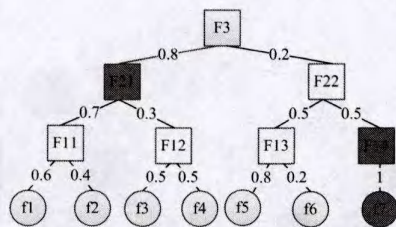


图2 故障模型 M_F 的可诊断情况

对于智能、自适应以及自愈等系统,故障可诊断的精确程度尤为重要,它可以给系统自我调整、恢复或者修复提供更加准确的信息。为进一步量化衡量可诊断的精确程度,引入可诊断深度指标。

定义 11(可诊断深度) 设离散事件系统 FSM 模型为 G , 故障模型为 M_F , 从 M_F 根节点到任一叶子节点 f_i (基本故障) 路径上的最低层次可诊断故障节点的深度称为该基本故障的可诊断深度, 记作 $h_{f_i}^d$ ($h \geq 0$), 故障模型中所有基本故障可诊断深度的平均值与故障模型高度之比称为故障模型的可诊断深度, 记作 $H_{M_F}^D$ 。

根据定义可得, $H_{M_F}^D = (\frac{1}{n} \sum h_{f_i}^d) / H$, 其中, H 为故障模型 M_F 树的高度。可诊断深度量化反映了系统在故障模型 M_F 下故障可诊断的精确程度。

根据图 1 中 M_F 可诊断故障情况可得:

$$H_{M_F}^D = \frac{1}{n} \sum h_{f_i}^d / H = 5/7$$

同样, 通过计算可得图 1 中 M_F 可诊断深度为 $1/3, 5/7 > 1/3$ 说明图 1 中故障模型的可诊断深度大于图 2 中故障模型的可诊断深度, 这量化反映了图 1 中故障模型的可诊断精确程度优于图 2。

4.2 可诊断度与可诊断深度求解算法

在实际应用中, 故障模型 M_F 建立之后, 系统设计结果决定了 M_F 中节点的可诊断状态。树状结构的故障模型 M_F 可以采用双亲孩子兄弟表示法, M_F 中节点可诊断度与可诊断深度可分别用算法 1 与算法 2 求解。

算法 1(节点可诊断度求解算法)

输入: 故障模型 M_F 任一节点

输出: 节点可诊断度

```
float computeTreeNodeD (TreeNode p) {
    float d=0;
    IF (p. pData. nodeColor == D) { //节点可诊断
        d=p. weight * 1;
    }
    ELSE{
        IF(p. pFirstChild == NULL) {
            d=0;
        }
        ELSE{
            FOR (TreeNode child=p. pFirstChild; child != NULL; child
            =child. pNext) { //遍历计算孩子节点
                d+= computeTreeNodeD(child);
            }
        }
    }
    RETURN d;
}
```

算法 2(节点可诊断深度求解算法)

输入: 故障模型 M_F 任一节点

输出: 节点可诊断深度

```
float computeTreeH(TreeNode p) {
    float h=0; float total=0; int n=0;
    TreeNode q;
    FOR (q=LookupFirstLeaf(p); q != NULL; q=q. pNext) { //遍历
    孩子节点
        total += computeLeafNodeH(q);
        n++;
    }
    h=total/n/H(p);
}

float computeLeafNodeH(TreeNode p) {
    float h=0;
    IF (p. pFirstChild != NULL) {
        RETURN-1;
    }
    ELSE {
        IF (p. pData. nodeColor == D) { //节点可诊断
            h=H(p); //H(p)为节点 p 在树中的高度
        }
        ELSE{
            p=pParent;
            WHILE (p != NULL) {
                IF (p. pData. nodeColor == D) {
                    h=H(p);
                    BREAK;
                }
                ELSE{
                    p=pParent;
                }
            }
        }
    }
    RETURN h;
}
```

在实际求解时, 首先需要将 M_F 中节点权值归一化处理, 然后将模型 M_F 的根节点分别输入算法 1 与算法 2 即可分别得到系统可诊断度与可诊断深度。

以上两个算法的时间复杂度与树的遍历时间复杂度相同, 假设故障模型中节点数为 n , 则以上两个算法的时间复杂度均为 $O(n)$ 。

5 实例分析

下面以一个完整的离散事件系统 FSM 模型实例, 来说明系统可诊断性量化评价指标的意义, 并讨论量化评价指标对实际系统设计的指导作用。

设系统 FSM 模型 G 如图 3 所示, 其中, $f_1, f_2, f_3, f_4, f_5, f_6, f_7$ 为基本故障事件, $o_1, o_2, \dots, o_{10}$ 为可观测事件, uo_1 为不可观测事件。

- [2] 赵相福, 欧阳丹彤. 离散事件系统基于模型诊断的研究进展[J]. 计算机科学与探索, 2011, 5(2): 114-127
- [3] Sampath M, et al. Diagnosability of discrete-event systems [J]. IEEE Transactions on Automatic Control, 1995, 40(9): 1555-1575
- [4] Contant O, et al. Diagnosability of Discrete Event Systems with Modular Structure[J]. Discrete Event Dynamic Systems, 2006, 16(1): 9-37
- [5] Zhou C, et al. Decentralized modular diagnosis of concurrent discrete event systems[C]// 9th International Workshop on Discrete Event Systems, Goteborg, Sweden, 2008: 388-393
- [6] Lafortune S, Chen E. A Relational Algebraic Approach to the Representation and Analysis of Discrete Event Systems[C]// Proceedings of American Control Conference, Boston, MA, USA, 1991: 2893-2898

- [7] 单锦辉, 徐克俊, 王戟. 一种软件故障诊断过程框架[J]. 计算机学报, 2011, 34(2): 371-382
- [8] 朱荣, 徐拾义. 软件测试中故障模型的建立[J]. 计算机工程与应用, 2003, 39(17): 69-71
- [9] Kwong R H, Yonge-Mallo D L. Fault Diagnosis in Discrete-Event Systems: Incomplete Models and Learning [J]. IEEE Transactions on Systems, Man and Cybernetics, 2011, 41(1): 118-130
- [10] Fijany A, Barrett A C, Vatan F. A fast model-based diagnosis engine[C]// 2012 IEEE Aerospace Conference, Big Sky, MT, 2012: 1-11
- [11] Mahulea C, et al. Fault Diagnosis of Discrete-Event Systems Using Continuous Petri Nets[J]. IEEE Transactions on Systems, Man and Cybernetics, 2012, 42(4): 970-984
- [12] 王晓宇, 欧阳丹彤, 赵剑. 不完备模型下的离散事件系统诊断方法[J]. 软件学报, 2012, 23(3): 465-475

(上接第 160 页)

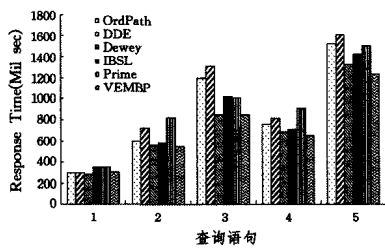


图 8 查询测试

3.3 更新性能分析

为了充分测试 VEMBP 编码方案的插入更新性能, 利用 D2 作为测试数据集, 在 site/people[1] 节点之后分别连续插入 10、20、50、80、100 个节点, 更新时间如图 9 所示。Dewey 编码的更新时间最长, 因为为了维持有序 XML 文档的节点之间的有序关系, 该编码需要对插入位置之后的节点以及它的后裔节点进行二次编码; OrdPath、DDE 虽然不需要进行二次编码, 但因为它们的查询定位时间长, 所以总的更新时间也较长; Prime 编码不需要进行二次编码, 但为了维持节点的有序关系需要修改关系索引表而进行 SC 的计算, 所以总的更新性能也较低; IBSL 编码更新性能低于 VEMBP 的主要原因是因为 IBSL 编码长度大于 VEMBP 而消耗了更多的查询时间; VEMBP 方案具有最好的更新性能, 因为该方案不需要像 Dewey 方案进行二次编码, 也不像 DDE 等方案进行复杂的结构关系判断。

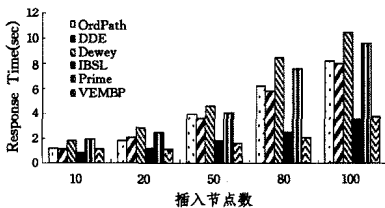


图 9 插入新节点

结束语 本文分析了已存在的支持有序 XML 文档操作的主要编码方案的特性, 并指出了它们的优点和缺点, 提出了 VEMBP 编码方法。该方法利用向量二元组表示节点的顺序关系, 采用素数表示节点之间的结构关系, 该方法在没有牺牲查询性能的前提下, 取得了很好的更新性能, 并且编码空间也得到了有效约束。下一步的工作是在该编码方案的基础上研

究如何进行 XML 关键字检索。

参考文献

- [1] Tatarinov S, Viglas D, Beyer K J, et al. Storing and Querying Ordered XML Using a Relational Database System[C]// Proc of the ACM SIGMOD 2002. Los Alamitos, CA: IEEE Computer Society, 2002: 204-215
- [2] Li Q, Moon B. Indexing and Querying XML Data for Regular Path Expressions[C]// Proc of the 27th Intl Conf Very Large Data Bases (VLDB), New York: ACM, 2001: 361-370
- [3] Wu X, Lee M, Hsu W. A Prime Number Labeling Scheme for Dynamic Ordered XML trees[C]// Proc of the 20th Int Conf Data Engineering (ICDE'04). Los Alamitos, CA: IEEE Computer Society, 2004: 66-78
- [4] O'Neil P, O'Neil E, Pal S, et al. ORDPATHs: Insert-Friendly XML Node Labels[C]// Proc of ACM SIGMOD 2004. Los Alamitos, CA: IEEE Computer Society, 2004: 903-908
- [5] Xu L, Ling T W, Wu H, et al. DDE: From Dewey to a Fully Dynamic XML Labeling Scheme[C]// Proc. of the 35th SIGMOD International Conference on Management of Data, 2009: 719-730
- [6] Ko H, Lee S. A Binary String Approach for Updates[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(4): 602-608
- [7] Jiang Yi, He Xiang-jian, Lin Fan, et al. An Encoding and Labeling Scheme Based on Continued Fraction for Dynamic XML[J]. Journal of Software, 2011, 6(10): 2043-2049
- [8] Liu Jian, Maa Z M, Li Yan. Efficient labeling scheme for dynamic XML trees[J]. LNCS, 2007, 4653: 515-161
- [9] 陈子阳, 刘佳, 张刘辉, 等. DeweyTP: 一种面向概率 XML 数据的编码方案[J]. 通信学报, 2013, 34(11): 26-32
- [10] 汪陈应, 袁晓洁, 王鑫, 等. BSC: 一种高效的动态 XML 树编码方案[J]. 计算机科学, 2008, 35(3): 76-78
- [11] 周军锋, 孟小峰. XML 关键字查询处理研究[J]. 计算机学报, 2012, 35(12): 2459-2478
- [12] Deng Hi-hong, Xiang Yong-qing, Gao Ning. LAF: a new XML encoding and indexing strategy for keyword-based XML search [J]. Concurrency and Computation: Practice and Experience, 2013, 25(11): 1604-1621
- [13] 罗道峰, 孟小峰, 蒋瑜. XML 数据扩展前序编码的更新方法[J]. 软件学报, 2005, 16(5): 810-818
- [14] 覃遵跃, 黄云, 蔡国民, 等. 支持 XML 插入更新的编码方法[J]. 计算机应用, 2012, 32(12): 3540-3543