

一个面向对象的系统故障诊断模型*

李大琪 邹逢兴 强 静

(国防科技大学自动控制系 长沙 410073)

摘 要 本文介绍了一个基于面向对象方法的系统故障诊断模型,讨论了该故障模型的结构及其搜索策略和算法。

关键词 故障诊断,故障树,面向对象,人工智能,专家系统,知识库

分类号 TP206.3

An Object-oriented Model in System Fault Diagnosis

Li Daqi Zou Fengxing Qiang Jing

(Department of Automatic Control, NUDT, Changsha, 410073)

Abstract In this paper, an object-oriented model is presented for system fault diagnosis. We discuss the structure of the model, the strategy and algorithm based on the model.

Key words fault diagnosis, fault tree, object-oriented, AI, expert system, knowledge library

目前,常用的故障诊断专家系统大致分为两类:基于规则和基于模型的诊断系统。这两类系统已获初步成功,但各自存在着无法弥补的缺陷,如针对一个动态系统的故障,这两类系统就很难达到诊断的实时性和诊断专家知识库易于更新的要求。这时,用面向对象的故障树模型代替上述两类诊断系统很有意义。基于这类故障树的推理诊断迅速,在这样的故障树中修改数据也很容易,只要修改对象类的成员数据表和利用成员函数去修改整个故障树中的有关信息即可。

这里介绍面向对象的故障树的推理诊断系统。它非常适用于动态变化的系统,系统变化时,只要非常简单的操作便可更新故障树信息。因此,它可以准确反映当前系统状态,保证了当系统发生故障时,诊断系统将会利用最新的系统信息去进行诊断工作。

1 面向对象的故障模型的树结构

对于故障树的建立,可以理解为这样一种分析技术,当系统运行的一种不期望状态

* 1996年3月4日收稿

发生时，对系统进行分析，利用故障树提供的数据、信息，找到所有可能的使期望状态发生的途径。由此分析的结果，可以描述为一棵信息树的形式，不期望事件或者称为故障事件是其根结点，有可能引起这一事件的原因是其子结点。在对树的每一枝节进行分析时，一旦发现某个结点无法继续分析（即无子结点），那么该结点称为叶结点，这些叶结点被认为是引起故障的基本事件，往往比较容易处理。

所有的非叶结点可以看作是一个逻辑门，其关系有 AND 和 OR 两种，见图 1、2。

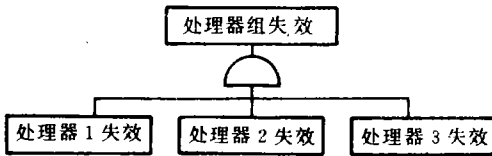


图 1 AND 结点示意图

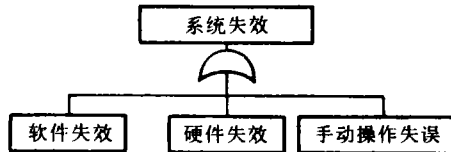


图 2 OR 结点示意图

除了事件间的逻辑关系外，其它一些信息也与树结点有关，比如一个结点事件发生引起相关故障事件发生的可能性，子结点事件与父结点事件发生的时间间隔等，含有这些信息的故障树叫扩充故障树。这种带附加信息的能力使得这种故障树非常适合于进行故障诊断，这些额外信息可以提供故障发生的可能性，从而对诊断工作起指导作用。

我们已建立了一个有效的故障树系统和一个维护程序用以动态刷新与结点事件有关的信息。在这个系统中，每个故障树结点事件由一个对象表示（见图 3），其成员数据表中包括该结点的代号、内容、类型，指向父、子结点的指针，以及所代表故障事件的细节。故障事件的细节有该事件的可能性、对顶重要性、同级重要性和故障间隔时间这几项内容。

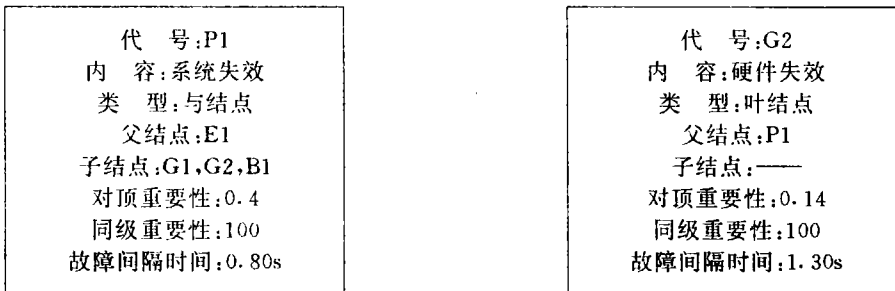


图 3 故障树结点对象例

事件对顶重要性是指该结点事件的发生对根结点事件的发生的重要程度，它反映了该结点事件及其子树各结点事件对根结点事件发生的相对责任。事件的同级重要性是指该结点事件与兄弟结点事件相比，对其父结点事件发生的重要程度。故障间隔时间对于 OR 结点来说是指该结点发生到父结点事件发生的时间，对于 AND 结点来说则是指所有兄弟结点事件发生到父结点事件发生的时间。显然，根结点没有对顶重要性、同级重要性和故障间隔时间这三项内容，而叶结点的子结点指针项为空指针。这些附加在系统初

始工作之前，要靠专家或专业技术人员的经验获取，在系统工作期间，这些信息会根据系统当前状态和系统工作历史来动态刷新。

2 故障树模型诊断算法

面向对象的故障树模型应用于系统诊断之所以比基于规则的方法处理得快，是因为根据对象进行诊断并从中获取各类信息要比在规则库中检索的方式更为有效。

诊断系统首先初始化各类信息，建立系统故障树，然后与监测系统相联。监测系统完成对系统故障的检测，对故障现象报警。故障报警与人的交互界面可以是系统控制面板上设置的指示灯，信号正常时绿灯亮，表示故障尚未发生，信号不正常时红灯亮，表示故障已发生，每一组指示灯就代表一个故障事件。此外，监测系统还与诊断系统进行实时通讯，及时传送故障信息。

诊断系统接收到监测信息后，对每一个不正常信号记录其对应事件发生时间，对每一个正常信号记录其对应事件尚未发生的最新时间，使用这一事件向前、向后的时间链，便可确定其它相关事件发生与否，这些时间信息随着系统的运行不断地发生变化。当诊断系统接收到故障信息时，便自动选择一个诊断起始点，利用建立好的链路去寻找一系列与待诊事件有关的基本故障事件。

运用传统的基于规则的诊断方法，其规则库也可以由故障树生成。每一棵故障树均可以转化为一系列的 if-then 规则，比如，一个 AND 结点事件可以用这样的规则描述：所有子结点连接起来作为输入事件，父结点则作为输出事件，见表 1。

表 1 由 AND 结点转化的规则

代号	对顶重要性	类型	同级重要性	时间间隔	子事件代号	失效子事件
(Rule1)	0.42	if (AND	(100	1.2	P1	Processor 1 is failed)
			(100	1.2	P2	Processor 2 is failed)
			(100	1.2	P3	Processor 3 is failed)
			失效事件			
		then	(System failed))			

在基于规则的诊断中，故障树必须转化为一系列的规则而存于数据库中，当一个系统需要得到一个诊断规则时，要以故障事件为关键字进行一次对规则库的查询。这一规则库的创建与查询工作都是非常繁重的。而面向对象的故障树诊断方法用对象代替规则，用高效的算法代替繁琐的查询，减轻了这一负担，每一个对象都同时扮演着三个角色：规则，事件，故障树结点。

故障树诊断系统每一个对象实际上与 if-then 规则所包含的信息是一样的。对象中事件就是规则的结果，一个对象的子对象事件就是规则的条件事件，对象中其它成员数据就是规则的类型等其它参数。在面向对象的诊断系统中，所有对象都存放于一个简表中，对应事件位置在表中固定，对象指针表同样指向对象所在位置，这种安排使得系统很快即可根据故障信息找到故障对象，并根据指针表遍历故障树。

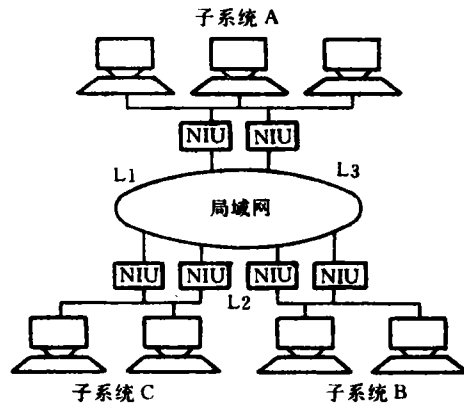
在诊断过程中，如果待考虑事件和期望规则是同一个对象，这时就不需再继续查询，系统已经有了诊断结果。而要获得其它规则，要根据对象树的向前、向后链去遍历故障

树。由于在故障树中每一个结点对象只表示一个故障事件，因此根据事件对象的父结点指针表可以找到以该事件为前因的所有事件（注意，一些结点有多个父结点，也就是说，有些事件可能引发多个事件。这时，其时间间隔是一个表结构，这点在实现诊断算法时要特别注意）。同样，一个事件的所有前因事件都可在该事件对象的子结点表中找到。由此可见，查找事件的前因、后果，只需一个指针表即可。在查询子结点表时，依据事件最大重要性原则，优先考虑对顶重要性较高的结点，显然，这样的原则可以提高查询的一次命中率。

诊断算法中还有一个重要的工作，就是要尽快、准确地更新故障对象的信息。在一个动态环境中，分布式计算机系统内部任何给定事件发生的可能性及其重要性都会随系统重构而发生变化，用面向对象的故障树诊断模型使对这种变化的适应变得非常容易，新的数据获取和计算也非常快，这一变化过程由事先定义好的算法来实现。例如若某一结点对应的故障事件发生的可能性发生变化，那么由于这一事件所引发的事件发生的可能性也会相应变化，结点的变化导致祖先结点发生变化，从而决定了各结点的新的属性值。而传统的基于规则的诊断系统就不具备这种更新方式，想要完成类似功能，需要相当多的推理。

诊断算法中一部分推理过程就是对故障事件发生最早与最新时间的处理。每一故障事件至少有一个时间参数，根据该故障事件结点及其父、子结点的时间参数的计算和比较，可能直接确定一条或较少几条诊断途径，从而避免了在非解途径上的诊断工作，这也是面向故障对象的诊断方法处理更为高效的原因之一。

如果诊断系统未发现任何基本故障事件，而的确有较高层故障事件发生，这说明所建故障树尚不完备，需要对其进行补充，找到该较高层故障事件，在其子结点组中加入一基本故障事件，除代号与内容外，该事件其余信息的获取均可通过算法实现，从而减少了人为的干预，更重要的是，所要进行的操作只是加入一新结点，以及改动几个与之相关链路即可，而在传统的基于规则的诊断系统中，要导致多个规则的改动。由此可见，我们的新方法在系统知识结构方面更易于维护。



NIU—Network Interface Units 网络通讯部件

L_n —Link Road ($n=1, 2, 3$) 通讯链路

图4 柔性制造系统中的计算机数据处理与控制系统

3 诊断实例

为更好地解释面向故障对象的诊断方法，我们选择一简单的系统来加以说明。

该系统为柔性制造系统中的计算机数据处理与控制系统，它由计算机局域网络连接三个子系统构成。（见图4）

子系统 A 由三个处理器构成，子系统 B、C 各由两个处理器构成，每个处理器组通

过 NIU (网络交互部件) 相联, 为简化故障树模型, 对系统作一些假设: 所有三个子系统均正常, 整个系统才能正常工作; 每一个系统正常工作的条件是: 1. 所有处理器正常; 2. A 从 B 获取信息正常, C 从 A、B 同时获取信息正常; 3. 每组 NIU 至少要有有一个正常工作。最后一点假设是, 局域网络可以提供一个组间的单独故障链路, 故障发生后, 组间通讯被及时禁止。

这个系统故障树模型见图 5 (结点类型采取简便表示)。

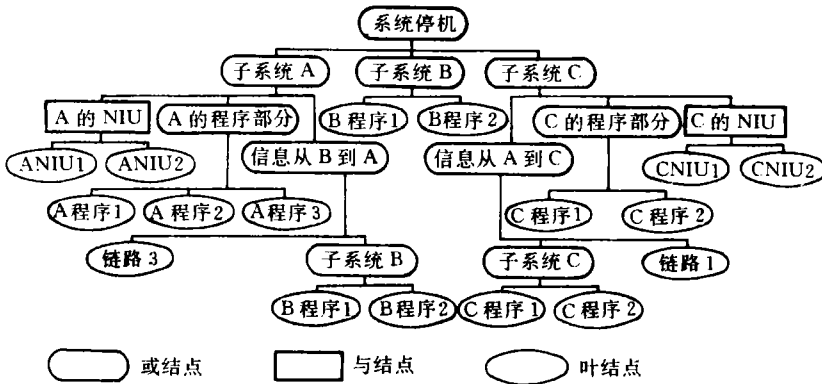


图 5 系统故障树模型

假定在时间 10 系统部分失效, 根据监测信息表明子系统 C 在时间 8 发生故障, 而且得知子系统 C 的 NIU 组在时间 8 正常工作, 子系统 A 在时间 9.5 正常工作, 这一信息以报警方式送至诊断系统, 见表 2, 图中以 | 作为信息分割符。

表 2 送至诊断系统的初始故障信息结构

失效事件	发生时间	正常信号	不正常信号	故障树模型
发现故障	计算机控制系统停机	10	((组 C 的 NIU, 8) (组 A, 9.5))	(组 C, 8) 计算机控制系统

初始正常信号组为 $\{(Cluster A, 9.5) (MIU Cluster C, 8)\}$, 初始不正常信号组为 $\{(Cluster C, 8)\}$, 仅依此信息还不足以推理发现组 C 的故障导致整个系统失效。根据父结点指针表发现其父结点对象为系统失效, 是一个或事件结点, 由此便可断定组 C 故障便是整个系统失效的原因。

接着, 故障诊断系统利用正常信号表中的信息, 断定哪些故障事件尚未发生。因为子系统 C 的 NIU 对象是一个与结点, 在时间 8 正常工作, 因此系统可以得知在时间 $8-t$ (t 为组 C 的 NIU 故障间隔时间) 时, 组 C 的 NIU 均正常。同理可以推知, 子系统 A 的 NIU 均正常, 组 A 程序运行正常, 信息从 B 到 A 正常等等, 直至确认树中所有尚未发生的故障事件。这些事件加入正常信号表中, 用以指导下一步诊断工作。在传统的基于规则的诊断方法中, 由于没有类似的信号表, 因此对规则的检索相比之下有很大的盲目性, 不可避免地会发生重复检索的现象。而利用面向对象的诊断方法在很大程度上解决了这一问题, 例如: 在确定组 B 正常后, 在研究组 C 故障时, 对从组 B 接收信息这一子事件

便可不用考虑。就是说,凡是当前已列入正常信号表的事件,它的子树便可以不用考虑。当然,这一正常信号表也不是固定不变的,如果一旦发现表中的事件新近发生了故障,就要及时地更新该信号表。

现在,诊断系统已经发现计算机控制系统失效的一个高级别原因,并已确认哪些故障事件尚未发生,接下去要寻找那些造成系统失效的基本事件。这时诊断从组 C 开始沿向后链的方向进行,将组 C 的不在正常信号表中的子结点事件作为新的高级故障事件,由此递推分析,直至找到造成系统失效的最基本事件(组)。在递推分析过程中,遇到多个可能的故障原因时,优先考虑同级重要性较高的事件;一旦一个枝节分析结束而又无解时,要利用树的回溯技术,这也值得注意。另外,我们实际上也隐含了一个假设条件,那就是假设在同一时刻,系统中只有一个基本故障事件发生,在实际运行的一般系统中,同时发生两个故障的概率一般较小,所以可满足条件。当然也不能排除一些特殊系统会同时发生多个故障,那就需将所有可能的基本故障均找到。

4 结束语

本文提出了一种基于面向对象的系统故障树诊断模型,着重介绍了该故障树的建立和结构,以及基于该树的搜索策略和算法。文中各部分随时用该方法与传统方法进行比较,表明使用该方法使故障诊断系统能力获得提高,省去了用传统方法在规则库中搜索的大量时间,而且这样的诊断系统在待诊系统结构发生变化时,很容易对故障树模型进行更新,保证了诊断系统的准确性。

整个诊断方法突出一个智能化,考虑如何去减少对分散的、人为的信息以及对系统和环境数据的过分的依赖性。今后的工作重点将放在如何将软件集成于系统模型之中,如何利用混合的建模方法使系统模型更为准确,更具通用性,以及如何进一步提高诊断算法中自动处理数据的能力等等。

参 考 文 献

- 1 Jaume D, Verge M. A model-based diagnosis in machine tools: application to the milling cutting process, EASAM&CNAM, 1990
- 2 William B D, Michael J R. Diagnosis expert system for PLC controlled manufacturing equipment, Int. J. Computer Integrated Manufacturing, 1994, 7: 116~122
- 3 Brynjolfsson S, Arnstrom A. Error detection and recovery in flexible assembly system, Int. J. Manuf TechNol, 1990, 5: 112~125
- 4 Hardy N W, Barnes D P, Lee M H. Automatic diagnosis of task faults in flexible manufacturing system, Robotica, 1989, 7: 25~35
- 5 Cheng S C, Comella J. Phaih-lan law, expert toolkit: an expert system shell for building diagnosis system. Application of Artificial Intelligence, 1989, 7
- 6 FMS 可靠性研究. 国防科技参考, 1992, (2)

(责任编辑 张 静)