

# 基于 CLIPS 的发射车诊断系统的设计与实现

徐姣姣

(北京航空航天大学自动化科学与电气工程学院 北京 100191)

**摘要:** 以发射车为例,采用基于故障树和基于案例的故障诊断方法,利用 CLIPS 专家系统开发工具设计专家系统的知识库和推理机,借助 C# 编程语言开发人机接口界面以及管理模块等,实现一种高效的故障诊断专家系统。将故障树的结构信息和上下层节点之间的逻辑关系,以及案例知识均以 CLIPS 的事实语法结构进行保存。系统借助 CLIPS 开发推理机,实现基于故障树的正反向混合推理,以及基于含有字符串型和枚举型属性的案例的推理,从而提高了故障诊断和推理的准确性与可靠性。

**关键词:** 故障诊断; CLIPS; 故障树; CBR

**中图分类号:** TP277 **文献标识码:** A **国家标准学科分类代码:** 520.2070

## Design and realization of diagnosis system based on CLIPS for launch vehicle

Xu Jiaojiao

(School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China)

**Abstract:** An expert system of fault diagnosis based on fault tree and case is introduced. The system uses CLIPS to achieve the knowledge base and inference engine, and applies C# to realize the man-machine interface and management modules. The structure information of fault tree, the logical relationships among the nodes case knowledge are preserved by fact grammatical structure of CLIPS. The system uses CLIPS as the inference engine to actualize the forward and backward reasoning based on fault tree and the reasoning based on case comprising string-type attributes and enum-type attributes, in order to improve the accuracy and reliability.

**Keywords:** fault diagnosis; CLIPS; fault tree; CBR

### 1 引言

发射车由于其复杂的机械结构以及特殊的工作环境,极易发生故障,使得对其既能安全运行又便于维修的要求也日益提高<sup>[1]</sup>。故障诊断技术是维持系统可靠运行和提高作业质量的重要技术保障。故障诊断系统能够快速地进行故障检测和故障定位,发现系统的薄弱环节,制定更加合理有效的维修方案,从而缩短了维修时间,保证了发射车系统稳定、可靠、高效地工作,为实现导弹发射车的健康管理提供重要的支撑技术和基础<sup>[2]</sup>。为调高诊断的可靠性和准确性以及满足特定条件的故障诊断,国内外专家进行了大量研究,不断改进和完善故障诊断方法,设计不同的故障诊断专家系统<sup>[3]</sup>,但是这些系统均有各自的局限性和弊端。本文对发射车系统采用基于故障树和基于案

例推理的故障诊断方法,并采用 C# 语言和 CLIPS 技术相结合的方式设计和实现一种高效的故障诊断专家系统<sup>[4]</sup>。

### 2 系统总体设计

#### 2.1 软件模块结构

发射车的故障诊断系统的总体软件结构如图 1 所示,其中知识库和推理机为故障诊断系统的核心<sup>[5-6]</sup>。知识库存储来自领域专家的经验知识,包括故障树知识库以及案例知识库,其中故障树知识库包括故障树结构以及上下层故障节点之间的逻辑关系;知识库管理模块通过人机接口界面,对知识库不断更新和扩充;推理机根据知识库和综合数据库中当前的参数信息,先利用正反向混合推理完成基于故障树的推理,若未获得匹配信息,则进行基于案例的推理,若案例推理的结果仍不尽理想,则通过人机交互诊断给

收稿日期:2014-11

出诊断结果和修复措施,案例推理通过自学习将该新案例保存至知识库,从而不断完善案例知识库;综合数据库用来存储发射车的运行状态参数信息以及推理机推理过程中的中间信息等;推理结果模块显示来自推理机的推理结果,并给出相应的解决方法和维修措施,指导用户快速准确地排除故障;人机接口界面:一方面实现领域专家对经验知识的录入、维护、修改和更新等相关操作;另一方面将诊断结果反馈给操作人员或领域专家,实现人机交互的功能。

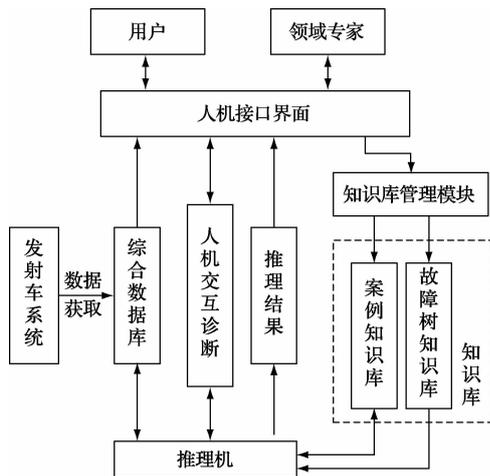


图1 故障诊断系统总体结构

## 2.2 软件总体实现方法

利用 C# 实现人机界面的开发、数据的计算处理以及与数据库和外设的数据交换,通过 CLIPS 完成专家系统中的知识库和推理机部分,采用应用于托管形式下的 ClipsNet.dll 组件将 CLIPS 项目整合到 .NET 框架下,从而实现功能强大、界面友好的专家系统。为防止由于功能的耦合性以及数据信息的交叉性,造成系统在兼容性和扩展性上的不足、运行效率的降低,以及难以有效的实现系统功能模块的重配置和可移植,在进行系统设计和实现时,不仅要保证系统实现基于应用层的分割,还需在应用接口层实现真正的功能独立。因此本文采用基于底层动态链接库的功能分割模块化设计,能够有效的解决由于功能耦合而导致的设计与实现难度的增加,以及兼容性、适应性不足等方面的问题。该系统细分为以下几个功能模块:绘制故障树模块、已有案例管理模块、数据采集模块、实时监控模块、基于故障树的推理模块以及基于案例的推理模块。将上述功能模块分别分装成动态链接库,在实现需要的功能时,采用显性调用的方式,实现所需功能的“动态”加载和调用,从而节约了系统资源,提高了运行效率。

## 3 具体实现步骤

### 3.1 知识库的设计

#### 3.1.1 故障树知识库

故障树分析法是分析系统可靠性和安全性的一种重

要方法<sup>[7-8]</sup>。借助计算机进行可视化故障树建模,可以解决人工建树效率低、易出错等问题。文献[9]给出了基于 GDI+ 和 pictureBox 控件的可视化建模方法,该方法虽易于设计和实现,但人界界面不够友好,绘图操作较为复杂。采用 Visio Drawing Control ActiveX 组件,通过借助 Visio 强大的绘图能力和良好的用户操作界面,完成故障树的绘制。通过广度优先搜索和深度优先搜索相结合的方式完成绘制的故障树进行搜索,将故障树的结构信息以及上下层节点之间的逻辑关系以 CLIPS 的事实语法结构进行保存,事实模板格式如下:

```
(deftemplate node(slot name) (slot type) (slot upnode)(slot gateType)(slot node1) (slot node2) ... (slot method))
```

其中,name 槽为当前节点的名称;type 为该节点的类型,包括顶事件/中间事件/底事件;槽 upnode 为该节点关联的上层节点;gateType 为该节点与下层事件关联的逻辑门的类型;node1, node2, ... 为该节点关联的下层事件;槽(slot method)为底事件的对应解决方法。文献[10]中仅给出当前节点关联的下层节点,这对反向推理带来一定的不便。节点定义模板,明确指出当前节点的上下层结构与逻辑关系,易于实现基于故障树的正反向推理。

#### 3.1.2 案例知识库

一个案例通常有两部分组成:故障的描述部分和故障的解决方法部分,其中故障的描述部分主要用于新案例的匹配,解决方法部分主要用于指导用户对故障进行排查和维修。根据发射车本身特点,将案例的模板定义如下:

```
(deftemplate case(slot name)(slot nameWeight (type FLOAT)) (slot system) (slot systemWeight (type FLOAT)) (slot time) (slot timeWeight (type FLOAT)) (slot phenomena) (slot phenomenaWeight (type FLOAT)) (slot mode) (slot modeWeight (type FLOAT)) (slot method))
```

其中,槽 name 为案例的名称,这是案例的标识符;槽 system 为故障发生的系统,如车控系统、液压系统等;槽 time 为故障发生的时间,如发射车展车调平时、起竖时等;槽 phenomena 为该故障的现象描述;槽 nameWeight、systemWeight、timeWeight 和 phenomenaWeight 为案例名称、故障发生系统、故障发生时间和故障发生现象的权重值,且为方便相似度的计算均定义为浮点型;槽 method 为该案例的故障解决办法,为匹配的新案例提供故障的解决和维护指导。

### 3.2 推理机的设计

#### 3.2.1 基于 CLIPS 的故障树推理机设计

文献[11]给出了基于故障树的正反向混合推理方法,即先进行正向推理得到结论,再以这个结论作为证据进行反向推理,直到推理成功或失败为止。该方法虽然结合了正反向推理的优点,但存在如下问题:一方面并非故障树的中间事件均存在故障征兆,例如该事件可能仅为某些原

因的概括总结或者该事件并无相关传感器采集其状态信息等;另一方面由于故障树本身信息的不完整性和不准确性,或者发射车某些传感器存在故障等原因,造成上层事件故障征兆发生,但其关联的下层事件的故障征兆未发生,即上层事件的误报警,或者下层故障征兆发生,但却未触发其关联的上层事件的故障征兆发生,即上层事件未报警等情况。

针对以上问题,本文先根据底事件的故障征兆状态进行反向推理,确定所有中间事件或顶事件的故障征兆状态是否正确并记录发生的错误,对于不存在故障征兆的中间事件,其故障征兆状态由其关联的下层事件的故障征兆状态决定。反向推理完成后,若顶事件的故障征兆状态正确,再利用深度优先搜索进行正向推理,仅搜索故障征兆发生的下层事件,直到底事件为止,从而获取故障原因,指导用户进行维修和处理。领域专家和用户可以根据反向推理的错误情况:一方面进行相关的传感器的检查;另一方面对故障树进行修改和完善。

如下规则用于反向推理,检查上层事件的故障征兆状态是真的发生:

```
(defrule detect-event-happen
  ?e<-(event(name ?name)(value ?value&-up) (faultType ?faultType&-nil))
```

;当前节点的逻辑门为或门,其关联下层节点只需有一个故障征兆发生即可

```
(or(and(node(name ?name)(type ?type&-~bottom)
(node1 ?node1)(node2 ?node2) (gateType ?gate&-or))
```

;假设有2个关联下层事件

```
(or(event(name ?node1)(upnode ?name)(value ?value1&-up) (faultType ?faultType1&-~nil))
```

```
(event(name ?node2) (upnode ?name) (value ?value2&-up) (faultType ?faultType2&-~nil))))
```

;当前节点的逻辑门为与门,需其关联下层节点的故障征兆均发生

```
(and(node(name ?name)(type ?type&-~bottom)(node1 ?node1)(node2 ?node2) (gateType ?gate&-and))
```

```
(event(name ?node1) (upnode ?name) (value ?value1&-up) (faultType ?faultType1&-~nil))
```

```
(event(name ?node2) (upnode ?name) (value ?value2&-up) (faultType ?faultType2&-~nil))))
```

=> (modify ?e(faultType 0));当前节点的故障征兆确实发生

其中,为了方便程序编写,设置如下规则: faultType = 0,故障征兆状态正确; faultType = 1;误报警, faultType = 2,未报警。

如下规则用于正向推理,寻找引发当前事件故障征兆发生的原因:

;引发顶事件故障征兆发生的原因

```
(defrule rule-top-infer
```

```
(node (type ?type&-top)(name ?name)
  (node1 ?node1)(node2 ?node2))
  (event(name ?name)(value ?value&-up))
```

=>

```
(assert(assertEvent(name ?node2))) ;添加原因名称
(assert(assertEvent(name ?node1)))
```

;引发中间事件故障征兆发生的原因

```
(defrule rule-middle-infer
```

```
(assertEvent(name ?name))
```

```
(node(type ?type&-middle)(name ?name)(node1 ?node1)(node2 ?node2))
```

```
(event(name ?name)(value ?value&-up))
```

=>

```
(assert(assertEvent(name ?node2)))
```

```
(assert(assertEvent(name ?node1)))
```

;标记引发顶事件故障征兆发生的底事件

```
(defrule rule-bottom-infer
```

```
(assertEvent(name ?namep))
```

```
(node(type ?type&-bottom)(name ?name))
```

```
(event(name ?name)(value ?value&-up))
```

```
?e<-(bottomEvent(name ?name)(selected ?selected&-nil))
```

=> (modify ?e (selected true));当前底事件标记为原因

### 3.2.2 基于CLIPS的案例推理机设计

根据案例相似性的量度方法,利用案例库的索引机制进行案例的检索,从案例库中找到与当前案例最相似的旧范例,为当前案例提供解决办法<sup>[12]</sup>。其中案例相似的计算公式定义如下:

$$Sim(X, Y) = \sqrt{\sum_{i=1}^n w_i \times (1 - D(x_i, y_i))^2} \quad (1)$$

式中:  $X = [x_1, x_2, \dots, x_n]$  和  $Y = [y_1, y_2, \dots, y_n]$  分别表示新案例和旧案例,  $x_i$  和  $y_i$  分别表示新案例和旧案例的第  $i$  个属性,  $w_i$  表示第  $i$  个属性的权重值,  $D(x_i, y_i)$  表示新旧案例第  $i$  个属性的距离。本文案例属性的类型包括字符串类型,如故障名称和故障现象,和枚举型,如发生故障的系统和发生故障的时间。因此需根据不同类型的特点,设计不同的计算公式。对于字符串类型,属性距离计算公式为:

$$D(x_i, y_i) = 1 - \frac{2N}{N_x + N_y} \quad (2)$$

式中:  $N$  表示字符串  $x_i$  和字符串  $y_i$  中相同字符的个数,  $N_x$  和  $N_y$  分别表示字符串  $x_i$  和字符串  $y_i$  中所含字符的个数。

对于枚举型,属性距离计算公式为:

$$D(x_i, y_i) = \begin{cases} 1 \rightarrow x_i \neq y_i, \text{即 } x_i \text{ 和 } y_i \text{ 中内容不同} \\ 0 \rightarrow x_i = y_i, \text{即 } x_i \text{ 和 } y_i \text{ 中内容相同} \end{cases} \quad (3)$$

为较少 LHS 和 RHS 的重复表达式,相似度求解方法均采用 CLIPS 自定义函数结构实现。如下函数用于字符串类型的属性距离的计算:

```
(deffunction string-match ( ?aString ?bString)
  (bind ?matchNum 0);设置初始值
  (loop-for-count ( ?num1 (str-length ?aString)) do
    (loop-for-count ( ?num2 (str-length ?bString)) do
      (if(eq(str-compare (sub-string ?num1 ?num1 ?aString)
        (sub-string ?num2 ?num2 ?bString)) 0)
        then(bind ?matchNum (+ ?matchNum 1));匹配,
        则加 1
        (break))))
  (bind ?matchValue (- 1(/ (* ?matchNum 2) (+
    (str-length ?aString)))));完成属性距离值的计算
  (return ?matchValue));返回属性距离值
```

如下函数用于枚举类型的属性距离的计算:

```
(deffunction enum-match ( ?aEnum ?bEnum)
  (if(eq ?aEnum ?bEnum)
    then(return 0);相等,则返回 0
```

else(return 1));不等,则返回 1

#### 4 发射车故障诊断系统验证

发射车故障诊断系统的设计是以某型号导弹发射车为背景,根据领域专家知识和经验数据得到规则如下:

1) 发生主泵大泵回路压力小于 4 MPa 的故障,可能原因是供电电源异常、电控回路故障或液压回路故障;

2) 液压回路故障可能由阀件卡滞或损坏、阀件或油路漏油、液压高度异常或油源质量异常造成;

3) 若电缆/0102-40/2W4 输出端故障则应使用电烙铁、斜口钳、焊锡等,在液压系统主阀箱内测试相应短路点或短路点,并进行维修;

.....

这些规则均是以自然语言进行描述的,需要将其转换为专家系统知识库中的规则形式。规则存储完成后,需对其进行使用验证。当系统接收到主泵大泵回路压力小于 4 MPa 的故障报警时,先搜索故障树知识库中有无该故障树,当检测到后,则进行基于故障树的正反向混合推理,推理界面和推理结果如图 2 所示。

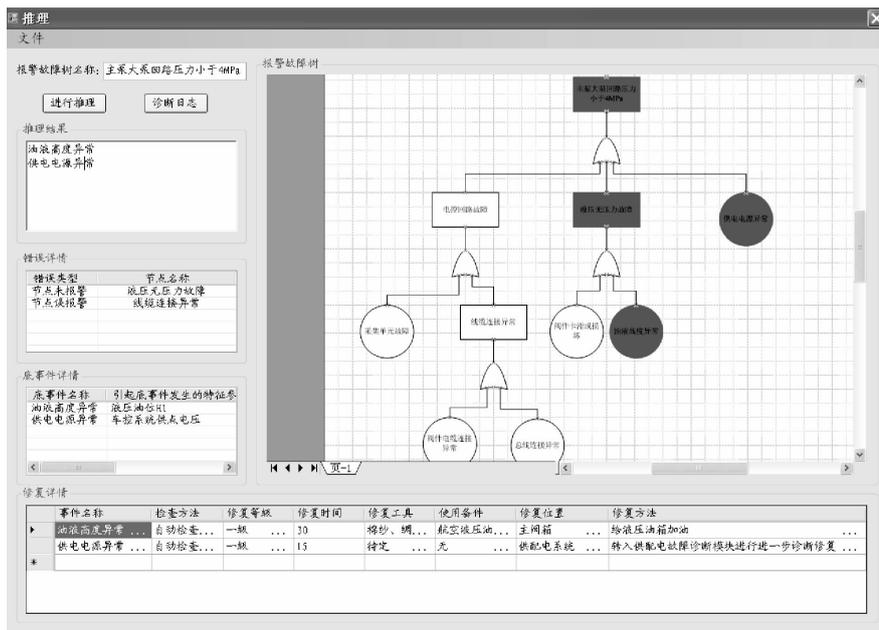


图 2 基于故障树的推理界面

图中推理结果部分给出引发顶事件发生的底事件,错误详情中给出经反向推理所得的误报警和未报警情况,供领域专家和用户进行传感器的检查或故障树的修改完善。报警故障树中标红的节点为经过正向推理推得所需底事件的路径。修复详情中给出详细的解决方法和修复措施供用户查看。

若收到的故障报警电缆(0102-90/2W9)YA206 输出端故障时,系统中无该故障树,则需进行基于案例的推理,推理界面和推理结果如图 3 所示。界面中给出了当前案例与已有案例的基本信息和匹配情况,用户可根据修复建议进行维修,维修结束后,用户可根据实际情况,选择是否保存该新案例至案例库。



图3 基于案例的推理界面

统计结果表明,诊断结果与实际故障原因相吻合,给出的修复方法能有效地指导用户解决所发生故障。基于CLIPS的发射车故障诊断系统能够有效地完成故障诊断工作,诊断的故障覆盖类型广,平均诊断时间短,一定程度上解决了车用设备安全性和可靠性较低的问题。

## 5 结论

以CLIPS专家系统开发工具和.NET平台相结合进行开发,实现人机界面友好,基于故障树和基于案例的复杂推理的发射车的故障诊断系统。利用CLIPS实现故障诊断系统知识完备的知识库,以及可靠、高效的推理机。此外借助动态链接库技术,提高系统的可移植性和可扩展性。对基于发射车的故障诊断系统的设计和实现进行了一定的理论研究和科研实践,并且为类似系统的开发提供了一定的实践积累和经验借鉴。

### 参考文献

[1] 王潇一,毛德强,王晓峰.某型导弹自驾仪的故障诊断研究[J].电子测量与仪器学报,2015,29(1):146-153.  
 [2] 彭宇,刘大同.数据驱动故障预测和健康管理综述[J].仪器仪表学报,2014,35(3):481-495.  
 [3] 陈安华,蒋玲莉,刘义伦,等.基于知识网络的故障诊断专家系统模型[J].仪器仪表学报,2009,30(11):2450-2454.  
 [4] 王华,李鹏波.基于CLIPS和.NET的惯性导航系统的故障诊断专家系统[J].中国惯性技术学报,

2008(6):78-80.

[5] 朱传敏,周润青,陈明,等.故障树与案例推理在数控机床故障诊断专家系统中的应用研究[J].制造业自动化,2011,33(10):21-24.  
 [6] 蔡猛,张大发,张宇声,等.基于故障树分析的核动力装置实时智能故障诊断专家系统设计[J].原子能科学技术,2010(44):373-377.  
 [7] 潘红兵,蔡云龙.基于故障树及LabVIEW的雷达设备故障诊断[J].电子测量技术,2013,36(9):115-118.  
 [8] 陈雷.基于故障树的供水监控诊断系统的仿真[J].国外电子测量技术,2014,33(12):35-39.  
 [9] 丁晓磊,王秋生,袁海斌,等.可视化故障树建模分析的研究与设计[C].第25届中国控制与决策会议论文集,2013.  
 [10] 康敏辉.基于CLIPS的煤层气集输过程故障诊断专家系统设计[D].大连:大连理工大学,2011.  
 [11] 邢智辉.基于专家系统的电梯故障诊断技术研究[D].哈尔滨:哈尔滨工业大学,2010.  
 [12] 徐遐龄,林涛,高玉喜,等.基于CBR和OAPID的互联网区间模式振荡预警[J].电力自动化设备,2013,33(8):88-93.

### 作者简介

徐姣姣,1993年出生,工学硕士,硕士研究生。主要研究方向为检测技术、故障诊断与健康管理等。  
 E-mail: xujiaojiao\_2008@163.com