

基于 CLIPS 的矿车故障诊断专家系统设计与实现

廖清清 谢振钊 黄云逸

福建宏大时代新能源科技有限公司, 福建 厦门 361000

[摘要]随着矿用车辆智能化程度的提升,其故障模式日益复杂,传统的人工诊断方法在效率和准确性上难以满足现代矿山生产的需求。针对矿车故障诊断难题,文中设计了一套基于 CLIPS (C Language Integrated Production System) 专家系统的智能诊断方案。通过故障树分析构建知识模型,采用“框架+产生式规则”混合表示方法建立知识库,并实现了反向与混合推理策略。系统以 C#.NET 开发界面,通过动态链接库集成 CLIPS,结合数据库进行管理。实际应用表明,该系统能准确推断故障原因并提供处理建议,为矿车智能化运维提供了实用可行的解决方案。

[关键词]矿车; 故障诊断; 专家系统; CLIPS; 知识库; 推理机

DOI: 10.33142/ect.v3i10.18186

中图分类号: TP393

文献标识码: A

Design and Implementation of an Expert System for Mining Car Fault Diagnosis Based on CLIPS

LIAO Qingqing, XIE Zhenzhao, HUANG Yunyi

Fujian Hongda Era New Energy Technology Co., Ltd., Xiamen, Fujian, 361000, China

Abstract: With the improvement of the intelligence level of mining vehicles, their fault modes are becoming increasingly complex, and traditional manual diagnostic methods are difficult to meet the needs of modern mining production in terms of efficiency and accuracy. A set of intelligent diagnosis scheme based on CLIPS (C Language Integrated Production System) expert system is designed in this paper to address the problem of fault diagnosis of mining cars. A knowledge model was constructed through fault tree analysis, and a knowledge base was established using a hybrid representation method of "framework+production rules". Reverse and hybrid reasoning strategies were implemented. The system is developed with a C #. NET interface, integrated with CLIPS through dynamic link libraries, and managed in conjunction with a database. Practical application has shown that the system can accurately infer the cause of faults and provide processing suggestions, providing a practical and feasible solution for intelligent operation and maintenance of mining cars.

Keywords: mining car; fault diagnosis; expert system; CLIPS; knowledge base; inference engine

引言

矿车是矿山生产的核心装备,其技术集成度提高使得故障更隐蔽且关联复杂。传统依赖经验的诊断方式存在效率低、知识传承难等局限。为此,本文设计并实现一个基于 CLIPS 的矿车故障诊断专家系统。重点研究内容包括:构建故障知识模型,设计系统架构,在 CLIPS 中实现矿车故障推理,解决 CLIPS 与 .NET 平台的集成问题,旨在为矿车智能诊断提供可借鉴的技术方案与原型。

1 相关技术与工具概述

1.1 专家系统与故障诊断

专家系统是应用大量人类专家的知识和推理方法求解复杂的实际问题的一种人工智能计算机程序^[1]。专家系统一般由知识库、推理机、人机界面等核心模块组成。

将专家系统技术应用于矿车故障诊断,能够有效整合和固化专家经验,实现诊断过程的标准化与自动化,显著提升诊断的准确性与响应速度。典型的故障诊断专家系统工作流程为:通过人机界面或数据采集模块获取设备状态信息或故障现象(征兆);推理机将这些征兆与知识库中的知识进行匹配和推理;最终输出故障定位、原因分析及维修建议。目前,故障诊断专家系统在发动机、船舶机舱设备以及惯导系统故障分析中均有研究及应用^[1,2,5]。

1.2 CLIPS 专家系统开发工具

CLIPS (C Language Integrated Production System) 由美国国家航空航天局 (NASA) 开发,因其完全开源、可移植性强、推理效率高(采用 Rete 匹配算法)以及支持

规则、面向对象、面向过程等多种编程范式而得到广泛应用^[3]。CLIPS 由 C 语言编写,可以编译为动态链接库(DLL),方便被 C、C++、C#、Delphi 等多种主流编程语言调用,从而能够隐藏复杂的推理过程,与友好的图形界面相结合^[4]。然而,CLIPS 规则本质上是正向链的,即当规则左部(LHS)的条件被事实满足时,触发右部(RHS)的动作。为了实现故障诊断中常用的反向推理(目标驱动),需要在系统架构和规则设计层面进行额外设计,例如通过消息传递机制或元规则来控制推理流程^[3]。

1.3 故障树分析法

故障树分析法(Fault Tree Analysis, FTA)是一种自上而下的、演绎的故障分析方法。它以一个不希望发生的系统故障事件(顶事件)作为分析目标,通过逻辑门(与门、或门等)逐层向下寻找所有直接和间接的原因(中间事件),直至不可再分的基本原因事件(底事件)^[2]。FTA 不仅能清晰地展示故障传播路径,其树形结构本身也是一种良好的知识组织形式。将故障树转化为专家系统的知识库,可以自然地建立故障现象与原因之间的因果链,并为推理策略提供依据。因此,本文采用 FTA 作为构建矿车故障诊断知识库的基础方法。

1.4 开发平台及语言

作为.NET 平台上的核心语言,C#以其高性能和面向组件的特性,使开发者能够直接、充分地利用.NET 框架在数据通信、图形界面、硬件交互及数据库服务等方面的强大功能。此外,CLIPS 6.30 以上版本提供了可由.NET 平台直接调用的动态链接库(Dynamic Link Library),为在.NET 平台上集成 CLIPS 提供了强大基础,借助.NET 平台丰富的组件,可以实现故障诊断系统的快速集成开发。

2 系统总体设计

2.1 设计目标

本系统的设计目标是开发一个能够辅助甚至替代部分人工诊断的矿车故障智能诊断平台。具体需求包括:

2.1.1 准确性

诊断结论应具有高可信度,对常见故障的诊断准确率需达到 90% 以上。

2.1.2 高效性

响应速度快,从数据输入到结果输出应在数秒至 20 秒内完成,以满足现场快速响应的需求。

2.1.3 易用性

提供直观的图形用户界面,使现场技术人员能够方便地进行故障查询、诊断和知识维护。

2.1.4 安全性

实现用户分级管理,普通操作员只能进行诊断查询,专家或管理员方可修改核心知识库,确保知识安全可靠。

2.1.5 可维护与可扩展性

知识库应易于更新和扩展,支持通过结构化界面添加新车型、新故障知识及诊断规则等。

2.1.6 便捷性

考虑到矿区网络条件复杂,系统核心功能支持完全离线运行;同时提供在线模块,用于知识库同步、工单上传等,确保系统的可用性与先进性。

2.2 技术路线选型

当前实现智能诊断主要有规则引擎、专家系统和 AI 大模型三种技术路径。经综合对比(如表 1 所示),本系统选择基于规则的专家系统路线。主要基于以下考量:矿车故障诊断领域知识(故障码、判定条件、处理流程)结构化程度高,可由领域专家明确表述为规则;专家系统推理过程透明、结论可解释,符合安全关键系统的要求;专家系统集成成本相对较低,且能快速形成诊断能力。未来,在积累足够多故障案例数据后,可考虑引入 AI 大模型进行故障预测与深层关联分析,形成“专家系统初步筛选+AI 模型深度分析”的混合智能诊断模式。

表 1 故障诊断技术路线对比

方法	特点	适用场景	选型考量
规则引擎	基于预定义规则,灵活可调,性能高,可解释性强	设备监控、流程自动化	规则是核心,但需更上层的知识表示与管理系统
专家系统	集知识表示、推理、解释于一体,针对特定领域,用户友好	复杂故障诊断、辅助决策	选用完美匹配将专家经验转化为系统能力的需求,提供完整人机交互
AI 大模型	数据驱动,自适应强,能预测复杂非线性故障	预测性维护、大规模监控	对数据量和算力要求高,初期条件不成熟,作为远期演进方向

2.3 系统架构设计

系统整体架构主要划分为 UI 层、服务层和数据层。

2.3.1 UI 层

采用 C#.NET 平台的 WPF 界面框架开发,负责提供现代化图形操作界面与用户交互。主要模块包括:系统登录、在线连接、信号显示、故障推理、故障工单、知识查询、知识管理(专家用户)、知识发布(专家用户)及系统设置等。界面开发采用 MVVM 设计模式,实现 UI 设计与业务逻辑分离,方便软件的维护与扩展。

2.3.2 服务层(业务逻辑层)

封装系统的核心业务逻辑,是连接 UI 层与数据层的枢纽。关键服务包括:

CAN 通讯服务:实现通过 CAN 工具将系统软件与矿车的在线连接读取车辆 CAN 总线数据。信号绘制服务:将.asc 文件内保存的 CAN 总线数据以“信号曲线图”的形式展示给用户。故障推理服务:封装 CLIPS 推理引擎,负责加载规则知识库(.clp 文件)与接收数据处理服务提供的事实数据,执行故障推理并返回诊断结果。故障工单服务:负责将用户提供的故障工单数据保存至服务器的故障工单表内,便于数据与其他用户共享。知识管理服务:为 UI 层提供对故障知识库、配置知识库的增删改查接口,并管理本地 SQLite 数据库。软件更新服务:实现联网环境下的一键更新,通过对比本地与服务器文件版本,自动下载并更新知识库文件与应用程序本身。

2.3.3 数据层

负责数据的持久化存储,本系统采用混合数据存储策略。

本地知识库:使用 SQLite 数据库存储故障知识表、知识文件配置表(关联车型、故障码、规则文件)。故障

知识以独立数据库文件(如 fault.db)形式存储,内含各车型的故障信息表。规则文件:以.clp 文本文件形式存储,使用 CLIPS 语言编写,按车型和部件(如 hc105ei_bms.clp)分类存放。服务器数据库:采用 MySQL,集中存储所有知识库文件的最新版本、文件更新历史以及现场提交的故障工单信息等,作为数据分发的中心。

2.4 系统功能模块设计

基于上述架构,系统主要功能模块如下:

2.4.1 在线连接模块

用户通过 CAN 工具将本系统所运行的计算机与矿车进行物理连接,配置 CAN 通讯参数后链接到矿车 CAN 总线后在线读取 CAN 报文,如图 1 所示。

2.4.2 信号显示模块

用户获取到车辆 CAN 报文存储的.asc 文件后,可导入到系统内,通过选择产品型号及所需查看的信号后可生成信号曲线图,如图 2 所示。



图 1 在线连接模块

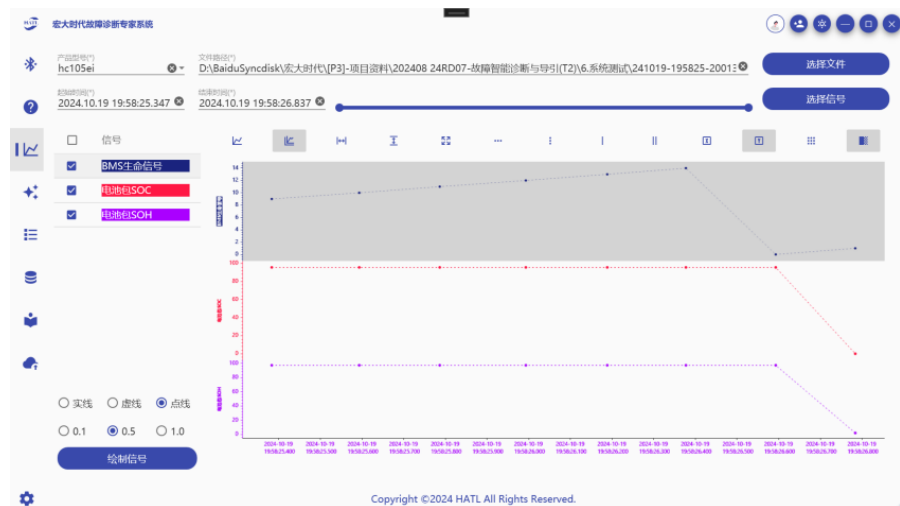


图 2 信号显示模块模块

2.4.3 故障推理模块

核心功能模块。用户选择车型、输入故障码或选择故障数据文件后,系统自动解析数据,将关键特征转化为事实,调用 CLIPS 推理引擎进行诊断,并以结构化形式展示故障原因、触发条件、影响及处理措施,如图 3 所示。

2.4.4 故障工单模块

故障工单模块的核心作用是实现现场故障信息的集中化管理与共享。该模块允许现场用户(如维护人员或操作员)准确填写并提交故障详情,这些信息将实时同步至服务器数据库中进行安全存储。与此同时,其他授权用户(如技术支持团队或管理人员)可通过系统及时获取这些现场故障数据,从而有效打破信息壁垒,提升团队协作效率与问题处理透明度,如图 4 所示。

2.4.5 知识库管理模块

知识库管理模块主要有两大块内容:

知识库查询:面向所有用户,提供本地及服务器上的

数据库内容的查询服务。

知识库管理及发布:面向专家用户,提供对知识库中故障知识、知识配置、规则(文件)知识及 CAN 数据库(DBC 文件)的编辑、审核、发布功能。通过设计结构化的编辑界面,降低直接编写知识文件内容的难度,如图 5 所示。

3 关键技术实现

3.1 知识库设计

知识库是专家系统的基石。本系统构建了多维度、结构化的知识库体系,主要包括故障知识库、规则文件库以及用于辅助系统运行的文件配置库以及知识文件信息库。

3.1.1 故障知识库

存储具体的故障案例知识,以关系型数据库表的形式组织。如表 2 所示,每条记录对应一个唯一的故障码,包含了故障现象、判定条件、对车辆的影响以及详细的处理措施。该库支持按产品型号、故障单元(如 BMS、VCU)、故障等级进行多维度查询,查询功能展示如图 6。



图 3 故障推理模块

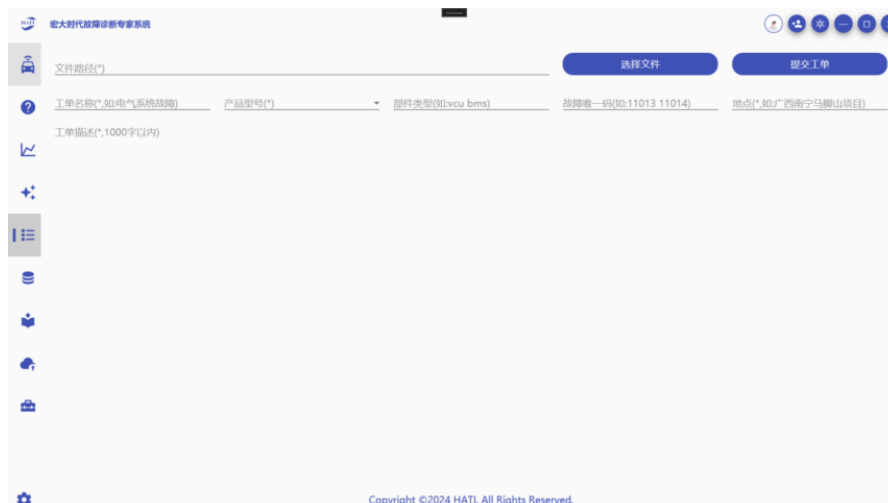


图 4 故障工单模块



图 5 知识管理模块



图 6 故障查询功能展示图

表 2 故障知识表示例

故障码	故障单元	故障内容	故障判定条件	故障影响	故障处理措施
23016	BMS	电池 SOC 过低报警 (3 级)	$SOC \leq 8\%$	限制牵引功率 0%	拖车拖往充电点充电
11011	VCU	蓄电池电压高 (1 级)	蓄电池电压持续介于 28V 至 30V 之间	可能损坏车辆低压设备	使用万用表检查蓄电池输入及输出端电压

3.1.2 规则文件库

使用 CLIPS 语言编写产生式规则。每条规则对应一种故障的诊断逻辑。规则左部 (LHS) 定义触发条件 (模式匹配), 右部 (RHS) 定义诊断结论和行动。例如, 针对故障码 23016 的规则核心逻辑为: 如果存在 (fault-current (fcode 23016)) 事实 (表示当前报出此故障码), 且存在 (bms-info (pack-soc ?soc <= ?soc 8))) 事实 (表示 SOC 值小于等于 8%), 则触发规则, 在右侧断言诊断结

果 (diagnosis “原因: 电池 SOC ?soc%, $\leq 8\%$ ”) 并给出处理建议。规则文件按车型-部件由系统模块化管理。规则文件部分内容如图 7 所示。

3.1.3 文件配置库

作为“导航表”, 建立了车型、部件、故障码列表、相关 CAN ID 列表与对应规则文件路径之间的映射关系。推理服务启动时, 首先查询知识文件配置库, 以确定需要加载哪些规则文件来处理当前诊断任务。

3.1.4 文件信息库

作为系统的“知识库”核心枢纽, 集中管理文件、车型、部件与版本之间的关联关系。为保障数据一致性与实时性, 系统采用本地与服务器的双重存储架构。当需要进行知识更新时, 系统会首先自动比对本地与服务器端文件的版本信息。若服务器存在更高版本, 则触发更新流程, 将最新文件同步至本地, 从而实现知识库的持续优化与双

向同步。

```
;;x18FFC13A
(deftemplate x18FFC13A
  (slot h-time); 事实时间
  (slot tms-sta);TMS工作状态
  (slot tms-hv-cont);TMS高压继电器状态
  (slot out-temp);出水温度
  (slot in-temp);回水温度
  (slot tms-power);TMS需求功率
  (slot tms-fcode);TMS故障码
  (slot tms-flevel);TMS故障等级
)

;;;;;;;;
;;涉及ID x18C4EFF3、x1884EFF3、x1885EFF3、x1886EFF3、x1887EFF3、x18FFC13A
;;;;;;;;
;;21161 单体过压报警 (1级)
(defrule bms-fault-161
  (fault-current (fcode ?code&(<= ?code 21161)))
  (x1887EFF3 (max-cell-volt ?macv&(>= ?macv 3.8)))
=>
  (bind ?result (format nil "原因: 单体电芯电压最大值 %.3f V, ≥3.8V" ?macv))
  (assert (diagnosis ?result))
  (printout t ?result crlf)
)

;;22081 单体过压报警 (2级)
(defrule bms-fault-081
  (fault-current (fcode ?code&(<= ?code 22081)))
  (x1887EFF3 (max-cell-volt ?macv&(>= ?macv 3.85)))
=>
  (bind ?result (format nil "原因: 单体电芯电压最大值 %.3f V, ≥3.85V" ?macv))
  (assert (diagnosis ?result))
  (printout t ?result crlf)
)
```

图7 规则文件内容展示图(部分)

3.2 推理机设计

本系统采用 NASA 开源的 CLIPS 推理引擎作为推理机核心。其工作流程如下:

初始化: 根据诊断请求中的产品型号和故障唯一码, 从配置知识库中定位并加载对应的.clp 规则文件至 CLIPS 引擎。事实断言: 数据处理服务将指定时间段内的 CAN 数据切片, 转化为一组 CLIPS 事实, 并通过 API 断言到 CLIPS 引擎的工作内存中。模式匹配与冲突消解: CLIPS 引擎采用高效的 Rete 算法进行模式匹配, 寻找所有条件被满足的规则(被激活的规则)。当多条规则同时被激活时, 采用默认的深度优先策略进行冲突消解。规则执行与推理: 执行被选中的规则右部动作, 可能包括断言新的中间事实、修改已有事实或输出诊断结果。这一过程循环进行, 形成链式推理, 直至无新规则被激活。结果获取: 推理结束后, 从引擎中检索所有诊断结果事实, 格式化后返回给几面显示显示。

3.3 数据处理与 CAN 解析

矿车故障数据主要来源于车载 TBOX 记录的 CAN 总线数据(.asc 格式)。数据处理是连接物理数据与符号推理的桥梁。

3.3.1 DBC 解析

系统内置各车型的 DBC 数据库文件。DBC 定义了 CAN 报文中信号的位置、长度、缩放因子、偏移量等解

析规则。

3.3.2 报文解码

读取.asc 文件中的每一帧报文, 根据 DBC 定义, 将十六进制原始数据解码为具有工程意义的物理值(如电压、电流、温度)。

3.3.3 数据切片与事实生成

为实现时序推理, 系统将解码后的数据按固定时间窗口(如 100ms)进行切片。对每个切片, 根据配置知识库中定义的 CAN ID 与事实模板的映射关系, 生成一组 CLIPS 事实, 供推理机使用。

3.4 知识库的构建与维护

知识库的物理存储由多组 CLIPS 规则文件(.clp)和数据库文件组成。初始知识库通过人工编写和从故障树转换而来。为了便于维护, 系统开发了知识库管理模块。该模块提供图形化编辑界面, 允许专家用户通过类表单填写等方式, 结构化地描述推理规则、条件和结论, 后台服务自动将这些描述转换为标准格式文件, 并存入数据库中。这种设计有效缓解了“知识获取瓶颈”问题, 使领域专家能够快速的参与知识库的更新与优化。

3.5 CLIPS 与 .NET 平台集成

CLIPS 推理引擎需要被集成到基于 C#.NET 开发的图形化应用中以供调用。本系统采用动态链接库(DLL)嵌入方式, 这是最常用且稳定的集成方案。具体步骤为:(1) 将 CLIPS 源代码编译为可供 C#调用的动态链接库(如 ClipsNET.dll 或自定义封装)。(2) 在 C#项目中添加对该 DLL 的引用。(3) 在服务层的推理引擎服务中, 实例化 CLIPS 引擎对象。通过该对象提供的 API(如 Load、AssertFact、Run、GetResult 等), 可以实现从 C#端加载 CLIPS 规则文件、断言事实、启动推理循环、获取并解析推理结果等全套操作。这种集成方式使得 CLIPS 强大的符号推理能力与.NET 平台丰富的 UI 控件、网络通信和数据库访问功能完美结合。

4 结束语

本文设计并实现了基于 CLIPS 的矿车故障诊断专家系统, 主要工作与贡献包括: ①设计了模块化、可扩展的三层系统架构, 明确了各层功能与交互关系。②采用故障树分析与“框架+产生式规则”混合知识表示法, 在 CLIPS 中构建了结构清晰、易于维护的知识库, 并开发了知识管理模块以降低维护门槛。③解决了 CLIPS 推理引擎与 C#.NET 上位机软件的集成问题, 实现了符号推理与图形界面的有效结合。④开发了完整原型系统, 并通过实际故

障数据验证了其可行性与有效性。然而,系统目前知识库规模有限,主要依赖规则,对需深度机器学习识别的复杂故障处理能力不足,且缺乏自学习与知识自动挖掘功能。未来研究将围绕以下方向展开:①扩充与优化知识库,引入案例推理(CBR)技术,实现基于历史案例的诊断。②融合神经网络、支持向量机等数据驱动模型与专家系统,构建混合智能诊断系统以处理更广泛故障。③拓展为云-边协同架构,实现多车数据汇聚分析、知识库云端统一管理与动态分发,提升系统整体智慧水平。

[参考文献]

[1]吴英建.基于.NET的发动机网络化故障诊断专家系统

研究[D].陕西:西北工业大学,2007.

[2]杨家涛.基于CLIPS的船舶机舱设备故障诊断专家系统研究[D].湖北:武汉理工大学,2012.

[3]沈大伟,庄诚,王学雷.基于CLIPS的故障诊断专家系统开发[J].化工自动化及仪表,2012,39(4):450-453.

[4]王华,李鹏波.CLIPS嵌入.NET平台技术与实现[J].计算机系统应用,2006(11):45-48.

[5]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,2006,14(6):78-80.

作者简介:廖清清(1994—),男,江西人,汉族,本科学历,工程师,主要从事新能源矿车设计开发工作。