

基于可视化规则引擎的轨道交通车辆智能预警系统研究 ——以 CJ6 型动车组为例

仲继生, 陈之恒

中车株洲电力机车有限公司, 湖南 株洲 412001

DOI: 10.61369/TACS.2026040047

摘 要 : 针对轨道交通车辆预警规则变更频繁、硬编码开发响应滞后且维护成本高的难题, 本文提出一种基于 Drools 可视化规则引擎的智能预警方法, 并以 CJ6 型动车组为例开展应用研究。其核心创新在于, 通过图形化规则编排界面, 将复杂的运维经验抽象为可拖拽的“参数节点”与“函数算子”, 并自动编译生成标准 DRL 文件热加载至诊断引擎, 实现了业务逻辑与底层代码的深度解耦。实车测试表明, 该系统能稳定处理单列车 15475 个测点、每秒数万条数据的并发流, 99 分位规则判断延迟仅 91ms, 故障上传端到端延迟控制在 672ms 以内, CPU 平均使用率低于 10%, 全面满足车载环境实时性与可靠性要求。该研究为构建高灵活性的轨道交通智能运维体系提供了可行的技术路径。

关 键 词 : 规则引擎; Drools; CJ6 型动车组; 故障诊断; 实时预警; 智能运维

Research on Intelligent Early Warning System for Rail Transit Vehicles Based on Visualized Rule Engine --Take the CJ6 type EMU as an example

Zhong Jisheng, Chen Zhiheng

CRRC Zhuzhou Locomotive Co., Ltd., Zhuzhou, Hunan 412000

Abstract : To address the challenges of frequent warning rule changes, delayed response of hard-coded development, and high maintenance costs in rail transit vehicles, this paper proposes an intelligent warning method based on the Drools visual rule engine, taking the CJ6 EMU as a case study. The core innovation lies in abstracting complex operational experience into draggable "parameter nodes" and "function operators" through a graphical rule configuration interface. These are automatically compiled into standard DRL files and hot-loaded into the diagnostic engine, achieving deep decoupling of business logic from underlying code. Real-vehicle tests demonstrate that the system can stably handle concurrent data streams from 15,475 measurement points per train set, processing tens of thousands of data points per second. The 99th percentile rule evaluation latency is only 91ms, and the end-to-end fault upload delay is controlled within 672ms, with average CPU usage below 10%, fully meeting the real-time and reliability requirements of the onboard environment. This research provides a feasible technical pathway for building a highly flexible intelligent operation and maintenance system for rail transit.

Keywords : rule engine; Drools; CJ6 EMU; fault diagnosis; real-time warning; intelligent operation and maintenance

一、概述

随着轨道交通装备智能化水平的不断提升, 车载监测系统所产生的数据量呈指数级增长, 对故障诊断的实时性与准确性提出了更高要求^[1]。传统的故障诊断方法多采用硬编码方式将业务逻辑嵌入软件系统, 导致规则变更周期长、维护成本高, 难以适应快速变化的运维需求。

规则引擎作为一种将业务决策逻辑与应用程序代码解耦的中间件技术, 通过预定义规则库与高效推理机制, 能够实现对复杂逻辑的快速响应与动态更新。在轨道交通领域, 已有研究表明, 规则引擎在列控数据验证方面展现出良好的应用前景^[2], 能够实现验证逻辑与代码分离, 有效提升系统的敏捷响应能力^[3]。

本文以 CJ6 型动车组为研究对象, 基于实际运维文档《CJ6 型动车组预警报警模型》, 设计并实现了一套基于 Drools 规则引擎的智能预警报警系统。系统涵盖齿轮箱、电机、轴承、制动、高压等 15 大类、40 余条预警规则, 采用边缘计算架构实现传感器数据的实时处理与分级预警^[4]。实际运行结果表明, 该系统能够支持单列车 600 个传感器、每秒 600 条数据的并发处理, 诊断延迟控制在 100ms 以内, 预警准确率满足现场运维需求, 为轨道交通装备的智能化运维提供了可行路径^[5]。

二、CJ6 型动车组预警报警需求分析

本节围绕 CJ6 型动车组预警报警系统的工程需求展开分析。

一方面，车载监测普及带来海量并发数据，对系统吞吐量和实时处理能力提出更高要求，亟须引入高效计算方案；另一方面，传统故障诊断软件因业务逻辑与代码强耦合，导致开发周期长、维护成本高，难以灵活响应运维规则的快速变化。基于上述问题，本节将从监测现状、预警模型特征及传统开发模式局限性三方面进行分析，为后续引入规则引擎奠定需求基础。

（一）CJ6型动车组监测概况

CJ6型动车组是我国自主研发的时速160公里动力分散型电力动车组^[4]，主要服务于长株潭城际铁路等线路，具有启停频繁、运行密度高、周转时间短的特点。车辆采用“2动2拖”4节编组形式，支持灵活重联运行，最大可实现8节编组，对列车运行安全性和可靠性提出了更高要求。

全车部署了高密度信号采集网络，以4节编组为单位，全车信号总数达到15475个，其中传感器信号约占50%，其余为状态信号、控制信号和通信信号以及结果变量。监测范围覆盖走行部、高压系统、牵引系统、制动系统、辅助系统等关键部件，数据采集频率为1Hz，每日单列车产生的监测数据量可达GB级别。这些海量实时数据为车辆状态感知与故障预警提供了丰富的信息基础，但同时也对数据处理能力提出了严峻挑战。

面对如此庞大的实时数据流，传统的人工阈值判定或简单的固定规则判断已难以满足复杂工况下的精准预警需求。首先，数千个传感器测点之间存在复杂的物理耦合关系，单一测点的数值异常可能是多种故障模式的表现，需要结合多源信息综合判断。其次，车辆运行环境复杂多变，环境温度、线路条件、负载状态等因素都会影响设备运行参数，要求预警规则具备自适应能力。再者，机械电气设备的故障往往具有时序演化特征，早期故障表现为缓慢指标漂移，突发故障则表现为参数剧烈突变，需要系统具备不同时间窗口的处理能力。

这些复杂特性使得传统基于硬编码的预警系统面临诸多瓶颈：规则变更需要修改源代码并重新部署，单次调整周期往往需要数天时间，难以适应运维模型的快速迭代；时序窗口计算、多源数据融合等复杂逻辑在传统编程范式下实现困难且效率低下；面对15475个信号的实时流入，层层嵌套的条件判断语句导致CPU开销急剧增加，难以满足车载系统毫秒级的实时诊断要求。因此，迫切需要引入灵活高效的规则引擎技术，将业务逻辑与系统实现解耦，实现预警规则的动态配置和高效执行。本文以CJ6型动车组为例，基于Drools规则引擎构建智能预警报警系统，以解决上述复杂工况下的实时诊断难题。

（二）预警诊断模型概述

如表1所示，CJ6型动车组的预警诊断模型呈现出覆盖面广与逻辑高度复杂的显著特征。全车预警共划分为12大类、39项核心规则，跨越机械传动、电气与制动等多个子系统，深度覆盖齿轮箱、电机、轴箱轴承、总风系统等关键部件。在判定逻辑上，该模型突破了单一的固定阈值判断，广泛融合了环境温度耦合、同源设备均值比较、时序变化率追踪以及跨系统状态组合等多维判定条件。这种多变量交叉关联、时序窗口嵌套的综合诊断机制，导致业务逻辑的分支与状态链路极其繁杂，对底层系统的高效计

算与解耦能力提出了极大的挑战。

序号	部件/系统	规则数量	主要监测参数与判定维度
1	小/大齿轮箱	10	绝对温度、温差、均衡温差、温度突变（结合环温）
2	电机定子与轴承	7	绝对温度、温差、均衡温差、温度突变
3	轴箱轴承	4	绝对温度（分级判定）、温差、均衡温差、突变
4	WTD 通讯设备	1	连续离线时间（>3天）
5	总风系统	1	单车压力异常、重联列车管不贯通、打风频率异常
6	高压系统	1	网压异常持续时间
7	蓄电池	3	均充/浮充单节电压偏低（<均值0.9倍）、充放电电压降
8	基础空气制动	1	强迫缓解操作后的制动缸（BC）压力残留
9	EP 阀系统	1	列车管减压量及持续时间异常
10	变流器水冷系统	5	冷却水入口温度（分级）、进出水温差、系统流量及压力
11	变压器系统	3	变压器油温超限（>90℃ / >95℃）
12	空转滑行系统	2	持续空转/滑行预警、频繁空转/滑行预警
合计 共计 39 条核心诊断逻辑			

表 1: CJ6型动车组预警项点分布

（三）传统模型开发困难

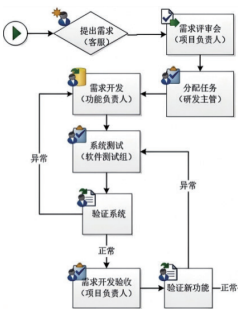


图1 传统故障诊断模型开发流程图

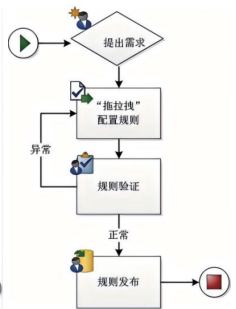


图2 基于规则引擎的诊断模型开发流程

在传统的轨道交通预警报警系统开发与维护中，预警规则往往通过硬编码的方式直接固化在软件系统的底层逻辑中。如图1所示，传统的故障诊断模型开发流程极为繁琐，涉及需求提出（客服）、需求评审（项目负责人）、任务分配（研发主管）、代码开发（功能负责人）、系统测试（软件测试组）以及最终的验收与发布等多个串行环节。一旦现场工况发生变化或需要调整预警阈值，研发人员必须对软件内部代码进行修改，甚至可能引发底层功能的重构。这不仅导致开发成本高昂、涉及人员众多，而且每次更新都需要重新编译、全面测试并重启发布软件，整体响应周期长，难以满足实际运维中业务规则频繁变更的需求。

为了突破传统硬编码带来的研发瓶颈，引入规则引擎成为一

种行之有效的解决方案。如图2所示,使用规则引擎后的开发流程得到了根本性的简化。新的流程将易变的业务逻辑与底层系统架构深度解耦,业务人员或系统管理员只需通过“拖拉拽”等可视化界面的配置方式,即可完成规则的定义与修改。新规则配置完成后,可直接在系统中进行闭环验证,验证通过后即可实现规则的热发布。整个过程无需修改系统底层代码,也不涉及复杂的跨部门协同与软件重启发布。这种范式的转变从根本上降低了预警逻辑修改的技术门槛与复杂性,大幅削减了开发成本和人力投入,使得系统能够敏捷地应对现场需求的快速迭代。

三、基于 Drools 的规则引擎系统设计

(一) 规则引擎定义

规则引擎是一种旨在将易变的业务决策逻辑与底层应用程序深度解耦的基础软件组件。其核心架构通常包含规则库、工作内存与推理机,通过引入如 Rete 或 Phreak 等高效的模式匹配算法,系统能够对连续输入的“事实”数据与预设规则条件进行高速增量计算与状态评估。在轨道交通车辆的高频并发监测场景中,规则引擎摒弃了传统的硬编码模式,有效规避了复杂条件嵌套带来的性能瓶颈;它不仅能够以毫秒级低延迟处理多源传感器交叉融合、时序窗口分析及环境参数耦合等复杂预警逻辑,还使得运维专家可以借助可视化或近自然语言的方式直接参与规则的动态配置与迭代,从而显著提升了智能预警报警系统的可扩展性与工程实用性。

(二) 总体架构

针对 CJ6 型动车组车载设备传感器节点密集、数据采样频次高(1Hz)及预警逻辑多维耦合等典型特征,本文提出一种微服务架构。该架构旨在实现故障诊断逻辑与底层业务代码的深度解耦,保障海量并发状态下的低延迟推理响应。结合边缘诊断与中心聚合的分层设计思想,系统自下而上划分为数据接入层、核心处理层、数据存储层、服务治理与调度层以及展示与交互层,系统总体架构如图3所示。

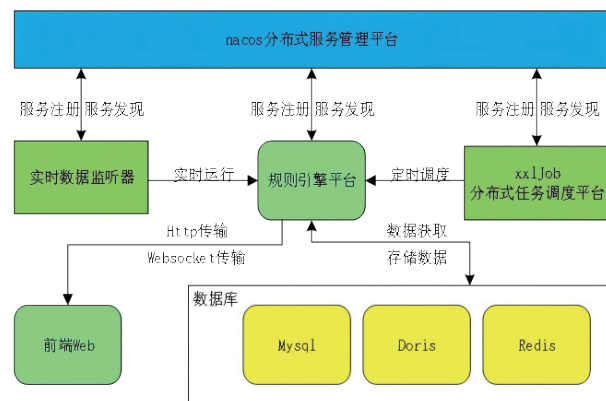


图3 基于规则引擎的智能预警报警系统总体架构

系统各核心模块及其数据交互流向具体描述如下:

- 数据接入层(实时数据监听器):作为车载异构数据的统一入口,该层主要承担 CJ6 型动车组 15475 个信号的实时采集任

务。数据实时接入组件对高速流入的原始报文进行协议解析、异常值清洗与标准化封装,将其转化为规则引擎可识别的事实对象(Fact),随后通过高吞吐量的实时流推入核心处理中枢。

- 核心处理层(规则引擎平台):作为本系统的智能诊断推理核心,该层深度集成了 Drools 规则引擎。通过预加载包含环境耦合、时序变化率及系统级差值等多维度计算逻辑的预警规则集,引擎利用高效的模式匹配算法(Rete)对连续输入的数据流执行高速逻辑推理,进而毫秒级输出符合 A/B/C 分级标准的预警诊断结果。

- 数据存储层(数据库):为适配不同模态数据的读写特性,系统构建了多元化的混合异构存储架构。其中,MySQL 等关系型数据库用于持久化存储预警规则元数据、车辆基础配置及系统权限模型;高性能分析型数据库(如 Apache Doris)专门承载海量历史监测数据与报警日志,以支撑复杂的多维数据挖掘与事后故障追溯;同时,系统引入 Redis 构建内存级缓存,用于管理高频访问的事实对象、车辆实时状态快照以及计算温升突变所需的时序滑动窗口,从而有效降低高频计算场景下的 I/O 延迟。

- 服务治理与调度层(任务调度平台):在分布式部署架构下,本层依托 Nacos 分布式服务管理平台提供完善的服务注册与发现机制,确保“数据接入”“规则计算”等微服务节点的通信可靠性与动态弹性扩缩容能力。此外,系统集成 xxlJob 分布式任务调度框架,负责管理长周期组合规则的定时校验任务与离线分析调度。

- 展示与交互层(前端 Web):该层采用 Web 化视窗为车辆运维人员提供全景可视化的交互接口。前端应用通过 HTTP 协议与核心服务层通信,实现诊断规则的动态编排与历史运维数据检索;同时,利用 WebSocket 技术建立全双工长连接,确保车辆实时工况及突发报警指令能够以极低延迟推送至终端界面,形成完整的监测预警闭环。

四、CJ6 型动车组应用案例分析

(一) 规则库设计

本案例的规则库设计,采用的是图形化界面设计,以此来降低规则开发门槛。如图4所示,规则库摒弃了传统的代码硬编码模式,转而采用面向领域专家(车辆运维人员)的“节点化、可视化”规则编排设计理念。



图4 规则配置主界面

在底层抽象设计上,系统将 CJ6 型动车组复杂的物理架构

（如牵引系统、制动系统、通风监测等）及其海量传感器测点映射为标准化的“参数节点”，同时将各类诊断所需的固定阈值和计算逻辑封装为高内聚的“常量节点”与“函数算子”（如四则运算、关系运算及逻辑与或非运算等）。

在规则构建机制上，预警诊断模型的设计过程被巧妙转化为直观的拓扑连线操作。业务专家无需具备专业的编程背景，只需依据实际的预警诊断模型，提取关键监测参数、设定阈值条件，并通过连线建立数据流转与逻辑校验链路，即可编排组合出跨系统、多维度的复杂预警规则。

在系统执行引擎层面，当可视化规则拓扑定型并提交后，平台后台的核心解析器会自动将其编译并转化为 Drools 规则引擎原生支持的标准化规则描述语言（DRL 脚本）。生成的 DRL 文件随后被无缝下发并热加载至车载边缘计算节点中。这种“所见即所得”的规则库设计机制，不仅在物理层面实现了业务诊断逻辑与底层软件架构的绝对解耦，更从工程实践角度打通了从“运维经验”到“机器执行”的快速转化通道，赋予了系统极高的动态扩展与敏捷迭代能力。

（二）引擎诊断流程

系统在 CJ6 型动车组上共部署预警规则 39 条。以图中所示的故障诊断流程为例，该流程清晰地展示了基于规则引擎的预警机制如何实现从传感器数据采集到最终诊断结果输出的完整闭环。

如图 5 所示，诊断流程始于计算结果变量以及多源传感器数据的实时采集。车载报文数据（如计算结果变量、传感器数据等）首先被下发至地面解析程序，数据解析后上传至 Kafka 存储。随后，规则引擎系统监听解析后的新数据，并存储到工作内存中，引擎根据预加载的预警规则集进行逐条匹配与逻辑推理。

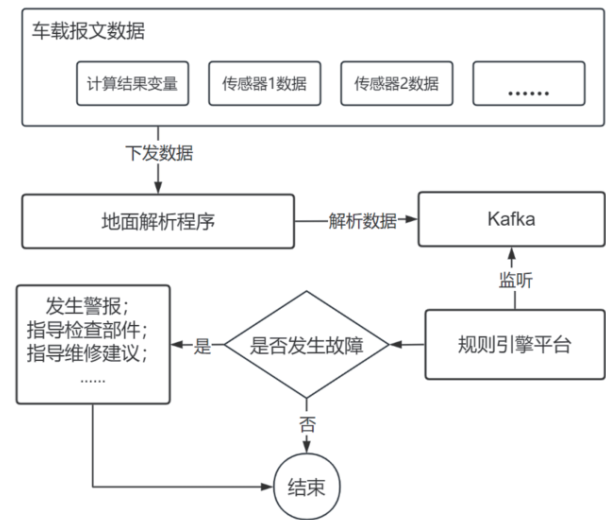


图5 故障诊断流程图

在规则执行过程中，一旦某条规则的条件被触发（例如，轴箱温度超过绝对阈值且与相邻轴箱的温差超出允许范围），引擎立即执行该规则对应的动作集。这些动作包括：发出分级警报（如 A 类严重报警、B 类中度报警等）、请求规则引擎继续评估关联规则、遍历其他相关规则以确认是否存在复合故障，以及向运维终端提示具体的检查部件（如“建议检查轴箱轴承”）、输出结

构化的诊断结果，供运维人员及时处置。

该机制的核心优势在于，复杂的跨系统计算与阈值判定逻辑被完全封装在规则引擎的可配置规则集中。若后续需要调整诊断逻辑（例如修改温差阈值或增加新的判定条件），运维人员仅需在可视化规则配置界面中修改相应节点或连线，系统即可自动完成规则的热加载与生效，无需对底层软件代码进行任何修改或重新部署，从而实现了业务敏捷性与系统稳定性的统一。

（三）系统性能测试分析

为验证基于规则引擎的智能预警报警系统在真实车载环境下的处理能力，我们在实验室搭建了模拟测试环境，对系统进行了全面的性能压力测试。测试环境配置如下：服务器采用 Intel Xeon Gold 5218 CPU（48 核心）、96GB DDR4 内存、6TB SAS 存储，操作系统为 CentOS7.9。被测系统完整部署了 Drools 规则引擎及配套的数据接入、存储与展示模块，规则库包含 CJ6 型动车组全部的 39 条预警逻辑，覆盖阈值判定、差值分析、时序窗口及多条条件组合等复杂类型。

测试数据源模拟了 15 列 CJ6 型动车组的实时监测流，每列车按照实际信号规模生成 15475 个测点数据（其中约 50% 为传感器数据），数据更新频率为 1Hz，即系统需并发处理约 20 万条数据 / 秒的接入与规则匹配。测试持续 2160 小时，重点观测规则引擎的处理延迟及系统资源占用情况。

延迟指标定义如下：

1. 规则判断延迟：从数据进入规则引擎工作内存到规则匹配完成并输出诊断结论的时间间隔（不包括数据预处理与动作执行）。
2. 故障上传执行延迟：从数据到达系统入口到最终预警信息推送至前端界面或存储数据库的完整端到端耗时，包含数据解析、规则判断、动作触发及网络传输。

测试结果表明，系统在满负荷压力下表现稳定，各项延迟指标均满足车载实时诊断的严苛要求。具体数据如表 2 所示。

从表 2 可以看出，规则判断延迟平均为 42ms，最坏情况不超过 100ms；故障上传执行延迟平均为 218ms，最坏情况控制在 1s 以内。这一性能表现完全满足 CJ6 型动车组 1Hz 数据更新频率下的实时诊断需求，且留有充足余量应对突发峰值。

统计指标	规则判断延迟	故障上传执行延迟
平均值	42	218
P95	76	456
P99	91	672
最大值	98	987

表 2 系统处理延迟实测结果（单位：ms）

在系统资源消耗方面，CPU 平均使用率低于 10%，峰值情况下亦未超过 15%，充分体现了规则引擎高效的模式匹配算法对计算资源的友好性。内存占用方面，常用内存（活跃使用）小于 40GB，运行内存常驻占用率稳定在 12GB 以内，主要包含规则引擎的工作内存及滑动窗口缓存。系统在 2160 小时连续运行中未发生内存泄漏或规则执行错误，表明其具有良好的稳定性与可

靠性。

五、结束语

本文针对轨道交通车辆高频海量监测数据处理与复杂预警逻辑解耦的工程难题，提出并构建了一种基于 Drools 规则引擎的 CJ6 型动车组智能预警报警系统。通过引入边缘计算架构与可视化规则编排技术，系统彻底突破了传统硬编码模式下规则变更困难、响应滞后的瓶颈，实现了故障诊断逻辑的动态配置与快速热发布。测试与实车应用表明，系统能够高效处理涵盖齿轮箱、变

流器、高压系统等关键部件的 39 条核心规则，在单列车每日 GB 级数据量的并发压力下，平均规则判断延迟仅为 42ms，完全满足现场 100ms 级的高可靠、低延迟诊断需求。

本研究的成功实施不仅验证了规则引擎在实现业务决策与底层代码深度解耦方面的巨大工程价值，更为轨道交通装备的智能化运维体系提供了一套高可扩展的范式。面对未来日趋复杂的行车工况，下一步研究将聚焦于突破现有静态计算模型的局限，探索预警规则的自适应学习机制与预测性维护算法，并计划引入大模型辅助自动化生成 DRL 代码，以期进一步降低现场运维门槛，推动车载智能诊断技术向全面自动化与认知化演进。

参考文献

[1] 赵焱, 曾云峰, 陈龙, 等. 基于大数据的城轨车辆 PHM 智能运维系统的研究与实践 [J]. 铁道车辆, 2022, 60(6): 133-138. DOI: 10.3969/j.issn.1002-7602.2022.06.029.

[2] 崔岩. 规则引擎在列控产品配置数据验证中的应用 [J]. 铁路通信信号工程技术, 2020, 17(7): 29-34. DOI: 10.3969/j.issn.1673-4440.2020.07.007.

[3] 杜星熠, 胡静, 蒋伟, 等. 基于 Drools 规则引擎的物联网数据查询方法设计 [J]. 测控技术, 2022, 41(8): 71-77. DOI: 10.19708/j.ckjs.2022.08.012.

[4] 同强强, 何伟. 一种基于行动规则的仿真推演引擎设计与实现 [J]. 舰船电子工程, 2022, 42(12): 85-88, 112. DOI: 10.3969/j.issn.1672-9730.2022.12.019.

[5] 王钰. 基于规则引擎的气象台站元数据质量控制方法 [J]. 江西科学, 2023, 41(6): 1198-1201. DOI: 10.13990/j.issn1001-3679.2023.06.029.