

# 汽车制造工艺中的 EOL 序列便捷编辑工具研究

郝梦园 杨东旭 张晓光 陈越

中国第一汽车集团有限公司, 中国·吉林 长春 130011

**摘要:** 为了提高汽车制造工艺中 EOL (End of Line) 序列的编辑效率和准确性, 论文研究一种基于 Web 网页的序列编辑工具, 以可视化操作方式, 允许用户快速构建和生成汽车诊断流程, 有效指导电检程序开发和电检工艺落位。论文详细研究了该工具的设计和实现, 旨在为汽车诊断提供一个切实可行的序列编辑解决方案。

**关键词:** 汽车制造; EOL; 序列编辑; 诊断流程

## Research on Convenient Editing Tool for EOL Sequence in Automotive Manufacturing Process

Mengyuan Hao Dongxu Yang Xiaoguang Zhang Yue Chen

China FAW Group Co., Ltd., Changchun, Jilin, 130011, China

**Abstract:** In order to improve the efficiency and accuracy of EOL (End of Line) sequence editing in automotive manufacturing processes, this paper studies a web-based sequence editing tool that allows users to quickly construct and generate automotive diagnostic processes through visual operations, effectively guiding the development of electrical inspection programs and the positioning of electrical inspection processes. This paper provides a detailed study of the design and implementation of the tool, aiming to provide a practical and feasible sequence editing solution for automotive diagnosis.

**Keywords:** automobile manufacturing; EOL; sequence editing; diagnostic process

### 0 前言

软件定义汽车趋势下, 电器功能不断增加, 诊断逻辑日益复杂, 对车辆下线 (End of Line, EOL) 测试序列提出的要求也越来越高。EOL 测试序列由一系列满足诊断通信协议 (如 ISO 14229) 的诊断服务指令和过程逻辑构成<sup>[1]</sup>。

在汽车制造过程中, 量产车型的控制器 (ECU) 诊断规范文件及 EOL 测试序列由研发端释放后, 往往需要由产线工程师对 EOL 测试序列进行二次编辑, 目的是匹配生产实际, 补充缺失的工艺要求、检测前提、并行诊断、失败处理逻辑等信息。现有序列编辑技术存在以下不足:

①大部分依赖手工编制, 使用 Microsoft Word 或 Excel 等传统文本编辑工具, 处理复杂逻辑关系力不从心, 转换效率偏低。

②生产实施中动态调整指令执行难度大, 仅按照经验进行时序排布, 准确性较差、无法快速压缩电检工时。

③面对测试车型变更, 已成型的序列指导文件变更适配较为烦琐, 管理混乱。

针对汽车制造工艺中存在的上述问题, 如何使得整个序列详细开发过程准确且高效, 进而有效指导电检程序开发和工艺落位, 对汽车产线工程师提出挑战。

论文旨在研究 EOL 测试序列的便捷编辑方案, 研究兼容数智化诊断工具链的软件应用, 如图 1 所示, 该项工作将使得编辑操作更加便捷、易读、可视, 助力满足产线电检工艺需求。

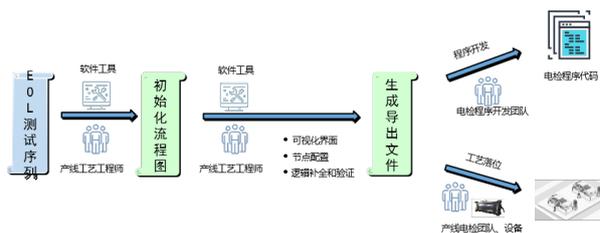


图 1 EOL 序列便捷编辑方案

### 1 汽车 EOL 及诊断测试相关研究

汽车制造企业的 EOL 测试序列经过标准化文件模板的创建和记录, 确保了测试的一致性和可追溯性, 一般含有如下重要信息:

- ①整车信息: 适用特定车系或车型。
- ②控制器 (ECU) 信息: 提供 ECU 名称、物理请求 ID、物理响应 ID、功能请求 ID、DoIP 逻辑网关地址、DoIP 逻辑功能地址和 DoIP 逻辑 ECU 地址等控制器相关信息。
- ③版本和更新信息: 记录了文档的版本和更新内容, 包括版本号、更新描述、日期和变更人。
- ④测试序列信息: 详细列出了测试序列的执行过程, 包括序列名称、测试子项、步骤、诊断服务请求、期望响应、特征值、前提条件、建议工位和备注等。
- ⑤特殊检测信息: 包括 ECU 标定、自学习、初始化、自检等特殊检测流程, 以及对应的说明文件要求。
- ⑥其他信息: 如对时间参数、安全算法、指令生效要求等的附加注解。

在现有研究中, 汽车诊断数据的编辑和管理可依靠制定

的标准来执行。OTX (Open Test Sequence Exchange Format) 是一种基于 ISO 13209 标准的开放式测试序列描述格式, 允许创建从基础功能测试到完整测试应用的诊断测试序列<sup>[2]</sup>。能够与 ODX (Open Diagnostic Data Exchange) 格式集成, 且具有高度的可扩展性, 支持 HMI 访问调用, 使得诊断测试序列开发过程更容易, 更有成效。

一些研究开始探索基于图形化界面的序列逻辑编辑方法。例如, 利用流程图建模, 通过加载资源文件和与工作流引擎的交互, 实现了流程图的可视化编辑和属性配置。然而, 这些研究大多集中在一般性的流程图编辑上, 对于汽车诊断序列这一特定应用场景的深入研究和优化还不够充分<sup>[3,4]</sup>。

此外, König 等人还提出一种在软件定义汽车制造中使用 BPMN 和 Web 服务编排汽车测试工作流的方法<sup>[5]</sup>。利用 BPMN 2.0 符号来构建测试流程模型, 可进一步在车载诊断程序中使用流程驱动进行执行分析, 有益之处是可视化增强, 但需要复杂的初期建模和过程理解<sup>[6]</sup>。

论文基于上述参考, 考虑产线工程师二次编辑需要, 最终通过设计图形化界面, 集成开发一款 Web 端便捷工具, 提高了序列转换效率和准确性。

## 2 便捷编辑工具设计

### 2.1 系统架构

论文提出的便捷编辑工具采用 B/S 架构, 见图 2, 包括前端编辑器、后端服务器和数据库。前端编辑器提供用户界面和编辑功能, 后端服务器处理业务逻辑和数据管理, 而数据库负责存储编辑好的诊断序列和相关配置信息。



图 2 便捷编辑工具架构

### 2.2 功能模块设计

如图 3 所示, 系统主要包含以下功能模块。

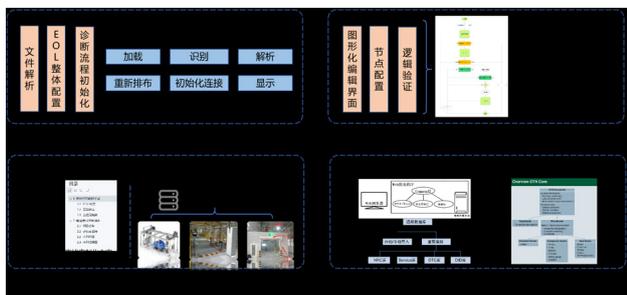


图 3 工具功能模块设计

### 2.2.1 文件解析模块

①文件解析: 加载并解析由研发端释放的 EOL 测试序列, 提取对应的诊断流程构成元素, 特别是序列名称、测试子项、步骤、诊断服务请求、期望响应、特征值、前提条件、建议工位等信息。

② EOL 整体配置: 同一车型下, 所有测试序列都含所属 ECU 和所属工位两个属性。据此, 按两个维度显示已解析出的所有测试序列, 并允许对特定维度下的序列进行重新排序、复制、增删, 修改 EOL 配置。

③诊断流程初始化: 按每个序列的诊断步骤进行初始化连接, 得到至少一个诊断序列流程, 并以流程图的可视化形式显示。

### 2.2.2 流程图编辑模块

①图形化编辑界面: 基于 HTML5 和 JavaScript 技术, 提供画布式界面, 实现流程图的拖拽、连接和编辑功能。

②节点配置: 允许用户对每个诊断步骤对应节点进行详细配置, 包括诊断服务请求、参数设置等。

③逻辑补全和验证: 在编辑过程中实时验证流程逻辑的正确性, 确保测试序列的合理性。

### 2.2.3 数据管理模块

① OTX 协议支持: 确保编辑的测试序列符合 ISO 标准, 便于与现有诊断系统兼容。

②数据交互: 通过 RESTful API 与后端服务器进行数据交互, 实现测试序列的保存、更新和检索。

### 2.2.4 生成导出模块

①格式转换: 将编辑好的测试序列流程转换、保存为 XML、SVG 等格式, 以适应不同的应用需求。

②文件导出: 提供导出接口, 允许用户将编辑好的详细序列导出为符合产线实际的可读性文件, 便于后续传递至电检程序开发团队。

## 2.3 技术路线

前端技术: 采用 HTML5、CSS3、JavaScript 和 SVG 技术, 实现响应式和交互式的编辑界面。

后端技术: 使用 Java EE 和 Spring 框架, 实现业务逻辑的处理和数据管理。

数据库技术: 采用 MySQL, 存储测试序列和配置信息。

数据交换格式: 使用 JSON 作为前后端数据交换的格式, 确保数据的一致性和实时性。

## 3 便捷编辑工具技术实现

本节研究用于完整描述测试逻辑的简单元语言和图形化表示, 并结合软件开发技术说明在 Web 端的显示、交互和生成原理。

### 3.1 流程编辑器实现

基于 B/S 架构的流程编辑器具有易集成、开发工作量小的特点。采用一种轻量级流程编辑技术, 前端页面采用框

架 Vue.js (基于标准 HTML、CSS 和 JavaScript 构建, 提供一套声明式、组件化编程模型) 开发, 支持流程节点的拖拽、连接和属性编辑。用户可以通过拖拽预定义的图形元素进行流程搭建, 通过连接线定义节点间的逻辑关系。

主界面由四个区域组成:

①菜单区: 显示当前 EOL 测试序列流程的名称、ECU 名称和所属关系, 对应某一车型或某一工位。

②组件选区: 是流程编辑所需组件的集合, 根据不同属性进行分类, 为诊断服务 (SessionControl、ECUReset、ReadDataIdentifier、WriteDataIdentifier、TesterPresent、ClearDiagInformation、InOutControl、RoutineControl、SAResetSeed、SASendKey、CommunicationControl、TesterPresent、ControlDTCSetting、DTC-ReportNum 等 UDS 命令) 组件、逻辑 (开始、结束、判断、循环、并行、赋值、回调等) 组件、操作 (延迟、提示等) 组件。此外还附有变量区和输入输出参数区。选择要使用的组件并将其拖拽到画布区构成流程节点, 进一步可通过拖放来修改单个组件的位置 / 大小。

③画布区: 用于绘制流程图的空间。

④属性编辑区: 用于对当前节点进行配置, 每个节点都提供丰富的配置选项, 包含下拉框、输入框、弹框等形式, 允许用户配置节点的诊断服务指令和参数、访问不同的数据或执行不同的操作。例如, 通过输入框定义一个是否执行 0x19 服务读取 DTC 的条件表达式。

本流程编辑器中的拖拽操作是利用 HTML5 提供的 API, 通过监听 dragstart、dragover、drop 等事件, 控制节点的拖动行为和放置位置; 连接线生成操作则是通过鼠标事件监听和 SVG 路径绘制。为避免连接线的交叉和重叠, 还开发实现基于图论的智能路由算法, 自动寻找最短且不交叉的路径。

此外, 为满足动态操作, 由 Vue.js 框架根据节点的类型和配置信息动态生成属性编辑区的 UI 组件; 并利用双向数据绑定技术, 确保用户在属性编辑区中的更改能够实时反映到流程图中, 触发后续的逻辑处理。

### 3.2 数据交互与存储

在 Web 客户端设计和编辑序列流程图后, 将提交流程图数据到服务器端, 保存数据到数据库。

前端 js 代码中, 每个 EOL 测试序列流程的全部流程元素都以 XML 形式存储, 包含名称和图标等基本信息, 以及节点步骤、每个节点的组件和每个组件的配置。用户操作的每一步 (添加步骤、拖放组件、修改配置等) 都会更新此 XML 并将其存储。在编辑或访问某测试序列时, XML 将由该测试序列 ID 读取并解析, 以形成最终的渲染效果。

系统通过 AJAX (异步 Javascript 和 Xml 技术, 数据传输) 技术与后端服务器进行数据交互, 实现测试序列的实时保存和更新。后端服务器通过 JDBC (Java Database

Connectivity) 技术与数据库进行交互, 存储和管理诊断序列数据, 实现通用和跨平台访问, 实现访问标准数据库获取外部诊断数据, 如否定响应信息、控制器故障码信息、安全算法信息等。

特别的, 在提交数据前, 前端要进行格式和逻辑初步验证, 并在后端进行更为严格的数据完整性、一致性和业务规则校验, 确保接收到的数据符合测试序列可用性要求。设计 RESTful API, 实现前端与后端的数据交互, 包括测试序列的增删改查操作。

论文设计合理的数据模型, 存储测试序列的节点信息、连线信息和配置参数。例如, 节点间连线信息表征路径流向逻辑, 该流向逻辑及所附条件数据可存储为一组表单数据等。论文选择 MySQL 数据库, 该数据库是一款成熟的、开源的关系型数据库, 可以使用 SQL 语句进行处理, 在满足存储需求外能够快速、安全地处理大量数据。又如, 对故障码的存储, 在 MySQL 中建立故障码解析的数据 DTC\_table, 其结构如表 1 所示。其中, DTC 为键值, 通过该值对故障码进行快速检索。

表 1 故障诊断码数据

DTC	DESC_C	DESC_E
C110216	电池电压过低	Battery Voltage Low
C110921	点火信号故障	Ignition Signal Error
C111201	EPS 传感器电源电压故障	Torque and Angle Sensor Supply Voltage Error
C126204	温度传感器故障	Temperature Sensor Error
C16034B	过热保护	Over Heat Protection

### 3.3 导出功能实现

系统支持导出功能, 支持通过解析编辑器中的 SVG 图形和配置信息, 将编辑好的诊断序列导出为所需工程文件格式。

将内部数据模型转换为 XML、SVG 等格式需要开发格式转换器, 此处格式转换机制利用了 OTX 协议 XML Schema, 确保导出的 XML 文件符合预期的 OTX 描述格式和标准。开发算法将流程图的 SVG 表示转换为适用于工业标准的 SVG 文件, 包括对图形属性和样式的精确控制。

特别的, 本方案导出的上述工程文件格式可以通过映射编辑器以图形方式重新创建和编辑。

在后端实现的导出服务, 还允许用户根据模板导出测试序列, 使得人工易读, 该模板为汽车制造产线梳理的定制化格式, 包含整体序列表单预览 (工位级 / ECU 级)、ECU 信息、各序列流程图、各序列文字描述及附图。

## 4 工具验证

基于已开发的软件, 论文使用主机厂真实 EOL 测试实

例进行工具验证。过程为：首先，于 EOL 产线测试环境部署软件于服务器及 Web 端，且该软件版本已经过前期系列测试；其次，协调软件测试人员及产线工程师，参照某车型正常生产过程文件，共同使用该软件进行序列编辑；最后，输出序列编辑结果，由电检程序开发团队进行评审，记录情况。

其中，举例对某车型的“静检工位\_ECAS 初检\_高度传感器检查”EOL 序列进行详细开发，从最终编辑效果来看，诊断流程编辑为完全的可视化操作，搭建流程图过程简单流畅，能够满足对车辆进行 EOL 测试序列二次编辑并生成结果文件的要求。

本研究在此过程中得到优化，基于收集的大量测试用例来进行工具验证和使用情况评审。

## 5 结论和建议

论文提出的基于 Web 的汽车诊断序列编辑工具，支持可视化的拖拽操作，简化了汽车制造工艺中 EOL 测试序列的编辑过程，提高了编辑效率和准确性。使得序列编辑结果关联产线实际，为电检程序开发提供指导，实现准确工艺落位。

未来的研究可以在工具智能化、自动化方面进行深入研究，进一步优化用户体验和系统性能。

## 参考文献：

- [1] 马里,金良伟,谭辉,等.智能汽车制造下线EOL检测架构及应用机制研究[J].汽车知识,2024,24(4):138-141.
- [2] ISO G. Switzerland: Road vehicles—Open Test sequence eXchange format (OTX)[J].2012.
- [3] 李莉.基于C2B的汽车诊断配置方法设计[J].汽车与新动力,2022,5(4):58-60.
- [4] 李娇娇.电动汽车电池运行状态远程诊断技术研究[D].焦作:河南理工大学,2020.
- [5] König S, Vogel-Heuser B, Hradecky A, et al. Executing automotive test workflows with BPMN and web service orchestration in software-defined car manufacturing[J]. Production Engineering, 2024:1-15.
- [6] Mrasek R, Mülle J, Böhm K, et al. User-friendly property specification and process verification—a case study with vehicle-commissioning processes[C]// Business Process Management: 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings 12. Springer International Publishing, 2014:301-316.

作者简介：郝梦园（1991-），男，中国吉林人，本科，工程师，从事汽车电器诊断研究。