由大语言模型赋能的设计阶段功能安全与安全性在汽车系统中的应用

Nenad Petrovic*、Vahid Zolfaghari*、Fengjunjie Pan*、Alois Knoll*
* 机器人学、人工智能与实时系统教席慕尼黑工业大学，慕尼黑，德国 电子邮箱：{nenad.petrovic、v.zolfaghari、f.pan、k}@tum.de

arXiv:2601.02215v1 [cs.SE] 5 Jan 2026

摘要—本文介绍了一种由大语言模型赋能的工作流，用于支持软件定义汽车（SDV）的软件开发，涵盖安全感知的系统拓扑设计，以及事件驱动的决策代码分析。对于代码分析，我们采用事件链模型，为功能安全的系统化验证提供形式化基础，同时考虑关键组件之间交换的消息的语义有效性，包括 CAN 和车辆信号规范（VSS）。拓扑的安全性分析依托于与模型驱动工程（MDE）方法和对象约束语言（OCL）规则的协同。评估过程中同时考虑了可本地部署的解决方案和专有解决方案，并将其应用于与高级驾驶辅助系统（ADAS）相关的场景。

索引词—**网络安全、功能安全、事件链、大语言模型（LLM）、软件定义汽车。**

I. 引言

在汽车创新领域，快速原型开发、验证和部署先进的车辆功能，同时保障功能安全和网络安全至关重要；对于日益复杂的软件定义汽车（SDV）而言尤其如此。传统工作流程依赖人工编写的规范、状态机和专用建模工具，能够提供符合 ISO 26262 和 ISO 21434 要求的可追溯性。然而，这些流程需要大量专家介入，而且随着系统复杂性的提高，越来越难以扩展，尤其是在集成不同传感器模态、分布式电子控制单元和可变时序行为时 [1]。

大语言模型（LLM）为加速这些流程带来了新机遇，可通过提取自然语言需求中的信息来生成初步行为模型。然而，若使用不当，大语言模型无法保证正确性、故障安全行为、时序确定性和安全态势。直接由大语言模型生成的逻辑可能会引入虚构的信号、违反架构接口，或忽视对危害分析（HARA）、故障传播建模或威胁分析与风险评估（TARA）至关重要的因果约束。

本研究获得了欧洲芯片联合事业根据第 101139789 号框架合作协议（HAL4SDV）提供的资助，其中包括德国联邦研究、技术与空间部根据资助编号 16MEE00471K 提供的国家资助。本出版物内容由作者负责。

为弥补这些不足，我们提出了一种基于事件链、由大语言模型引导的工作流程，专门用于支持功能安全和安全分析。事件链以结构化方式表示汽车系统中安全机制和时序要求所依托的因果关系、时间关系及数据流关系。通过将事件链置于大语言模型工作流程的核心，我们对安全论证所必需的约束加以强制执行：允许的信号流、安全相关步骤的顺序、端到端延迟预算，以及安全关键元素与非关键元素的分离 [2] [3] [4]。因此，事件链充当形式化锚点，对大语言模型的输出加以约束，并防止不安全或未经验证的行为进入生成制品。

在此框架中，大语言模型可协助自动完成通常需要领域专家知识的任务：根据文本需求推导候选安全机制，使功能意图与车辆信号规范（VSS）保持一致，识别潜在危险或不安全交互，并生成诊断或缓解机制的初稿。检索增强生成（RAG）进一步将大语言模型的推理范围限制在可信数据源内，例如经批准的 VSS 目录、安全手册和架构描述，从而降低虚构信号或未经评估的数据路径出现的可能性。这确保自动生成的模型能够追溯至安全与网络安全审计所要求的权威来源。

事件链模型与大语言模型的协同作用，可实现对功能安全（例如违反时间预算、因果链断裂、缺少安全响应）和网络安全（例如未经授权的信号路径、信任边界不一致或意外的控制影响）的自动化预分析。在任何代码部署到目标平台之前，工作流程都会执行与 ISO 26262 第6部分（软件架构约束） [5] 和 ISO 21434（安全的信号与接口处理） [6]相符的一致性检查。因此，大语言模型生成的制品不仅更加可靠，也能直接融入既有的安全工程工作流程。

总体而言，所提出的流程有助于实现汽车代码安全与安保分析流程的自动化，从而减少耗时。这旨在减少早期设计阶段的工作量，同时保留满足现代安全与网络安全需求所必需的严谨性。作为

演示，我们将该工作流程应用于基于我们的物理测试台 [7]和仿真平台的高级驾驶辅助系统场景。

II. 背景与相关工作

大语言模型（LLM）正日益融入自动驾驶汽车系统的开发生命周期，以应对安全与安保方面的双重问题。从安全角度来看，GPT-4 和 CodeLlama 等模型被用于自适应巡航控制（ACC）等功能的代码协同生成，并通过自动化评估流水线进行评估；这些流水线采用与模拟相集成的软件在环（SIL）环境，从而实现快速的初步验证。在安全防护方面，安全测试自动化框架（STAF）等专用框架将 GPT-4.1 和 DeepSeek-V3 等大语言模型集成到具有自我纠错能力的检索增强生成（RAG）架构中，以便根据攻击树自动推导出可执行的安全测试用例。此外，OpenAI 的 Codex 等大语言模型还用于生成形式化安全断言，包括针对交通信号控制器和 AES 加密模块等底层自动驾驶汽车硬件组件的 SystemVerilog 断言（SVA）；在提供全面设计上下文时，其成功率明显更高。与此相对，HackerGPT 等定制模型能够自动生成针对虚拟控制器局域网（CAN）、蓝牙接口和遥控钥匙系统的网络攻击脚本，这凸显了大语言模型的双重用途特性，并表明亟须采取稳健且积极主动的汽车安全措施。

表 I 基于大语言模型的汽车安全解决方案总结

参考文献	模型	用例
Nouri 等人 [8]	CodeLlama, CodeGemma, DeepSeek-r1, DeepSeek-编码人员, Mistral, GPT-4	Code 协同生成 and 初步评估流程 用于安全相关 高级驾驶辅助系统/自动驾驶 系统。评估 基于四个 汽车 功能， 包括 自适应巡航控制（ACC） and 碰撞 避免 by 规避机动（CAEM）
Khule等人 [9]	GPT-4.1, DeepSeek-V3	自动 执行动作的生成 可执行的安全测试用例（Python 脚本）和线性时序逻辑 （LTL）攻击树属性 用于汽车安全测试，采用 安全测试自动化 框架（STAF）。
Louati 等人 [10]	OpenAI Codex (code-davinci-002)	形式化 安全断言（SystemVerilog 断言），用于自动驾驶车辆 CLE 硬件子系统，包括 交通信号控制器、AES 加密模块和寄存器 访问控制机制
Usman 等人 [11]	HackerGPT（微调 经过调优的基于 GPT 的 LLM）	模拟 and 执行 of 针对车辆系统的网络攻击 通过 自动化 生成 of 攻击 脚本 针对 虚拟 CAN（vCAN）、蓝牙 子系统和远程无钥匙 进入（RKE）/遥控钥匙系统。

另一方面，本文聚焦于设计阶段的安全与安保，旨在防止与高级驾驶辅助系统相关的代码在目标平台上执行之前产生恶意行为。当检测到威胁或不安全行为风险时，我们的工作流还会提出纠正措施，以消除这些风险。这样，我们的目标是最大限度地降低恶意代码对环境造成损害的潜在风险。该方法建立在我们此前多项工作中观察到的良好实践之上。在 [12], 中，我们介绍了一种集成工具链：该工具链从文本需求出发，依托大语言模型，并结合使用Eclipse建模框架（Eclipse 建模框架，EMF）、Ecore和对象约束语言（OCL）的模型驱动方法，生成Python格式的高级驾驶辅助系统模拟代码。研究重点是车辆配置（传感器和执行器）以及环境方面（车辆、行人和障碍物的布置）。有关各个模型驱动工程相关步骤实现的更多细节，请参阅我们的其他论文：元建模 [13], 、实例模型 [14]以及约束生成 [15]。另一方面，在 [16], 中，我们介绍了一种面向软件定义汽车区域ECU、在物理测试台中进行大语言模型辅助C++代码生成的工作流，并利用事件链验证车辆行为。两种解决方案均依赖现有的 n8n [17] 工作流框架，以实现工具链集成。此外，在 [18] 和 [19],中，我们展示了如何利用大语言模型和视觉语言模型（VLM）驱动的VSS接口规范提取，以减少软件定义汽车代码和测试中幻觉的发生。

最后，在 [20] 中，我们采用基于检索增强生成（RAG）的检索与重排序方法，以实现基于大语言模型的问答，并依托选定的汽车标准化文档。在本文中，我们利用了一种基于 [16] 的检索与重排序流水线，其中Se ntenceTransformer模型（all-MiniLM-L6-v2） [21] 对场景查询和VSS/CAN 条目进行编码，并依据嵌入相似度检索语义相似度最高的前k个信号。随后，使用交叉编码器（ ms-marco-MiniLM-L-6-v2） [22]对这些候选项进行重排序，之后将排名最高的条目分块，以适应大语言模型输入约束。

III. 实现概述

A. 代码分析工作流

所提出的方法将检索增强大语言模型与模型驱动工程（MDE）相结合，以支持复杂软件定义汽车系统在早期开发阶段的安全可靠的系统设计。其目标是在部署前自动提取相关信号、推理可能存在的危险或不安全事件链，并使用形式化设计规则对其进行验证。图 1 展示了所提出的工作流，其中每个步骤对应一个或多个编号元素，后文将对此进行说明。

输入准备（*1a/1b*）：车辆信号目录（VSS）和 CAN 消息目录（1a、1b）是权威且结构化的信息来源（信号名称、取值范围、消息标识符、语义）。它们会被传递给相应的目录解析器，后者生成仅包含相关信息、适用于后续安全和功能安全分析的机器可读信号/消息表示——

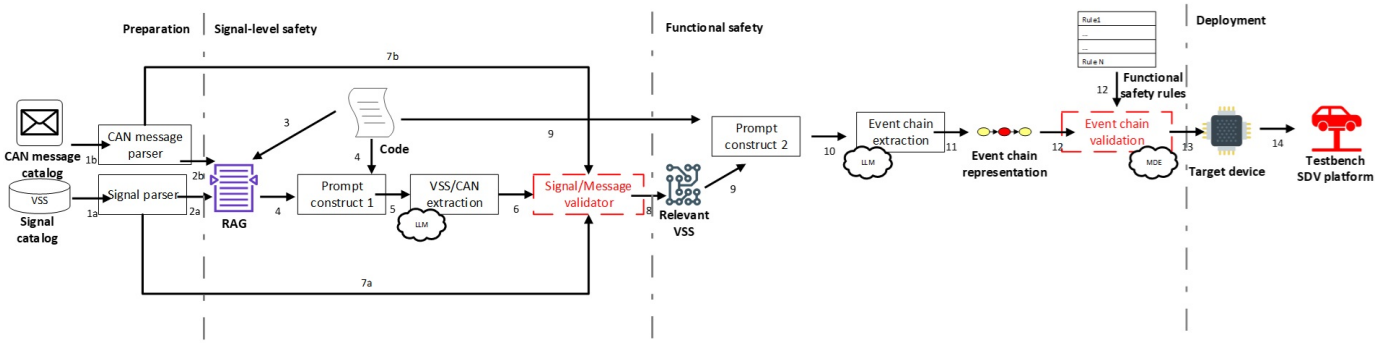

图 1. 面向汽车的、由大语言模型赋能的基于事件链的功能安全设计工作流 .

形式。这两个目录并非互斥使用，因为在软件定义汽车系统的不同部分中运行的某些程序可能依赖不同抽象级别的信息。对于较高抽象级别（软件定义汽车中间件的常见情况，例如 commAPI），通常会使用基于 VSS 的信号表示，将信号作为复杂对象，并配有相应的处理程序（1a）。另一方面，在涉及较低级别的脚本时，也可能直接在程序中使用或引用 CAN 消息（1b）。

基于检索增强生成的上下文构建（*2a/2b，3*）：在实践中，VSS 目录通常包含数千个条目；此外，系统中可能会使用数百种不同类型的 CAN 消息及其变体，这会超出大语言模型的上下文容量，并可能导致巨大的令牌消耗，从而使流程成本高昂。因此，我们采用检索增强生成层对目录和先前的产物进行索引，使大语言模型进一步生成的结果始终以具体且最新的数据为依据，同时最大限度地减少幻觉。检索增强生成会根据即将由我们的工作流分析的软件定义汽车组件配置源代码，检索相关的目录上下文。

VSS/CAN 消息提取（*4-8*）：提示构造 1 旨在从代码/配置中提取信号/消息。VSS/CAN 提取会生成一组精简的信号/消息，这些信号/消息与当前安全分析相关。为此，除了信号/消息验证器（基于规则的检查器或形式化检查器）之外，还会验证提取的信号/消息是否确实存在于 VSS/CAN 目录中，并且是否符合预期格式和取值范围，然后才会进入后续阶段。这样可以防止运行包含潜在恶意信号值和消息值的代码与配置。经过验证的信号集（相关 VSS）将作为事件级推理的输入，以进一步进行功能安全验证。

提示词构建1 – VSS/CAN报文提取。

您正在根据给定的源代码提取 VSS信号和CAN消息 {代码}。
对于每个信号/消息步骤，

提取条目：名称、类型、值、协议。

事件链构建（*9-11*）：另一方面，提示构造2旨在生成基于代码的事件链表示，将预期效果表示为事件序列，从而轻松识别不安全或危险的结果。大语言模型（结合检索增强生成上下文）会输出结构化的事件链（例如，“CAN报文X被伪造 →执行器命令发生变化 →车辆状态进入危险状态”）。该方法基于我们先前的工作 [16] 事件链提取模块将大语言模型文本转换为机器可读的事件链表示，而在我们的案例中，我们采用带有一些扩展的简易PlantUML活动图表示法。我们先前的工作 [13], 表明，与Eclipse建模框架的Ecore等其他方案相比，大语言模型更新PlantUML [24] 表示法时更不容易出错。

提示构造2 – 基于给定代码的事件链生成。

你正在更新汽车事件链的PlantUML活动图，不添加注释或说明，给定内容如下{当前事件链}，并基于以下源代码： {代码}。，同时考虑 {相关消息/信号}。对于每个事件链步骤，以下参数将作为备注考虑：输入、输入_格式、输出、输出_格式。

功能安全验证 *(11-12)*：在 MDE 环境中，事件链以形式化模型（例如状态机、时序图、SysML/UML 或特定领域模型）表示。在我们的案例中，我们使用 PlantUML 活动图，并通过模型到模型转换将其转换为 JSON 格式，以便进行规则检查。基于此 JSON 表示，我们验证事件在链中是否按正确顺序出现，以确保功能安全。对于以 e_1 **之前** e_2 指定的规则，事件 e_1 必须先于事件 e_2 发生。使用之后关系的规则则反之，其中要求 e_1

在 e_2之后发生。”功能/安全规则（Rule1 ... RuleN）以约束的形式存在，而 MDE 工具会根据这些规则验证事件链。我们假设这些规则基于安全参考资料和良好实践，可由人工定义，也可借助大语言模型制定（类似于我们此前的工作）。规则使用示例：检查摄像头检测到行人时车辆是否不会加速，因此，在受检事件链中，“行人之后加速”这条规则不应成立。

部署与测试（*13-14*）：经验证的制品（软件定义汽车模块的代码、测试脚本）将部署到目标设备，并在测试平台/软件定义汽车平台上执行。此过程会执行假设的事件链，并检查具体的安全/安保后果。

反馈循环（*7a/7b* 和 *12* 循环）：如果验证失败，反馈会沿红色虚线标示的步骤返回给用户。这一迭代循环弥合了大语言模型假设与经过正式验证、可测试制品之间的差距。可以由用户手动采取纠正措施，也可以借助大语言模型，同时考虑安全分析的结果以及代码本身。如果采用自动纠正方式，我们会构造另一个提示并将其提交给大语言模型执行，称为提示构造 3。

提示构造 2b – 自动纠正。

基于代码分析结果 {结果}，纠正以下代码 {代码} ，以消除检测到的功能安全相关问题。

部署阶段（步骤 *13-14*）：生成的代码将部署到目标仿真环境（例如开放可用的 digital.auto 和 CARLA）或嵌入式设备（例如汽车电子控制单元或区域控制器），既可以直接部署，也可以通过空中下载部署。

B. 系统拓扑安全分析

该工作流整合用户输入、元模型、安全指南、大语言模型辅助的模型构建以及基于模型驱动工程的设计分析，以自动判断所提出的汽车架构是否满足通信相关安全要求。它以我们此前的工作为基础[14]，[15]，[13]。通过这种方式，我们提供了一个协同的半自动化汽车安全分析环境，能够在进一步实施之前验证复杂的系统设计。

系统建模输入：（步骤 *1*）：用户提供与系统拓扑相关的描述，包括系统组件、通信关系、预期用于通信的底层协议，以及其他相关的架构假设。提示构造 3 基于这些输入构建，并将其转换为适用于大语言模型的结构化提示。

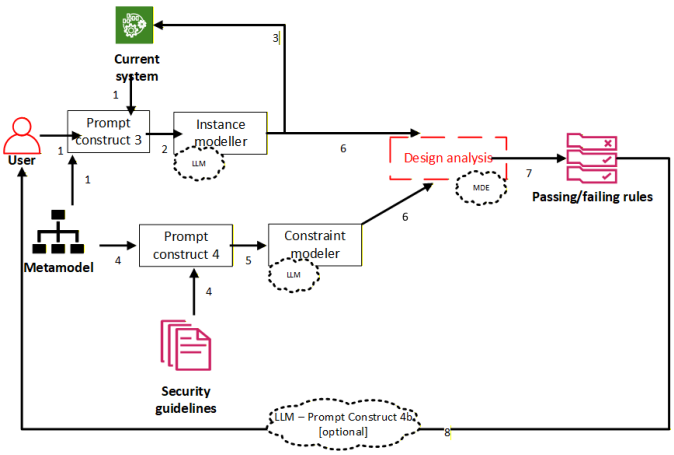

图 2. 大语言模型驱动的拓扑级安全内建工作流。

提示构造 3 – 系统模型创建/更新。

更新模型实例 {当前系统}，依据 {元模型}，基于需求{用户输入}。

. 大语言模型辅助的实例模型生成（步骤 *2*）：大语言模型接收结构化提示并生成实例模型，即基于预定义元模型对汽车系统进行的具体表示，其中包含有关电子控制单元、传感器、总线、通道和协议的信息。此步骤以 [14]和 [13]为基础。

当前系统反馈（步骤 *3*）：向用户展示使用 PlantUML 类图表示的当前系统表示，以便将新增功能或对现有功能的改进纳入考量，并通过返回步骤 1 和 2 迭代更新模型。

安全指南集成（步骤 *3–5*）：文本形式的安全指南（例如安全通信约束、消息格式，以及特定车辆组件之间允许使用的协议）会借助提示构造 4，按照给定的元模型转换为 OCL 格式的形式化规则。这项工作基于我们此前的研究 [15]。

提示构造 4 – 系统模型约束生成。

根据 {元模型}，基于参考规范 {安全指南}生成汽车系统安全约束。

.

模型驱动的安全感知设计分析（步骤 *6-8*）：将先前生成的符合 EMF 标准的系统模型与以 OCL规则 表示的安全指南进行检查。在此之前，使用模型到模型转换，将先前生成的 PlantUML 模型转换为 EMF Ecore 格式，如 [13] 所述。分析结果显示，

系统会告知用户，根据所提供的设计是否发现了安全缺陷，并重点指出未通过的规则。这些规则既可由机器读取，也便于人类理解，因此用户可以采取纠正措施——更新系统架构，并据此重新生成模型、重新运行分析，直至满足所有与通信相关的安全约束。根据提示构造 4b 的定义，大语言模型也可以选择根据分析结果对系统模型实例提出修正建议。

提示构造 4b – 系统模型修正。

根据{元模型}，基于当前表示{当前系统} 和分析结果 {对象约束语言通过/失败列表}，更新汽车系统模型。

.

IV. 案例研究概述

A. 功能安全分析

为了展示功能安全方面的概念验证，我们基于旨在在摄像头或激光雷达检测到行人时执行紧急制动的高级驾驶辅助系统，设计了多个实验场景。每个场景都从车辆决策模块的角度进行考察，并且都存在应由我们提出的大语言模型赋能框架检测出的功能缺陷。

- 场景 1：检测到行人时，车辆加速而不是制动。

- 场景2：当激光雷达检测到行人时，车辆停止，但传感仅使用摄像头。

- 场景3：车辆在检测到行人之前停止。

下文给出了基于 PlantUML 的活动图，用于说明图 3所示的给定场景。对于每个场景，图中影响功能安全的关键部分均以红色标记。需要强调的是，为了便于说明和提高可读性，我们在概念层面展示了简化表示，图中本身并未直接包含准确的 CAN-FD报文或完整的 VSS信号名称。

在第一个场景中，风险在于车辆将撞上行人，因为车辆在检测到行人时不会停止，反而会加速。对于第二个场景，车辆根据系统其余部分提供的激光雷达信息作出制动决策，而系统中实际存在的仅是基于摄像头的检测。因此，如果确实出现行人，车辆很可能不会停止。在第三个场景中，制动发生在决策之前，导致车辆过早停止。

针对场景中每个事件链所检查的底层规则如表 II 所示。

B. 拓扑分析

假设用户指定了组件级连接需求：“仿真计算机通过以太网连接到车载高性能计算机。此外，

表 II 功能安全场景示例

场景	规则	描述
1	加速 之前 摄像头检测到行人 并在之前加速 激光雷达检测到行人	加速度 指令 应仅在 .....之前 行人被 激光雷达或摄像头检测到。
2	（制动器 之后 摄像头-行人 检测到 *and* 摄像头感知 之前 摄像头检测到的行人-检测到）或（制动器之后 激光雷达行人 检测到 *and* 激光雷达感知 之前 激光雷达检测到行人）	制动 发生 要么 作为摄像头或 基于激光雷达的检测结果，而对于其中每一项，都必须先完成这两项工作 传感 和分析，然后再 触发执行器。
3	制动器 之后 摄像头 检测到行人	制动 仅在行人被检测到之后才会发生 摄像头。

车载高性能计算机通过以太网连接到区域ECU，并向其发送符合 IEEE 1722 标准的消息。另一方面，高数据速率传感器（例如低分辨率摄像头、激光雷达）通过以太网连接到区域ECU，而简单传感器和执行器则通过 CAN-FD 连接。高分辨率前后摄像头也通过以太网连接到高性能计算机。此外，应支持从 ZoneECU 到所连接设备的 VSS消息级通信，并且通信内容应包括状态信息、执行器命令以及传感信息。”底层元模型如图 4 所示。接下来给出用于封装设计级安全主要约束的 OCL规则示例。

*SteeringCommandWithinLimits*规则确保发送给转向执行器的任何消息都包含处于 -15 至 +15 度安全范围内的转向角命令，从而在通信层面防止不安全的转向命令，因为角度过大可能会损坏物理测试台。

HPCtoZoneEther-netIEEE1722 规则强制执行架构约束：从高性能计算机发送到区域ECU的所有消息都必须通过以太网传输，并符合 IEEE 1722 标准，以确保正确使用骨干网络。*TargetSpeedWithinSafetyLimit* 规则保证定义目标车速的 VSS消息不会超过 30 km/h，无论发送方或网络为何，都对速度命令施加全局安全限制。

拓扑级安全分析 OCL规则。

上下文 消息 不变量 SteeringCommandWithinLimits: self.target.oclIsTypeOf(SteeringActuator) implies let angle : 实数 =self.payloadValue.toReal() in angle >= -15.0 and angle <= 15.0

上下文 消息 不变量 HPCtoZoneEthernetIEEE1722: self.source.oclIsTypeOf(高性能计算机) and

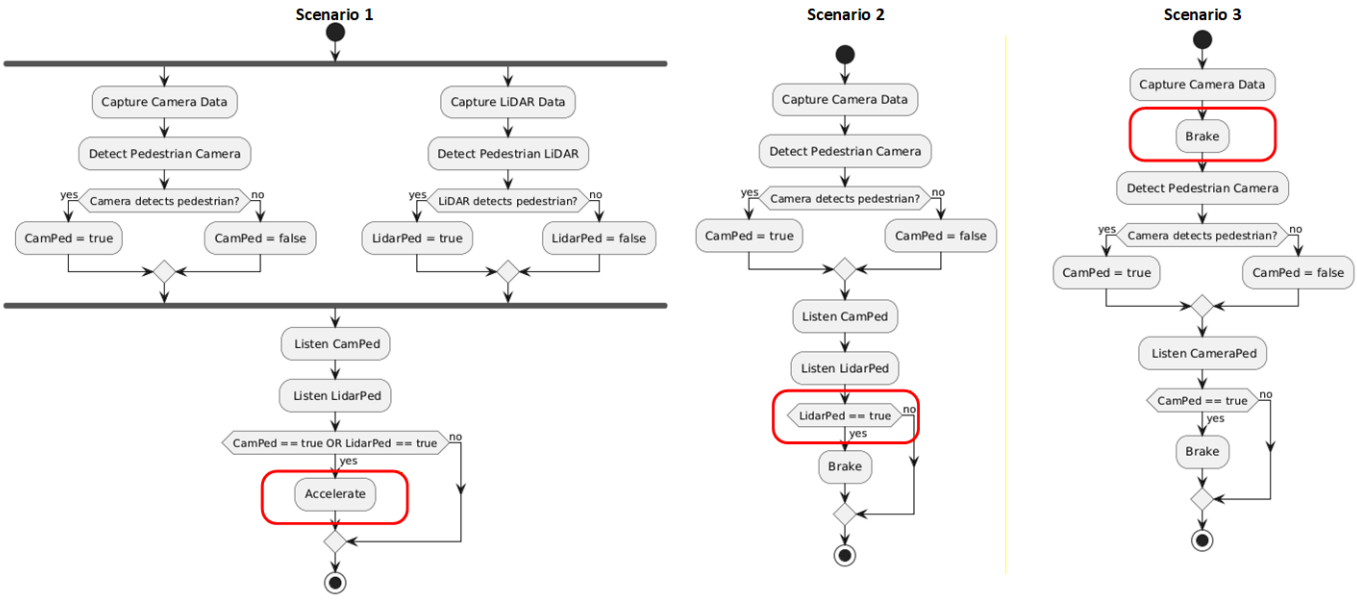

图3. 活动图表示法中的功能安全场景概览。

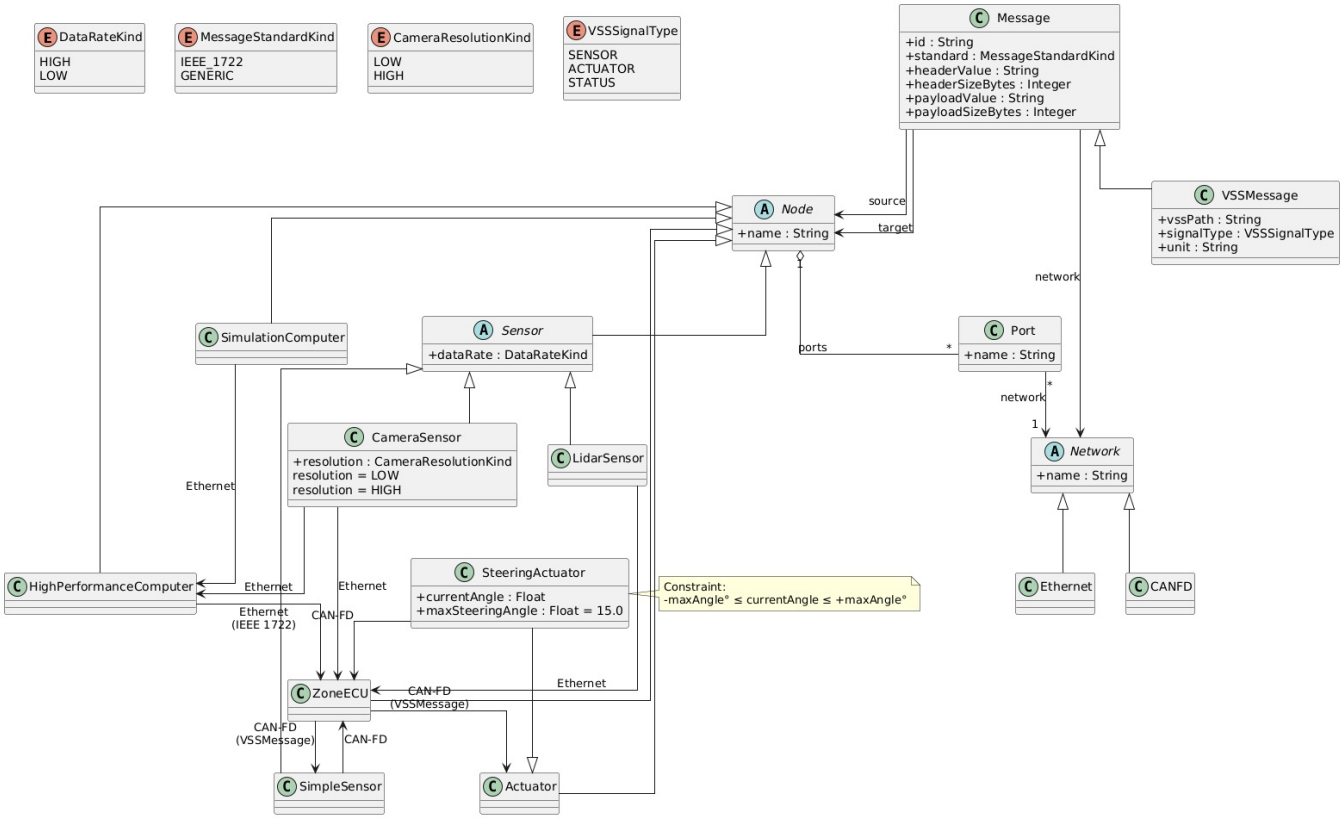

图4.　安全感知拓扑分析案例研究背后的元模型　　例研究。

self.target.oclIsTypeOf(ZoneECU) 表示 self.network.oclIsTypeOf(以太网) 且 self.standard = 消息标准类型::IEEE-1722

上下文 VSS消息 不变量 TargetSpeedWithinSafetyLimit:
self.vssPath = 'Vehicle.Speed.Target' 表示
sSelf.payloadValue.toReal() <= 30.0

.

V. 实验与评估

鉴于与SDV相关的需求和源代码通常受到严格的知识产权保护法规和限制，采用可在本地部署的基于大语言模型的解决方案可能极具价值。在本文中，关于方法评估，我们比较了两种广泛采用的模型的有效性：商业版 OpenAI 的 GPT-5 [25]与可在本地部署的 Meta 的 LLaMA-3.3-70B-指令版 [26]。结合我们此前针对需求处理和建模开展的工作，并假设用于问答的基于检索增强生成的系统固定不变，本文所介绍的实验将重点关注功能安全和安全分析中至关重要、但未在我们以往发表成果中涵盖的互补方面——基于给定软件制品的 VSS/CAN信号映射，以及基于事件链的功能安全验证。经评估的由大语言模型赋能的集成工具链同样基于 n8n[17], , 与我们在 [12] 和 [16]中的工作类似。

对于所分析的源代码，我们涵盖了两种基于 Python 的执行环境：1) 与图形化 CARLA 自动驾驶模拟器集成的测试平台 [7] 以及 2) digital.auto 软件定义汽车仿真平台 [23]。表 III 涵盖了三类实验：1) CAN-FD映射——根据检索增强生成生成的候选信号筛选结果，从源代码中识别 CAN消息及其取值（有效载荷、报头）；2) VSS映射——与 1) 类似，但针对 VSS信号对象；3) 基于事件链的功能安全——根据 Python 源代码构建事件链，以便进一步进行分析，同时考虑图3中此前说明的场景。对于 1) 和 2)，我们进一步考虑了两种实验变体：假设由基于检索增强生成的系统生成的筛选信号数量较少或较多；本文不对检索增强生成系统本身进行评估。表 III 的第一列表示前文所述的实验类别。第二列给出了使用 GPT-5模型时观察到的结果。第三列则列出使用可本地部署的 LLaMA模型时各项实验的观察结果。最后一列汇总了特定实验的模型性能，以相同条件下 10 次运行中成功完成的百分比表示。对于 VSS 和 CAN-FD映射，只有在识别出所有正确的信号及其对应取值时，才视为成功完成。对于事件链分析，成功完成意味着所构建的事件链在语法和语义上均

表 III 大语言模型在功能安全场景中的评估

方面	GPT-5	LLaMA 3.3 70B 指令	性能
CAN-FD mapping	正确地识别 CAN-FD 信号和 maps 执行器和传感器事件到消息：值对 with 最小值 跌义。	识别 most CAN-FD信号正确地，but 偶尔产生冗余的 or 宽松地 范围限定的消息:值映射。	GPT (20): 90% LLaMA (20): 80% GPT (50): 90% LLaMA（50）: 70%
VSS 映射（digital.auto）	准确映射感知, 传感, and 驱动 事件 to 标准化 VSS 路径（例如, Vehicle.Speed, ADAS.Brake)。	提供部分 VSS 映射; some 领域特定的 信号需要 手动校正或规范化。	GPT（20）: 100% LLaMA (20): 100% GPT (50): 90% LLaMA (50): 90%
事件链分析（digital.auto）	始终如一地重构正确的因果链（感知 → 检测 → 决策 → 执行) 与.....保持一致功能安全规则。	捕获 main 事件排序但可能会遗漏复杂规则中的前置条件复杂规则。	GPT: S1-90%, S2-90%, S3-100% LLaMA: S1-70%, S2-80%, S3-80%

正确，因此根据每个场景（标记为 S1-S3）的预定义规则识别出了功能安全缺陷。对于两个模型，每个提示的令牌限制均设为 4096，并使用默认温度值。执行本地可部署模式时，我们使用配备 80GB 显存的 NVIDIA A100 GPU。

VI. 结论

本文提出了一种将大语言模型与模型驱动工程相结合，用于软件定义汽车安全和 Python代码功能安全分析的方法，并针对商业模型和本地可部署模型在基础高级驾驶辅助系统功能中的应用进行了评估。根据所得结果可以发现，两种模型在 VSS信号提取方面的性能相近，而商业模型在处理 CAN消息方面略胜一筹。这可以归因于 VSS表示法采用结构化格式且信号层级明确，因此更易于解读；而要理解 CAN-FD 格式消息中的隐含含义，则需要表达能力更强的模型。在考虑较长的候选信号列表时，模型性能会下降，而 GPT 在这种情况下略有优势。不过，此时最终结果仍主要受检索增强生成系统生成的所选信号列表影响。另一方面，在基于事件链的代码分析方面，商业方案略胜一筹；两种模型在令牌消耗方面则没有显著差异。

根据结果，较大的可本地部署模型（例如本文评估的模型）即使不进行微调，也能够以令人满意的性能支持软件定义车辆安全和安全相关任务，

前提是提供上下文信息（例如相关信号和消息）。因此，随着可本地部署模型能力不断增强，预计基于大语言模型的工具将在汽车安全和功能安全领域占据主导地位，因为这类模型能够限制潜在机密和受保护的软件资产的流动。

未来，我们计划将所提出方案的应用扩展到 Rust 编程语言的软件定义汽车生态系统中，这正成为该领域一个颇具前景的发展方向。由于 Rust 编程语言从设计上确保了内存安全，我们将比较这种方法与 C++ 和 Python 在安全与可靠性分析方面的整体效果和复杂度，因为基于编程语言相关的设计决策，这种方法有望简化此类活动。此外，我们还计划更详细地评估检索增强生成系统的替代方案、各种参数配置，以及本地可部署模型微调的影响。

参考文献

[1] N. Petrovic 等，《汽车软件开发中的生成式人工智能综述：从需求到可执行代码》，2025年第二届国际生成式人工智能与计算语言建模会议（GACLM），西班牙瓦伦西亚，2025年，第184–197页，doi: 10.1109/GA- CLM67198.2025.11232000。[2] P. M¨unzenberger 和 O. Heckmann，《使用流图对分布式实时系统进行时间约束仿真》，2021年。在线。获取地址：https://www.inchron.com/wp-content/uploads/2021/12/ESE21 Muenzenberger Heckmann.pdf。[访问日期：11月 21, 2025]。[3] INCHRON¯ GmbH，《白皮书——流图与时序》，2022年。 [在线]。获取地址：https://www.inchron.com/wp-content/uploads/2022/08/0020 219 Whitepaper-Flow-Diagrams-And-Timing DD1.pdf。 [访问日期：11月 20, 2025]。[4] S. Dingler、P. Rehkop、F. ¯Mayer 和 R. M¨unzenberger，《面向自动驾驶与高级驾驶辅助系统的事件链分析：确保安全并满足法规时序要求》，2025年。 [在线]。获取地址：https://arxiv.org/pdf/2511.18092。 [访问日期：12月 18, 2025]。[5] ISO，《ISO 26262：道路车辆——功能安全》，2018年。 [在线]。获取地址：https://www.iso.org/standard/68383.html。 [访问日期：12月 18, 2025]。[6] ISO/SAE，《ISO/SAE 21434：道路车辆——网络安全工程》，2021年。[在线]。获取地址：https://www.iso.org/standard/70918.html。 [访问日期：12月 18, 2025]。[7] C. Wu 等，《ViL-TUM：一种基于测试平台的车辆在环测试方法，用于中央汽车服务器验证》，DTF 研讨会，2025年。[8] A. Nouri、B. Cabrero-Daniel、Z. Fei、K. Ronanki、H. Sivencrona 和 C. Berger，《用于安全自动驾驶汽车的代码协同生成中的大语言模型》，预印本，即将发表于国际计算机安全、可靠性与安全保障会议（*SafeComp*）论文集，2025年。[9] T. Khule、S. Marksteiner、J. Alguindigue、H. Fuchs、S. Fischmeister 和 A. Narayan，《STAF：利用大语言模型自动生成基于攻击树的安全测试》，2025年。[10] A. Louati、H. Louati 和 E. Kariri，《面向智慧城市系统中自动驾驶汽车的大语言模型引导安全声明生成》，*IEEE Access*，2025年11月20日录用，doi: 10.1109/ACCESS.2025.3636576。[11] Y. Usman、P. K. Gyawali、S. Gyawali 和 R. Chataut，《人工智能的阴暗面：大语言模型作为针对车辆系统发起网络攻击的工具》，2025年。[12] N. Petrovic、F. Pan、V. Zolfaghari、K. Lebioda、A. Schamschurko 和 A. Knoll，《面向汽车软件开发的生成式人工智能：从需求到车轮》，arXiv 预印本 arXiv:2507.18223，2025年。[13] N. Petrovic、F. Pan、V. Zolfaghari 和 A. Knoll，《汽车领域基于大语言模型的元建模迭代方法》，载于*2025*年第二届国际生成式人工智能与计算语言建模会议（*GACLM*），西班牙瓦伦西亚，2025年，第266–271页，doi: 10.1109/GA- CLM67198.2025.11232159

[14] F. Pan、N. Petrovic、V. Zolfaghari、L. Wen 和 A. Knoll，《大语言模型赋能的实例模型生成》，载于*2025*年美国计算机协会/电气电子工程师学会第*28*届模型驱动工程语言与系统国际会议伴会（*MODELS-C*），美国密歇根州大急流城，2025年，第586–595页，doi: 10.1109/MODELS-C68889.2025.00082。[15] F. Pan、V. Zolfaghari、L. Wen、N. Petrovic、J. Lin 和 A. Knoll，《用于对象约束语言约束生成的生成式人工智能：数据集收集与大语言模型微调》，载于*2024*年电气电子工程师学会系统工程国际研讨会（*ISSE*），意大利佩鲁贾，2024年，第1–8页，doi: 10.1109/ISSE63315.2024.10741141。[16] N. Petrovic、N. Kroth、A. Torschmied、Y. Song、F. Pan、V. Zolfaghari、N. Purschke、S. Kirchner、C. Wu、A. Schamschurko、Y. Zhang 和 A. Knoll，《软件定义汽车系统中高级驾驶辅助系统的事件链驱动代码生成：大语言模型赋能》，技术报告，*TUM mediaTUM*，2025年。 [在线]。获取地址：https://mediatum.ub.tum.de/1086674?show id=1837104。[17] n8n，《n8n÷工作流自动化平台》，获取地址：https://n8n.io/，访问日期：2025年11月。[18] M. Pavel、N. Petrovic、Ł. Mazur、V. Zolfaghari、F. Pan 和 A. Knoll，《基于大语言模型的代码生成中的幻觉：汽车案例研究》，arXiv 预印本 arXiv:2508.11257，2025年。[19] D. Zyberaj、L. Mazur、N. Petrovic、P. Verma、P. Hirmer、D. Slama、X. Cheng 和 A. Knoll，《软件定义汽车平台中基于生成式人工智能的测试用例生成与执行》，arXiv 预印本 arXiv:2509.05112，2025年。[20] V. Zolfaghari、N. Petrovic、F. Pan、K. Lebioda 和 A. Knoll，《将检索增强生成用于大语言模型辅助的未来车辆设计》，载于第二届基础模型与大语言模型国际会议论文集（*FLLM*），阿拉伯联合酋长国迪拜，2024年，第437–442页，doi: 10.1109/FLLM63129.2024.10852467。[21] sentence-transformers，《all-MiniLM-L6-v2》，获取地址：https: // huggingface.co/sentence-transformers/all-MiniLM-L6-v2，访问日期：2025年11月。[22] sentence-transformers，《ms-marco-MiniLM-L-6-v2》，获取[地址：https:// huggingface.co/sentence-transformers/ms-marco-MiniLM-L-6-v2，访问日期：2025年11月。[23] digital.auto联盟，《digital.auto：软件定义汽车开发者生态系统》， [在线]。获取地址：https://digital.auto/ [24] PlantUML，《活动图测试版语法》，2025年。 [在线]。获取地址：https://plantuml.com/activity-diagram-beta。[25] OpenAI，《GPT-5》，2025年。 [在线]。获取地址：https://openai.com/gpt-5。访问日期：2025年12月20日。[26] Meta AI，《LLaMA 3.3 70B Instruct》，2024年。 [在线]。获取地址：https:// huggingface.co/meta-llama/Llama-3 20, 2025 3 70. - B-Instruct。访问日期：12月。