

# 基于融合模型的车辆监控系统设计与实现

高雪 臧金环

(中汽信息科技(天津)有限公司,天津300000)

【欢迎引用】高雪,臧金环.基于融合模型的车辆监控系统设计与实现[J].汽车文摘,2023(6):57-62.

【Cite this paper】GAO X, ZANG J H. Design and Implementation of A Vehicle Monitoring System Based on Fusion Model[J]. Automotive Digest (Chinese), 2023(6): 57-62.

【摘要】随着大数据的发展,对于车辆的监控除了车辆信息、备案转发情况等业务功能需求外,还有对于车辆实时数据分析和统计的需求。为了解决该问题,提出了一种通过Java和Python整合技术,使业务平台和算法平台相结合的融合模型方案,通过命令行在Java代码中调用Python脚本,并对外提供程序接口(API)供业务平台调用,该方案使业务平台和算法平台之间实现了集成,同时均作为独立服务部署,相互解耦,更好地监控车辆信息、实时数据、故障信息和运行情况。系统设计与实现符合大数据时代汽车行业数字化转型需求,提出了汽车监控系统技术优化方案,并对融合模型的技术设计进行了详细阐述。该融合模型在车辆监控系统中的应用对于汽车行业智能化发展具有重要价值。

关键词:Java;Python;融合模型;解耦;车辆监控

中图分类号:TP311.1;U471.1 文献标识码:A DOI:10.19822/j.cnki.1671-6329.20220164

## Design and Implementation of A Vehicle Monitoring System Based on Fusion Model

Gao Xue, Zang Jinhuan

(China Auto Information Technology Co., Ltd., Tianjin 300000)

【Abstract】With the development of big data, in addition to the needs of business functions such as vehicle information and record forwarding, the monitoring of vehicles also requires analysis and statistics of real-time vehicle data. In order to solve this problem, a fusion model scheme is proposed that combines the business platform and the algorithm platform through the technologies of Java and Python integration. The Python script is called in the Java code through the command line, and an Application Program Interface (API) is provided externally for the business platform to call. The solution integrates the business platform and the algorithm platform, and at the same time, they are deployed as independent services, decoupled from each other, and better monitoring vehicle information, real-time data, fault information, and operating conditions. The design and implementation of the system are in line with the digital transformation needs of the vehicle industry in the era of big data. The technical optimization scheme of the vehicle monitoring system is proposed, and the technical design of the fusion model is elaborated in detail. The application of the fusion model in vehicle monitoring system is of great value to the intelligent development of vehicle industry.

Key words: Java, Python, Fusion Model, Decoupling, Vehicle monitoring

## 0 引言

近年来,大数据技术发展迅速,车辆运行数据采集和积累也达到了大数据的应用水平。在此背景下,汽车行业开始数字化转型,向人工智能迈进。汽车主机厂也开始借助大数据技术来研究车辆监控系统<sup>[1-8]</sup>。目前行业内对车辆的监控已经不满足于简单的数据监控,对大数据分析也有更高的要求,实现车

辆更智能化的全面监控是目前汽车行业的主流需求。

从技术层面来看,Java作为面向对象编程语言的代表,已经是应用系统的主流开发语言<sup>[9-10]</sup>,目前已在互联网(WEB)应用开发领域得到广泛应用。Python语言在实现面向对象编程同时,还提供了高效的高级数据结构,具有丰富和强大的类型库<sup>[11-12]</sup>。结合2种编程语言各自优势,对2者进行整合开发,既满足了车辆监控系统设计功能要求,又提高了开发灵活性和开

发效率。但是经过调研发现,目前整车企业所应用的车辆监控系统还处于起步、不断更新阶段,有很多可优化空间。

本文提出了基于融合模型的车辆监控系统,实现了在车辆监控系统中集成业务平台与算法平台的功能,并且在运维、部署时互相解耦、互不影响,保证在满足需求的前提下,对融合模型的技术方案做了创新优化。融合模型的研究具有行业重要价值,实时数据和模型分析相结合才真正起到了车辆运行监控的目的和意义;营销数据和数据分析相结合的销量监控是汽车行业销售市场上提高竞争力的有力工具。

### 1 系统技术架构

车辆监控系统架构主要是由业务平台和算法平台融合而成。车辆运行数据由车载通信终端(Telematics Box, T-Box)实时上传,数据报文通过卡夫卡(Kafka)接入服务器,报文分为2路:

(1)与重型车排放远程监控平台(以下简称:国家平台)交互进行备案转发;

(2)解析入库到大数据集群中,抽取到关系型数据库和远程字典服务(Remote Dictionary Server, Redis)缓存中,供平台进行数据读取。

实时数据既做业务平台实时监控展示,又输入到算法平台中进行数据统计分析,并将算法模块嵌入到监控系统中,实现基于融合模型的车辆监控系统设计开发,系统总体技术架构如图1所示。

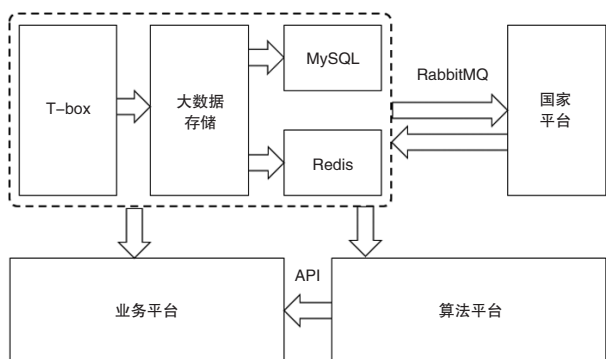


图1 系统总体技术架构

在车辆监控系统技术架构中用到了大数据存储、关系型数据库、缓存和中间件技术栈,在大数据存储部分依赖于Hadoop大数据架构,与国家平台交互时采用消息队列中间件(Rabbit Message Queue, RabbitMQ)。业务平台和算法平台部署成2个独立的服务,通过应用程序接口(Application Program Interface, API)进行通信。

图2所示为该系统的WEB端展示。菜单中分为数据质量监控、车辆异常监测、大数据分析、细分市场、用户行为和系统监控等共9项,每项数据可以展开分析,其中“用户行为”可以展开为11项分析数据。



图2 系统WEB端展示

### 2 车辆监控系统设计与实现

车辆监控系统设计与实现部分,从总体功能到分模块进行阐述,模块划分为2部分:系统业务平台和系统算法平台。

#### 2.1 系统总体功能

车辆监控系统是集数据采集、数据分析、数据计算、数据展示、数据监控和预警为一身的监控平台。系统总体功能分为业务模块和算法模块2大部分。系统总体功能架构如图3。

##### (1)业务模块

业务模块是由Java代码编写实现,主要功能为:对车辆总体情况的监控展示、数据监测、报表功能和系统管理。

##### (2)算法模块

算法模块是由Java嵌套Python代码来实现,对业务模块提供API调用,主要功能为:细分市场分析、加油点分析、电池安全预警、路况分析、载重分析等。

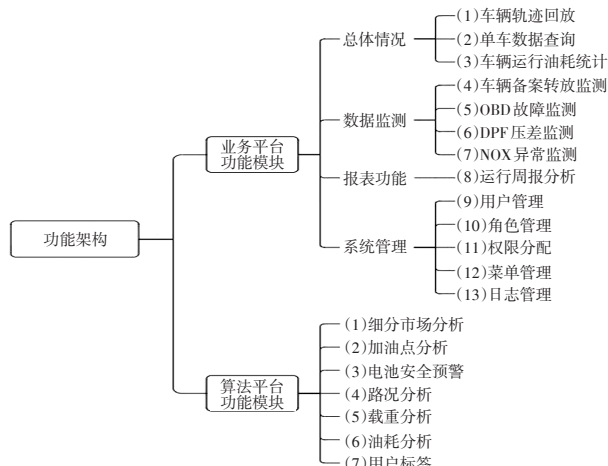


图3 系统总体功能架构

## 2.2 系统业务平台

业务平台采用Java编码实现。主流开发框架采用SpringBoot+VUE架构,前后端分离部署,关系型数据库采用MySQL,缓存采用Redis,消息队列采用RabbitMQ,中间件采用Kafka,大数据存储采用Hadoop集群技术栈,实现了对车辆数据监控与预警,系统与T-Box和国家平台的3方交互。技术架构如图4所示。

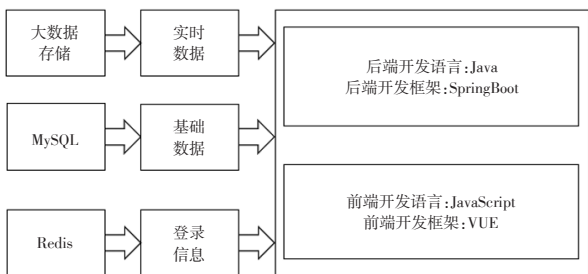


图4 业务平台技术架构

## 2.3 系统算法平台

算法平台采用Java集成Python脚本的编码实现,这样设计的目的是既能充分利用Python编码在大数据分析上的灵活性、扩展性,又能充分利用Java框架的外壳使其自成独立服务,并且能通过对外提供的API和业务平台进行融合,达到融合模型的效果。整合方案采用的技术方式为:Python代码作为文件夹放到Java框架的资源文件夹(Resource)下,通过代码中命令行进行调用执行Python脚本,技术架构如图5所示。

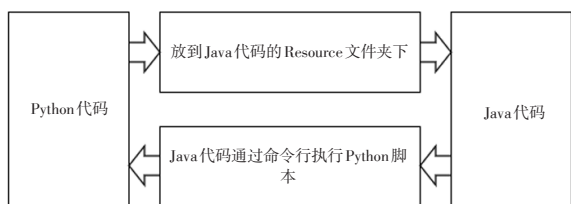


图5 算法平台技术架构

## 3 Java、Python 整合方案技术实现

针对本研究中采用的融合模型有2种开发语言:Java和Python。常用的融合方式主要有2种方案:

- (1) Jython(直接在Java中执行Python语句);
- (2) 通过命令行运行Python代码。

本研究采用的是第3种开发语言方案,即将Java调用Python的代码封装一下,对外提供API,然后与业务平台融合(方案(3))。

第3种封装Python代码算法方案和第2种方案的不同点在于,第3种方案算法部分封装了一层Java外壳,可以作为一个单独的算法平台进行独立部署,而

第2种方案是和业务平台代码融合在一起,耦合性更强<sup>[13-20]</sup>。

### 3.1 方案对比

将3种方案从需求满足、可维护性、技术实现的难易程度3个维度进行对比,如表1所示。

表1 方案对比

方案对比项	方案(1)	方案(2)	方案(3)
	Jython		调用Python命令行
能否满足	能	能	能
可维护性	差	差	好
难易程度	一般	一般	一般

经过验证显示,3种技术方案都能满足功能需求,技术实现难度都不大。

但是从可维护性上考虑,Jython和通过Python命令直接调用的开发语言,都是将算法平台代码和业务模块代码集成在一起,不论哪一部分出现问题都会直接影响监控系统的正常运行。同时Jython依赖于Python版本,如果停止维护导致版本过旧,则会影响功能开发的实现。本文优化方法为把调用Python代码算法部分封装Java外壳,对业务平台提供API,实现了代码解耦,算法模块出现故障不会影响监控系统正常运行,故采用优化后的方案3。

### 3.2 技术方案实现

通过方案对比分析后,本文提出的算法模块封装方案为最优选择,在该方案中共采用了4个API调用,如图6所示为技术方案实现逻辑。

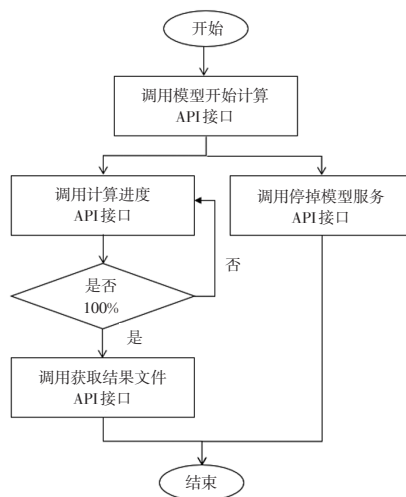


图6 技术方案实现逻辑

先调用模型开始计算接口,触发算法模型运算。运算过程中可以通过调用计算进度接口来获取运算进度比例,也可以通过调用停掉模型服务的接口随时

停止计算。当模型计算进度为 100% 时,便可以通过获取结果文件接口的方式获取最终生成的运算结果。

### 3.3 代码实现

除了功能测试外,也对 3 种技术方案分别进行了性能测试,通过 API 调用的方式实现了从模型计算到获取结果文件的全过程。本小节对每部分实现的核心代码进行展示。

#### (1) 模型开始计算

//API 调用时,输入模型计算参数,体现了模型灵活性,参数可调节。

```
result=remote.callRemoteServer("/market/market/startMarket?partition="+ fileName + "&leps="+ leps + "&lradius="+ lradius + "&heps="+ heps + "&hradius="+ hradius,null);
```

```
//更新数据库表中计算状态 marketService.updateStatusByFileName(fileName);
```

#### (2) 获取计算进度

//通过 API 返回的字符串信息,获取计算进度比例

```
result=remote.callRemoteServer("/market/market/progress?partition="+fileName,fileName);
```

```
fileStatus = result.getString("message");
```

#### (3) 停掉模型

//杀死进程,终止模型计算过程

```
result=remote.callRemoteServer("/market/market/restart",null);
```

#### (4) 获取结果文件

//通过获取文件存放路径,并返回给前端 url 访问路径的方式来实现。结果文件以 csv 格式生成。

```
String url = httpurl+fileName+".csv";
```

```
returnurl;
```

## 4 融合模型应用

在大数据时代,算法模型应用非常广泛。本文中提出的融合模型在车辆监控领域应用广泛,主要集中在对车辆运行数据和车辆销量数据的监控<sup>[21-25]</sup>,下面分别对 2 方面应用做如下阐述。

### 4.1 车辆运行数据

基于融合模型的车辆监控系统可应用于车辆运行数据监控,将车辆运行数据及算法分析结合起来。该应用使对车辆运行数据的监控更智能化,满足客户对于车辆运行监控的全面要求。能够更及时掌握车辆运行情况,包括对运行中的故障车辆进行响应。

车辆运行数据包括的业务板块是车辆运行实时数据监控,算法模块是对车辆数据传递丢包率、丢条率、细分市场、电池状况、路况信息、用户画像、油耗分析监控。通过 Kafka 发布订阅的方式接入汽车团队软件过程(Team Software Process, TSP)服务器原始数据信息,按分区进行大数据存储供监控系统使用。目前,包括家用乘用车在内的汽车是很普遍的交通工具,车辆数量很多,每天产生的运行数据量大。车辆故障问题、运行问题也引起广泛重视,人工已经无法分析海量运行数据,所以通过车辆监控系统中调用算法模型的方式能进行海量数据分析。

### 4.2 车辆销量数据

基于融合模型的车辆监控系统同样适用于销量数据监控,对于整车企业来说,车辆运行是一方面,车辆的销量也同等重要。

车辆销量数据包括的业务模块是对车辆库存、出库、回款、未销数据、存销比、销售目标完成情况的监控,算法模块做车辆销量的预测分析、销量的聚集省份、城市分析。可以通过 API 请求、文件导入方式获取车辆品牌销量,按月维度进行统计和更新。由于营销数据单独通过 Excel 工具进行分析,数据量大,工作繁琐,所以此时通过车辆监控系统中调用算法模型的方式更简单、高效。

### 4.3 监控系统展示

本文基于融合模型的车辆监控系统是以车辆运行数据为基础的设计与实现。图 7 和图 8 所示,是该系统将算法模块嵌入到业务系统后的可视化界面展示,即模型计算界面与模型列表界面。

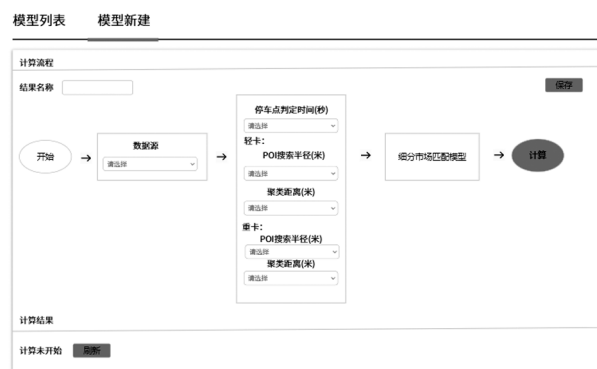


图 7 模型计算界面

本文所设计实现的系统中,通过融合模型解决方案,将业务平台与算法平台结合,并在 WEB 端做了算法模型操作的可视化界面,为用户提供了执行算法模型运算的入口。如图 8 是系统中细分市场模型的展示,该部分设置了模型列表展示和模型新建功能。

模型列表 模型新建

序号	结果名称	创建人	创建时间	状态	操作
1	细分市场示例1	管理员1	2022-12-10 10:36:30	已完成	查看 暂停 分析
2	细分市场示例2	管理员1	2022-12-10 11:26:31	计算中	查看 暂停 分析
3	细分市场示例3	管理员1	2022-12-12 09:12:12	计算暂停	查看 恢复 分析
4	细分市场示例4	管理员1	2022-12-13 08:00:01	未计算	查看 暂停 分析

图8 模型列表界面

在模型列表展示菜单下可以对所有操作过的运算进行查看,查看已保存或正在运行模型的运算参数,并可以进行暂停操作,也可通过点击分析跳转到模型新建页面。列表中清晰地展示了模型名称、创建人、创建时间和模型运行状态,方便用户查看。

模型新建页面采用流程图形式展示,可以让用户清晰地看出模型运算环节,并可以进行模型参数配置,数据源选择,点击“计算”即可触发一次模型运算,保存后便可在模型列表中新增一条数据,也可通过计算结果来查看运行进度,导出最终结果文件。

## 5 结论

本文通过Java开发的业务平台和Python开发的算法平台进行融合,研究出了一套通过技术方案优化的融合模型,并基于此实现对车辆监控系统设计。该融合模型是经过了汽车行业现状调研、对用户需求收集,以及对Java和Python融合方案对比调试后提出的。从功能上、技术架构上以及技术运维、业务应用方面,都实现了改进和优化,主要成果和创新点总结如下:

(1)业务需求:满足了用户对于监控的需求,既能监控车辆的实时状态(实时运行数据、车辆备案状态等),又能实现对车辆运行数据的统计分析(细分市场分析、工况分析、加油点分析等),实现了对车辆的全方位可视化监控。

(2)技术实现:充分结合了2种开发语言优点。相比于直接调用Linux命令行触发算法执行来说,将命令行和Python代码部分嵌入到Java框架部署独立的服务,提供对外访问API,可以实现代码解耦、相互独立、互不影响。如果算法部分出现问题,不影响WEB系统接口请求。不论是从功能上还是性能上都进行了优化。

(3)部署运维:业务基本功能和算法部分进行隔离开来,算法部分作为单独服务,和WEB可视化系统分开,便于代码部署和后期系统运行维护。

(4)应用广泛:本文提出的基于融合模型监控系统既适用于车辆运行数据监控,也适用于对车辆销量数据的监控,应用数据范围广。

(5)用户体验好:该融合模型设计与实现给用户提供了操作算法模型执行的可视化管理界面,提供了控制模型开始计算、计算进度获取、运行结果导出、停止模型运算的可操作按钮。这些均将算法模块可视化,为提高用户体验带来便利。

(6)可拓展性强:模型平台界面做了算法模型可配置设置,用户可以对运算模型参数进行调整,这个设计理念也体现出了该融合模型可拓展性,参数可配、可调、可修改和可查看。

综上所述,本文所设计与实现的基于融合模型的车辆监控系统实现了对车辆全面监控功能,也在整车企业中得到应用,具有实际参考价值与探索意义。融合模型的车辆监控系统不仅能在车辆行业中得到广泛应用,而且可以推广到其它领域,比如财务、医学等。

## 致谢:

感谢在此论文编写过程中中汽信息科技有限公司前瞻业务团队对作者的支持与协助,团队共创是对一项研究成果最好的支撑。

## 参考文献

- [1] 程学旗, 靳小龙, 王元卓, 等. 大数据系统和分析技术综述[J]. 软件学报, 2014, 25(9): 1889-1908.
- [2] 赵力. 基于大数据处理的车辆监控系统的设计与实现[D]. 北京: 北京交通大学, 2019.
- [3] 曾宪宇. 基于大数据技术的车辆监控系统的优化与实现[D]. 长春: 吉林大学, 2016.
- [4] 李国杰, 程学旗. 大数据研究: 未来科技及经济社会发展的重大战略领域——大数据的研究现状与科学思考[J]. 中国科学院院刊, 2012, 27(6): 647-657.
- [5] 曾雷. 大数据研究综述[J]. 软件导刊, 2015, 14(8): 1-2.
- [6] 付岩, 李洪茹. 关于整车企业数字化转型系统架构的研究[J]. 汽车文摘, 2022(7): 20-26.
- [7] 张文杰, 李长龙, 洪宇, 等. 基于大数据技术的远程诊断系统在汽车上的应用[J]. 汽车文摘, 2022(4): 30-33.
- [8] 陈亮. 基于云平台的电动汽车电池监控系统研究[D]. 上海: 上海工程技术大学, 2017.
- [9] 王茹葳. Java编程语言在大数据开发中的应用[J]. 电子技术, 2022, 51(1): 160-161.
- [10] 郭阳, 常英贤. 浅谈Java语言在计算机软件开发中的应用[J]. 数字通信世界, 2022(1): 88-90+94.

- [11] 张佳琪. 基于Java Web的车辆试验管理信息系统的设计与实现[D]. 北京:北京交通大学, 2016.
- [12] 卢绍兵. 基于Python的混合语言编程及其实现研究[J]. 科技资讯, 2022, 20(14): 31-33.
- [13] 张巍. 基于Python的车辆检测算法的实现[J]. 机械管理开发, 2019, 34(12): 258-261.
- [14] 程颀, 崔继强, 王睿莹. 基于Python的重型自卸车工况分析[J]. 汽车实用技术, 2019(20): 58-60.
- [15] 赵志升, 张晓, 马腾. Java与Python整合方案的研究与实现[J]. 网络安全技术与应用, 2015(1): 146+148.
- [16] 张春光. 面向Java和Python混源软件综合质量评估系统设计及实现[D]. 南京:东南大学, 2021.
- [17] 黄敬轩, 刘康军, 云辰太, 等. 基于Java和Python的辅助评标系统的设计与实现[J]. 工业控制计算机, 2021, 34(3): 18-20.
- [18] 沈骛, 王磊, 王西点, 等. Jython技术在网管支撑系统中的应用研究[J]. 电信工程技术与标准化, 2013, 26(5): 52-56.
- [19] 谭翼, 王开, 周兰江. 使用Jython快速开发Web应用[J]. 电脑知识与技术, 2006(35): 50+72.
- [20] 曾浩. Python在其它语言中的应用研究[J]. 大众科技, 2012, 14(7): 13-14+19.
- [21] 王文扬, 陈正, 陈祥威. 新能源汽车监控平台软件的设计[J]. 装备制造技术, 2014(12): 46-48.
- [22] 张瑜, 车晓波, 王勇, 等. 新能源车辆远程监控系统优化设计[J]. 现代电子技术, 2019, 42(3): 96-100.
- [23] AOYAMA K I. Universal traffic management system (UT-MS) in Japan[C]//Proceedings of VNIS'94 - 1994 Vehicle Navigation and Information Systems Conference, Yokohama, Japan: IEEE, 1994.
- [24] OSTROWSKI K, BIRMAN K, DOLEV D. Extensible Architecture for High-Performance, Scalable, Reliable Publish-Subscribe Eventing and Notification[J]. International Journal of Web Services Research, 2007, 4(4): 18-58.
- [25] YANG J. Web service componentization[J]. Communications of the ACM, 2003, 46(10): 35-40.

#### 【作者简介】

高雪, 现就职于中汽信息科技(天津)有限公司, 主要研究方向为商用车数字化平台系统开发。

E-mail: gaoxue2021@catarc.ac.cn

臧金环, 现就职于中汽信息科技(天津)有限公司, 主要研究方向为数字化平台系统开发。

E-mail: zangjinhuan@catarc.ac.cn