

# Sensor Web 支持下的租赁车辆远程监控系统设计与实现

张亚平, 刘学锋

(上海大学 通信与信息工程学院, 上海 200444)

**摘要:** 对于大型的租赁公司来说, 租赁车辆类型多种多样, 不同车型安装的定位设备多来自于不同的生产商, 这些设备在数据传输协议和数据存储格式上存在很大差异, 导致资源难以互通互联; 现有的关于租赁车辆远程监控方案只是针对单一定位设备进行数据管理, 难以实现多源定位设备获取数据的融合; 为解决以上问题, 开展多源定位设备接入管理研究, 构建一个兼容多源定位设备的租赁车辆远程监控系统; 通过对 GT220 和 GT710 这两种来自不同厂商的定位设备建模注册, 利用 Sensor Web 标准中的传感器观测服务 (SOS, Sensor Observation Service) 对多源定位设备及其观测信息进行统一管理; 基于 Java、AJAX、MINA 框架等技术研发了租赁车辆远程监控系统; 基于 HTML、CSS、JavaScript 等技术开发了简洁友好的前端监控页面, 实现了车辆实时位置查询、历史轨迹回放、电子围栏报警等功能; 对 GT220 和 GT710 两种定位设备的仿真实验表明, 该系统能准确获取租赁车辆的定位信息, 并实现实时跟踪、越界报警。

**关键词:** Sensor Web; 传感器观测服务; 租赁车辆; 电子围栏; 远程监控

## Design and Implementation of Remote Monitoring System for Rental Vehicles Based on Sensor Web

Zhang Yaping, Liu Xuefeng

(College of Communication and Information Engineering, Shanghai University, Shanghai 200444, China)

**Abstract:** For large leasing companies, there are many types of rental vehicles. The positioning devices installed in different models are mostly from different manufacturers. These devices have great differences in data transmission protocols and data storage formats, which makes resources difficult to share. The existing remote monitoring scheme for rental vehicles only performs data management for a single positioning device, and it is difficult to achieve integration of data acquired by multi-source positioning devices. In order to solve the above problems, research on access management of multi-source positioning equipment is carried out, and a remote monitoring system for rental vehicles compatible with multi-source positioning equipment is constructed. Through the modeling and registration of two GT220 and GT710 positioning devices from different manufacturers, the multi-source positioning device and its observation information are managed by using the Sensor Observation Service (SOS) in the Sensor Web standard; based on Java, AJAX, MINA framework and other technologies have developed a remote monitoring system for rental vehicles; based on HTML, CSS, JavaScript and other technologies, a simple and friendly front-end monitoring page has been developed to realize real-time location query, historical track playback, and electronic fence alarm. The simulation experiments on two kinds of positioning equipments of GT220 and GT710 show that the system can accurately obtain the positioning information of the rental vehicle and realize real-time tracking and cross-border alarm.

**Keywords:** sensor observation service; rental vehicle; electronic fence; remote monitoring

## 0 引言

由于车辆流动性强, 对租赁车辆的管理难度大。传统的车辆管理方式缺乏对于车辆的实时监控, 造成信息的透明度低, 信息不能及时反馈等问题<sup>[1]</sup>。同时, 由于没有统一的租赁车辆管理平台, 管理人员不能有效地对车辆进行管理<sup>[2]</sup>。另一方面, 大型的租赁公司通常存在多种车型租赁业务, 不同的车型安装的定位设备来自不同的厂商, 这些设备在数据传输协议和数据存储格式上存在很大差异,

导致现存资源不能互通互联<sup>[3-5]</sup>。现有的租赁车辆远程监控系统只能解决单一定位设备数据管理, 不能很好地来自不同厂商设备获取的数据融合到一起实现数据共享<sup>[6]</sup>。

2005年, OGC (Open Geospatial Consortium) 提出了一种国际通用的 Sensor Web 规范—SWE (Sensor Web Enablement), 在该框架中描述了传感器的接口、协议和编码规范, 可从远程接入传感器, 打造一个“即插即用”的基于 Web 的传感器网络标准平台<sup>[7]</sup>。传感观测服务 (SOS) 是 SWE 的最重要且最基础的规范之一, 它作为一种 Sensor Web 服务, 致力于为异构传感器数据仓库和使用这些数据的应用程序之间提供互操作性<sup>[8]</sup>。它提供了一种标准方法, 用于访问来自传感器信息和传感器系统的观测数据解决信

收稿日期: 2019-04-09; 修回日期: 2019-04-28。

作者简介: 张亚平(1994-), 女, 河南通许县人, 硕士研究生, 主要从事于异构移动传感器的接入与应用方向的研究。

息孤岛问题, 实现数据的共享<sup>[9-11]</sup>。

本文将来自不同厂商的定位设备进行建模, 遵循 SensorML 规范生成标准的 XML 建模文档, 注册到 SOS 中, 研究基于 Sensor Web 的租赁车辆远程监控系统, 通过 SWE 提供的传感器观测服务 SOS 实现移动传感器及其观测信息的管理, 在此基础上开发集成来自不同厂商定位设备数据的

统一租赁车辆监控管理平台, 实现车辆定位、历史轨迹回放、电子围栏报警等功能。

## 1 远程监控系统设计

从租赁车辆远程监控系统的实际应用需求出发, 将该系统功能需求分为设备管理、车辆定位、电子围栏越界报警、用户管理等。围绕这些功能, 将系统划分成感知层、数据解析层、数据管理层、逻辑层、应用层五部分。如图 1 所示。

感知层由安装在租赁车辆上的定位设备(以下统称为移动传感器)组成, 主要负责定位信息采集并提供将采集到的信息上传至远程设备的接口。数据解析层主要包括移动传感器信息描述模型和数据解析模块, 主要用于传感器注册和观测数据插入。数据管理层主要分为共享数据中心和业务数据中心两个部分, 共享数据中心由传感器观测服务 SOS (Sensor Observation Service) 和抄收数据库组成, 主要负责对传感器以及观测数据的存储, 这部分数据是可以共享的; 业务数据中心由业务数据库组成, 主要负责业务功能实现所需数据和产生数据的存储。逻辑层主要由 SOS 核心操作实现模块和业务功能实现模块组成, SOS 核心操作实现模块主要负责生成标准的 XML 文档和解析标准的 XML 文档提取所需数据以供业务功能实现模块使用; 业务功能实现模块主要负责应用层功能的实现。应用层主要提供了租赁车辆远程监控页面, 通过该页面用户可以很方便的使用设备管理、车辆位置查询、围栏设置、报警管理、用户权限管理等功能。

定位设备通过传输协议将观测数据上传, 服务器端接收的消息进行解析生成标准的 InsertObservation 操作文档插入 SOS。SOS 为数据的插入和使用提供了方便交互的 Rest 接口, 所有数据的使用者都需要到数据中心获取对应的传感器数据及其观测数据。系统应用层分为内部数据交互和外部数据交互。内部数据交互是指前端与业务数据库进行交互, 使用 SpringMVC 搭建 Web 应用, 用户从前端页面调用后端接口获取或添加数据。外部数据交互即前端与 SOS 的交互。应用层通过 SOS 服务暴露的 REST 接口获取 SOS 对应的传感器数据, 然后传输到前端界面上做展示。

## 2 远程监控系统实现

### 2.1 数据解析层

数据解析层主要包括移动传感器信息描述模型和数据解析模块。在 Sensor Web 中, 集中管理传感器和其观测信息的基础是对传感器建模, 并将传感器注册到 SOS 中<sup>[12]</sup>。

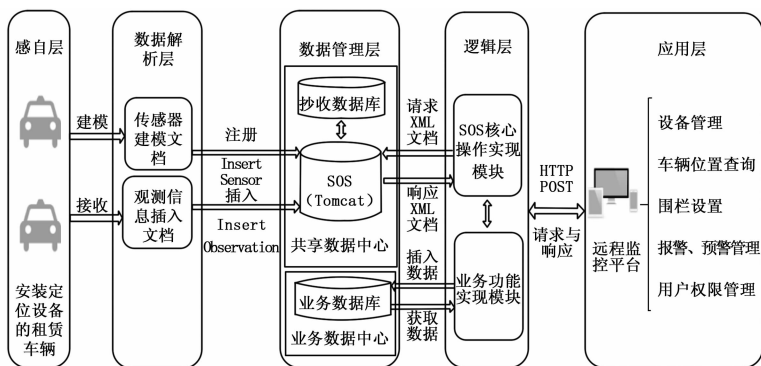


图 1 租赁车辆远程监控系统框架图

移动传感器信息描述模型是一个基于 SensorML 标准的文档, 是从移动传感器分析并提取的建模信息根据 SensorML 标准进行抽象转化所生成的建模文档。对传感器建模之前, 首先要对移动传感器进行分析, 提取建模所需基本信息, 然后遵循 SensorML 规范, 将信息整合成标准的 XML 格式的传感器建模文档。对于移动传感器需要抽象化的信息主要分为两大类: 一类是传感器基本信息, 另一类是传感器的观测信息。完成移动传感器建模后将通过 SOS 提供的 InsertSensor 操作注册到 SOS 服务中。

定位设备以一定的时间间隔将采集到的观测数据通过传输协议发送给服务器, 对数据发送模块所使用的不同传输协议, 解析器都有与之匹配的解析模块<sup>[13]</sup>。通过解析器提取出数据内的关键信息, 发送给部署于后台的文档生成模块。文档生成模块对数据进行解析并通过 SOS 获取已注册传感器信息, 通过对比已注册传感器来判断请求接入的传感器是否合法, 若合法则生成标准的观测信息插入文档, 通过 SOS 提供的 InsertObservation 操作接入数据库中, 完成移动传感器的数据接入。数据接入流程如图 2 所示。

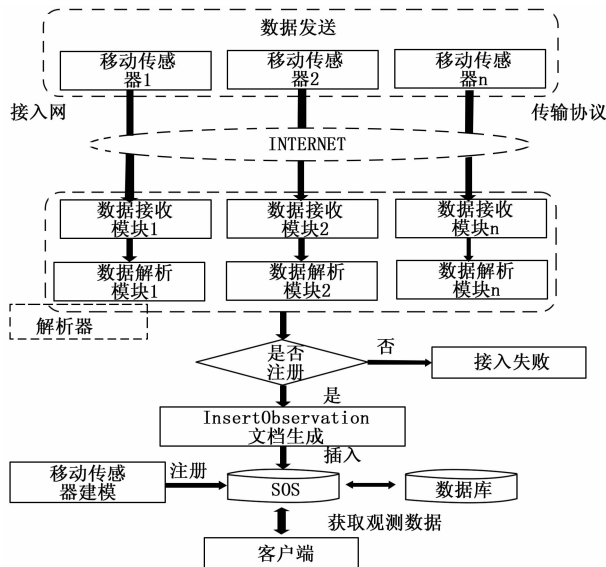


图 2 观测数据接入流程图

## 2.2 数据管理层

数据管理层主要分为共享数据中心和业务数据中心两个部分, 所用数据库为 MySQL 数据库。共享数据中心由传感器观测服务 SOS 和其绑定的数据库组成, 由于 SOS 能够集中管理来自移动、原位和遥感传感器的观测数据, 并提供一个接口, 使得传感器和传感器观测数据能够通过一个基于 Web 的界面被存取, 能很好的实现数据共享。将 52North 公司提供的 SOS 服务 war 包部署到 Tomcat 虚拟服务器上, 当 Tomcat 运行时会自动解析 war 包并提供 SOS 服务的功能。移动传感器观测数据记录了移动传感器观测值的获取的时间、传感器的经纬度等信息, 通过 SOS 提供的 InsertObservation 操作存储到部署 SOS 时绑定的数据库中。

业务数据中心部分数据是不共享的, 主要为具体功能实现而设计。本文根据系统要求, 业务数据库设计主要分为设备管理模块、围栏设置模块、报警预警模块、定位信息模块、权限设置模块 5 个模块。

设备管理模块包括汽车信息表、汽车设备绑定表、汽车状态改变表。汽车信息表用来存储汽车基本信息, 前端调用后端车辆添加和车辆删除等接口时会操作该表单。汽车设备绑定表用来存储汽车与移动传感器绑定信息, 调用后端设备绑定和设备解绑等接口时会操作该表单。汽车状态改变表用来存储汽车状态的变更信息, 调用后端车辆信息变更接口时会对该表单进行操作。

围栏设置模块包括围栏信息表、车辆围栏绑定信息表、围栏状态改变表。围栏信息表用来存储围栏设置基本信息, 前端调用后端围栏创建和围栏修改等接口时会对该表单进行操作。车辆围栏绑定信息表用来存储车辆与电子围栏绑定信息, 前端调用后端围栏绑定和围栏解绑等接口时会操作该表单。围栏状态改变表用来存储围栏状态变更信息, 前端调用后端围栏修改和围栏绑定等时会对该表单进行操作。

报警预警模块包括报警预警信息表、报警信息下发表、预警信息下发表、邮件下发表。报警预警信息表用来存储报警预警基本信息, 前端调用后端报警信息查询和预警信息查询等接口时会对该表单进行操作。报警信息下发表用来存储报警处理信息, 前端调用后端报警信息查询和报警信息处理等接口时会操作该表单。预警信息下发表用来存储预警处理信息, 前端调用后端预警信息查询和预警信息处理等接口时会操作该表单。邮件下发表用来存储报警邮件下发信息, 前端调用后端邮件下发接口时会操作该表单。

定位信息模块包括定位信息表, 用来存储车辆定位信息, 前端调用后端查询车辆信息等接口时会操作该表单。权限设置模块包括用户注册表、权限表、用户权限关联表。用户注册表用来存储用户注册信息, 前端调用后端添加用户和删除用户接口时操作该表单。权限表用来存储权限设置信息, 前端调用后端添加权限和删除权限接口时操作该表单。用户权限关联表用来存储用户与权限相关联信息,

前端调用后端分配权限接口时操作该表单。

根据不同模块实现功能, 依照设计原则对各个表进行设计, 以围栏设置模块中围栏信息表为例, 表结构如表 1 所示。

表 1 围栏信息表

表名	f_fencing				
序号	参数名	中文名称	必选	数据类型	长度
1	id	围栏编号	Y	Bigint	20
2	fencing_name	围栏名称	Y	Varchar	20
3	fencing_data	围栏点集合	Y	Blob	
4	fencing_detail	围栏描述	N	Varchar	200
5	add_user	添加用户	Y	Varchar	20
6	add_date	添加日期	Y	Datetime	
7	update_user	更新用户	N	Varchar	20
8	update_date	更新日期	N	Datetime	

## 2.3 逻辑层

逻辑层主要由 SOS 核心操作实现模块和业务功能实现模块组成。当传感器信息与传感器观测数据插入到与 SOS 绑定的数据库之后, 即可通过 SOS 提供的 3 个核心操作 GetObservation、GetCapabilities 和 DescribeSensor 来获取传感器元数据和目标传感器观测数据<sup>[14]</sup>。通过 SOS 核心操作实现模块屏蔽了 SOS 相关操作的复杂细节, 用户无需构建复杂的 SOS 操作请求的 XML 文档就可以获取传感器信息和观测数据。SOS 核心操作实现模块通过 Java 类实现, 构建两种 Java 类: 一种是 Request 类, 另一种是 Response 类。通过 Request 类生成 SOS 核心操作的标准 XML 请求文档, 并且发送给传感器观测服务 SOS。Response 类实现了将 SOS 核心操作的响应 XML 文档进行解析提取所需数据供业务功能实现模块使用。图 3 为类详细结构图。SOS 服务向外界提供获取数据的方法接口是基于“请求—响应”的交互模式, 一次请求只能获得一次响应, 响应完成后结束会话。

业务功能实现模块主要负责业务功能逻辑处理, 本文中租赁车辆远程监控系统主要实现业务功能包括: 设备管理、车辆位置查询、围栏设置、报警预警管理、用户权限管理等功能。本系统基于前后端分离的思想来构造, 服务器后端使用 SpringMVC+Hibernate 实现。其中 SpringMVC 处理请求分发, Hibernate 进行与数据库交互。整个后端系统通过 Spring 来构造和管理实例达到系统的整合。前端使用 HTML+CSS 来构造用户页面, 前端逻辑使用 JavaScript 来编写。前后端的交互使用 Rest 架构风格, 前端使用 AJAX 请求后端数据, 后端返回对其请求结果的 Json 对象。

用户在应用层进行操作就会调用后端业务功能相应的接口, 以电子围栏创建和车辆越界报警功能为例, 在电子围栏设置部分高德地图 API 在前端画图时提取经纬度信息, 这些信息保存成字符串形式, 通过 ajax 保存到后台数据库。

Request	功能
+ XMLDoc: String	表示返回XML文档
- getCapabilities () : String	生成GetCapabilities操作的标准XML请求文档
- describeSensor(String sensorID): String	生成DescribeSensor操作的标准XML文档
+ getObservation(String SensorID,String Properties,String StartTime,String EndTime): String	生成GetObservation操作的标准XML文档
Response	功能
+ result: String	表示解析后的数据
+ getCapabilitiesResponse () : String	解析GetCapabilities操作返回的XML文档
- describeSensorResponse(String sensorID): String	解析DescribeSensor操作返回的XML文档
+ getObservationResponse(String SensorID,String Properties,String StartTime,String EndTime): String	解析GetObservation操作返回的XML文档

说明: 类成员的可见性分别用四种符号表示其访问权限

- + : public
- : private
- # : protected
- 无 : default

图 3 Request 类和 Response 类结构图

创建电子围栏调用电子围栏创建接口, 请求协议为 HTTP 协议, 请求类型为 post 请求, URL 全径为 CarPatures/FencingController/addFencing, 请求参数包括: fencing\_name (围栏的名称)、fencing\_detail (围栏的备注)、fencing\_data (围栏的点集合)、user\_name (用户名), 响应参数包括: code (请求状态码, 备注: 200 请求成功, 500 请求失败)、message (返回信息, 备注: 成功: 添加成功。失败: 添加失败+围栏信息)。操作表单为业务数据库围栏设置模块的围栏信息表。在对车辆进行越界报警监控前首先要进行设备和围栏绑定, 设备和地图都放在内存中间用 map 的形式保存, 绑定的租赁车辆的每个定位信息都会和对应的围栏信息进行判断。前端和后端使用 websocket 保持长连接, 后端发现越界直接推送信息到应用层。

### 2.4 应用层

为了用户更直观地观测到定位数据, 引入高德地图 API 作为重要的技术支持, 对车辆监控最核心的需求都是围绕它来进行开发。利用 HTML 构建页面框架, CSS 设置页面样式, 通过 JavaScript 将高德地图嵌入到网页中, 使用 jQuery 框架的 AJAX 方法实现前后端参数传递。租赁车辆远程监控系统应用层主要设计实现了设备管理、车辆位置查询、围栏设置、报警预警管理、用户权限管理等页面。

## 3 功能测试

### 3.1 仿真环境搭建

根据本模型系统总体架构的设计与实验实现需要, 开发环境如下:

- 1) Java 开发包和运行环境: JDK1.8.0\_102, JRE1.7
- 2) Java 集成开发环境: Eclipse Java EE IDE Photon
- 3) 前端开发环境: HBuilder9.1.25
- 4) Web 服务器: Apache Tomcat 7.0
- 5) 数据库: MySQL5.3
- 6) SOS 版本: 52n-sos-4.0

7) 服务器配置: Windows Server 2008 R2 Enterprise, CPU E5-2403 1.8 G, 内存 8 G

8) 不同厂商的定位设备: GT220, GT710

### 3.2 仿真结果与分析

#### 3.2.1 数据采集与数据接入

通过在自行车上安装来自不同厂商的定位设备来模拟测试车辆, 首先对定位设备进行建模注册, 在上述测试环境下, 设置 GT220 定位设备每 10 s 上传一次位置信息, 设置 GT710 定位设备每 5 s 上传一次位置信息。由于 GPRS 网络具备长距离传输的能力, 因此本文选用 GPRS 进行无线接入。定位设备使用 TCP 协议传输数据, 不同厂商的定位设备上传数据包格式有所差别, 通过定义不同的类型的数据包解析模块来实现多源定位设备的接入, 移动传感器(定位设备)注册及其观测数据插入结果如图 4 所示。

传感器注册列表		
2 T	SHUTEST01	(Null) Offering for sensor
3 T	SHUTEST02	(Null) Offering for sensor
4 T	SHUTEST03	(Null) Offering for sensor
5 T	SHUTEST04	(Null) Offering for sensor
6 T	SHUTEST05	(Null) Offering for sensor

传感器id为SHUTEST03的经度值 (GT220)

procedureid	observablePropertyId	value
4	8	123.4531
4	8	120.1212

传感器id为SHUTEST05的电池电量 (%) (GT710)

procedureid	observablePropertyId	value
6	11	80
6	11	79
6	11	78

图 4 传感器注册和数据插入结果

上述数据展示了移动传感器在后台的存储方式, 每个传感器注册都有唯一的传感器标识, 也就是 SensorID, 通过对应属性参数之间的关联, 展示不同的传感器性能。通过将传感器注册 id 与设备一一对应来实现对定位设备及其观测数据进行统一管理。

#### 3.2.2 业务功能测试

##### 1) 设备管理功能测试:

传感器注册后每个传感器都有唯一的传感器 ID, 传感器 ID 所对应定位设备也有唯一的设备编号, 为了方便用户管理和查询设备信息, 屏蔽后端设备绑定的一些复杂操作, 开发前端页面来方便用户管理, 管理员通过设备添加操作实现设备与虚拟传感器绑定, 通过设备删除操作实现设备与虚拟传感器的解绑, 另外可以通过设备列表查询获取后端数据库中设备与传感器绑定所有信息, 如图 5 所示。

管理员在管理页面进行上述操作, 后端数据库会操作相应的表单进行数据的插入与删除, 说明该设备管理模块可以成功的实现设备绑定的添加和删除功能。

##### 2) 车辆位置及其行车轨迹查询功能测试:

根据设置, 定位设备会定时上传经纬度观测信息插入 SOS 进行管理, 通过 SOS 服务 GetObservation 操作获取这

设备编号	设备型号	传感器ID	入库时间	入库用户
GT220-01	GT220	SHUTEST01	2019-03-04	zhang
GT710-01	GT710	SHUTEST04	2019-03-04	zhang

设备编号	设备型号	传感器ID	入库时间	入库用户
GT220-01	GT220	SHUTEST01	2019-03-04	zhang
GT220-02	GT220	SHUTEST02	2019-03-04	zhang

设备编号	设备型号	传感器ID	添加时间	添加用户	最后更新时间	更新人
GT220-01	GT220	SHUTEST01	2019-03-04	zhang	2019-03-04	zhang
GT220-02	GT220	SHUTEST02	2019-03-04	zhang	2019-03-04	zhang
GT220-03	GT220	SHUTEST03	2019-03-04	zhang	2019-03-04	zhang
GT710-01	GT710	SHUTEST04	2019-03-04	zhang	2019-03-04	zhang

图 5 设备管理

些观测信息, 后台封装了 DeviceLocationImpl 类来实现车辆位置查询功能, 当前端发起请求, 该类将 GetObservation 操作响应解析后数据进行过滤上传至前端页面进行显示。因为数据采集的经纬度坐标比较密集, 为了更直观的观测车辆位置信息以一分钟为间隔进行取点在地图上显示, 并且以虚线绘制行车路径, 如图 6 所示。



图 6 车辆位置信息与行车轨迹

如图 6 所示, 设备编号为 GT220-03, 传感器标识号为 SHUTEST03 在开始时间 2019-03-05-00:00:00 到结束时间 2019-03-06-00:00:00 这一时间段内的车辆位置信息及其行车轨迹。由图可以看出该系统历史行车轨迹显示及其车辆位置查询功能是可以很好的实现的。

### 3) 电子围栏越界报警预警功能测试:

后端封装了 FencingImpl 类来实现添加围栏、删除围栏和添加车辆与围栏绑定等功能。创建一个 MessageImpl 类实现报警预警管理功能, 该类中创建 isInFencing (是否驶入围栏)、isOutFencing (是否驶出围栏)、isCloseFencing (是否靠近围栏) 3 个成员变量来存储车辆位置判断后状态信息, 调用该类相应的成员方法对车辆位置进行判断如果满足驶入围栏、驶出围栏、靠近围栏条件, 将相对应的 isInFencing、isOutFencing、isCloseFencing 参数赋值为 true。然后启用新的线程对上述 3 个参数进行实时监控, 如

果该参数值为 true 则进行报警预警, 并将报警预警信息写入相应数据库。报警判断条件设置为: 驶入围栏报警、驶出围栏报警。预警判断条件设置为当车辆靠近围栏 50 米处进行预警。其中对于报警信息会进行邮件下发通知管理者, 预警信息则不进行邮件下发。

首先进行电子围栏创建, 点击围栏设置跳出电子围栏创建页面, 设置围栏名称, 选择围栏位置。如图 7 (a) 所示, 选择上海市宝山区为围栏功能测试区域, 点击提交完成电子围栏创建。将监控车辆与电子围栏进行绑定, 建立长连接, 对测试车辆进行持续 3 个小时远程监控, 车辆运行轨迹如图 7 (b) 所示。



图 7 电子围栏创建与车辆运行轨迹图

从图 7 (b) 中可以看出车辆 4 次驶入驶出围栏, 在车辆驶入驶出围栏前都会先靠近围栏, 当车辆靠近围栏 50 出会先进行预警, 预警信息不会下发邮件通知, 驶入驶出围栏报警会下发邮件进行通知管理员。针对上述测试, 对应后台数据库如图 8 所示。

id	deviceId	sensorId	messageType	messageContext	uploadTime	user
1	GT710-02	SHUTEST05	靠近围栏预警	请注意车辆靠近围栏!	2019-03-05 12:0:	zhang
2	GT710-02	SHUTEST05	驶出围栏报警	请注意租赁车辆驶出围	2019-03-05 12:1:	zhang
3	GT710-02	SHUTEST05	靠近围栏报警	请注意车辆靠近围栏!	2019-03-05 12:5:	zhang
4	GT710-02	SHUTEST05	驶入围栏报警	请注意租赁车辆驶入围	2019-03-05 12:5:	zhang

图 8 报警预警信息结果显示

从图 8 可以看出, 定位设备 GT710 编号为 GT710-02, 对应传感器 id 为 SHUTEST05, 绑定设备每次报警预警信息都会入报警预警信息信息表, 用户通过前端请求以获取相应数据在管理页面进行显示。

针对以上测试可以看出, 对于来自不同厂商的定位设备进行建模注册并通过 SOS 对定位设备及其观测数据进行管理是可行的, 在此基础进行租赁车辆远程监控并对其设备添加和删除、车辆位置查询、历史轨迹回放及电子围栏报警预警等功能进行测试, 说明本系统能够很好地实现租赁车辆远程监控。

## 4 结束语

本文基于 Sensor Web 设计并实现了租赁车辆远程监控 (下转第 119 页)