The Internet of Things for Petroleum Transportation

Hui-ran WANG¹ and Rui-fang MA²,*

¹College of Computer Science, Xi’an Polytechnic University, Xi’an 710048, China
²School of Software Engineering, Xi’an Jiaotong University, Xi’an 710049, China

*Corresponding author

Keywords: Petroleum, Transportation, Internet, Information, Model.

Abstract. The Internet of Things for petroleum transportation is still in the early stages of development. One of the technical key problems is there is still no information model. We put forward the infrastructure for the Internet of Things for Petroleum Transportation. It covers oil depots, gas stations, tank trunks and control center. We studied the information model. The basic unit of the information model is a monitored object. A monitored object has 6 attributes. The information model for the Internet of Things for Petroleum Transportation contains one root, 5 branches, and a lot of leaves. Each branch stores a group of monitored objects. And each leaf stores a single monitored object. We discussed the implement of information model. We have developed a type of the Internet of Things for petroleum transportation and it has worked five years.

Introduction

With the rapid development of national economy of China, road transportation system has been rapidly developed. Until the end of 2013, the total mileage of the national highway network has reached 4.35 million kilometers, 100 thousand kilometers of which are highway. Highway transportation consumes a large amount of gasoline and diesel oil. As of the end of 2014, the number of China's highway gas stations exceeded 100 thousands. And the number of oil transport vehicles exceeded 30 thousand. We are going to develop the Internet of Things for petroleum transportation to manage these vehicles.

The Internet of Things for petroleum transportation is the combination of the Internet and the traditional petroleum transportation systems. It consists of mechanical structures, sensors, computer software and hardware, computer networks and other components. Nowadays the Internet of Things for petroleum transportation is still in the early stages of development. Its technology is not perfect and mature. There are many technical key problems to be solved. One of the technical key problems is there is still no information model for the Internet of Things for petroleum transportation. Information model is applied to organize and store information of oil transportation systems [1-4]. It defines the types of information, the structure of information and the relationship between varieties of information in oil transportation systems. This paper studies the information model and its realization method in the Internet of Things for petroleum transportation.

The Internet of Things for Petroleum Transportation

As shown in Figure 1, the Internet of Things for Petroleum Transportation covers oil depots, gas stations, tank trunks and control center. A tank trunk is filled with gasoline at an oil depot. Then the tank trunk is driven to a gas station and the gasoline is pumped into a ground tank. The whole process is controlled and managed by the control center.

In the Internet of Things for Petroleum Transportation, the Control Center exchanges a varieties of information with oil depots, gas stations and tank trunks. We organize this information by means of an information model.
Design of Information Model

The basic unit of the information model used in the Internet of Things for Petroleum Transportation is a monitored object. As seen in Figure 2, a monitored object has 6 attributes, i.e. a descriptor attribute, an identifier attribute, a syntax attribute, an access attribute, a status attribute, and a definition attribute.

Figure 1. Internet of Things for Petroleum Transportation.

The descriptor attribute is the textual name of a monitored object. The identifier attribute is self-explanatory, and has to be unique in the whole information model. It is of OBJECT IDENTIFIER data type. The syntax attribute specifies the data type and structure associated with this monitored object.

Table 1 lists several data types frequently used in the Internet of Things for Petroleum Transportation. The access attribute specifies pattern how the control center access this monitored object. Its available values include read-only indicating that control center may read this object only, write-only indicating that control center may write this object only, read-write indicating that control center may both read and write this object as well, not-accessible indicating that control center can not access this object. The status attribute provides the current status of a monitored object. It has 3 optional values, i.e. mandatory, optional, and obsolete. The mandatory status shows that this object has to be implemented by every station of the Internet of Things for Petroleum Transportation. The optional status indicates that this object may be implemented optionally. The obsolete status means that this object has been replaced by other object. The definition attribute is a human-readable textual definition of the monitored object. We have to implement this monitored object exactly on this definition to keep its consistent semantics between different systems.

Figure 2. Attributes of monitored objects.
The monitoring objects are described in ASN.1 (Abstract Syntax Notation version 1) \cite{5}, and encoded using BER (Basic Encoding Rules) \cite{6}. The BER specifies how instances of monitored objects are encoded and sent over the Internet. It adopts the so-called TLV (Type, Length, and Value) approach.

The information model for the Internet of Things for Petroleum Transportation contains one root, 5 branches, and a lot of leaves. Each branch stores a group of monitored objects. And each leaf stores a single monitored object.

Table 1. Data types and their tags.

| Data type     | Tag            | Description                                      |
|---------------|----------------|--------------------------------------------------|
| Integer       | Universal 2    | An arbitrary integer                             |
| Object Identifier | Universal 6  | A sequence of integer components that identify a monitored object |
| Real          | Universal 9    | An arbitrary real number                         |
| Sequence      | Universal 16   | An ordered collection of one or more types       |
| PrintableString | Universal 19 | An arbitrary string of printable characters      |
| GeneralizedTime | Universal 24 | Time in generalized format                       |

These 5 branches include oil depot branch (1.1), gas station branch (1.2), tank trunk branch (1.3), driver branch (1.4) and control center branch (1.5).

The oil depot branch (1.1) stores information about an oil depot, as shown in table 2. This branch contains 6 monitored objects. Object ykID stores the number of this oil depot. Object ykName stores the name. Object ykLocation stores the position. Object ykContact stores the Contact. Object ykTelephon stores the telephone number. Object ykIPAddr stores the IP Address of this oil depot.

Table 2. Monitored objects in the oil depot branch.

| identifier | descriptor | syntax | access    | status    |
|------------|------------|--------|-----------|-----------|
| 1.1.1      | ykID       | Integer| read-only | mandatory |
| 1.1.2      | ykName     | String | read-write| optional  |
| 1.1.3      | ykLocation | String | read-write| optional  |
| 1.1.4      | ykContact  | String | read-write| optional  |
| 1.1.5      | ykTelephone| String | read-write| optional  |
| 1.1.6      | ykIPAddr   | IPAddress | read-write| mandatory |

The gas stations branch (1.2) contains information about a gas station, as shown in table 3. This branch includes 6 monitored objects.

Table 3. Monitored objects in the gas stations branch.

| identifier | descriptor | syntax | access    | status    |
|------------|------------|--------|-----------|-----------|
| 1.2.1      | jyzID      | Integer| read-only | mandatory |
| 1.2.2      | jyzName    | String | read-write| optional  |
| 1.2.3      | jyzLocation| String | read-write| optional  |
| 1.2.4      | jyzContact | String | read-write| optional  |
| 1.2.5      | jyzTelephone| String | read-write| optional  |
| 1.2.6      | jyzIPAddr  | IPAddress | read-write| mandatory |
The tank trunk branch (1.3) stores information about an oil depot, as shown in table 4. This branch contains 20 monitored objects.

Table 4. Monitored objects in the tank trunk branch.

| identifier | descriptor | syntax | access     | status     |
|------------|------------|--------|------------|------------|
| 1.3.1      | gcID       | String | read-only  | mandatory  |
| 1.3.2      | gcType     | String | read-write | optional   |
| 1.3.3      | gcVolume   | Integer| read-write | optional   |
| 1.3.4      | gcCabin    | String | read-write | optional   |
| 1.3.5      | gcDatePrd  | Time   | read-write | optional   |
| 1.3.6      | gcRKTable  | Sequence of | not-accessible | mandatory |
| 1.3.6.1    | gcRKEntry  | Sequence | not-accessible | mandatory |
| 1.3.6.1.1  | gcRKID     | Integer | read-write | optional   |
| 1.3.6.1.2  | gcRKStatus | Integer | read-write | optional   |
| 1.3.6.1.3  | gcRKTime   | Integer | read-write | optional   |
| 1.3.7      | gcHDFTable | Sequence of | not-accessible | mandatory |
| 1.3.7.1    | gcHDFEntry | Sequence | not-accessible | mandatory |
| 1.3.7.1.1  | gcHDFID    | Integer | read-write | optional   |
| 1.3.7.1.2  | gcHDFStatus| Integer | read-write | optional   |
| 1.3.7.1.3  | gcHDFTime  | Integer | read-write | optional   |
| 1.3.8      | gcCYKTable | Sequence of | not-accessible | mandatory |
| 1.3.8.1    | gcCYKEntry | Sequence | not-accessible | mandatory |
| 1.3.8.1.1  | gcCYKID    | Integer | read-write | optional   |
| 1.3.8.1.2  | gcCYKStatus| Integer | read-write | optional   |
| 1.3.8.1.3  | gcCYKTime  | Integer | read-write | optional   |

The driver branch (1.4) stores information about drivers of vehicles, as shown in table 5. This branch contains 6 monitored objects.

Table 5. Monitored objects in the drivers branch.

| identifier | descriptor | syntax | access     | status     |
|------------|------------|--------|------------|------------|
| 1.4.1      | jsyID      | Integer| read-only  | mandatory  |
| 1.4.2      | jsyName    | String | read-write | optional   |
| 1.4.3      | jsyIdentif | String | read-write | optional   |
| 1.4.4      | jsyGender  | String | read-write | optional   |
| 1.4.5      | jsyBirthday| Time   | read-write | optional   |
| 1.4.6      | jsyTelephone| String | read-write | optional   |

The control center branch (1.5) stores information about control center, as shown in table 6. This branch contains 6 monitored objects.
Table 6. Monitored objects in the control center branch.

| identifier | descriptor | syntax     | access   | status     |
|------------|------------|------------|----------|------------|
| 1.5.1      | kzzxID     | Integer    | read-only| mandatory  |
| 1.5.2      | kzzxName   | String     | read-write| optional  |
| 1.5.3      | kzzxLocation | String    | read-write| optional  |
| 1.5.4      | kzzxContact | String    | read-write| optional  |
| 1.5.5      | kzzxTelephone | String    | read-write| optional  |
| 1.5.6      | kzzxIPAddr | IPAddress  | read-write| mandatory  |

Implement of Information Model

We take the oil depots branch as example to show how the model is realized. We use C language as programming tools [7].

Firstly, we define a structure named as infobody as following:

```
struct infobody {
    char id[30]; //to store identifier attribute
    char name[30]; // to descriptor attribute
    char type[20]; // to store syntax attribute
    char value[50]; // to store value of a monitored object
    char access[20]; // to access attribute
    char status[20]; // to store status attribute
    char defin[100]; // to store definition attribute
};
```

This structure is used to store parameters about a monitored object in the oil depot branch. The 7 elements represent 6 attributes and its value of a monitored object respectively.

Then, we defining an array of this structure as following:

```
struct infobody ykinfo[6]
```

Those 6 members indicate the 6 monitored objects (as shown in Table 2) of the oil depots branch. Element ykinfo[0] corresponds to monitored object ykID. Element ykinfo[1] through Element ykinfo[5] correspond to monitored object ykName, ykLocation, ykContact, ykTelephone, ykIPAddr respectively.

We process the monitored objects in 4 steps as following:

1. Initializing ykinfo[6]
   We input initial value by means of keyboard through scanf function of C language. After encoding by BER rules, we store these values in the 7 elements of a member of array ykinfo[6].

2. Receiving a request from the control center
   Suppose a component of in the Internet of Things for Petroleum Transportation receives a operation request from the control center over the network.
   If the request belongs to a read operation, we check the access attribute of the requested member of array ykinfo[i]. If this access attribute equals to “read-only” or “read-write”, we send this value of the member of array ykinfo[i] to the control center. Otherwise we reject this request.
   If the request belongs to a write operation, we also check the access attribute of the requested member of array ykinfo[i]. If the access attribute equals to “write-only” or “read-write”, we store the value sent by the control center into this structure array ykinfo[i]. Otherwise we reject this request.

3. Backup parameters in local disc
   Before closing the system, we use the fwrite function of C language to store the value of the structure array ykinfo[6] into a disk file. In this way, all the attributes of the monitored objects are protected for recovery in the future.
(4) Restoring parameters from local disc

When the system restarts, the fread statement of C language can be used to retrieve all the information of the monitored objects from the disk file to the structure array ykinfo[6], and then system can make various operations on them.

Summary

We put forward the infrastructure for the Internet of Things for Petroleum Transportation. It covers oil depots, gas stations, tank trunks and control center. We studied the information model. The basic unit of the information model is a monitored object. A monitored object has 6 attributes. The information model for the Internet of Things for Petroleum Transportation contains one root, 5 branches, and a lot of leaves. Each branch stores a group of monitored objects. And each leaf stores a single monitored object. We discussed the implement of information model. We have developed a type of the Internet of Things for petroleum transportation and it has worked five years.

Acknowledgement

This research was financially supported by Science and Technology Department of Shaanxi Province, project no. 2017XT-014 and by Science and Technology Bureau of Xi'an Beilin District, project no. GX1607.

Reference

[1] Ye Fei. Optimization of Interface Data Refresh Method in Power Dispatch SCADA System. Automation of Electric Power Systems, v 39, n 18, 2015. Pp 69-74.

[2] Yan Li-mei. Power system state estimation of quadrature Kalman filter based on PMU/SCADA measurements. Electric Machines and Control. V 18, n 6, 2014. Pp 79-84.

[3] Xiong Huang, Huang Xiao-li, “Design and realization of power dispatching data network in Central China”, Telecommunications for Electric Power System, n 12, 2008.

[4] Yang Hai-jing1; Zhang Dong-yng; Wu Qiong; Liu Wen-yng; Yang Yi-hng; Zou Pin-yung, Structure Design and Function Analysis of Power System Flexible SCADA, Power System Technology, n 15, 2006.

[5] CCITT. Recommendation X. 208: Specification of Abstract Syntax Notation One (ASN.1).

[6] CCITT. Recommendation X. 209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

[7] Jean J. Labrosse, MicroC/OS-II the Real-Time Kernel, second edition, CMP Books, 2002.