Research on Data Storage Structure of Object Pool for Intelligent Virtual Terminal Based on ISO11783

Mingzhu Zhang[1,2,a,*], Mingming Cui[1,b] and Tingting Wang[1,c]

[1]School of Mechtronics Engineering, Henan University of Science and Technology, Luoyang, Henan Province, China

[2]Collaborative Innovation Center of Machinery Equipment Advanced Manufacturing of Henan Province, Luoyang, Henan Province, China

[a] ming2000@126.com, [b]15670301150@163.com, [c]1083321568@qq.com

Keywords：ISO11783, Virtual Terminal, object pool, data storage structure

Abstract: To facilitate the intelligent virtual terminal to analyze the object pool data, the basic characteristics of object pool data are studied and the data storage structure of object pool in the virtual terminal is presented. According to the technical standard of virtual terminal based on ISO11783-6, the characteristic function of intelligent virtual terminal is determined. The functional characteristics of each object in the object pool of the working group are analyzed. The subordinative relation between the objects is determined at different levels, and the object hierarchy relation allowed by the object pool is established. According to the properties and characteristics of the object pool, the data storage structure of object pool and structure expression form of each object are designed based on C language. A unique Object ID to each object in its object pool is assigned, so that each object is uniquely addressable, and the operation of querying and changing data can be achieved efficiently, so as to better satisfy the requirements of intelligent virtual terminal analysis the object pool. This study lays a theoretical foundation for the further development of the intelligent virtual terminal.

1. Introduction

The most commonly used embedded electronic equipment and remote sensing technology in modern agricultural machinery have been adopted by precision agriculture system. The tractor virtual terminal is the most commonly equipment [1,2]. The virtual terminal(VT) is the critical core electronic control equipment for the automation and intelligent operation between tractors and farm implements. It was usually installed in the cab, according to the operator's needs, displaying the status information of electronic control unit(ECU) in tractor or farm tools by words or graphics on the display screen, and inputting operational data and control information to each ECU through keys, buttons, etc.

The ECU of the farm implements or working set provides the man-machine interface through VT, which includes a large variety of display input and output information. The ISO11783-6 standard defines various kinds of information as objects, which have specified attributes, data, and corresponding behavior functions [3]. A working set has many objects that are called working set's objects pool.

The object pool is considered as one large block of data with each object and its attributes making up a single variable length record. This paper studied the characteristics of the object data and the relationship between the data, established the object hierarchy relation that allowed by the object pool, proposed a data storage structure of object pool to organize data so that the virtual terminal can receive, write, read, store and efficiently parse the object pool.

2. Characteristic Function of Intelligent Virtual Terminal

Intelligent VT is a man-machine interface device, the main is to provide a platform for the

information processing of the vehicle ECU bus network based on ISO11783, for displaying the ECU data and the alarm information, and the real-time control on the operating state and information resource management functions, etc[4]. VT occupies a very important position in the whole standard, and satisfies the purpose of sharing the information between the tractor and each ECU of the farm implements. A example of intelligent VT as shown in Figure 1.

Figure 1 Example of Intelligent Virtual Terminal

Intelligent virtual terminal mainly involves some of the following features:

1. The ECU data information collection functions. It requirements that VT can received the any object pool data that is standardized and had the ability to store, restore, and delete a specific object pool. When the vehicular electronic equipment is reconnected，VT may call the appropriate object pool data in memory, without retransmitting it again.
2. Display. Analytic the object pool and displaying on the screen, to facilitate real-time monitoring of the equipment which linked to the bus. The display information includes not only the vehicle speed, engine speed, fuel and other instrument information, also agricultural operation information. For operating and monitoring conveniently, VT not only displays active work set, but also displays one or more inactive work set.
3. Alarm Function. VT allows the work set to display alarm information at any time. The alarm information including speed, oil mass, error and fault information, etc. It is convenient for the operator to process the exception situation in a timely manner. If there are several woke set's alarm masks were activated, VT should display the alarm mask in priority order.
4. Control Function. Operators can adjust the operating parameters and working state by VT, according to the state information of the current time. VT supports auxiliary functions such as buttons, switches, joystick, etc. Operators can set up the auxiliary function through VT, so as to realize the more convenient control, and also to obey the habit of the operators.

3. The Object Pool of Intelligent Virtual Terminal

The agricultural implements or work set's ECU provides the man-machine interface through VT. This contains a large variety of input and output information, and defines all kinds of information as object[5], ISO11783-6 standards set a lot of different object types for us to use. The object type is shown in Table 1.

Table 1 Object types supported by the ISO 11783-6 standard

Working Set object	Input List object	Number Variable object
Data Mask object	Output String object	String Variable object
Alarm Mask object	Output Number object	Font Attributes object
Container object	Output Line object	Line Attributes object
Window Mask object	Output Rectangle object	Fill Attributes object
Soft Key Mask object	Output Ellipse object	Object Pointer object
Key object	Output Polygon object	Macro object
Button object	Output Meter object	Auxiliary Function
Input Boolean object	Output Linear Bar Graph	Auxiliary Input object
Input String object	Output Arched Bar Graph	
Input Number object	Picture Graphic object	

The operator interface definition for a device of one or more implements represented by either a single ECU or a Working Set consists of a set of objects. The Working Set assigns a unique Object ID to each object in its object pool so that each object is uniquely addressable. Object IDs shall be unique within a single object pool but may not be between different Working Sets. The visible object of an object pool can be arranged in a hierarchical structure. The parent object contains child objects, and the ISO11783-6 standard defines the allowed object hierarchy, as shown in Figure 2.

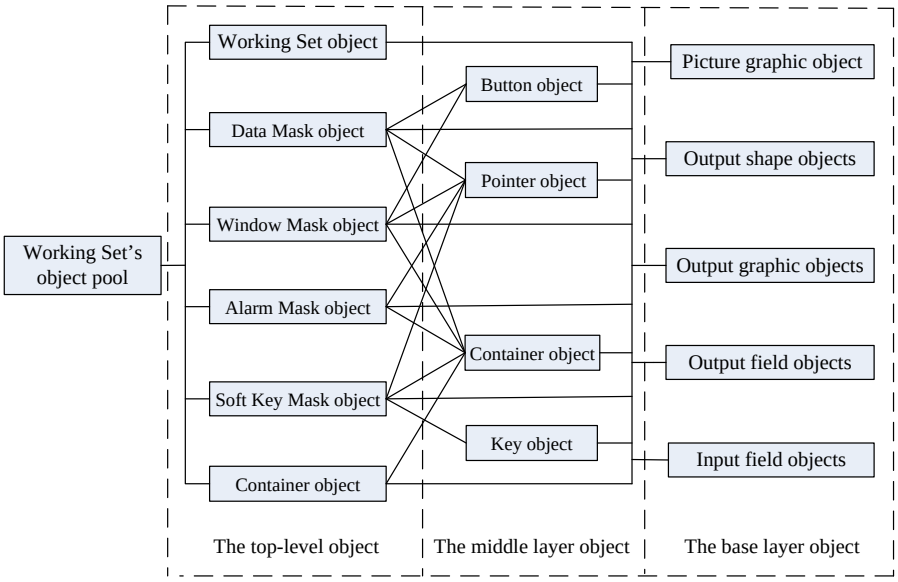

Figure 2 Object hierarchy diagram of Working Set's object pool

According to the allowed objects containing relation of object pool, object pool is divided into three levels: the top-level object, the middle layer and the base layer object. The hierarchical structure of visible objects in the object pool is arranged legitimately. The designer can arrange a object pool base on needs according to the following level relationship.

4. Data Storage Structure of Object Pool

The intelligent VT receives the objects pool data that sent by working set's master, and stores it in a non-volatile memory. When receives the activated message, VT loads the object pool data into memory to processing, display, and other operations. According to the standard description of each object, the object storage process is summarized with the following characteristics:

a) The object pool should include the working group name, workgroup version, workgroup number and other information.
b) The Working Set assigns a unique Object ID to each object in its object pool so that each object is uniquely addressable.
c) The record size is not fixed, because the object contains a different number of other objects.
d) Each object consists of the object ID, its type, and its related attributes.

According to the above features, the object pool stores in tabular form[6], as shown in Table 2, which includes Working Set Name, Working Set versions, Working Set numbers, and Working Set object pool bytes, etc.

Table 2 The index table of Working Set's object pool storage address

The index table of Working Set's object pool storage address			
Contents	Variable name	Type	Example
Working Set Name	NAME	Char	
Working Set versions	Version	Char	
Working Set numbers	WS_Num	Int	
Working Set object pool bytes	Databyte	Int	
Object storage address	DataADD	Void*	0x080000

A unique Object ID to each object in its object pool so that each object is uniquely addressable. It makes more convenient to process object data, such as replacing object data, deleting object data, and so on. Because the top-level object contains the different number of objects, the record size is not fixed. And stored the data into two data storage area, one contains the basic attribute information and the record size is fixed, such as Object ID, type, and so on. Another is a storage area according to the number of objects with size is not fixed. As shown in Figure 2.

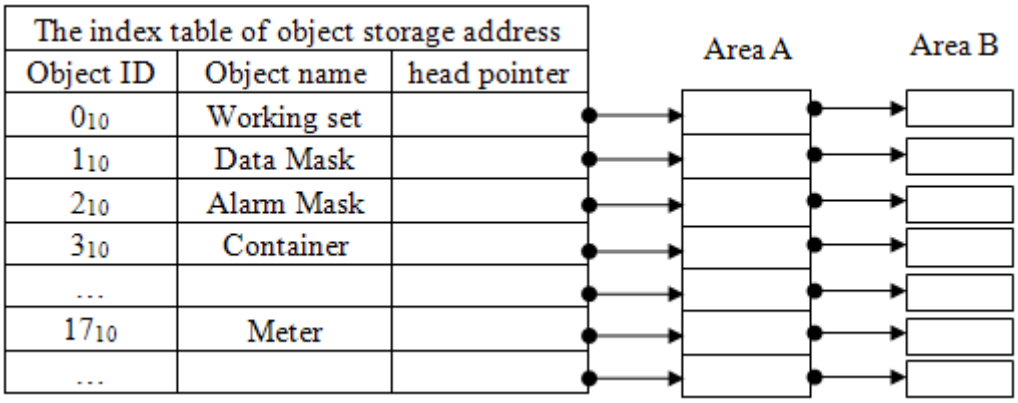

Figure 2 The diagrammatic sketch of object storage

Object storage including the index table of object storage address, data area A and data area B. VT can retrieves each object through the index table of object storage address and the storage address of the object data based on the address header pointer of each object. The advantage of this is that each object can be independently addressed, not only make good use of the features of the object pool, but also achieved data query, data change and other operations efficiently.

Each object contains different attributes and data. For example, Working set object includes Object ID, type, background color, activation mask, followed child objects, event ID, and so on; Meter object includes Object ID, type, width, pointer color, border color, arc and scale color, number of calibration, maximum, minimum and other information. The information of each object is stored in data area A in the form of structure, but the including concrete object and macro object are not contained in the structure, and can be stored in data area B.

5. Concrete Realization of Data Storage Structure

Based on C language, the data storage structure of object pool is designed. It contains textual content, which is Char type, for declaring the name and version of the Working Set. Using type Int to indicate Working Set numbers, and type Void to represent the object storage address pointer which point to the storage address of the object. The specific data storage structure is as follows:

```
Struct    WorkSet_s
{
    Char    NAME[20];              //    Working Set Name          //
    Char    Version[20];           //    Working Set versions      //
    Int    WS_Num ;                //    Working Set numbers       //
    Void*    dataADD;              // the pointer of the index table //
} WorkSet_a;
```

The object data storage structure is designed. It contains multiple objects. Each object is represented by Object ID as the storage identifier, and the object storage address index table is given. Using Int type to represent object ID, Char type to declare object name and Void type to represent the storage address pointer of each object which point to the storage address of the corresponding object. The specific data storage structure is as follows:

```
Struct    DataADD_s
{
    Int      obID；                   //                    Object ID                          //
    Char      obName；             //                     Object Name                       //
    Void*    obData              //    the pointer of the object storage address    //
} DataADD_a；
```

Each object contains different attributes, data, and the storage byte. Take the Working Set object as an example: including Object ID, object type, background color, Activation mask and the numbers of following object, the storage types of which were unsigned short type, unsigned short type, unsigned chart type, unsigned short type, unsigned short type, etc. The specific data storage structure is as follows:

```
typedef struct
{
    unsigned short    objectID;                        //                  Object ID                        //
    unsigned short    type;                            //                Object Type                      //
    unsigned char    backgroundColor;           //           BackgroundColor                  //
    unsigned short    selectable;                   //                  Selectable                    //
    unsigned short    activeMask;                 //              ActiveMask                      //
    unsigned short    includeobjects;           //    Number of objects to follow          //
    struct    IncludeObject；                    //    Information of objects to follow      //
} WorkingSet;
struct IncludeObject；
{
    unsigned short    objectID;                            //             Object ID to follow               //
    signed short      X_Location;                        //              X Location of object          //
    signed short      Y_Location;                      //         Y Location of object              //
    unsigned short    EventID;                         //          Event ID to follow                //
    unsigned short    MacroID;                        //        Number of macros to follow      //
    unsigned char    languageCodes;               //             Language Code                   //
}Object_a;
```

Other objects have similar structure storage formats based on the corresponding attributes and data, and then achieve the storage of the entire object pool. Designers can choose the corresponding objects constituted object pool, which can be used by intelligent VT and achieve the aim of reception, storage and analysis, etc. In the process of object pool stored, we should set up the status flag of VT status message is 1, indicating that the current VT is in a stored data state. In accordance with the transmission order of ISOBUS, VT stores the data of each object sequentially. And then stores the address head pointer of each object's store address to the index table of object storage address, next stores the head pointer of the object storage address index table to the index table of working Set's object pool storage address. In this way，the object pool data is stored into the VT memory.

6. Conclusion

The data storage structure presented in this paper applies to object pool which conformed to ISO11783 standard. Combining the functional features of each object, the subordinative relation between the objects at different levels is determined. The Working Set assigned a unique Object ID to each object and expressed each object in the structure expression form. The method of data storage and data management for the object pool was realized, organized and stored data effectively

into VT. With pointer data pointing to the storage address of each object, the efficiency of retrieving and processing object pool is improved definitely.

Acknowledgment

This paper is supported by National Natural Science Foundation of China(Grant No51375145) and the Project for Key Science and Technology Research of Henan Province(Grant No162102210205).

References

[1] Ham W, Enkhbaatar T, Luubaatar B, et al. DESIGN AND IMPLEMENTATION OF VIRTUAL TERMINAL BASED ON ISO11783 STANDARD FOR AGRICULTURAL TRACTORS[J].

[2] Tumenjargal E, Badarch L, Kwon H, et al. Embedded software and hardware implementation system for a human machine interface based on ISOAgLib[J]. Journal of Zhejiang University SCIENCE C, 2013, 14(3): 155-166.

[3] ISO11783-6, Tractors and machinery for agriculture and forestry — Serial control and communications data network， Part 6: Virtual terminal[S], 2014.

[4] Zhiqiang X. Research on Virtual Terminal of Tractor Based on ISO11783[D], Henan University of Science and Technology, 2007.

[5] Mingzhu Z, Zhili Z, Zhiqiang X. Design of Virtual Terminal for Agriculture Machinery Based on ISO11783[C]//Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on. IEEE, 2009, 2: 852-855.

[6] Xiaoguang Y. The example tutorial of data structure[M].Beijing.Beijing Jiaotong University Press, 2015.6:16-54.