Design and Implementation of Aviation Bus ICD Data Management in Flight Testing

Jiang Xie(✉), Chenli Ji, and Ruchang Huang

Chinese Flight Test Establishment, Xi'an, China
784592220@qq.com

Abstract. To address the needs of post test data analysis and real-time monitoring analysis in flight tests, an aviation bus ICD design and management function has been developed. Analyze flight test testing requirements, determine the requirements framework, complete the top-level design of ICD data management, and form an overall design plan. Complete the data design module to create, edit, retrieve, and export various ICD data functions. At the same time, manage the version of ICD data and design data tables, establish an ICD bus database to meet the centralized management of ICD and real-time sharing of data, provide data support for the subsequent work of flight test and improve the efficiency of flight test data processing.

Keywords: Control interface · database data sheet · Manage Retrieval

1 Introduction

Through the analysis of flight test requirements, the database management system should have various ICD data design functions, which can easily input, edit, store, query interface data between the avionics system and other systems, and be able to export according to the specified file format [1]. Record and manage interface data at levels such as data blocks, signals, and fields, and standardize unique identification. The field type should be consistent and standardized with the requirements, provide multiple query methods, and provide a data interface for the testing system as the basis for avionics system data collection and testing [2, 3]. The system supports the management of 1553B, RS422, ARINC429, AFDX and other bus ICDs, and the management of non bus quantity A/D, D/A, power signal, discrete signal, analog signal, reference signal, pulse signal, audio signal, video signal and serial digital signal. The security requirements of the system mainly include user permission management, version management of ICD data, and a log system for recording daily operations. Design aircraft bus ICD design and management functions, establish an ICD bus database, meet the centralized management of ICD and real-time data sharing, and improve work efficiency [4].

D. Kumar et al. (Eds.): IEIT 2023, AHSSEH 10, pp. 1251–1259, 2023.
https://doi.org/10.2991/978-94-6463-230-9_151

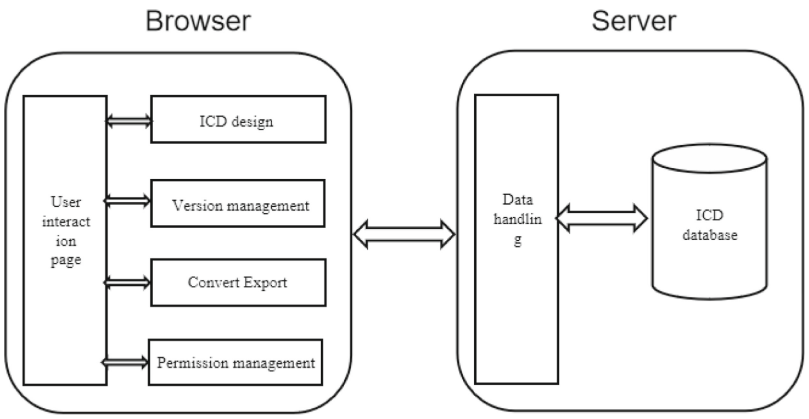


Fig. 1. System framework diagram

2 Scheme Design

The ICD design and management module is designed in the flight test data management system. With the unified data structure as the core idea, the interface definitions of various aviation bus standards are unified through this ICD structure to form an ICD database which can support the definition of various aviation bus standard interfaces. Information in ICD management software is stored structurally by message type, data block, data frame and specific signal [5].

ICD design and management functions generally adopt B/S architecture. In order to meet the functional requirements of the system, various functional modules and support modules need to be developed and organized organically [6]. The overall system frame diagram is shown in Fig. 1.

The ICD database is divided into an ICD server and an ICD browser. The ICD server serves as the core of the entire software, achieving data storage and user application data synchronization. The ICD browser is an interactive page for software users, which enables operations such as inputting, editing, and querying ICD data, as well as managing permissions and versions through it.

The design and management function of avionics ICD based on bus technology should consider the issue of data interface with other applications or business units. Using XML files as a universal and easy to read data interface file can avoid a series of problems caused by deviation in document understanding, as shown in Fig. 2, which is a data interface file in XML format [7].

3 Database System Design

According to requirements, this database design needs to meet the requirements of ICD file and bus data storage to ensure data integrity. Data tables should be designed with scalability to record detailed data information. The relationships between tables need not be too close to prevent the coupling between tables from being too large, causing

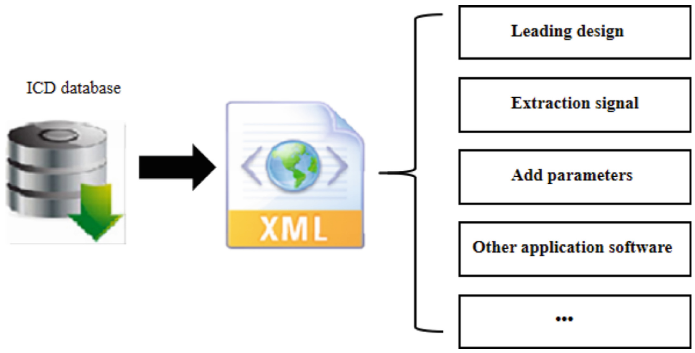


Fig. 2. XML format data interface file

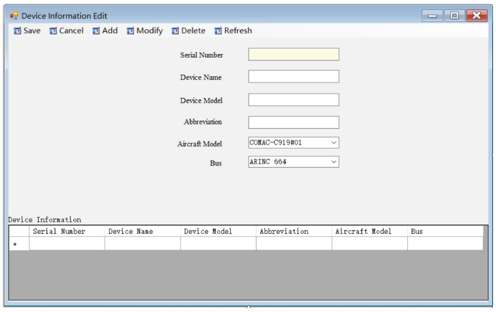
changes in one table to affect other tables, and the design of the primary key and foreign key of each table is reasonable [8].

3.1 Data Design Module

The data design module mainly completes the functions of creating, editing, retrieving and exporting various ICD data [9]. This module interacts with users through a graphical interface to input and set data attribute information and related relations, through access to the database, completes various ICD data addition, deletion and alteration checking operations, and identifies user's operation rights on various resources through access rights management module.

3.1.1 Data Entry Interface

The ICD file database management tool module is used to implement the data entry and modification functions of ICD files. The main module interface is shown in Fig. 3. The device information editing interface is used to input or edit subsystem information on the bus. Add the bus option to distinguish which acquisition channel the current subsystem is connected to. Mainly including data blocks, data words, field information, etc.

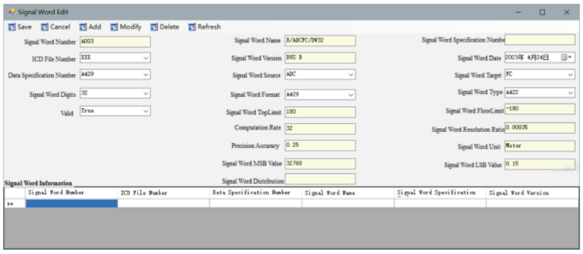


Serial Number	
Device Name	
Device Model	
Abbreviation	
Aircraft Model	COMAC-C919#01
Bus	ARINC 664

Device Information

	Serial Number	Device Name	Device Model	Abbreviation	Aircraft Model	Bus
*						

Fig. 3. Device information editing interface

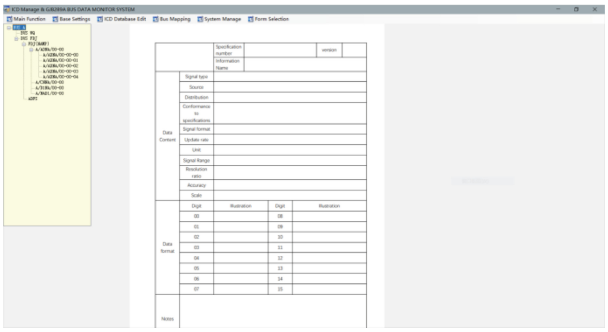


Signal Word Number	A003	Signal Word Name	B/ADCFC/DW32	Signal Word Specification Numbe	
ICD File Number	XXX	Signal Word Version	BUS B	Signal Word Date	2023年 4月24日
Data Specification Number	A429	Signal Word Source	ADC	Signal Word Target	FC
Signal Word Digits	32	Signal Word Format	A429	Signal Word Type	A422
Valid	True	Signal Word TopLimit	180	Signal Word FloorLimit	-180
		Computation Rate	32	Signal Word Resolution Ratio	0.00035
		Precision Accuracy	0.25	Signal Word Unit	Meter
		Signal Word MSB Value	32768	Signal Word LSB Value	0.15
		Signal Word Distribution			

Signal Word Information

Signal Word Number	ICD File Number	Data Specification Number	Signal Word Name	Signal Word Specification	Signal Word Version

Fig. 4. Signal word information editing interface



		Specification number		version	
		Information Name			
Data Content	Signal type				
	Source				
	Distribution				
	Conformance to specifications				
	Signal format				
	Update rate				
	Unit				
	Signal Range				
	Resolution ratio				
	Accuracy				
	Scale				
Data format	Digit	Illustration	Digit	Illustration	
	00		08		
	01		09		
	02		10		
	03		11		
	04		12		
	05		13		
	06		14		
	07		15		
Notes					

Fig. 5. Signal word field information editing interface

The field of a semaphore is the most basic unit for parsing data. Generally speaking, there are two types of fields, one is numeric, which can be determined by the start, end and MSB/LSB values. The other is an enumeration type, which typically uses one or more values to represent states or options. As shown in Fig. 4, in the editing interface, the field type option makes it easy to distinguish between the two types of fields by enabling the corresponding input areas according to the user's choice.

The user can access the signal word field information editing interface, as shown in Fig. 5. View the ICD file that already exists in the database according to the tree structure of the data block. This module also has some other panels for setting the basic information of ICD files, such as the input and operation panels for system status information, communication format information, signal word status information, signal word type information, and signal word field information.

3.1.2 Data Retrieval

Using resource browser based data retrieval, it provides a multi project and multi mode browsing format in the form of tree nodes. Under the subsystem nodes of the project node, the data block information sent and received by each subsystem can be viewed step by step, as well as the signal information associated with the data block. Under the sub nodes of data blocks in the project node, all data blocks and their associated signal information can be viewed step by step. Under the signal sub node of the project node, all signals and their domain information can be viewed step by step. Under the signal type node of the project node, all signal information can be viewed by length [10].

Using editor based association retrieval, various association information can be viewed based on the association list in various ICD data editors, and all data block information sent and received can be viewed in the subsystem editor. In the data block editor, you can view all signal information associated with it, as well as its source and target system information. In the signal editor, you can view all the data block information associated with it, as well as all the domain information it contains.

3.2 ICD Data Version Management

Version management module mainly completes ICD data modification history and project data version management functions. This module interacts with users through a graphical interface to browse history, restore data, version management, and compare version data. It completes data storage and update by accessing databases and version databases. Version management is recorded in the database with each submission by the user, allowing you to view the project's image at any historical moment. You can rewind the current version of the data; Local save does not immediately commit the database, and data that does not commit the database can be rolled back.

3.3 Entity Attribute Diagram Design

By analyzing the contents of ICD files and bus test data, the related entities of this database are extracted. By analyzing the related relationships, the entity-attribute map can be obtained. The following are the main entities selected to illustrate.

1) The model entity is used to describe the basic properties of the model under test, such as model name, model number, bus number, etc. The entity-property diagram of the model entity is shown in Fig. 6.
2) The airborne device entity describes the basic properties of the specific functional subsystem on the bus of the airplane aeronautical system, such as the device name, device model, abbreviated code number, ICD file name, corresponding RT address number, etc. The entity-property diagram of the airborne device entity is shown in Fig. 7.
3) The data block entity of an ICD file describes the properties of all bus message data blocks in an ICD file, including data block specification number, data block name, layout, date, source, target, communication format, transmission type, interrupt

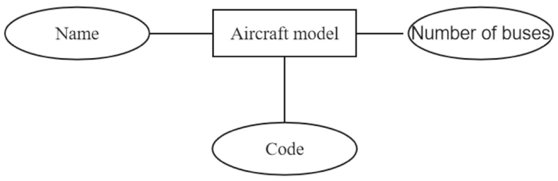


Fig. 6. Model Entity Attribute Map

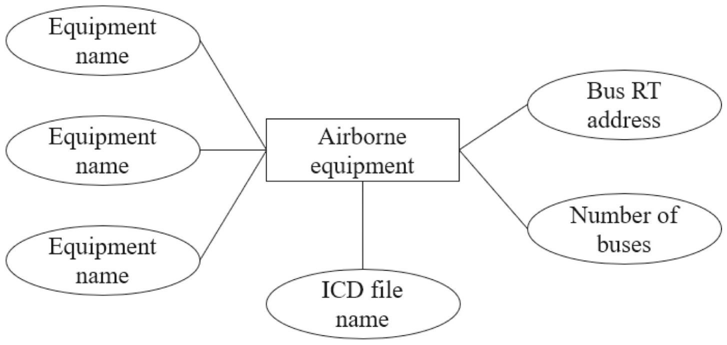


Fig. 7. Physical Attribute Map of Airborne Equipment

enablement, override permission, number of data block signal words, refresh cycle, maximum delay, send and receive flags, message type, corresponding RT subaddress, etc. The entity-property diagram of the ICD file data block entity is shown in Fig. 8.

4) The ICD file data word entity describes the specific properties of a data word contained in a data block, such as data font size, data word name, layout, date, type, source, target, format, upper and lower bounds, resolution, precision units, bus data word offsets, and so on. The entity-property diagram of an ICD file data word entity is shown Fig. 9.

5) Bus data entities describe the properties contained in the data that appears on the bus, including message number, time scale, message type, bus number, RT address, etc. The entity-property diagram of a bus data entity is shown in Fig. 10.

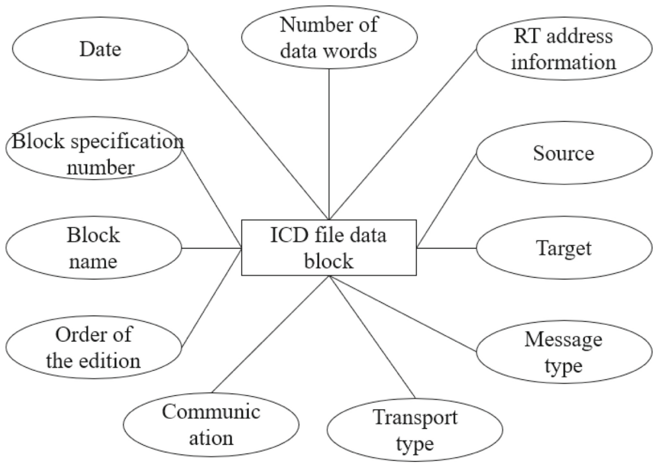


Fig. 8. ICD file data block entity attribute map

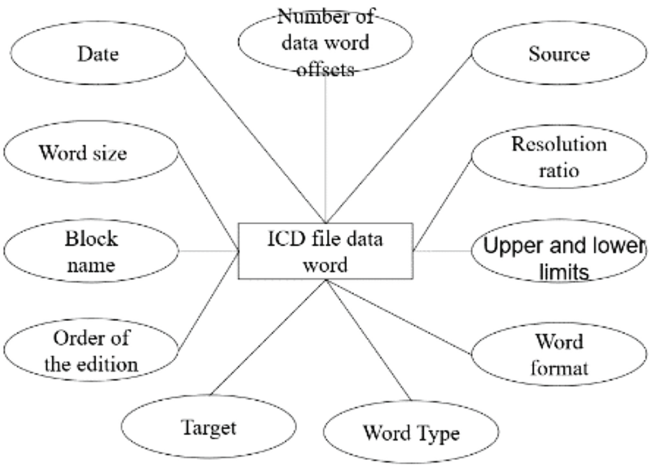


Fig. 9. ICD file data word entity attribute diagram

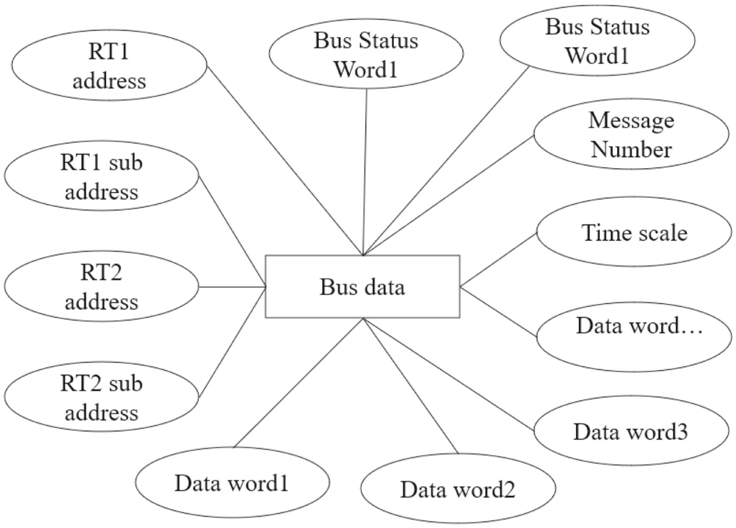


Fig. 10. Attribute Graph of Bus Data Entities

3.4 Data Table Design

Based on the entity attribute map, the corresponding database data table design can be made. In order to unify and standardize the fields of all the data tables in the whole database, some abbreviations are defined. As shown in Table 1, the Airborne Device Information Table stores all the RT information hooked to the bus in the database.

For an ICD file owned by an RT, due to possible version changes, a table for storing ICD file information, ICD file information data table, needs to be designed in the database. Each ICD file data block contains one or more data words, which represent the

Table 1. Airborne equipment information

Attribute Name	Chinese means	Data type	Key Description
D_no	serial number	int	PK
D_name	Airborne equipment name	char	
D_type	Airborne equipment model	char	
D_id	Airborne equipment abbreviation code	char	
D_pno	Model number	int	FK, PlaneType(PTno)
D_bno	Bus number	int	FK, BUS(Bno)
D_ifno	ICD file number	int	FK, ICDFile(IFno)
D_rtno	Airborne equipment RT address	int	

physical meaning of the bus data. Therefore, a table is needed to store the data digital information in the database, and the field design data table of the data word information table is needed.

In practice, not every data word simply represents a physical meaning. Some data words represent some state variables. A data word has 16 data bits and can be represented by one or more bits. At this time, if only one data word table can not meet the requirements of use. For this reason, the content of data word needs to be refined. After the physical meaning is refined to the field by the way of field, the database of ICD file has been basically completed. There are also some auxiliary tables in the database to store some common data types, formats and other information. Bus data tables are used to store data collected on the bus.

4 Conclusion

Through ICD design and management function, it can help flight test effectively and accurately apply bus interface control document. This function provides an efficient interface management platform for flight test testing. Just need to input and output the required information values (physical quantities) and the source and destination of each information transmission to automatically generate interface files, which greatly improves efficiency.

Due to its excellent versatility, this function can be used in flight testing of all models, reducing design iterations, shortening flight testing data processing time, and reducing human errors. The efficiency value it brings is enormous, and it can indirectly generate considerable economic benefits.

References

1. Chen zhixiong. Management of ICD data of civil aircraft. People's Communications Press, Science and Technology Innovation Guide. 2014

2. Wang xiaofei. Testing and Analysis of Aviation Bus Signal Based on Database Technology. Measurement and control technology. 2013:32(8).
3. Dai weibing. Attribute Reconstruction Technology for C919 Aircraft Avionics Bus Data Source. Flight Mechanics. 2019, 5(37).
4. Han Bing. Design of ICD Management System Platform Based on Avionics System ICD Analysis. Manufacturing Automation. 2020, 6(42).
5. Zhang Xiangfen. Database Management of Avionics interface control document. Aeronautical Computer Technique. 2000, 9: 3-31.
6. Han xiaodong. Research on Automatic Generation Method of Avionics Bus Test Software Based on ICD Configuration. Measurement and Control Technology. 2011:30(11).
7. Wang Peng, Qiu Zhiliang. Design of AFDX Switch and Key Modules of the Switching Chip. 2008. Shaanxi: Xidian University: 13–15.
8. Ding Lina. The research of AFDX system simulation model. Multimedia Technology (ICMT) International Conference, 2010. (5): 29–31.
9. M. Yao. Leaky bucket algorithms in AFDX Electronics Letters. AFDX Electronics Letters, 2009. (3): 543-545.
10. WU Hua. Design and Implementation of Communication Port in AFDX End-System [J]. Measurement & Control Technology, 2009. 28(3): 56-59.