

A Study of Cockpit HMI Simulation Design Based on the Concept of MVC Design Pattern

Zheng Liu*, Fei Li, Hongtao Liu, Chunze Wu and Jingjin Zhang

Shanghai Aircraft Design and Research Institute, Cockpit Integration and Industrial Design Research Department, Shanghai 201210, China

*Corresponding author

Abstract—A human-machine interaction (HMI) simulation design method for the aircraft flight deck has been introduced, including modular user interface design, logical dependency between display unit and control panel, and general aircraft model. Model-View-Controller (MVC) design pattern is a classic software architecture, which plays an important role in the interactive user interface design. This design pattern is to achieve the goal of decoupling the logic dependency and system model. The cockpit HMI design based on MVC design pattern has the features of convenient modification, modular design and strong portability. The design method has been applied in the prototype of autopilot system in the civil aircraft cockpit.

Keywords—MVC pattern; human-machine interaction; cockpit simulation; software engineering

I. INTRODUCTION

Cockpit is the primary place for crew members to contact with the aircraft, a platform for pilots to control the aircraft, and also offers pilots an interface and means for the human-machine interaction with the aircraft. The main function of the cockpit is to ensure that the pilot can get enough information about the aircraft and the flight environment, manipulate the aircraft, and guarantee the safety for each flight cycle [1, 2]. Therefore, the rational design of the cockpit has played a key role in completing flight tasks, ensuring flight safety and providing information services.

With the rapid development of information technology, the massive application of Computer Aided Software Engineering (CASE) in the flight simulation provides the technical support for the interactive simulation design of the cockpit [2]. Computer software can simulate the design of the control panel, screen, alarm message inside the cockpit and external terrain and climate change outside the cockpit. In the different flight scenarios, the flight attitude or the systems working modes are transmitted to the pilot through flashing or sound effects [2]. The pilot can also perform various flight tasks through the simulated control devices, such as take-off, approaching, cruising, tower communication, and so on. However, the interactive cockpit simulation needs a complete set of flight system models, airborne systems, and normal functions and failure modes. The complexity of the aircraft model and the parallel design of the cockpit and various subsystems have brought great challenges to the aircraft engineer [3].

According to the complexity of the aircraft cockpit structure and the unexpected changing design requirements, this paper proposes an idea of human-computer interaction simulation design based on MVC design pattern.

II. MVC FRAMEWORK

The common human-computer interaction model provides the system information to the user through the display unit and responds to the user's operation on the screen. For a software application, this model provides two main functions: visual or aural application interface and certain business logic [4]. If the two functions are mixed together, such a design is unfavorable to the maintenance, reusability, and expansion of the software. This is also a major defect in the traditional way of software design. Additionally, users hope to keep the function changeable and adjust the operation interface quickly and conveniently, which improves the scalability and reusability of the software [5].

The MVC pattern has the feature of the independent design of operation interface, state display and data model of the software design. In the other words, the adjustment of the display will not affect the data model by reducing the coupling between the software modules. It is a design pattern suitable for developing human-machine interaction framework. The MVC pattern, as shown in Figure 1, divides the system into three parts [5].

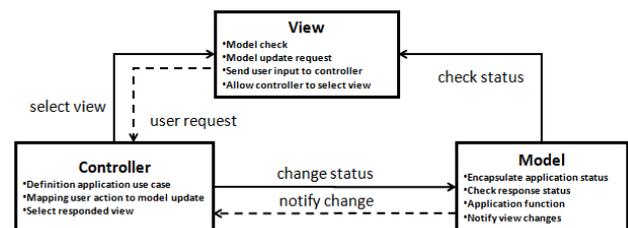


FIGURE I. MVC FRAMEWORK BASIC STRUCTURE

- 1) Model is a business logic processing module that contains data, behavior, and business rules needed to complete the business. It abstracts the object in the business process, encapsulates the object attributes and the business processing logic implied by the object.
- 2) View is a user view module consisting of multiple visualized elements. It provides an input to the user, triggers the controller to get access to the enterprise

data through model, and then displays them in some way.

- 3) Controller is a process module, which controls the various events and services of the system. It is also the cross link between the view and the model, which can get the data from the view and pass it to the model to execute the business logic. According to the results after execution, a suitable view is selected to show the result to the user.

The MVC design pattern decouples the business logic from the application interface from the structure, enabling both of them to expand independently and provide greater reusability [6]. At the level of the software code, because the relationship between transaction logic and data is loosely coupled, the reusability can be realized at the level of system easily. In the system development organization, because of the separation of data representation and transaction logic, the division of developer's labor field is clear and the efficiency of the whole development is improved [7].

III. RESEARCH ON RELATED FIELDS

The model view controller (MVC) is a fundamental design pattern for the separation between user interface logic and business logic. Since applications are very large in size these days and the MVC design pattern can weak the coupling among the different application tiers of application, a web application framework based on MVC in J2EE platform, and extends it with XML so that the framework is more flexible, expansible and easy to maintain [8]. To overcome the disadvantages of current application development technology of .NET, an application model of MVC in .net with three tier structures is put forward and realizing of the model in.NET environment [9]. Considering the similarities and differences between the MVC pat tern structures of three major architectures, stand alone, C/S and B/S, the presentation layer of the three-tier MVC pattern was extended by to construct the general purpose MVC pattern by adding models, thus solving the difficulties in conversion between these architectures and improving scalability, maintainability and reusability [10].

The design pattern of MVC in the field of aerospace has been widely used. Based on the strong real-time and reliability requirements of the avionics system, the multithread MVC mode is applied to the avionics software design, thus simplifying the design complexity [11]. Aiming at the difficulty of maintenance and expansion in the traditional control and recording simulation system, due to the inaccurate definition of model layer, logic layer and view layer, a stable MVC application framework has been introduced. The software designed by using this framework can simplify the development process and improve maintainability, so that the simulation system can be flexibly configured and expanded to meet the corresponding expansion needs for the future simulation applications [12].

The cockpit HMI simulation framework developed by MVC pattern can decrease the development complexity through decoupling the complex human-machine interactive logic from the general aircraft model.

IV. MVC FOR COCKPIT INTERACTIVE DESIGN

Model based MVC design pattern uses the model to abstract the characteristics of user interface from the top level, and then map it to the logic model and data model to achieve the purpose of [1] interactive design. This method is a research frontier in the current human-computer interaction.

The framework of the model based MVC design method proposed in this paper is shown in Figure 2, the core is integrated and modular design ideas, including the three main models: the model of aircraft (Aircraft Model) to model the core function, it is equivalent to MVC in the model; interactive logic model (HMI Controller) on user task modeling in the implementation, in order to control the interaction between the plane model and interface model, the equivalent control part of the MVC. The interface model (HMI View) is used to model the system state and human-machine interface, and realize the feedback to control the aircraft model and the state of the system, which is equivalent to the view part in MVC.

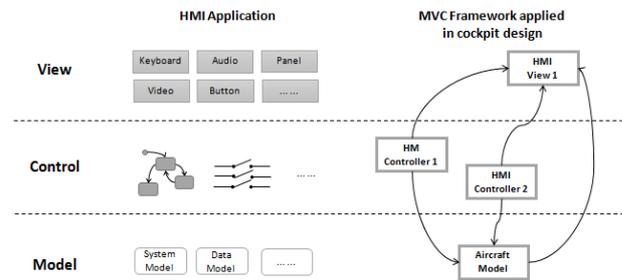


FIGURE II. MVC CONCEPT APPLIED IN COCKPIT SIMULATION

- 1) Model: flight models (including flight dynamics, aerodynamics and external environment), and airborne system models (including power system, power supply system, flight control system, avionics system, etc.), as well as system status of normal and failure mode, etc. Additionally, the system model can also be a data-driven model by the flight simulation data. The main purpose here is to drive the display interface.
- 2) Controller: display and control logic, which realize the system simulation with the normal or fault state and the display logic simulation. The information can be transferred to the crews through the sketch, indicating lamp, EICAS page, voice alarm or some other ways.
- 3) View: human-machine interface of the cockpit, which is mainly composed of cockpit devices and the control panel namely rudder, throttle, brake pedal, display control panel (DCP), multi-function keyboard (MKB), cursor control board (CCD), audio control panel (ACP), a tuning control panel (TCP), etc.

The cockpit is designed and implemented by MVC design pattern, namely the general aircraft model, human-machine interaction interface and control logic. In the background, according to each cockpit dynamic HMI application scenarios different operation model is to be established. Then according to the symbol library, GUI library, font library to realize the HMI cockpit display interface; control module processes the

external signal and makes influence on the model and display interface.

V. A HMI PROTOTYPE OF AUTO-PILOT SYSTEM

According to the design method described in the paper, a HMI prototype of the civil aircraft auto-pilot system is introduced below, as shown in Figure 3. The human-machine interface of the front stage is implemented by graphical design software. The logic control part is modeled and simulated by the toolbox of Stateflow and Simulink tools, and the model is a general aircraft model.

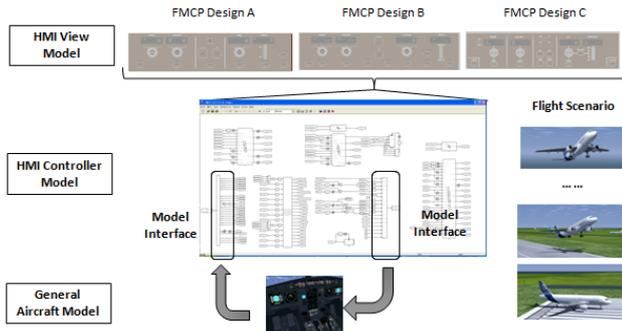


FIGURE III. A SIMPLE HMI PROTOTYPE OF AUTO-PILOT SYSTEM

The flight control system simulation includes multiple flight phases, such as take-off, climbing, cruising, descent, automatic approaching, automatic landing, and so on. The control of the automatic flight mode is achieved through the flight mode control panel (FMCP), which supports setting speed, height, track angle, and switching between manual and automatic flight mode. Figure 3 shows that, the FMCP Design prototype A, B, C shares the same HMI Controller Model with the Model Interface. Flight director (FD) simulates the heading, vertical velocity and track angle. When FD opens, the automatic flight system takes over the operation of the aircraft. Flight mode announcement (FMA) supports all the automatic flight modes and display on the PFD correctly. In this prototype example, the HMI View Model, the HMI Controller Model and the General Aircraft Model are developed decoupled and independently.

VI. SUMMARY

In this paper, a design method for the cockpit interactive simulation has been introduced with the core idea of MVC design pattern. The design method based on the MVC framework applies the modular design concept, which saves the development time and makes the aircraft cockpit design more convenient and flexible.

- MVC design pattern. Through the basic idea of decoupling design of system model, view and control unit, the three modules are designed with each other independently. Where there is a change in demand, the function can be modified at the minimum cost.
- Modular design. The modular design method provides an effective solution to meet the flexible requirements

of the product and the design modification at different stages.

In the automatic flight system example, a general aircraft model is used to simulate the FMCP mainly function. However, the logic behind the FMCP is very complex and involves in lots of aspects on different flight stages and external environment. Therefore, how to extract and manage the requirement from the simulation design in the cockpit man-machine interface; how to develop the algorithm logic of the controller more efficiently and other issues will be studied in the work of interactive simulation later.

REFERENCES

- [1] Duan Lin. The evolution and development tendency of human-machine interface in cockpit [J]. *Civil Aircraft Design and Research*, 2017 (1): 7-11.
- [2] Dong Dayong, Yu Jinhai, Li Baofeng, Chen Yingchun et al. Cockpit human factors of airworthiness verification technology [J]. *Chinese Journal of Aeronautics*, 2016, 37 (1): 310-316.
- [3] Qiu Weilong, Chen Guoxing. Key technology for modeling and simulation of aircraft virtual instrument [J]. *Aircraft Design*, 2014 (3): 49-54.
- [4] ZENG Yan-yang, LIU Shu-fen, GAO Wu-xing. Research of Applying MVC Framework in Command and Control Simulation System.
- [5] Fang Youqiang. Discussion on Reducing Coupling Measures in MVC Architecture[J], *Network and Information Engineering*.
- [6] Ningxia Personnel Testing, Ningxia, Yinchuan, Zhou Yongping. The study of MVC pattern in software design applications[J], *China Computer & Communication*, 2009.11
- [7] ZHANG Shao-lan, XING Guo-bo, ZHANG Su. Study on MVC design pattern [J]. Study on MVC design pattern, *JOURNAL OF SHANDONG UNIVERSITY OF ARCHITECTURE AND ENGINEERING*, June 2004.
- [8] Praveen Gupta et. al. / (IJCS) *International Journal on Computer Science and Engineering* Vol. 02, No. 04, 2010, 1047-1051 MVC Design Pattern for the multi framework distributed applications using XML, spring and struts framework.
- [9] XU Zhao-hui, FAN Yin-ting, Research and Implementation of MVC Design Pattern in Development of.NET with Three Tier Structure, Vol. 15 No. 2 *Journal of Bei jing Electronic Science and Technology Institute* Jun. 2007.
- [10] LIU Liang, HUO Jianqing, GUO Yugang, YUAN Quan, WANG Xiaopu. Design and actualization of universal MVC pattern [J]. *JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA*, Jun. 2010.
- [11] Wei Peng-zhi. Human-Machine Interactive Design and Realization based on the MVC Framework [S]. *Electronic Technology & Software Engineering*, 2001.
- [12] WU Hao. A Model-based User Interface Development Method [J]. *COMPUTER ENGINEERING & SOFTWARE*, 2015, 36 (8): 7-11.