

Study on virtual prototype for complex system

Gang Dong^a and Zhen Nie

School of Mechanical Engineering, Xi'an University of science and Technology, 710054, Xi'an, China

Abstract. In order to shorten the design period, and to reduce the cost of unmanned aerial vehicle system design, the design method of complex systems by using virtual prototype technology is studied. Firstly, the design environments of complex system which construct of Rhapsody, Matlab/Simulink, Oracle and VC + +. Secondly, a detailed design process of complex system based on model-driven development was researched. Finally, taking FCS of UAVS for example, a complex system was designed. The simulation results demonstrate that the complex system is feasible, stable, and it can simulate FCS from the architecture, function, behavior, performance and other aspects; support complex software model-driven development; improve the efficiency of the UAVS design.

Keywords: virtual prototype; UAVS; complex system; Rhapsody; model-driven development.

1 Introduction

Complex system is a typical complex embedded hybrid system, it is the core for unmanned aerial vehicle system (UAVS). With the complication of combat environments and updating of combat objects, the architecture, function, behavior, and data communication of UAVS, FCS are getting more complicated than yesterday. Meanwhile, test and verification process can only be carried out at the physical prototype of FCS design stage. So, traditional method of FCS design has high cost and low work efficiency. Virtual Prototype (VP) is an alternative to physical prototype, in which observation, analysis and manipulation of designs can be done in the early stage of design process[1]~[3]. It could provide powerful support for the design and manufacture of FCS. So, we researched the technology of virtual prototype design for UAVS complex system.

Complex system is a distributed simulation-based approach for FCS design and the simulation model is called complex system [4]~[6]. The development models in different engineering realm are combined together to imitate a physical prototype of FCS from the view of architecture, function, behavior and performance. In recent years, research of complex system had carried out in more and more domestic and foreign research institutes. Statemate was used to building models of aviation electronic products and VP verification by M. Romdhanic etc. in France^[7]. Formal methods applied in design and verification FCS[8], and trajectory planning and complex of autonomous flight of unmanned helicopter were developed by Koo etc. of university of California at Berkeley[9]. Space systems department of Honey-well designed the virtual prototype of Avionics systems uses Avionics Integrated Development Environment[10]; Jet Propulsion Laboratory developed and simulated the FCS of the spacecraft using Flight System Testbed[11]; National Luchten Ruimtevaart-laboratorium

^a Corresponding author : dg5594@aliyun.com

developed and experimented the FCS using Fire-By-Wire platform; Beijing University of Aeronautics and Astronautics simulated the non-redundant complex computer of A340, Boeing777 and other similar aircraft by using virtual prototype technology[12]. Comparing to the design and simulation of VP based on physics-platform, in this study based on graphical modeling language for COMPLEX SYSTEM with low-cost, high-efficiency, and it also support the complex software of model-driven development.

2 Design environment of complex system

In recent years, many design tools of VP design in control system are put forward, such as MATRIX family of integrated systems Inc, Statemate of I-Logix Inc, Rhapsody of IBM Inc. Matlab/Simulink is a widely used tool in the control engineering filed; Rhapsody is a technology of system engineering designer in the entire application of engineering-oriented, including State chart language based on formal semantics and powerful functions on behavior level design, verification and evaluation. So, we constructed a design environments of COMPLEX SYSTEM based on Rhapsody, Matlab / Simulink, Oracle, Doors and VC ++, and the overall structure is shown in Fig. 1.

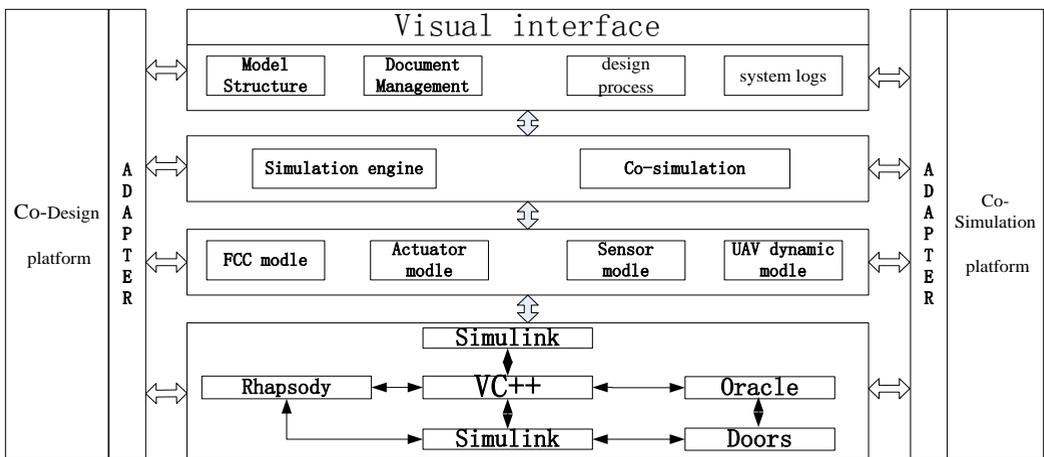


Figure 1. Design environment of COMPLEX SYSTEM

The environment had a four-level structure. The first level is the tool's software resources, including four software which is Rhapsody, Matlab / Simulink, Oracle, Doors and VC ++; The second level is the Virtual model layer , providing the sensors, complex computer, actuators and UAV dynamic model such as the virtual prototype model generation; The third level is environment that supporting for simulate of complex system and also support hardware-in-the-loop simulation; the fourth level is scheduling management, including the model structure management, document management, design process management and system logs management, this layer also provides visual interface.

The connection of software and function of each software are shown in Fig. 2. Platform of door used to managing the user requirements of FCS; Requirements model, static model, activity model and other discrete model are designed in Rhapsody; continuous model are designed in Matlab/Simulink, such as control law model and dynamic model. Above all, Rhapsody can communicate with Simulink by SYSML profile, this provide co-simulation of COMPLEX SYSTEM; Rhapsody and Simulink can also communicate with VC++ for visualization; all the designed models are stored in Oracle and managed in Doors.

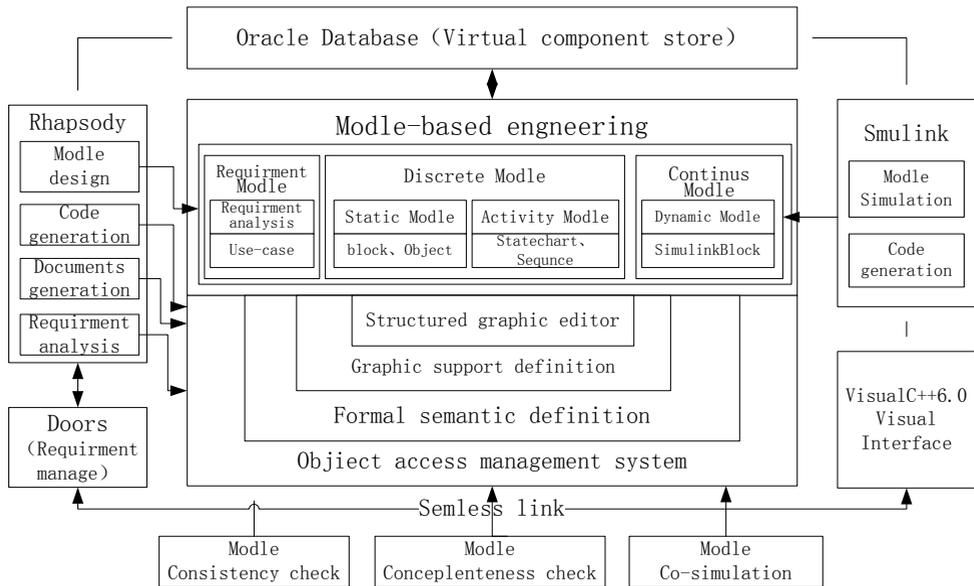


Figure 2. The connection of software in design environment of COMPLEX SYSTEM

3 Design process of complex system

The whole COMPLEX SYSTEM can be designed in this design environment, as shown in Fig.2, control law modules and the UAV dynamic models are designed in Matlab/Simulink, control logic modules, sensor models and actuator models are designed in the Rhapsody. All of the designed models can be stored in Oracle database and also can be managed in Doors.

Image approach is used to creating clear and accurate system specification in a typical Model-driven development (MDD) process [13], aim to discovering and removing various fuzzy conceptions and errors at the early development stages. As shown in Fig.3, fist of all, the SYSML language and Simulink language are used to describing the top level, according to the SYSML language and Simulink language established an abstract model of application (platform independent model). And then, through the interface region, build the physical model (platform specific model). Finally, through API mapping mechanism to generating application program on collaborative design platform, and through RTI mapping mechanism to generating application program on collaborative simulation platform.

Two kernel languages for building COMPLEX SYSTEM are run in this paper, such as SYSML and Simulink, to finishing design and verification for COMPLEX SYSTEM. The procedure involves the follow steps:

- 1) The FCS stakeholder requirements were analyzed, refined and translated into system requirements, system requirement is run in a top-down approach based on Use Case diagram, including boundary definition, actor, Use Case , and communication association between actor and Use Case.
- 2) The processes of level decomposition, definition and description of FCS function which based on Block diagram were analyzed, including boundary definition, functional description, and signal interface specification between modules, and so on.
- 3) Design of function and behavioral description based on Object diagram and State Chart diagram are used to design Activity behaviors in detailed.
- 4) During the design process, stochastic verification and validation for COMPLEX SYSTEM are run based on co-simulation experiment, in order to checking whether system functions are accurate.

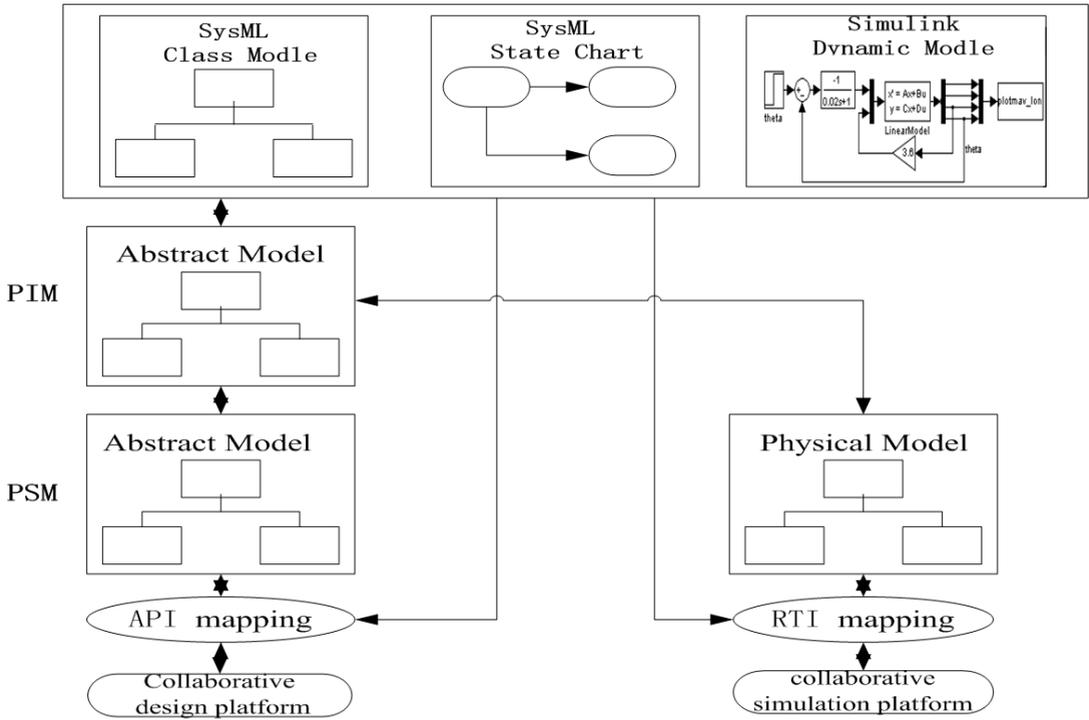


Figure 3. Model-driven development process

4 Design of UAVS complex system

A small UAVS is a common system platform, can add a camera, infrared instrument and other mission equipment for reconnaissance, patrol, or electronic interference and other special tactical missions. Put UAVS FCS as an example, detailed design of COMPLEX SYSTEM is discussed.

4.1 Requirements analysis of complex system

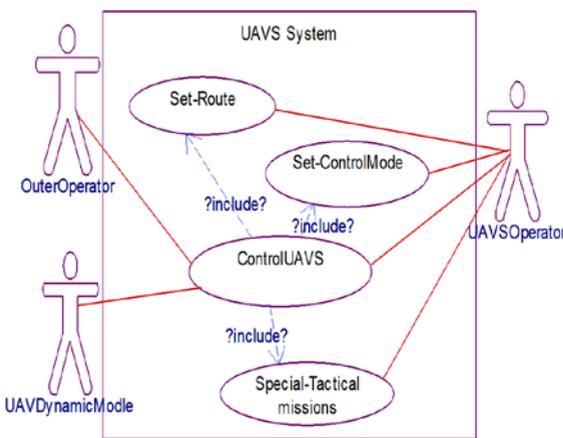


Figure 4. (a) Top Use Case Diagram of FCS

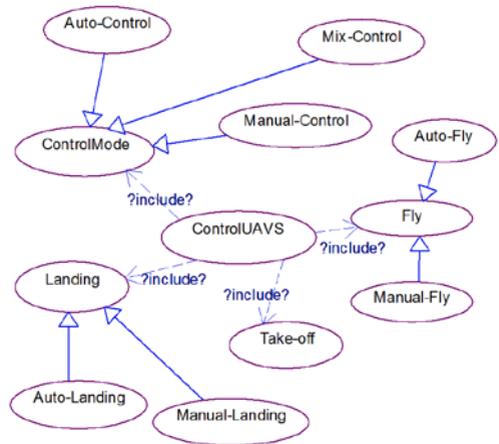


Figure 4. (b) Sub Use Case of "ControlUAVS"

COMPLEX SYSTEM requirements analysis is described by the Use Case diagram. Fig.4 (a) represents the top Use Case diagram of the FCS. Actors that interact with the FCS are "Outer

Operator”, “UAVS Operator” and “UAV Dynamic Module”. Within the FCS includes four other Sub-Use Cases, such as “Set-Control Mode” Use Case. Actor “Outer Operator” only communicated with Use Case “Control UAVS”, but actor “UAVS Operator” communicated with four Use Cases, such as “Set-Control Mode”, “Set-Route”, “Control UAVS” and “Special-Tactical missions”. Use Case “Control UAVS” includes other three Use Cases. Fig.4 (b) is the sub Use Case diagram of Use Case “Control UAVS”. The relations between Use Cases are “include” and “generalization”.

4.2 Level decomposition and definition of FCS function

According to the results of requirements analysis, top-level designed by the Block diagram. The main content includes FCS configuration and functional decomposition. The Block diagram of the FCS is shown in Fig.6. FCS is decomposed into 9 blocks, they are “A-Gyros”, “W-Gyros”, “GPS”, “HV-Sensor”, “HMR”, “Ground-Station”, “Remote Controller”, “Flight Control Computer” and “UAV Dynamic Module”, and “Flight Control Computer” is composed of “Flight Control Logic” and “Flight Control Law”. Blocks communicate with other blocks by port, flow port and interface in Block diagram. The function of every block is shown in Table 1.

Table1. Function of FCS Block

| Block of FCS | Function |
|-------------------------|---|
| A-Gyros | There Attitude angles measuring(roll/pitch/yaw) |
| W-Gyros | There Axes of Angular measuring(roll/pitch/yaw) |
| GPS | GPS Signal measuring(Latitude/ Longitude/ altitude) |
| HV Sensor | Barometric Height/Air Speed measuring |
| Flight Control Computer | Controller of UAVS |
| | |

The ‘’ mark in Fig.5 denotes that the module has a more detailed definition of decomposed by State chart, the detailed definition of block “Flight Control Law” and "UAV Dynamic Module" are described by Simulink block and others are modeled by state chart.

4.3 Designs of functions and behaviour descriptions

Detailed design is based on the top-level design. Each module was designed in a deeper level. The kernel of FCS is a complex computer, which is realized by SYSML and Simulink in a top-down development mode, it covers diversified discrete and continuous control algorithm.

State chart is a basic diagram in SYSML[14], including source state, target state, event trigger, guard condition and action. Behavior of discrete system, like complex computer, can be designed by State chart. The algorithm of state chart is:

$$SS > \text{---} E[G]/A \text{ ---} > TS >$$

Where, SS and TS are source state and target state, they are system states before and after the transition; E[G]/A represents the transition. The meanings of this model is that when event trigger E generates and guard condition G is satisfied at the same time, system enters the transition from SS state to TS state, and active A co-happens. The ‘>’ sign after state name shows the state has an addition reaction.

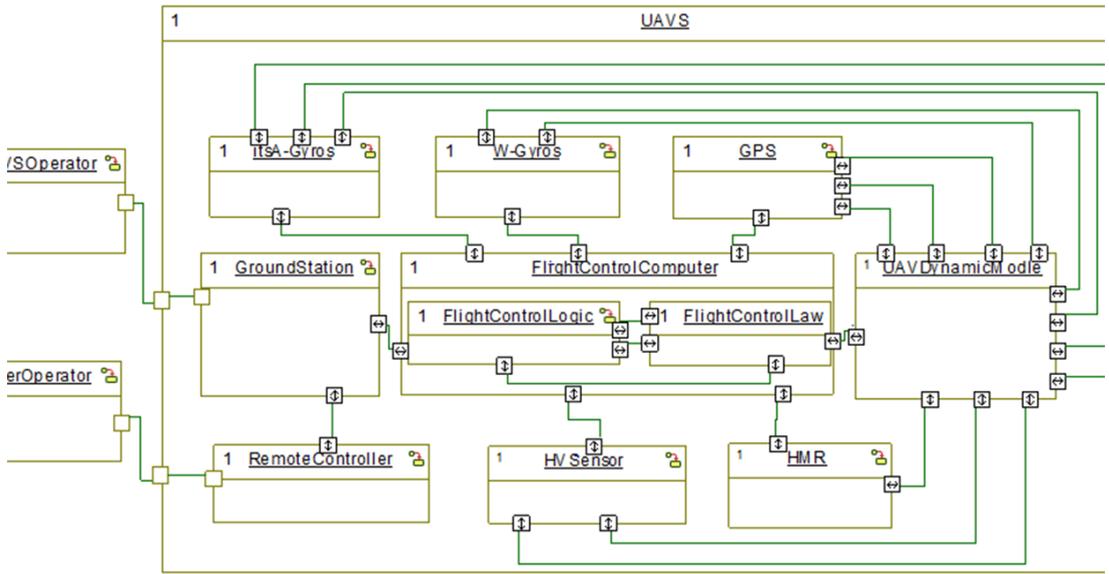


Figure 5. Block Diagram of FCS

As shown in Fig.5, complex computer decomposed into complex logic and complex law. Complex logic is realized by block and state chart; complex law is realized by Simulink. Take complex logic as an example, Fig.6 shown the function of complex logic, the operations, attributes and receptions are allocated to the “Flight Control Logic” block, and the ports, interfaces of complex logic are defined.

The behavior of complex logic is realized by state chart. Fig.7 shown a major state chart of block “Flight Control Logic”, Control Mode describes the states and states transition of UAV complex including “Auto-Control”, “Manual-Control” and “Mix-Control”, and every state has a sub-state chart.

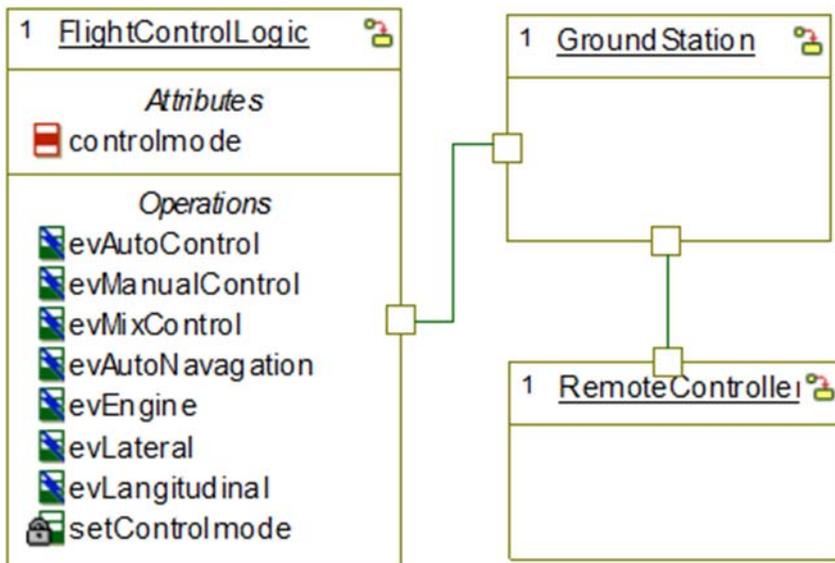


Figure 6. Function of complex logic

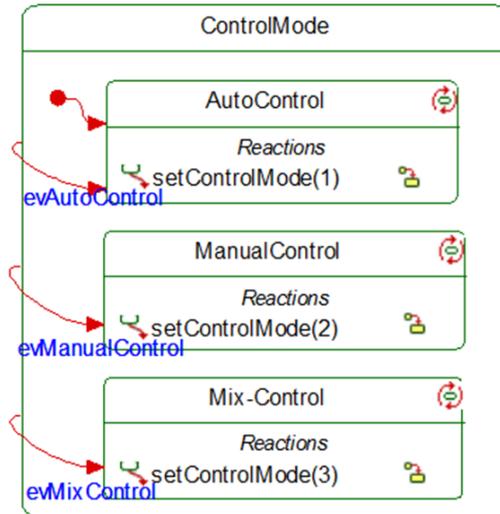


Figure 7. State chat of “Flight Control Logic”

5 Checking and co-simulation of complex system

In order to check whether COMPLEX SYSTEM functions are accurate and to satisfy the requirements of customers, stochastic verification and validation are run based on co-simulation experiment. The design environment of COMPLEX SYSTEM supports two levels check and simulation.

5.1 Syntax and grammar check of the model

Using Simulink\SYSML checker of the design environment check syntax and grammar of the model, consistency and integrity of each module traverse a quick automatically check to find out which there is a warning, error, and then describe and locate them. Designer processed the warning and error according to the location and description of them, some errors are due to incomplete model being there temporary, but some errors need to be corrected.

5.2 Co-simulation of complex system

Test of architecture, functions, behaviors and performance needs the support of UAV dynamic model, principle of bottom-up and first in the outer are generally used in this level test. Firstly, sub-module verification is processed, and then subsystem level and system-level of FCS simulation are held, finally, it is the co-simulation of COMPLEX SYSTEM.

To validate the UAV autonomous navigation capability along with the designated route reconnaissance, electronic interference task and other typical tasks, autonomous navigation flight test is processed.

Fig.8(a) and Fig.8(b) shows the simulation of state charts: the lighted state “Auto Control” is the current control model of “Flight Control Logic” and the lighted state “Auto Navigation” is the current state of “Auto Control”. At the same time in each sub-state associated with the system behavior and performance descriptions, such as the Auto navigation, Longitudinal, Lateral, Engine state, the choice of control law, sensor state, steering state, the current time of flight of UAV control parameter settings and so on. Fig.8(c) shows Autonomous navigation flight trace of UAV at the moment of “Auto Control” and “Auto Navigation”, by the A, B, C, D, E made of five interlinked dashed straight line is the task of route planning, UAV from the A point of entry, solid line represents

the flight trace simulation, we can see the UAV flight trace was very closely to the route planning. Approaches above are very closely with engineering, which can reduce design errors efficiently, improve synthetic design successfully and improve the FCS design speed.

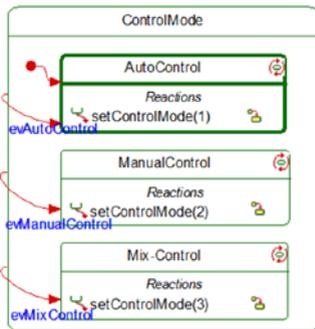


Figure 8. (a) Animated of Fig.7

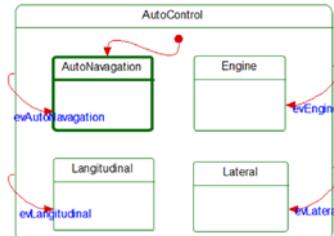


Figure 8. (b) Animated of Fig.8(a)

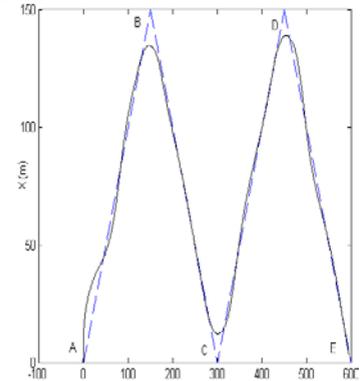


Figure 8. (c) flight trace of UAV

6 Conclusion

COMPLEX SYSTEM is a new technology which can be measured, and realizes a rapid, reliable design of FCS. Using this method we could analyze, verify and validate intended architecture, functions, behaviors and performance of the system much earlier in the development cycle and eliminate misunderstandings and errors in the system specification, support MDD of complex software effectively. It could be used in the design of different kinds of aircraft FCS.

References

1. LI Bo-hu, CHAI Xu-dong, XIONG Guang-leng, et al. Research and primary practice on virtual prototyping engineering of complex product [J]. *Journal of System Simulation*, **14** (3): 336 - 341, (2002)
2. DI Yan-qiang, LI Bo-hu, CHAI Xu-dong, et al. Simulation technology in virtual prototype for complex product [C]. *The Proceeding of International Conference on Agile Manufacturing*, 2003, 303-308.
3. LIU Xing-hua, CAO Yun-feng, SHEN Chun-lin. Model-driven high-level design and verification for complex reactive system [J]. *Journal of System Simulation*, **21**(14): 4284-4287, (2009)
4. YANG Zhong, SHEN Chun-lin, LIU Jiu-fu. A New Design and verification technique for virtual prototypes of UAV complexers [J]. *Information and Control*, **34**(1):81-85, (2005)
5. GONG Yi-fan, PU Xiao-bo, LIU Jun, et al. Rapid Prototype Design Method in Avionics System Based on UML [J]. *Journal of System Simulation*, **19**(4): 749-753, (2007)
6. LIU Xing-hua, CAO Yun-feng. Design of UAV complex system virtual prototype using rhapsody and simulink [C]. *International Conference on Computer Design and Applications*, 34-38, (2010)
7. Romdhani M, Jeffroy A, and Chazelles P.D, etc. Modeling and rapid prototyping of avionics using Statemate [C]. *Proceedings of the International Workshop on Rapid System Prototyping*, 62 - 67, (1995)
8. David T. ESDA: Virtual prototypes aid complex system design [J]. *Electronic Engineering*, (6): 69 - 72, (1995)
9. Bogdanov K, Holcombe M. Statechart testing method for aircraft control systems [J]. *Software Testing, Verification and Reliability*. **11**(1): 39-54, (2001)
10. VOGEL K J, JOHNSON D H. An Integrated COTS Prototyping and Test Environment for Spacecraft Control Systems [C]. *Proceedings of AIAA 93-4701*. Reston, Va. USA: AIAA, 1325-1331, (1993)
11. CASANI E K, THOMAS N W. The flight system test bed [C]. *Proceedings of SPIE*. Bellingham, Wash. USA: SPIE, 1994: 227-236.
12. CHEN Zong-ji, SUN Xiao-zhe, QIN Xu-dong. Study on Virtual Prototyping Testing Platform for Complex Computer System of Civil Aircraft [J]. *Journal of System Simulation*, **20**(15): 4057-4063, (2008)
13. LIU Xing-hua, CAO Yun-feng, SHEN Chun-lin. Model-driven high-level design and verification for complex reactive system [J]. *Journal of System Simulation*, **21**(14): 4284-4287, (2009)
14. Object Management Group. OMG systems modeling language(SysML), version1.0, specification, (2007)