Software Design Flexibility Based on Dual Environment Architecture: Aerial Case Study

A. AbdElHamid
College of Electronic and Information Engineering
Nanjing University of Aeronautics and Astronautics
China

P. Zong
College of Astronautics
Nanjing University of Aeronautics and Astronautics
China

***Abstract*-The software engineering is applying the engineering discipline to the design, development, maintenance of software. In this paper, a dual environment design is introduced to prove the power of the software engineering to integrate the advantages of two different environments (VB.NET and MATLAB). The proposed platform could be applied to any research activities with the same circumstances. Via the utilization of software engineering sub-disciplines (requirements, design architecture, construction, testing), this paper addresses the way steps towards the development of an interactive dual environment platform, using an aerial case study.This platform not only works as a training simulator, but also works as a test bench for the developed aerial modules. The utilized aerial case study is selected for its narrow time constraints, and its modelling complexity to realize the proposed platform performance in the worst case scenario. The MATLAB/Simulink works as a computational background engine to satisfy the required aerial dynamics, while the VB.NET is utilized for the graphical user Interface (GUI), trajectory planning, and hardware interfacing. The results show the ability of the proposed dual environment model to achieve stable data transfer between VB.NET and MATLAB, to satisfy the goal of the simulated mission training, and to provide a detailed high resolution reports about an under test navigation controller.**

Keywords-Software design; Aerial simulator; flexibleplatform simulation

I. INTRODUCTION

Over the last two decades, there is a great potential towards software engineering in all life sectors (commercial, industrial, and military). The software engineering can be classified into many sub-disciplines (software requirements, design, construction, testing, and software quality) [1]. The software design and the software construction determine the final application type either it is used for special purpose or it is used for different tasks. The software engineering plays a great role towards the interface between different applications as some simulator architectures are based on the interaction of individual programmed modules (flight gear and MATLAB/Simulink), and develop a custom 2D map application on MATLAB/Simulink [2]; while another simulator is based on real avionics devices simulation [3]. Moreover, MathWorks dynamic simulation package and Simulink used as a graphical modeling environment, and is seamlessly integrated with the RT-LAB and distributed simulation software through the MathWorks Real Time Workshop (RTW) [4]. However, the utilization of the hybrid architecture by integrating the Visual Basic (VB) and the MATLAB has a great potential in many research fields like: prediction system [5], machine vision /image processing mixture [6].

In this paper, dual environment platform construction is developed for different purposes (aerial sample case study); this platform main task is to support the simulated mission activitiesand other research activities. Therefore, the proposed platform is intended to integrate the power of MATLAB engine with the power of VB.NET in an interactive scenario (one control loop without MATLAB automatic code generation module). The proposed platform objectives are: (1) the utilization of Commercial off-the Shelf (COTS) components for cost-wise considerations, (2) the flexibility towards any future required modifications, and (3) expandable to any other types of vehicles (ground or water surface ones). The evaluation process is measured via the testing of a very comprehensive flight model. The results show the following: (1) the proposed platform design is valid for the aerial case study. (2) The proposed platform is expandable enough to deal with other tasks (used as aerial module test bench). (3) The GUI satisfies the simulated training missions. (4) The utilized case study (navigation controller) is simply integrated in the dual environment structure. (5) The signal playback provides accurate add-on capability for the evaluation process.

II. DUAL ENVIRONMENT COMPONENTS

The proposed platform deeply relies on software engineering, but also uses Hardware-In-the-Loop Simulation (HILS) to provide a realistic responsefor the researchers. The proposed dual environment has the power of two main packages (VB.NET and MATLAB). The MATLAB modules are considered one of the main components of the proposed platform. This mathematical model (which is a Simulink based model) provides the dynamics of the vehicle. Simultaneously, the GUI plays a great role of connecting all the components together, and merges the calculated results in a single visual interface. The proposed platform will initially developed for an aerial simulated missions, then extra flexibility case study will take a place.

A. MATLAB/Simulink Non-Linear Mathematical Model

The proposed platform utilizes the Flight Dynamics and Control (FDC) Toolbox as a background mathematical model, to include the dynamics of the "De-havilland DHC-2 (Beaver)" aircraft into the simulation process. The FDC toolbox for MATLAB/Simulink makes it possible to analyze aircraft dynamics and flight control systems within one software environment (MATLAB). This toolbox has been set up around a general non-linear aircraft model, developed and constructed in a modular way to provide maximum flexibility for the user. The core of the aerial mathematical model consists of twelve Ordinary Differential Equations (ODE's or state equations), and a large number of output equations [7]. The original open loop "OLoop1" simulation model is used to obtain non-linear aircraft responses due to custom control signal inputs.

B. VB.NET SOLUTION

Microsoft Visual Studio .Net is a perfect solution for building modern and fancy Graphical User Interfaces (GUI). Moreover, VB.NET is a multi-paradigm high level programming language, provides the ability for developers to produce rapid and convenient software applications [5]. Thus, VB.NET is selected to be the development language for the proposed platform for its high level capability towards rapid development. VB.NET development utilizes the modular concept development, not only for cost reduction, but also for providing more flexibility for any design changes or any uprising modifications [8].

The proposed platform GUI console is designed and implemented to provide an intuitive display as illustrated in figure 1. GUI is divided into three parts: the graphical area, the tabular area, and status area. The graphical area contains all static/dynamic types of data (maps, airframe, waypoints, and mission paths) in a graphical form. The tabular area contains numerical data, controls, and gauges. The status area contains all important data which must be in front of the operator.

The geographic reference map is one of the most important modules in the dual environment simulator, which could be deduced from the word "reference". It allocates all main important objects in the system (platform coordinates, airframe, and waypoints). Through the utilization of the GMap.Net module, multiple standard map providers are interfaced with the simulator. It is great and powerful, free, cross platform, open source .NET control. Hereby, through the integration of such module, the proposed GUI console achieves a resolution up to 10^{-6} degree with some selected map providers.

Although the geographic map reference module is settled and established, but the GUI could not achieve a mission success. There is no successful training mission without a powerful and accurate mission planning module. Therefore, whatever the type of the operation (real / simulated), or (remotely piloted/ autonomous); mission planning is an essential module. The proposed development of mission planning utilizes the routing and overlaying capabilities, in addition with the in-memory database like datasets. Moreover, as a standardize step, the mission planning can exchange mission files to/from Google Earth application using the Keyhole Markup Language (KML) format.

Adding hardware equipment to the simulation process provides more trusted results on different scales. Therefore, a real Remote Control (RC) guidance device (which is used by the ground pilot in the real remotely piloted mission) is interfaced to the proposed simulator architecture. Typically, most of the modern RC devices are interfaced via USB, the proposed dual environment simulator platform utilizes the Human Interface Device (HID) classes (developed by Lakeview Research) to interface with the standard remote control guidance device.

III. DUAL ENVIRONMENT STRUCTURE

As VB.NET has some weakness such as the limited calculating function and finite drawing ones, therefore, it is not convenient to implement complex algorithm using VB.Net. Conversely, MATLAB was designed to perform mathematical computation, effective matrix calculation, to analyze and visualize data, and to facilitate the development of new software programs. However, it has less capability to generate human-computer interactive graphic interface that cannot be independent of MATLAB environment. Thus, the integration between these two packages creates extreme advantages with tiny minor weaknesses (like the memory data exchange delay between the processes). As illustrated in figure 2, and due to the nature of the proposed simulator, the MATLAB automation server Component Object Model (COM) interface is used to create the communication link between the GUI console and its backbone computation engine [9]. The automation server interface could be created by returning a COM Object reference of the MATLAB engine. Simultaneously, a new MATLAB command window instance is activated as an evidence of a successful COM connection creation. The proposed simulator data flow diagram is illustrated in figure 3. It is worthy noted that figure 3 explains how the dual environment architecture works; as the GUI console (VB.NET environment) closes the open loop simulation model (MATLAB environment) by capturing the outputs from the workspace, updating each internal initial condition for the open loop model, providing a new control inputs and orders a new simulation loop.

IV. RESULTS

The testing results show that the dual environment platform satisfies the ground pilot training requirements. The simulator GUI illustrated in figure 1 shows a sample mission path in blue color, starts with a green balloon, passes through some middle way points, and ends with the red balloon point. The aircraft symbol is overlaid according to the mathematical model feedback coordinated. The tabular area contains the simulation controls that shows every parameter of the output state vector in front of the trainee; while the status area (illustrated as a narrow bar at the bottom of the GUI) indicates the connected RC device status, the current zoom level, and the aircraft coordinates. The sample simulated mission (trajectory following) shows the capability of the operator to control the aircraft and follow the designed path via the utilization of the dual environment architecture. Moreover, the proposed simulator has the following advantages over the original FDC toolbox: supporting multiple maps providers through an innovative GUI, and mission planning, integration of the RC guidance device, recording and signals playback for the whole mission.

To satisfy the second objective of the proposed platform, an extra case study is conducted as a sample of an uprising required modification. This case study states that the proposed platform (which is used as a simulator) is required to be used as a test bench, to test and evaluate a Simulink fuzzy logic navigation controller. figure 4 illustrates the platform flexibility to satisfy the simulator purpose, the test bench purpose, and any other future demands. For the test bench purpose,theunder-test controller is a MATLAB based module, so it is assigned to the MATLAB environment to be connected to the aircraft model inputs. The inputs for this controller are the range and the azimuth of the next predefined waypoint. As the waypoint is related to the mission planning, these parameters will be calculated in the VB.NET environment and then transmitted to under-test fuzzy controller. The recording/playback module is also based on dual environment structure as the VB.NET writes the recorded parameters in M files syntax, then through the COM connection MATLAB is commanded to run the recording file. figure 5 explains the performance of the fuzzy controller to achieve its target. The fuzzy navigation controller tries to minimize the range error with the next waypoint to zero (which means that the aircraft is moving towards the next waypoint as in figure5(a)), and also try to minimize the azimuth error of the next waypoint to zero (which means that the next waypoint is ahead figure 5 (b)). To achieve the navigation task, theunder-test controller generates the required guidance commands to the aircraft control surfaces (aileron and elevator) as shown figure5 (c) and (d).

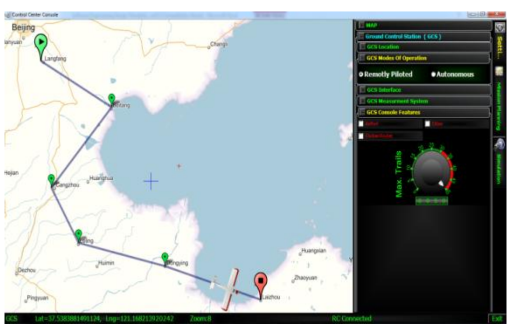

FIGURE I. PLATFORM GUI CONSOLE SHOWS TRAJECTORY FOLLOWING MISSION.

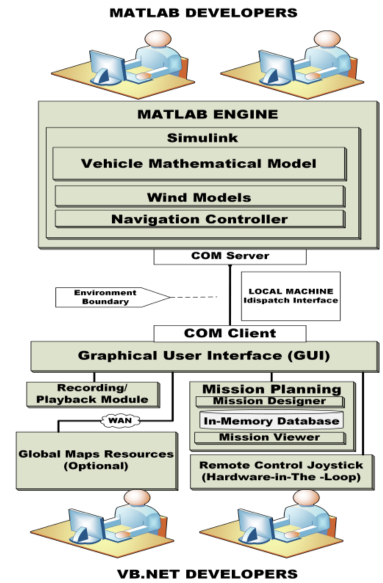

FIGURE II. DUAL ENVIRONMENT OVERVIEW.

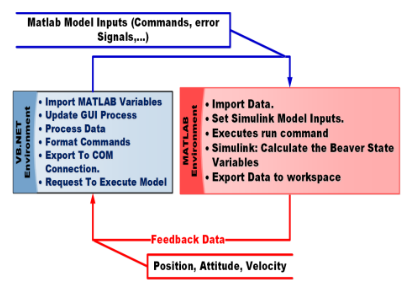

FIGURE III. DUAL ENVIRONMENT DATA FLOW DIAGRAM.

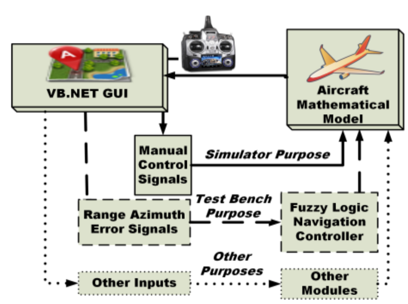

FIGURE IV. MULTI-PURPOSE PLATFORM.

V. CONCLUSION AND FUTURE WORK

This paper presented a new dual environment architecture software simulator. The dual environment architecture is approved via a non-linear aerial MATLAB simulation model with VB.NET GUI.The airframe dynamics are perfectly sensed via the GUI during the simulated mission. The ground pilot training process is recorded with a high resolution (for future analysis/evaluation process via signals playback module). It is worthy noted that the utilized mathematical model is just a sample, and could be changed/modified upon the research need. Thus, the dual environment architecture concept could be expanded to any other type of vehicles (road, sea surface, or even aerospace vehicles). The future expansion for this simulator is to inject the different realistic wind gust conditions to achieve more realistic environmental training.

ACKNOWLEDGEMENT

The authors would like to acknowledge the anonymous reviewers for their insightful comments and suggestions. In addition, the authors would like to thank the Chinese government "National 863 fund" (ref. 2013AA7010435) for their great support.

REFERENCES

[1] P. Bourque and R. Dupuis, "Guide to the Software Engineering Body of Knowledge 2004 Version," *Guide to the Software Engineering Body of Knowledge, 2004. SWEBOK,* 2004.

[2] A. Ajami, J. Balmat, J. P. Gauthier, and T. Maillot, "Path planning and Ground Control Station simulator for UAV," in *IEEEAerospace Conference*, pp. 1-13, 2013.

[3] W. Jianan, W. Wei, Z. Jinhong, and W. Bodong, "Research of a kind of new UAV training simulator based on equipment simulation," in *International Conference onElectronic and Mechanical Engineering and Information Technology (EMEIT)*, pp. 4812-4815, 2011.

[4] S. Zheng, Q. Huang, J. Jin, and J. Han, "Flight Simulator Architecture Development and Implementation," in *ICMTMA '09. International Conference onMeasuring Technology and Mechatronics Automation*, pp. 230-233, 2009.

[5] J. Zhang, R. Wang, and L. Pan, "The System of Lifetime Prediction for VFD Based on VB and MATLAB," in *AICI '09. International Conference onArtificial Intelligence and Computational Intelligence*, pp. 59-62, 2009.

[6] C. Lijun, B. Xuewei, and Z. Xiurong, "Software design of variable-spray control system based on VB and MATLAB," in *2nd International Conference onArtificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, pp. 2398-2401, 2011.

[7] M. Rauw, "FDC 1.2 – A Simulink Toolbox for Flight Dynamics and Control Analysis, 2nd edition," 2001.

[8] B. S. El-Haik and A. Shaout, *Software Design for Six Sigma: A Roadmap for Excellence*: Wiley, 2010.

[9] The MathWorks, "Matlab External Interfaces," *The MathWorks, Inc.,* 2013.