Simulation of Electronic Flight Instrument System of Boeing 787 aircraft

Andreas Stefanidis*
Supervised by: Miroslav Macik†

Department of Computer Graphics and Interaction
Faculty of Electrical Engineering
Czech Technical University in Prague

Abstract

Electronic Flight Instrument System (EFIS) known as glass cockpit is a key component of any modern aircraft. It visualises practically all cockpit instruments and serves as replacement for obsolete solution where most instruments were electromechanical. Flight crews must gain perfect knowledge of cockpit systems of each aircraft type in order to control it safely. For most aircraft types the available simulators are expensive, therefore difficult to access for most people. This paper describes realisation of a semi-professional simulator of Boeing 787 EFIS. Aim of the presented solution is to serve as simulator for demanding amateur users, for testing newly developed aircraft systems in the context of their future use and, in case of successful certification, also as a training device for flight crews. In this paper we present description of individual parts of the EFIS and discussion of currently available simulators for professional and amateur use. Furthermore, individual aspects of our solution are described in detail. Besides standard functions of EFIS it supports simulation of other aircraft systems, e.g. individual panels in the cockpit allowing interaction with the flight management system (FMS). Using a generic interface it allows integration with various systems for simulation of the external environment and physical model of the aircraft. Presented solution has been verified by functional testing and by specific usability tests with both professional and amateur users.

Keywords: Electronic Flight Instrument System, Flight Simulation, Flight Training, Human-computer Interaction, 2D Graphics

1 Introduction

Nowadays, the aviation is once again in the rise and global manufactures of civil aircrafts are introducing new variants of current aircraft models such as Boeing 747-8 or Boeing 737 MAX and even developing completely new models such as Airbus A350 XWB, A380 or Boeing 787. This trend also evokes increasing number of people interested in flying, not only as aircraft passengers, but also as those who like to try pilot experience.

After the events of 9/11 regulation in aviation security has tightened and access to the flight deck has become almost impossible for regular people. Flight simulators, either professional or amateur, provide more or less real flight experience, which solves this issue to some extend. Professional simulators usually provide experience close to a real flight, but access to them can be very costly. The other option is usage of amateur simulators, which can be used in a home environment for an affordable price. These simulators are usually not even close to the quality level of the professional devices. Mostly, they are product of the entertainment industry, such as PC games, which are mainly aimed on amateur users. Therefore they do not provide complex simulation.

The Boeing 787 is same as the vast majority of current airliners equipped with a glass cockpit – a cockpit consisting of LCD panels instead of electromechanical devices. This system is called Electronic Flight Instrument System (EFIS). Main goal of this work is to implement a simulation of Boeing 787 EFIS, which would allow all users, even those who do not have access to professional simulators, use a system providing highest possible quality of simulation of all systems in the cockpit.

This work focuses on following areas:

- On group of people who are interested in aviation and would like to experience control of an airliner but professional simulators are unavailable for them. Quality of home simulators is not sufficient for them, and thus they are seeking for a product which would provide, in home conditions, simulation comparable to professional solutions for an affordable price.
- To provide environment for testing of aircraft systems being developed in context of their future use.
- To serve as certified flight training device – Basic Instrument Training Device (BITD, see further) and/or Flight and Navigation Procedures Trainer (FNPT, see further).

*stefaan1@fel.cvut.cz
†macikmir@fel.cvut.cz

2 Background

This chapter introduces the Boeing 787 Dreamliner and basic terms from the field of aviation instrumentation necessary for comprehension of this text.

2.1 Boeing 787

Boeing 787 Dreamliner is currently the last model of civil aircraft from the Boeing Corporation. It is a twin-engine aircraft, which should eventually replace the Boeing 767 model. It is the first civil aircraft that use composite materials instead of aluminium in such a large scale. Furthermore, thanks to a new design of engines, the noise produced by the plane is decreased by 60% and the fuel consumption by 20% respectively [18]. Modern construction of this aircraft incorporates modern design of the EFIS in ins cockpit.

2.2 Electronic Flight Instrument System

In this work we focus mainly on the simulation of LCD panels which are present in an aircraft cockpit. The main part of the Boeing 787 glass cockpit consists of five fifteen-inch LCD panels. These LCD panels are then virtually divided into smaller parts (see Figure 1):

- Primary Flight Display (PFD) – Display that provides pilots with the most important flight data like airspeed, altitude, attitude, heading etc.
- Multi-Function Display (MFD) – MFD can display various panels according to data that are required at a particular situation. Specifically it can display navigation display (ND), system display (SYS), electronic checklist (CHKL), Control Display Unit (CDU), Information Display (INFO) and Communication display (COMM).
- Engine Indicating and Crew Alerting System (EICAS) – Provides flight crew with information about engines condition and also displays annunciations for the crew.

Figure 1: EFIS parts [1]

The cockpit is also equipped with Integrated Standby Flight Display (ISFD), Head-Up Displays (HUD) and Electronic Flight Bags (EFB). Normally, these systems are not vital for successful flight and description of these systems is out of scope of this work (however detailed information can be found in[14]).

2.3 Flight simulators

Flight simulators can be divided into two groups – amateur and professional. The former group incorporates mainly PC games, which allows users to experience the role of a pilot with no need for expensive flight training. This category is represented for example by Microsoft Flight Simulator [9] or XPlane [8]. The latter group represents professional simulators that are used for training of flight crews and therefore must satisfy strict requirements for accuracy of the simulation. Based on the level of the simulation, there are four main categories of professional flight simulators:

- Basic Instrument Training Device (BITD) – A ground based training device. It may use screen based instrument panels and spring-loaded flight controls, providing a training platform for at least the procedural aspects of instrument flight [7].
- Flight and Navigation Procedures Trainer (FNPT) – A training device which represents the flight deck or cockpit environment including the assemblage of equipment and computer programmes necessary to represent an aircraft in flight operations to the extent that the systems appear to function as in an aircraft [7].
- Flight Training Device (FTD) – A full size replica of a specific aircraft types instruments, equipment, panels and controls in an open flight deck area or an enclosed aircraft flight deck, including the assemblage of equipment and computer software programmes necessary to represent the aircraft in ground and flight conditions to the extent of the systems installed in the device [7].
- Full Flight Simulator (FFS) – A full size replica of a specific type or make, model and series aircraft flight deck, including the assemblage of all equipment and computer programmes necessary to represent the aircraft in ground and flight operations, a visual system providing an out of the flight deck view, and a force cueing motion system [7].

Created EFIS should meet the requirements for professional simulator from category BITD and/or FNPT, because for this category is not required complete simulation of all aircraft systems.

3 Related work

This chapter provides a brief review of currently available solutions providing relevant functionality.

Abacus 787 is a plug-in module for Microsoft Flight Simulator X that provides simulation of Boeing 787 aircraft. Although, this system provides visualisation of EFIS visually close to reality, the functionality is not optimal (see Figure 2). The panels show only fraction of the data needed in a real flight and the Flight Management Computer (FMC), which is a vital component, is missing completely. This prevents the system from providing realistic experience [17].

Figure 2: Abacus 787 [17]

Project magenta (PM) is EFIS simulator of Boeing 737NG, 747-400, 777 and 757/767 or Airbus A320, 330 and 340. Project Magenta provides not only simulation of the EFIS (see Figure 3) but also many other systems such as control panels or an instructor station. Most current amateur cockpit simulators are based on this project. However, it does not provide simulation of the Boeing 787 EFIS. It is not an open project, therefore it is not possible to extend it in order to simulate B787 EFIS.

Figure 3: Project Magenta [4]

Thales 787 is a professional simulator of Boeing 787 certified as category Level D from Thales Group (see Figure 4). Because its price around USD 15-20 million it is not available to most regular users.

Figure 4: Thales 787 [5]

4 EFIS simulator architecture

This section describes the architecture of the created simulator and some specific issues of the implementation.

The implementation is based on C++, OpenGL for graphical output and XAudio2 for audio output respectively. For communication with the Windows OS it uses WinAPI interface.

A modular design has been used in order to achieve an universal solution, where individual independent components are created as DLL libraries. The systems consist of three basic composes:

- The program core, which provides simulation of the EFIS itself.
- Module for communication with simulation platform that provides simulation of the external environment and physical model of the aircraft. The system can be extended in order to support another simulation platform by using appropriate version of this module.
- Module for communication with peripheral devices such as joystick, physical panels etc.

Because the program consists of several modules that work independently, it was necessary to ensure that these parts do not slow down each other, especially parts operating in precise time intervals. We used multi-threaded design to satisfy this requirement (see Figure 5).

EFIS essentially requires multiple displays, in our case five. Therefore it is necessary to allow our system to be displayed on multiple monitors as well. Graphic cards that support connection of more than two monitors at same time are still not very common. Therefore it was necessary to design the system to support multiple graphic cards at the same time.

On each monitor the content is displayed in an independent window, which guarantees hardware acceleration for each display. Therefore it was necessary to design rendering loop to be able to render into multiple windows simultaneously and work effectively as well. Our solution for

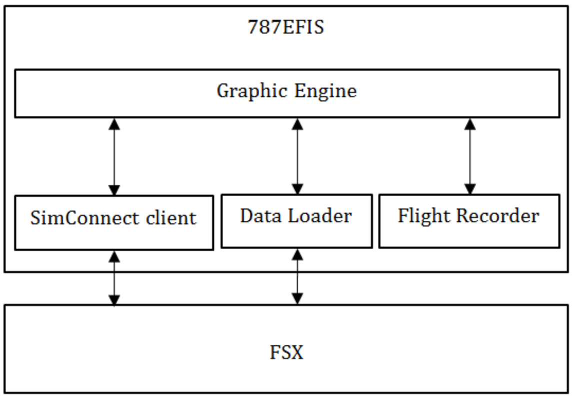

Figure 5: Structure of threads

this is to use an independent thread for each monitor (see Figure 6). All these threads contain an independent rendering loop. If we have more CPUs and/or GPUs this solution is significantly more efficient than a naive method (rendering all windows in one thread) [3]. On the other hand it requires synchronisation between threads and data sharing between different graphic cards. All created threads have access to shared memory where all necessary information about the current simulation state is stored.

It is also possible for users to choose on which monitor and position will be the individual parts of EFIS displayed. This customisation is possible by using configuration file.

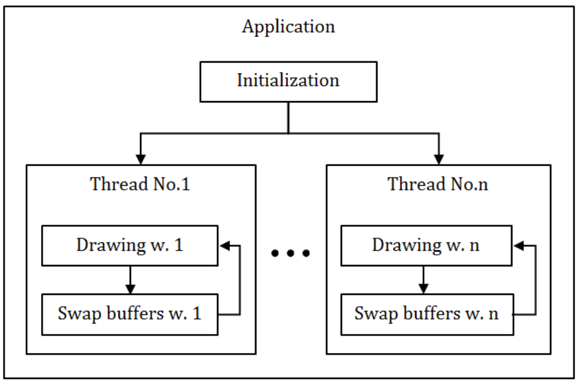

Figure 6: Rendering with multiple threads

Besides the rendering loop, any Windows® application contains its own loop for receiving messages through which it reacts to external events such as system messages, status information about mouse and keyboard etc. Due to the fact that our program contains multiple windows, it was necessary to adapt message loops of particular windows. Consequentially it was necessary to separate messages on those important for each application window and those important for the entire program (see Figure 7).

In order to allow interaction with the created EFIS our solution also contains simulation of some control panels from the cockpit. For their simulation we created a custom

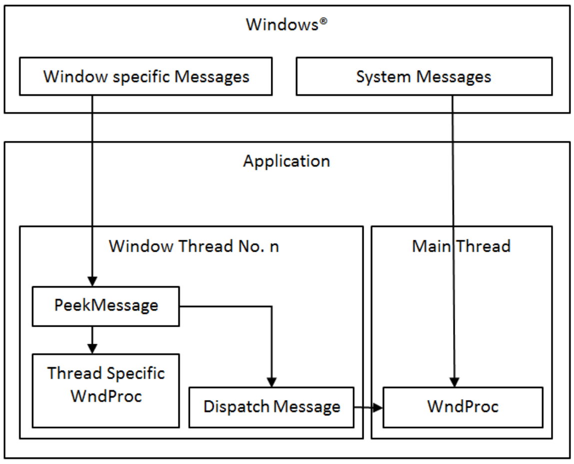

Figure 7: Processing messages

system. It is able to display these panels and react to events trigged by the user as well.

Another important component of the created simulation is Flight Management Computer (FMC). FMC contains three databases:

- Navigation database – contains informations about all airports and all navigation points.
- Performance database – contains performance characteristics of the aircraft.
- Airline modifiable information – contains airline specific data.

FMC is then able to calculate and provide important informations to crew and other systems. For example informations about minimum and maximum speed at particular situation or navigation information displayed on Navigation Display (ND) or Control Display Unit (CDU).

5 External simulation platform interface

Our system serves as simulation of the EFIS. In order to achieve complex simulation, it was necessary to integrate it with a platform that provides simulation of an external environment and physical model of the aircraft. Microsoft Flight Simulator X (FSX) meets most of our requirements, therefore we used this platform [3].

Various data about the external environment and the aircraft physical model are required for operation of the presented EFIS simulator. These data can be divided into three categories:

- Data whose accuracy is critical and hence require frequent updates. This category includes data like aircraft position, pitch, bank etc.

- Data that are not changing rapidly during the simulation and therefore does not require frequent updates. This group includes for example weather information or position of other aircrafts in a particular area.

- Data constant during the simulation, for example navigational information.

It was necessary to take into account these requirements when designing the communication interface between our simulator and FSX. To obtain time-critical data we used SimConnect [10] interface. This interface is part of the Microsoft FSX. It allows communication between FSX and other modules. SimConnect works on client-server principle. Information about the state of the simulated aircraft such as position, altitude etc. can be easily obtained by sending requests via the SimConnect interface into FSX [11]. This principle cannot be applied to all requirements of our system, because not all required data can be obtained using the SimConnects in this straightforward way (e.g. data for systems Traffic Collision Avoidance System (TCAS), Vertical Situation Display (VSD) or terrain radar). Further text describes solutions how to obtain required data that are not directly accessible using SimConnect interface.

For implementation of the TCAS it is necessary to get information about position of other aircrafts in the area. These data can be obtained via SimConnect using method as described in [6]. In the beginning it is sent request to FSX for list of all objects in the simulation. After that is sent for each object request for its actual position and altitude. These data are then requested periodically(see Figure 8). Also it is necessary to watch events when some object enter or leave simulation.

For the VSD, it is necessary to get data about terrain in front of aircraft. It is not possible to obtain data about the terrain using the SimConnect interface. However we can use similar principle which is used to simulate the TCAS. In our program we will first insert an object into FSX. This will serve as an invisible probe. Position of the probe will be then incrementally moved in the axis of the aircraft and at every change of the position we get height of the terrain (see Figure 9). Results of this solution are good and situation on VSD corresponds with relief in the FSX. However, because implementation of sampling does not use any buffer to store and/or preload the necessary data, after change of heading, a resampling of the new relief on the VSD is visible.

To obtain the static data, we decided to use decompilation [21] of FSX data files by using modified version of decompiler BGL2XML [2].

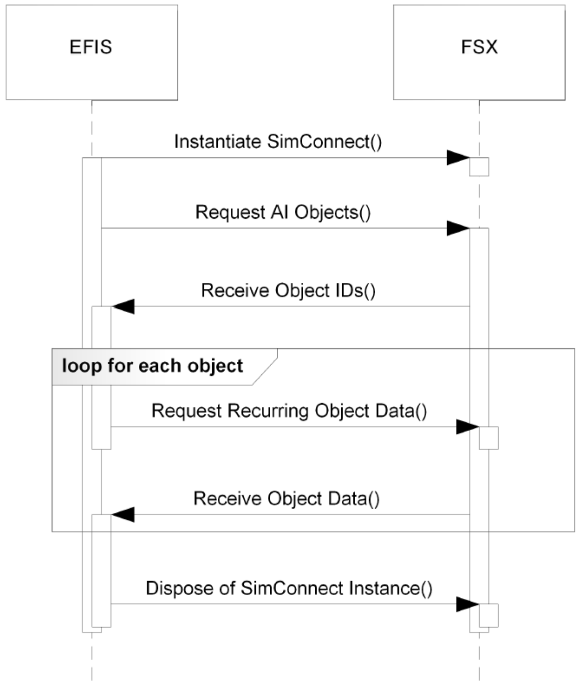

Figure 8: TCAS

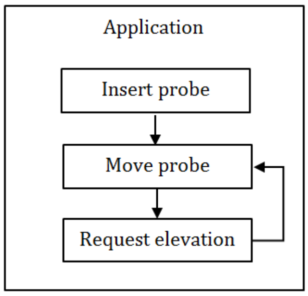

Figure 9: Using probe for getting terrain profile

6 Results and Evaluation

Created system fully implements following EFIS subsystems:

- Primary Flight Display (PFD)
- Navigation Display (ND)
- Engines Information and Crew Alerting System (EICAS)
- Traffic Alert and Collision Avoidance System (TCAS)
- Control Display Unit (CDU)
- Vertical Situation Display (VSD)
- Control panels: Mode Control Panel (MCP), EFIS Control Panel (ECP), Display Selection Panel (DSP), Multi-functional Keypad (MFK), Glareshield panel (GSP)

Furthermore some systems are supported partially, namely: FMC (Flight Management Computer), Synpoptics display (SYS), Auxiliary display (AUX), Centre Forward Panel.

Presented system has been tested for functional and performance aspects. Functional testing was performed by testing with users and by comparison with real system. For performance testing we used standard methods such as unit testing and profiling.

6.1 User interface testing

Because our system is a very specific product, testing of its user interface requires users with at least basic aviation experience. The testing was performed in a manner of an expert review with users who are familiar with function of an aircraft EFIS. Testing itself was made with two different user groups, those who gained their experience with EFIS in simulators and those with experience from a real aircraft. This way we get feedback from people with different perspective. The test also covered intended target categories of users.

The test was performed on a system consisting of two PCs. The first one was running Microsoft Flight Simulator X with one monitor. The other PC was running our EFIS simulator and was equipped with two monitors. Complete testing layout is shown in Figure 10. The system was controlled by a mouse and Saitek X52 Pro joystick with pedals.

Figure 10: Testing layout

The testing itself was divided into several parts:

- Introduction of the project
- Introducing the EFIS, its possibilities, limitations and the way of control
- Pre-test interview
- The actual test. The test consisted of several tasks, each of them aimed to verify a particular part of created system
- Semi-structured post-test interview

Overall assessment from the users was positive. However performed tests revealed certain factors, where it is necessary to improve the created system:

- Hardware problems - Due to hardware limitations, the layout used for testing was not same as layout in a real aircraft. This caused problems during interaction with the system.
- Problems caused by incomplete implementation - Because the created EFIS does not simulate all aircraft systems, it was not possible to test all sub-systems, especially the FMC.
- Problems caused by mistakes in implementation discovered during testing. Some of those issues were already fixed.

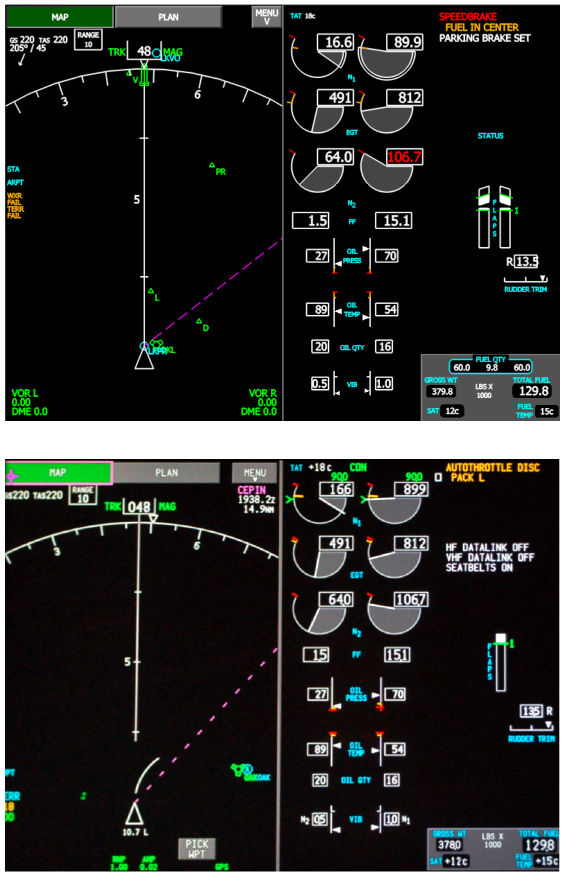

Figure 11: Comparison with the real system (top - our simulator, bottom - real system [5])

6.2 Comparison with the real system

It was practically impossible to compare our system directly with the real system in Boeing 787 or with higher category simulator. Also it was not possible to invite users with experience directly from Boeing 787. Therefore we compared the general functionality of individual systems with other planes. This allowed us to test the system by the functional aspect, because the basic function of the systems in different aircraft types is similar[20]. However comparison with the specific systems of B787 can only be obtained from available sources such as manuals for aircraft [14, 16, 15, 13], photos or videos. In Figure 11 and Figure 12 is a visual comparison of our system with photography of the real system. These figures depicts high fidelity of the simulated system. Most differences between these images are caused by slightly different state between simulated EFIS and the real system on the photography.

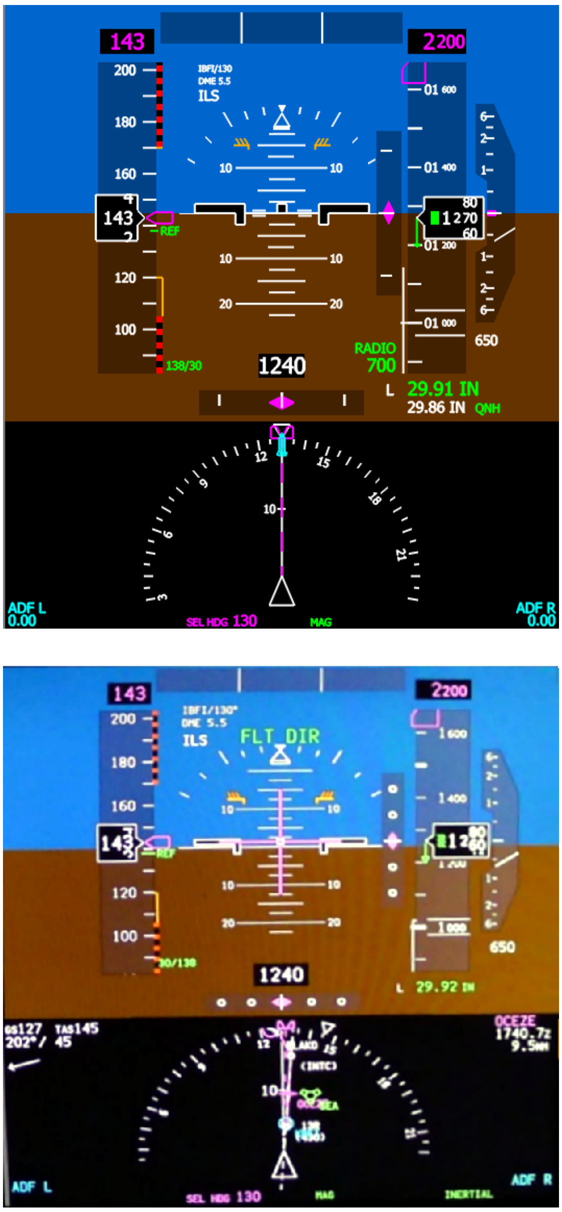

Figure 12: Comparison with the real system (top - our simulator, bottom - real system [5])

6.3 Unit testing and performance evaluation

A set of unit tests was performed in order to evaluate our implementation. Following subsystems were tested:

- Navigation database
- Thread management
- Interface between EFIS and FSX

Created system passed all performed unit tests. It is out of scope of this work to describe the results in detail, however the more information can be found here[3]. Moreover, the performance of the system was evaluated using profiling. Test itself was performed using integrated profiler in MS Visual Studio. During simulator testing we were running EFIS simulation on standard conditions for the period of one minute. From profiling results we concluded some recommendations to improve our system performance. The most significant one was to reduce data transfers between RAM and GPU. For example by using Vertex Buffer Objects or Vertex Arrays. Second recommendation was to improve synchronization between individual threads. The more detailed information and results can be found in [3].

7 Conclusions and Future Work

In this paper we presented design, implementation, evaluation and possible usage of our Boeing 787 EFIS simulator. Our implementation of EFIS could be used in the following scenarios:

- As a sophisticated EFIS simulator for advanced amateur users requiring high fidelity simulation.
- Thanks to its modular design it can serve as environment for laboratory testing of aircraft systems/instruments being developed or for testing interaction between cockpit systems and the flight crew.
- In case of successful certification from Federal Aviation Administration (FAA) [19] or European Aviation Safety Agency (EASA) [7] (or other national aviation authority) it could be used as a professional simulator.

Evaluation of our implementation of Boeing 787 EFIS proved that it provides functional simulation of a real system. We managed to design and implement quality simulation system, which includes most functions of the real EFIS and other aircraft systems.

Presented solution supports all basic function necessary for realistic simulation of an EFIS however the overall experience can be improved by finalizing some missing parts, namely: missing FMC functions, Synoptics display (screens representing status of hydraulic and electrical systems), Electronic Checklist and Head-up display.

Comparison with the real system has been performed using publicly available resources, however precise technical data are necessary for certification of the system as a professional simulator. This certification requires implementation of requirements described in Qualification Test Guide (QTG)[7]. Satisfying these requirements is then costly and may be difficult without information provided by aircraft manufacture.

Our system uses a modular design, therefore it is quite simple to enable support of another aircraft EFIS. There is already demand for EFIS simulator of Boeing 737 MAX. There is also demand for implementation of interface to the Flight Gear simulator – an open source project often used by the research community, e.g. in project [12].

8 Acknowledgements

This research has been done within project Automatically generated user interfaces in nomadic applications, funded by grant no. SGS10/290/OHK3/3T/13 (FIS 10-802900).

References

[1] Alan R. Jacobsen. 787 Flight Deck Displays, Boeing Commercial Airplanes, 2006.

[2] Alessandro G. Antonini. BGL to XML converter, Central Park Informatica, 2004.

[3] Andreas Stefanidis. Simulace systemu EFIS letounu Boeing 787, Czech technical university in Prague [online] https://sdip.felk.cvut.cz /browse/pdfcache/stefaan1_2012dipl.pdf. Master's thesis, 2012.

[4] Enrico Schiratti. Project Magenta [online] http://www.projectmagenta.com/, Accessed 08/11 2011, 2011.

[5] Jason Paur. Try Before You Fly: How Dreamliner Pilots Train Without Lifting Off [online] http://www.wired.com/autopia/2011/09/787-flight-training/, Accessed 11/13 2011 , 2011.

[6] Jessica Zorich. Monitoring AI Objects, Microsoft Corporation, 2008.

[7] Joint Aviation Authorities Training Organisation. JAR-FSTD A - Aeroplane Flight Simulation Training Devices, 2008.

[8] Laminar Research. XPlane, [online] www.x-plane.com/, Accessed 08/22 2011, 2011.

[9] Microsoft Corporation. Microsoft Flight Simulator X, [online] www.microsoft.com/games/flightsimulatorx/, Accessed 08/22 2011, 2008.

[10] Microsoft Corporation. SimConnect SDK Reference [online] http://msdn.microsoft.com/en-us/library/cc526983.aspx, Accessed 11/8 2011, 2008.

[11] Microsoft Corporation. Simulation Variables [online] http://msdn.microsoft.com/en-us/library/cc526981.aspx, Accessed 11/08 2011, 2008.

[12] Sipos, M. and Paces, P. and Reinstein, M. and Rohac, J. Flight attitude track reconstruction using two AHRS units under laboratory condition, Sensors, 2009 IEEE oct. 2009 675-678 ISSN=1930-0395.

[13] The Boeing Company. 777/787 Flight Crew Training Manual, Document Number FCT 777/787, Revision Number: 2, 2010.

[14] The Boeing Company. 787-8 Flight Crew Operations Manual, Document Number D615Z003-TBC, Revision Number: 4, 2010.

[15] The Boeing Company. 787 Flight Crew Training Manual, Document Number FCT 787, Revision Number: 2, 2010.

[16] The Boeing Company. 787 Quick Reference Handbook, Document Number D615Z003-TBC, 2010.

[17] Tim Capps. AVSIM Commercial Aircraft Review, [online] http://www.avsim.com/pages/0909/Abacus/787.htm, Accessed 08/23 2011, 2009.

[18] Tim Nelson. 787 Systems and Performance, Boeing Commercial Airplanes, Flight Operations Engineering, 2005.

[19] US Federal Aviation Administration. Federal Register / Vol. 71, No. 209 , 14 CFR Part 60 , [online] http://www.faa.gov /about/initiatives/nsp/media/consolidated_version.pdf, 2008.

[20] Wikipedia. ARINC 600 Series, [online] http://en.wikipedia.org/wiki/ARINC, Accessed 1/12 2012, 2012.

[21] Winfried Orthmann. FS X BGL File structure, [online] http://www.fsdeveloper.com/forum/showthread.php?t=3710, Accessed 9/13 2011, 2006.