Compact Computing Platform for Future General Aviation in the COAST Project

Pavel Zaykov, Jan Beran, Petr Axman
Honeywell International s.r.o.
Turanka 100, Brno, Czechia
E-mail: {pavel.zaykov, jan.beran, petr.axman}@honeywell.com

Abstract. The CS-23 category in the General Aviation domain requires affordable, greener technologies and support single-pilot operation. The CS2 COAST project addresses these needs by proposing i) a leap-change in computation performance, ii) reduction of development and operational costs, and iii) reduction of Size, Weight, Power, and Cost (SWaPC). In the COAST project, the findings in these technological areas are packaged in a common platform referred as a Compact Computing Platform (CCP). The CCP effort is split in two areas of development - Platform Engineering and Innovative Elements. The first area covers the development of the necessary Hardware (HW) and Software (SW) building blocks. The second area corresponds to the advanced features that provide product differentiation and competitive advantage on the market. In this paper, we briefly describe the most significant SW and HW components from both technological areas and their potential integration in a demonstrator.

1. Introduction
In recent years, Small Air Transport (SAT) has gained an ever-increasing importance across Europe and North America. The SAT concept refers to the usage of a fixed wing aircraft that has from 5 to 19 seats. European Union Aviation Safety Agency (EASA) has listed the SAT requirements in the CS-23 category specification that enables the transportation of people (or goods) between small airports over a regional range.

The principal goals in the CS-23 category are to combine cost-efficient and environment-friendly technologies. Major operational and environment costs are associated with the need of having two pilots on board for 9+ seat commuters. The salary of the pilots contributes approximately 30% of the overall operational costs. The weight of a pilot and the related equipment (e.g., displays and chairs) directly impact the fuel efficiency and thus increase the environment costs. Therefore, the future CS-23 avionics shall provide reduced pilot workload and support single-pilot operation. A way to achieve the requirements is to have: i) a leap-change in the computation performance, ii) a reduction of the development and operational costs, and iii) a reduction of the Size, Weight, Power, and Cost (SWaPC). In this context, the CS2 COAST project contributes to technology enablers that address these challenges. Moreover, the CS2 COAST technologies also consider the following aerospace specifics:

- Safety-critical as the growth of small-aircraft commercial operation is expected to be comparable to the CS-23 safety.
- Long-term support and maintenance.
• Low-volume market that pushes the extensive usage of Commercial-Of-The-Shelf (COTS) components that were already developed in other higher volume industrial domains (e.g. automotive) and invest into the innovations allowing the very same COTS components to be utilized for small-aircraft commercial operation.

In the context of the COAST project, we investigate, propose, prototype, and evaluate Software (SW) and Hardware (HW) solutions that improve the computing performance, and reduce the SWaPC. The findings in these technical areas are packaged in a common platform referred as a Compact Computing Platform (CCP). The CCP effort is split in two areas of development – Platform Engineering and Innovative Elements. The first area covers the development of the necessary HW and SW building blocks for the CCP operation. The second area corresponds to the advanced features that provide technology differentiation and competitive advantage on the market.

The remainder of the paper is organized as follows. Section 2 introduces the CS2 COAST project. Section 3 provides CCP overview and execution timeline. Section 4 discusses the Platform Engineering effort. Section 5 introduces the Innovation elements. Section 6 outline the test campaign. The paper concludes with Section 7.

2. CS2 COAST project

The COAST consortium includes the following four partners: Honeywell International (as a leader), Italian Aerospace Research Center (CIRA), Institute of Aviation (ILOT), and Rzeszow University of Technology. Each partner institution brings an expert knowledge for the selection and development of the most impactful technologies. Prior to the technology development, the COAST consortium worked on the identification of the most relevant enabling technologies for the design of affordable avionic system in the SAT vehicles cockpit. A brief summary of the technology streams is as follows:

• Tactical Separation System (TSS) is an ADS-B-based advanced self-separation system to extend traffic situational awareness. The system will provide the pilot with suggested maneuvers aimed to maintain the required separation minima. The main focus is to develop an affordable system with minimum aircraft coordination. Strong emphasis is given to maintain compatibility with the current and emerging ATR standards and to implement general aviation collision avoidance rules.

• Flight Reconfiguration System (FRS) is an emergency flight path management system that tackles pilot’s incapacitation. The FRS cooperates with related avionic systems, such as navigation systems, flight controls, airport database, data link, and additional sensors. The system does not require equipage by the Flight Management System (FMS); it however enables the integration with the FMS if one is installed.

• Advanced Weather Awareness System (AWAS) provides complete awareness of weather situation with (both observed and forecasted) information assisting the pilot in avoiding entry into atmospherically dangerous areas. The system consists of the on-board application, graphical human-machine interface, ground element, and integrated SatCom. The ground element is based on the MATISSE platform, capable to consolidate weather information from different data sources (as satellite, radar in situ data). The system performs short-term forecasts. The weather information is available on-demand, periodically or based on rapidly emerging adverse weather.

• Compact Computing Platform (CCP) is a scalable, reusable, and reliable platform for advanced cockpit functions, affordable for the CS-23. It features compact HW design, innovated SW architecture and enables simple customization for different aircraft platforms. The key technology elements are parallel-distributed computing, virtualization of I/O logic,
design and analysis tooling, on-board connectivity, and CPDLC (Controller-Pilot Data Link Communications) interface.

- High-Integrity Electronics (HIE) aims at enabling smart actuators and sensors, health monitoring and prognostics resulting in the reduced operational costs and reducing the aircraft integration complexity. The key technology elements are ultra-compact design, wireless connectivity, and small-footprint local processing.

3. Compact Computing Platform (CCP)

In this section, we briefly describe the CCP scope of activities in the COAST framework and their execution timeline.

3.1. Overview

Figure 1 introduces the CCP demonstrator where the most critical CCP elements are integrated. The CCP has two types of interfaces - safety-critical and open-world. The safety-critical interfaces rely on avionics communication standards (e.g., ARINC 429) to fetch data from the aircraft instruments and to deliver a ground communication link. The open world interfaces leverage widely accepted communication standards (e.g., Ethernet and WiFi) to connect to end-user Portable Electronic Devices (PEDs). Both, the input safety-critical and the output open-world interfaces support multiple active communication sessions at a time. Both, the safety-critical and open-world interfaces are protected by a Firewall such that CCP applications are not capable to corrupt the operation of the aircraft instruments.

![Figure 1: CCP demonstrator in COAST](image)

The CCP hosts user applications such as TSS and AWAS. The application business logic and the data storage are created with the help of a Software Development Kit (SDK). A Web browser hosted on a PED visualizes the application content and registers user inputs.

The CCP effort is split in two areas of development referred as Platform Engineering and Innovative Elements. The first area covers the development of the necessary HW and SW building blocks for the CCP operation. The second area corresponds to the advanced features that provide technology differentiation and competitive advantage on the market.

3.2. Design phases

The CCP development has followed the general COAST design process:
• Analyze the high-level CCP requirements and transform them to low-level requirements. Based on the CCP requirements, multiple architecture concepts are defined, and the most appropriate solution has been selected.

• Analyze the State of the Art (SoTA) of existing system, HW, and SW solutions that allow cost-efficient development of the CCP platform and its elements.

• Develop a HW solution - definition of the HW architecture, HW/SW relations, identification of the HW gaps, and development of the CCP target HW solution.

• Develop a SW solution - definition of the SW architecture, identification of the functional gaps, and development of the CCP target SW solution.

• Build a CCP demonstrator - integrate the HW and SW components to a common flight demonstrator.

The scope of the activities has been organized in the following timeline:

• System design - batch 1 covers the preliminary design of the algorithms, requirements refinement from the overall system requirements, revision of the I/O and demonstrator requirements.

• Prototype and Validation - batch 1 includes preliminary SW implementation of the algorithms, preliminary validation through fast time simulation in the simplified simulation environment, tuning of the algorithms parameters, preliminary design and SW implementation of the dedicated Human Machine Interfaces (HMIs).

• System Design - batch 2 includes tuning of the algorithms based on the results of the initial prototyping and validation, final SW architecture and final SW I/O interfaces definition, design update of the HMIs, final refinement of all the affected requirements.

• Prototype and Validation - batch 2 includes the final SW implementation of the updated algorithms, final validation of this SW through fast time simulation in detailed simulation environment, SW code generation in C++, integration of the main code with the HMI code and related validation.

• Integration and Validation (CCP or dedicated platform), where each technology and related HMIs are integrated and validated on the CCP or on the dedicated platform. The final validation is planned on an EV-55 aircraft by EVEKTOR [1].

4. CCP - Platform Engineering

4.1. Overview

The Platform Engineering effort covers the development of the following technologies:

• Hardware Platform that hosts the SW and addressing the HW gaps.
• Platform Software Board Support Package that provides low-level abstraction. It contains HW-specific drivers and routines needed for the selected RTOS.
• Platform Operating System Services that is an RTOS ensuring services such as the platform initialization, partition management, data storage management.
• Platform Middleware that is a set of SW enabling the access to adjacent systems, internal monitors, limiters and other low level functionalities.
• Platform Application Programming Interface (API) and Software Development Kit (SDK) that provide a universal interface to the underlying HW platform.
• Customer Application that is the SW running on the CCP. The customer applications provide the added value of the COAST project to the Pilot operation.

The Platform Engineering has followed two main directions:
Adopting existing technologies is focused on the reusability of HW/ SW from existing 3rd party solutions.

Platform development is integral part of the project and it is necessary for the integration of existing components with new innovative cockpit elements.

To enable an optimal split between the adoption of existing technologies and the platform development, the team has: i) analyzed the state of the art in the area of computing elements applicable for aerospace, ii) defined the expectations and requirements for the future innovative cockpit elements, and iii) built the innovative cockpit demonstrator allowing demonstration and verification of integrated technologies.

4.2. Software Development Kit

The CCP SDK is a package composed of static and dynamic libraries, modules, interfaces, and algorithms. The CCP SDK is developed using C++ and JavaScript. The purpose of the CCP SDK is to deliver easier and faster development of new applications running on the CCP and to demonstrate the capabilities of the underlying computing platform. The CCP SDK is composed of two types of modules – back-end and front-end. The back-end modules target the data processing and are designed to be hosted on the CCP platform. The front-end modules aimed at the data presentation. The front-end modules are supposed to be executed inside a Web browser on a PED.

All used technologies in the CCP SDK are selected after thorough analysis with criteria such as easy-of usage, widespread adoption, detailed documentation, up-to-date community maintenance, and non-restrictive licensing.

The back-end modules in the CCP SDK are as follows:

- Application manager that provides an application module interface class that easy the integration of new applications to the existing environment.
- HTTPServer that offers a full-duplex connection channel between the front-end GUI and the back-end logic. The primary function of the HTTPServer is to handle the HTML and other web resources.
- DataPool module that supports diversity of data sources and serves as a data storage for all SDK-based applications and modules.
- Connectivity manager that serves the connection to the DataPool module.

The front-end modules in the CCP SDK are as follows:

- The JSManager module with predefined examples for applications front-end logic and visualization.
- The graphic features enabled by CCP SDK in the front-end java script are based on an OpenLayers library [2].

5. CCP - Innovative Elements

The team identified technology gaps in several areas that would bring new features and functionalities leading to an increase in the resource efficiency, optimizing the memory allocation, and minimizing the power consumption. In the COAST project, the CCP team has spent research effort on the following innovative technologies:

- Memory Isolation – the performance in safety-critical applications is often defined by the time it takes to accomplish the application execution under worst-case conditions. The worst-case conditions might be defined as a combination of the longest execution path in a multi-path application, the most computation-intensive data input, and the worst contentions over the shared resources. In a multi-core processor, the architecture
contentions are often caused by the simultaneous accesses to a shared memory hierarchy. A shared memory hierarchy is often presented by caches and Dynamic Random Accessed Memory. The proposed innovation is a technology that may isolate the memory requests at each level of the memory hierarchy. The proposed technology has been deployed and validated on multiple potential CCP candidates. The Intellectual Property is protect by two US patents [3, 4].

- Composable performance assurance – this technology enabler addresses the problem of computing the worst-case bound of avionic applications executed on a multi-core processor. The worst-case bounds can be used for Composable Performance Assurance. By composable assurance, we refer to a case where the application performance is preserved independently from the presence or absence of other applications on the same or different cores in a multi-core processor. The composable performance assurance is provided by estimating the worst-case bounds of the application execution time. Based on a study covering the related works, we propose a statistical algorithm for estimating the application worst-case bound based on the Extreme Value Theory (EVT). Furthermore, our approach does not require any architectural modifications to the existing COTS processors neither any specific hardware. To prove the feasibility of our algorithm, we experimented with a synthetic and a real application and analyzed various timing datasets captured from COTS multi-core hardware. The findings of this work were disseminated in [5].

- Real-time simulation of multi-cores - this technology enabler proposes, demonstrates, and evaluates extensions to the QEMU (generic open-source machine simulator) that enable accurate timing estimations. By accurate timing estimations, we refer to estimating the execution time of demo applications on the QEMU which estimations are nearly the same as the execution time on a real target hardware using the processor and memory parameters. The proposed QEMU extensions (referred as mcQEMU) are able to track the timing across multiple simulated cores and keep them in sync. In Figure 2, we present a high-level figure of the mcQEMU and its interface with the QEMU. The mcQEMU regularly receives functional information and responds to QEMU time queries. The proposed QEMU extensions were demonstrated on an ARMv8 architecture and compared with in-order and out-of-order processor execution. The findings of this work were reported in [6].

![Figure 2: mcQEMU - time-accurate simulator](image)

- Improved thermal properties – the work has been carried within the THERMAC project, funded in the CS2 CfP08 wave. Under this activity, the team has developed an open-source testbed environment (referred as Thermobench tool) that is suitable for performance and thermal characterization of a processor under test. The Thermobench tool is independent to the underlying benchmark and is capable to significantly reduce the time necessary for the testing. In Figure 3, we introduce a high-level operation of the Thermobench process. It reads Sensors data (e.g., temperature, CPU frequency), controls the benchmarked process (e.g., memory intensive application), controls the CPU fan (thus reduces the time needed for testing) and logs the performance and thermal data in time-series. The findings of this work were reported in [7].
6. Test campaign
In this section, we introduce the test campaign for the CCP technologies. From the COAST partner perspective, the most important component of the CCP is the SDK. Thus, we provide a testing package of three exemplary applications (PathApp, WeatherApp, and TrafficApp). The purpose of the applications is to exercise the low-level drivers, the connectivity to the avionics instruments and PEDs, and the innovation elements. In the remainder of this section, we introduce a visual presentation of the developed application packages accompanied with a brief description.

Figure 3: Thermobench tool - overview

(a) Path tracking app
(b) Weather app
(c) Traffic app
(d) All demo apps

Figure 4: CCP SDK validation

Figure 4a introduces the PathApp demo application that serves for visualization of the airplane’s path. The path is listed by a thick blue colored line. The flight data such as actual longitude, latitude, and bearing are handled by the CCP SDK back-end. Figure 4b presents the Weather demo application. The Weather information serves for simulation of the collision of the airplane with the potentially hazardous weather phenomena. The hazard areas are displayed by circles of various colors indicating the level of severity. Figure 4c introduces the Traffic demo application. The Traffic information serves as a traffic visualization simulator. It shows the other airplanes and their intended flight paths on the moving map. The actual position and the aimed
landing targets of the airplanes are loaded and computed in the CCP SDK back-end modules and carried via transportation layer to be displayed in the web browser window. Figure 4d lists all three demo applications together in a single Web browser window. Applications can be either combined in any order or run independently. The basic controls allow user to track the airplane, center the view on the airplane, and zoom in/ out. The Table in the right section of the web browser window provides basic information about the airplane current position and actual height.

7. Conclusions
In this paper, we briefly described the CCP for Future General Aviation that is developed in the COAST project. We split the engineering from the research effort and refer to them as to Platform Engineering and Innovative Elements. The HW components identification process led to selection of the COTS components (computing elements, memories, interfaces, and other circuits) allowing long time, safe, and reliable operation with acceptable costs for the COAST program target market. The SW components identification process allowed us to select appropriate SW solution for the safety-critical and open-world computing platforms (e.g., RTOS, BSP, APIs) and to development the missing SW components (enablers) like SDK and Memory isolation. The SW enablers are developed also with the goal to allow easily integrate 3rd parties (e.g., COAST partners CIRA, ILOT, and PRz) technologies that are represented by AWAS, TSS, and FRS applications.

All HW and SW components / enablers are planned to be assembled and integrated into the COAST technology demonstrator and demonstrated on EV-55 aircraft by EVEKTOR.

Acknowledgments
This research has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 945535.

References
[1] “Evektor EV-55 outback aircraft,” https://www.evektor.cz/en/ev-55-outback, accessed: 2020-09-30.
[2] “Openlayers,” https://openlayers.org/, accessed: 2020-09-30.
[3] P. Zaykov and L. Matusova, “Apparatuses and methods for determining efficient memory partitioning,” Patent 10 366 007B2, 2019.
[4] ——, “Memory partitioning for a computing system with memory pools,” Patent 20 180 239 709 A1, 2018.
[5] P. G. Zaykov and J. Kubalcik, “Worst-case measurement-based statistical tool,” in In Proc. of the Int’l Aerospace conference (AeroConf), March 2019, pp. 1–10.
[6] H. Carvalho, G. Nelissen, and P. Zaykov, “mcQEMU: Time-accurate simulation of multi-core platforms using QEMU,” in In Proc. of the Euromicro Conference on Digital System Design (DSD), 2020, p. to appear.
[7] M. Sojka, O. Benedikt, Z. Hanzalek, and P. Zaykov, “Testbed for thermal and performance analysis in mpsoc platforms,” in In proc. of the Int’l Workshop on Cyber-Physical Systems (IW-CPS), 2020, p. to appear.