



(19) **United States**

(12) **Patent Application Publication**
Sivaratri et al.

(10) **Pub. No.: US 2020/0035116 A1**

(43) **Pub. Date: Jan. 30, 2020**

(54) **SYSTEM AND METHOD FOR
CLOUD-BASED FLIGHT MANAGEMENT
SYSTEM FAMILIARIZATION TRAINING**

(30) **Foreign Application Priority Data**

Jul. 26, 2018 (IN) 201841028216

(71) Applicant: **HONEYWELL INTERNATIONAL
INC.**, Morris Plains, NJ (US)

Publication Classification

(51) **Int. Cl.**
G09B 9/08 (2006.01)

(52) **U.S. Cl.**
CPC **G09B 9/08** (2013.01)

(72) Inventors: **Mahesh Sivaratri**, Bangalore (IN);
Rajeev Mohan, Bangalore (IN);
Srihari Jayathirtha, Bangalore (IN);
Raghu Shamasundar, Bangalore (IN);
Mythili Belle, Bangalore (IN); **Sujith
Kalarickal**, Kozhikode (IN); **Sushma
Bidrupane**, Bangalore (IN); **Subhransu
Sahoo**, Bangalore (IN)

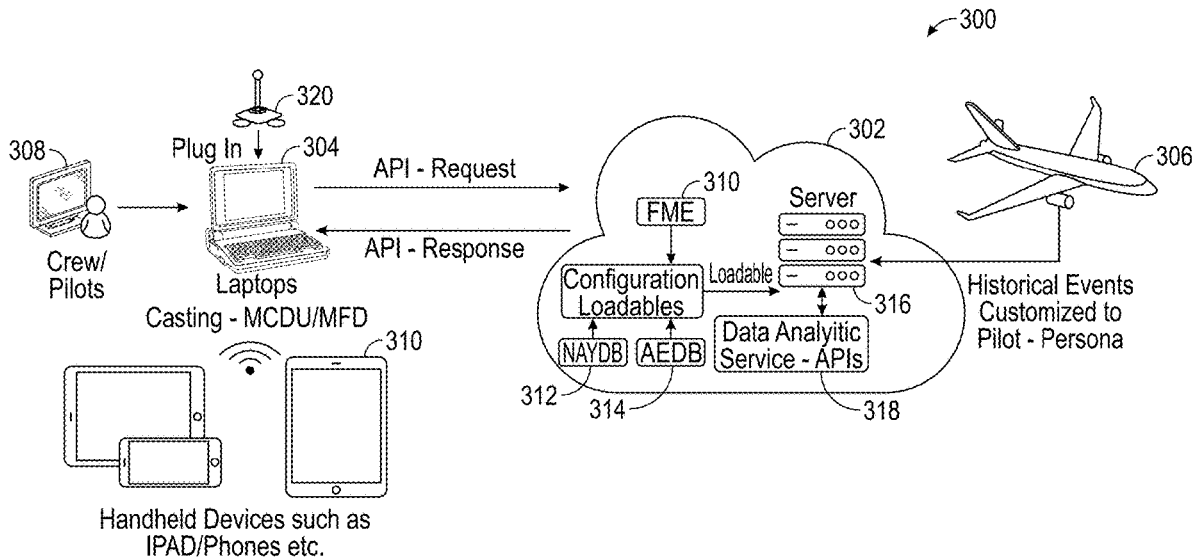
(57) **ABSTRACT**

Methods and systems are provided for flight management system (FMS) familiarization training. The system comprises a user application programming interface (API) that receives a request for a training scenario from a user. A flight management engine (FME) generates the training scenario that simulates the flight mission of an FMS onboard a designated aircraft. The training scenario is generated using the same algorithms as the FMS. A database configuration module retrieves performance and condition data that is provided to the FME for use in generating the training scenario. An external API retrieves external third-party data that is provided to the FME for use in generating the training scenario.

(73) Assignee: **HONEYWELL INTERNATIONAL
INC.**, Morris Plains, NJ (US)

(21) Appl. No.: **16/506,551**

(22) Filed: **Jul. 9, 2019**



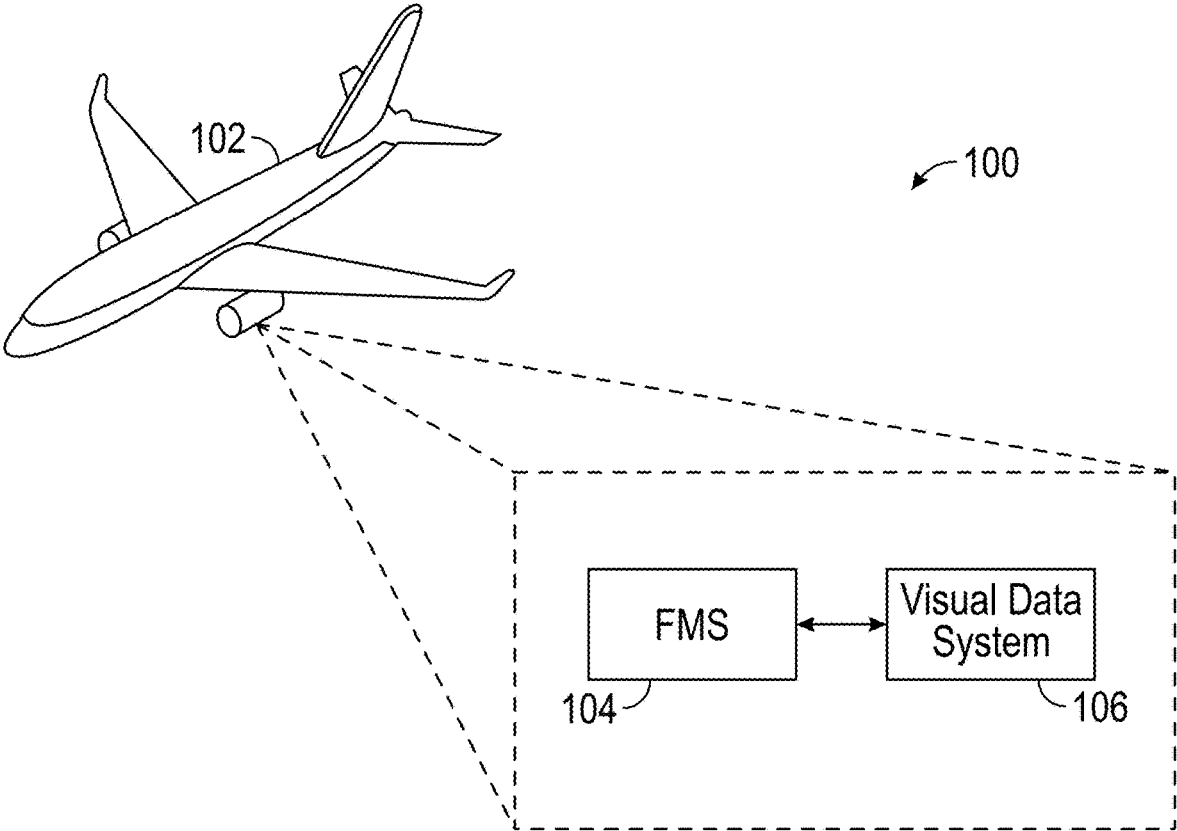


FIG. 1

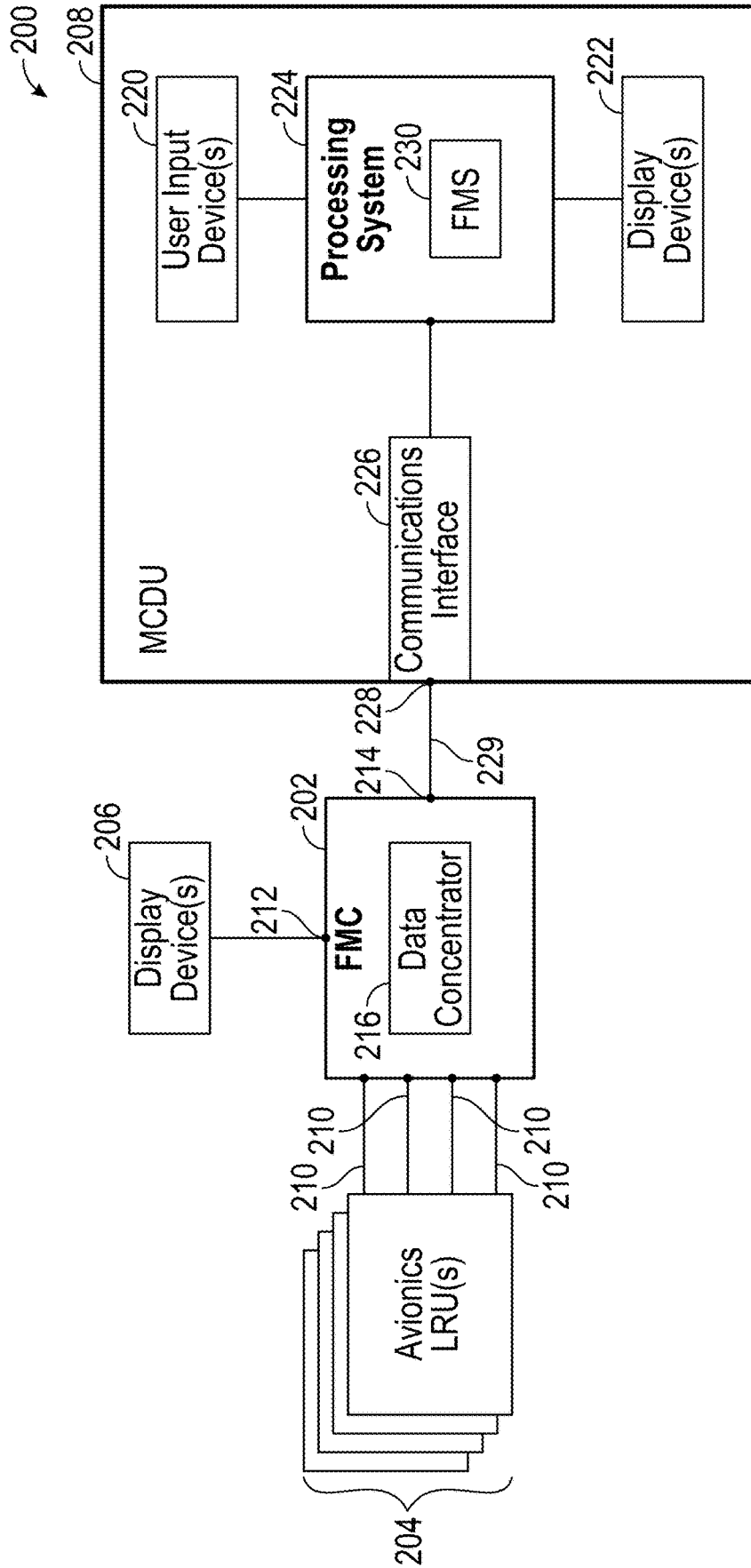


FIG. 2

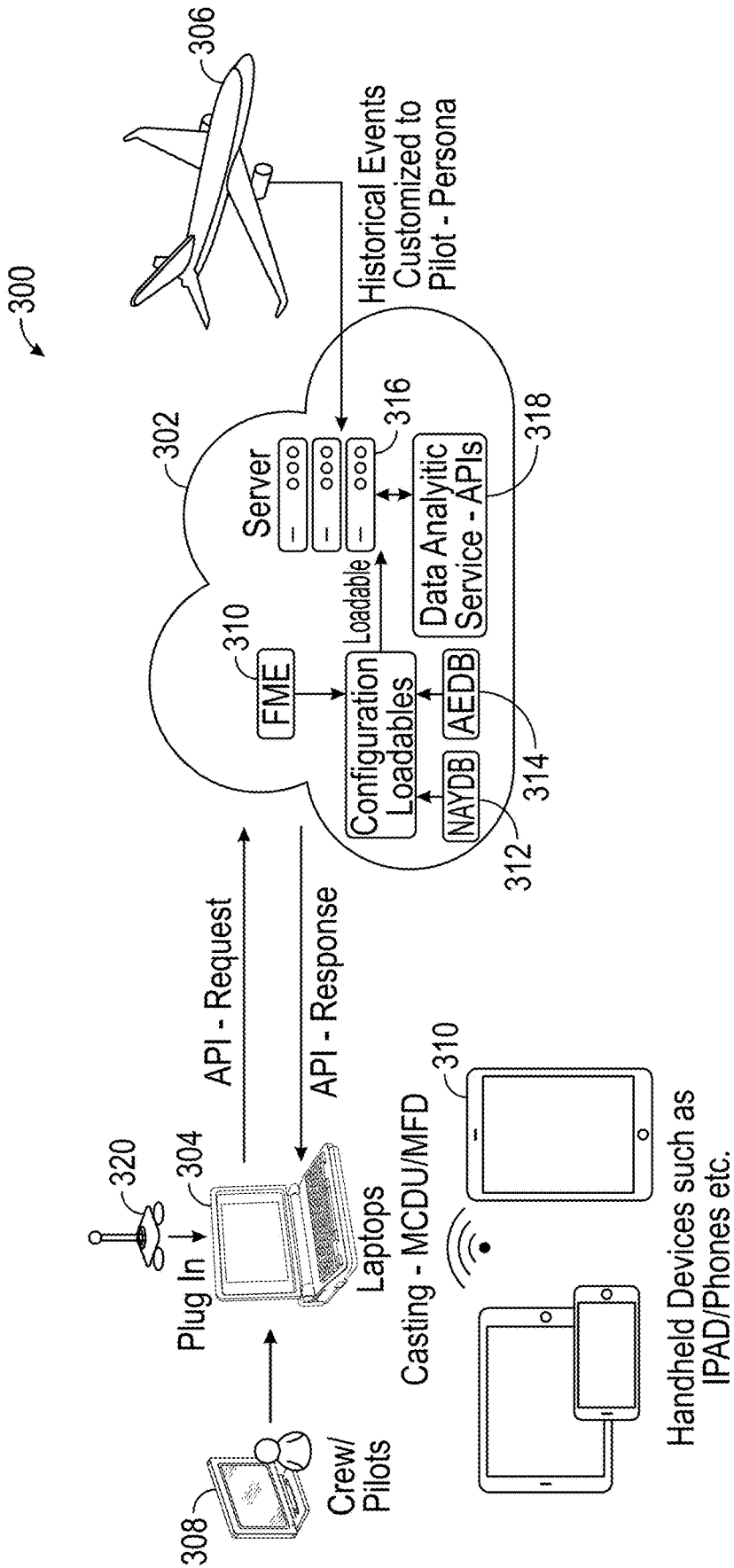


FIG. 3

400

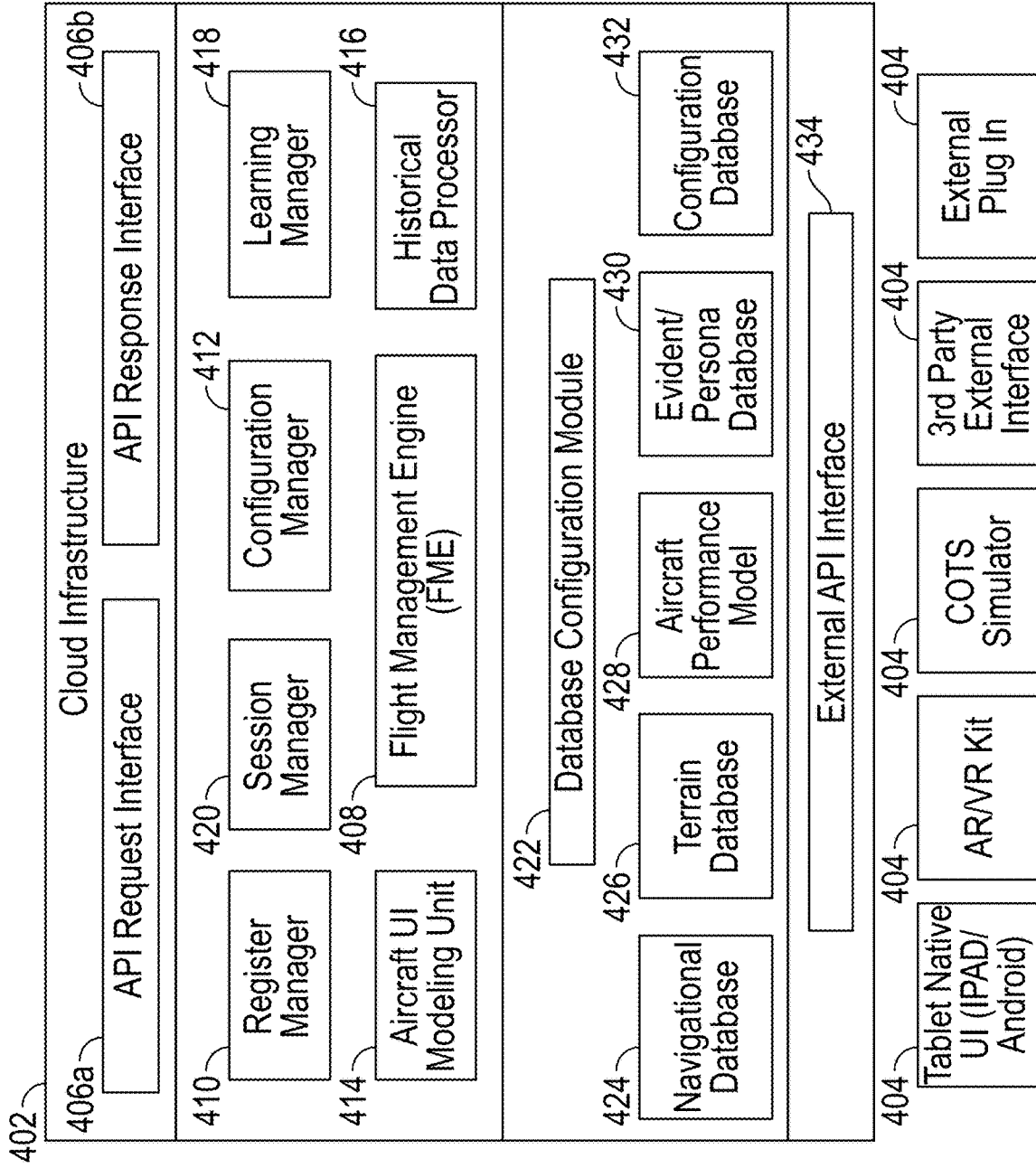


FIG. 4

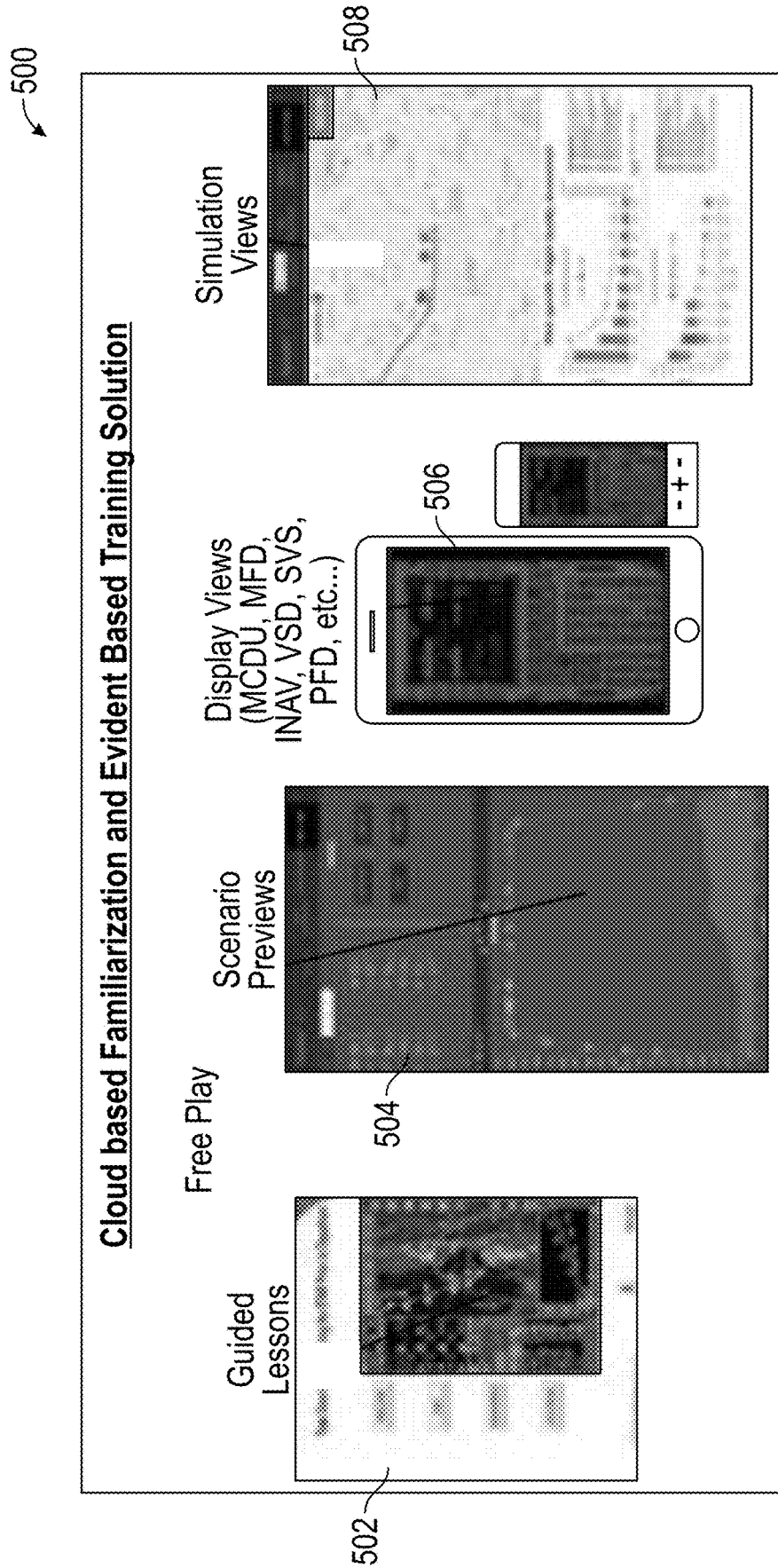


FIG. 5

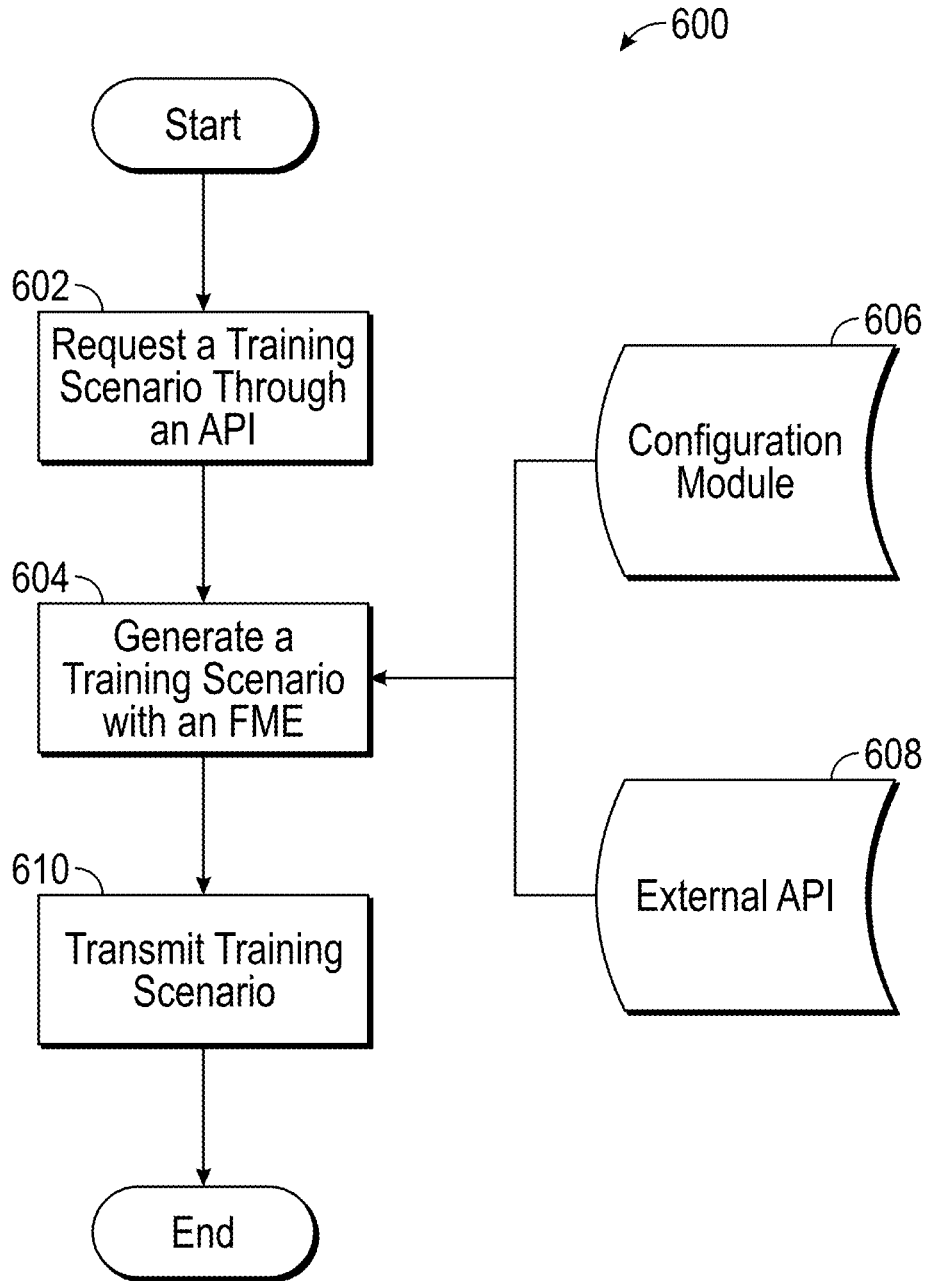


FIG. 6

SYSTEM AND METHOD FOR CLOUD-BASED FLIGHT MANAGEMENT SYSTEM FAMILIARIZATION TRAINING

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims priority from India Patent Application No. 201841028216, titled “SYSTEM AND METHOD FOR CLOUD-BASED FLIGHT MANAGEMENT SYSTEM FAMILIARIZATION TRAINING” that was filed Jul. 26, 2018

TECHNICAL FIELD

[0002] The present invention generally relates to aircraft training operations, and more particularly relates to a cloud-based flight management system (FMS) familiarization training.

BACKGROUND

[0003] Aircraft pilots need regular training with on-board avionics. Consequently, aircraft operators need to expend considerable cost and time in regularly training their pilots on the new and refurbished avionics. One piece of avionic equipment that requires such training is a Flight Management System (FMS).

[0004] An FMS is a specialized computer that automates a variety of in-flight tasks such as in-flight management of the flight plan. For example, the FMS may determine the aircraft’s position using various sensors and guide the aircraft along its flight plan using its navigation database. However, simulation software that is used for training may not accurately reflect the latest onboard version of an FMS. This can be of concern for pilots who need to understand the difference in training and actual behavior of the equipment. There may also be a lack of a vendor who can provide simulators by the time an aircraft is fitted or retrofitted with updated versions.

[0005] Hence, there is a need for providing a cost effective and accurate training solution for aircraft avionics.

BRIEF SUMMARY

[0006] This summary is provided to describe select concepts in a simplified form that are further described in the Detailed Description. This summary is not intended to identify key or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0007] A system is provided for flight management system (FMS) familiarization training. The system comprises: a user application programming interface (API) that receives a request for a training scenario from a user; a flight management engine (FME) where the FME generates the training scenario that simulates the flight mission of an FMS onboard a designated aircraft, and where the training scenario is generated using the same algorithms as the FMS; a database configuration module that retrieves performance and condition data that is provided to the FME for use in generating the training scenario; and an external API that retrieves external third-party data that is provided to the FME for use in generating the training scenario.

[0008] A method is provided for generating a training scenario for flight management system (FMS) familiarization training. The method comprises: requesting a training

scenario for a user through an application programming interface (API); generating a training scenario with a flight management engine (FME) that receives the request from the API, where the training scenario simulates the flight mission of an FMS for a designated aircraft using identical algorithms as used by the FMS; retrieving performance and condition data with a database configuration module, where the performance and condition data is provided to the FME for use in generating the training scenario; retrieving external third-party data from an external API, where the external third-party data is provided to the FME for use in generating the training scenario; and transmitting the training scenario to a computer system via a wireless communications link, where the user executes the training scenario on the computer system.

[0009] A system is provided for flight management system (FMS) familiarization training. The system comprises: a user application programming interface (API) that receives a request for a training scenario from a user; a flight management engine (FME) in operable communication with the user API, the FME operable to generate the training scenario that simulates the flight mission of an FMS onboard a designated aircraft, where the training scenario is generated using identical algorithms as used by the FMS; a database configuration module that retrieves performance and condition data from an aero-engine database (AEDB), from a navigational database (NAVDB), from a terrain database, from an aircraft performance model database, from an aircraft configuration database, and from a persona database, that is provided to the FME for use in generating the training scenario; an external API that retrieves external third-party data that is provided to the FME for use in generating the training scenario, the third-party data comprising present weather conditions, weather forecasts, and airport conditions; and a processor-based system that receives the training scenario from the FME via a wireless communications link, where the user executes the training scenario on the processor-based system.

[0010] Furthermore, other desirable features and characteristics of the method and system will become apparent from the subsequent detailed description and the appended claims, taken in conjunction with the accompanying drawings and the preceding background.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The present invention will hereinafter be described in conjunction with the following drawing figures, wherein like numerals denote like elements, and wherein:

[0012] FIG. 1 shows a diagram of an in-flight aircraft that contains an onboard flight management system (FMS) along with a visual data system in accordance with one embodiment;

[0013] FIG. 2 shows a block diagram of a flight management computer (FMC) (or flight management system (FMS)) onboard an aircraft utilized to communicate data between existing onboard avionics systems or line-replaceable units (LRUs) in accordance with one embodiment;

[0014] FIG. 3 shows a block diagram of a system for a cloud-based FMS familiarization training in accordance with one embodiment;

[0015] FIG. 4 shows a block diagram of the cloud-based infrastructure for a system for FMS familiarization training in accordance with one embodiment;

[0016] FIG. 5 show diagrams of display views for training scenarios for a system for a cloud-based FMS familiarization training in accordance with one embodiment; and

[0017] FIG. 6 shows a flowchart for a method of generating a training scenario for FMS familiarization training in accordance with one embodiment.

DETAILED DESCRIPTION

[0018] The following detailed description is merely exemplary in nature and is not intended to limit the invention or the application and uses of the invention. As used herein, the word “exemplary” means “serving as an example, instance, or illustration.” Thus, any embodiment described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments. All of the embodiments described herein are exemplary embodiments provided to enable persons skilled in the art to make or use the invention and not to limit the scope of the invention which is defined by the claims. Furthermore, there is no intention to be bound by any expressed or implied theory presented in the preceding technical field, background, brief summary, or the following detailed description.

[0019] Methods and systems for flight management system (FMS) familiarization training have been developed. The system comprises a user application programming interface (API) that receives a request for a training scenario from a user. A flight management engine (FME) generates the training scenario that simulates the flight mission of an FMS onboard a designated aircraft. The training scenario is generated using the same algorithms as the FMS. A database configuration module retrieves performance and condition data that is provided to the FME for use in generating the training scenario. An external API retrieves external third-party data that is provided to the FME for use in generating the training scenario.

[0020] Before proceeding further, it is noted that, for convenience, the following description is in the context of an aircraft environment. It will be appreciated, however, that the claimed invention is not limited to any particular aircraft environment, but may be implemented in numerous other vehicular and non-vehicular environments.

[0021] Turning now to FIG. 1, a diagram 100 is shown of an in-flight aircraft 102 that contains, among other systems, an onboard flight management system (FMS) 104 along with a visual data system 106 that is accessed by the FMS 104 in accordance with one embodiment. In alternative embodiments, the visual data system 106 may be integrated as part of the FMS 104. The FMS 104, as is generally known, is a specialized computer that automates a variety of in-flight tasks such as in-flight management of the flight plan. Using various sensors such as a global positioning system (GPS) or various other systems, the FMS 104 determines the aircraft's position and guides the aircraft along its flight plan using its navigation database. From the cockpit, the FMS 104 is normally controlled through a visual display device such as a control display unit (CDU) which incorporates a small screen, a keyboard or a touchscreen. The FMS 104 displays the flight plan and other critical flight data to the aircrew during operation.

[0022] The FMS 104 may have a built-in electronic memory system that contains a (non-illustrated) navigation database. The navigation database contains elements used for constructing a flight plan. In some embodiments, the navigation database may be separate from the FMS 104 and

located onboard the aircraft while in other embodiments the navigation database may be located on the ground and relevant data provided to the FMS 104 via a (non-illustrated) communications link with a (non-illustrated) ground station. The navigation database used by the FMS 104 may typically include: waypoints/intersections; airways; radio navigation aids/navigation beacons; airports; runway; standard instrument departure (SID) information; standard terminal arrival (STAR) information; holding patterns; and instrument approach procedures. Additionally, other waypoints may also be manually defined by pilots along the route.

[0023] The flight plan is generally determined on the ground before departure by either the pilot or a dispatcher for the owner of the aircraft. It may be manually entered into the FMS 104 or selected from a library of common routes. In other embodiments the flight plan may be loaded via a communications data link from an airline dispatch center. During preflight planning, additional relevant aircraft performance data may be entered including information such as: gross aircraft weight; fuel weight and the center of gravity of the aircraft. The aircrew may use the FMS 104 to modify the flight plan before takeoff or even while in flight for variety of reasons. Such changes may be entered via the CDU. Once in flight, the principal task of the FMS 104 is to accurately monitor the aircraft's position. This may use a GPS, a VHF omnidirectional range (VOR) system, or other similar sensor, in order to determine and validate the aircraft's exact position. The FMS 104 constantly cross checks among various sensors to determine the aircraft's position with accuracy.

[0024] Additionally, the FMS 104 may be used to perform advanced vertical navigation (VNAV) functions. The purpose of VNAV is to predict and optimize the vertical path of the aircraft. The FMS 104 provides guidance that includes control of the pitch axis and of the throttle of the aircraft. In order to accomplish these tasks, the FMS 104 has detailed flight and engine model data of the aircraft. Using this information, the FMS 104 may build a predicted vertical descent path for the aircraft. A correct and accurate implementation of VNAV has significant advantages in fuel savings and on-time efficiency.

[0025] In exemplary embodiments, an existing flight management computer (FMC) (or flight management system (FMS)) onboard an aircraft is utilized to communicate data between existing onboard avionics systems or line-replaceable units (LRUs) and one or more other modules coupled to the FMC, which support or otherwise perform new flight management functionality that is not performed by the FMC. For example, a multifunction control and display unit (MCDU) may support or otherwise perform new flight management functionality based on data from onboard avionics or LRUs received via the FMC. In such embodiments, the FMC is configured to receive operational or status data from one or more avionics systems or LRUs onboard the aircraft at corresponding avionics interfaces and convert one or more characteristics of the operational data to support communicating the operational data with the MCDU. For purposes of explanation, the subject matter may primarily be described herein in the context of converting operational data received from onboard avionics or LRUs in a first format (e.g., an avionics bus format) into another format supported by the interface with the MCDU, the subject matter described herein is not necessarily limited to format conversions or digital reformatting, and may be imple-

mented in an equivalent manner for converting between other data characteristics, such as, for example, different data rates, throughputs or bandwidths, different sampling rates, different resolutions, different data compression ratios, and the like.

[0026] FIG. 2 depicts an exemplary embodiment of an aircraft system 200 suitable for implementation onboard an aircraft 102 shown previously in FIG. 1. The illustrated aircraft system 200 includes a flight management computing module 202 communicatively coupled to a plurality of onboard avionics LRUs 204, one or more display devices 206, and a multifunction computing module 208. It should be appreciated that FIG. 2 depicts a simplified representation of the aircraft system 200 for purposes of explanation, and FIG. 2 is not intended to limit the subject matter in any way.

[0027] The flight management computing module 202 generally represents the FMC, the FMS, or other hardware, circuitry, logic, firmware and/or other components installed onboard the aircraft and configured to perform various tasks, functions and/or operations pertaining to flight management, flight planning, flight guidance, flight envelope protection, four-dimensional trajectory generation or required time of arrival (RTA) management, and the like. Accordingly, for purposes of explanation, but without limiting the functionality performed by or supported at the flight management computing module 202, the flight management computing module 202 may alternatively be referred to herein as the FMC. The FMC 202 includes a plurality of interfaces 210 configured to support communications with the avionics LRUs 204 along with one or more display interfaces 212 configured to support coupling one or more display devices 206 to the FMC 202. In the illustrated embodiment, the FMC 202 also includes a communications interface 214 that supports coupling the multifunction computing module 208 to the FMC 202.

[0028] The FMC 202 generally includes a processing system designed to perform flight management functions, and potentially other functions pertaining to flight planning, flight guidance, flight envelope protection, and the like. Depending on the embodiment, the processing system could be realized as or otherwise include one or more processors, controllers, application specific integrated circuits, programmable logic devices, discrete gate or transistor logics, discrete hardware components, or any combination thereof. The processing system of the FMC 202 generally includes or otherwise accesses a data storage element (or memory), which may be realized as any sort of non-transitory short or long term storage media capable of storing programming instructions for execution by the processing system of the FMC 202. In exemplary embodiments, the data storage element stores or otherwise maintains code or other computer-executable programming instructions that, when read and executed by the processing system of the FMC 202, cause the FMC 202 to implement, generate, or otherwise support a data concentrator application 216 that performs certain tasks, operations, functions, and processes described herein.

[0029] The avionics LRUs 204 generally represent the electronic components or modules installed onboard the aircraft that support navigation, flight planning, and other aircraft control functions in a conventional manner and/or provide real-time data and/or information regarding the operational status of the aircraft to the FMC 202. For example, practical embodiments of the aircraft system 200

will likely include one or more of the following avionics LRUs 204 suitably configured to support operation of the aircraft: a weather system, an air traffic management system, a radar system, a traffic avoidance system, an autopilot system, an autothrottle (or autothrust) system, a flight control system, hydraulics systems, pneumatics systems, environmental systems, electrical systems, engine systems, trim systems, lighting systems, crew alerting systems, electronic checklist systems, and/or another suitable avionics system.

[0030] In exemplary embodiments, the avionics interfaces 210 are realized as different ports, terminals, channels, connectors, or the like associated with the FMC 202 that are connected to different avionics LRUs 204 via different wiring, cabling, buses, or the like. In this regard, the interfaces 210 may be configured to support different communications protocols or different data formats corresponding to the respective type of avionics LRU 204 that is connected to a particular interface 210. For example, the FMC 202 may communicate navigation data from a navigation system via a navigation interface 210 coupled to a data bus supporting the ARINC 424 (or A424) standard, the ARINC 629 (or A629) standard, the ARINC 422 (or A422) standard, or the like. As another example, a datalink system or other communications LRU 204 may utilize an ARINC 619 (or A619) compatible avionics bus interface for communicating datalink communications or other communications data with the FMC 202.

[0031] The display device(s) 206 generally represent the electronic displays installed onboard the aircraft in the cockpit, and depending on the embodiment, could be realized as one or more monitors, screens, liquid crystal displays (LCDs), a light emitting diode (LED) displays, or any other suitable electronic display(s) capable of graphically displaying data and/or information provided by the FMC 202 via the display interface(s) 212. Similar to the avionics interfaces 210, the display interfaces 212 are realized as different ports, terminals, channels, connectors, or the like associated with the FMC 202 that are connected to different cockpit displays 206 via corresponding wiring, cabling, buses, or the like. In one or more embodiments, the display interfaces 212 are configured to support communications in accordance with the ARINC 661 (or A661) standard. In one embodiment, the FMC 202 communicates with a lateral map display device 206 using the ARINC 702 (or A702) standard.

[0032] In exemplary embodiments, the multifunction computing module 208 is realized as a multifunction control and display unit (MCDU) that includes one or more user interfaces, such as one or more input devices 220 and/or one or more display devices 222 (shown previously as 106 in FIG.1), a processing system 224, and a communications module 226. The MCDU 208 generally includes at least one user input device 220 that is coupled to the processing system 224 and capable of receiving inputs from a user, such as, for example, a keyboard, a key pad, a mouse, a joystick, a directional pad, a touchscreen, a touch panel, a motion sensor, or any other suitable user input device or combinations thereof. The display device(s) 222 may be realized as any sort of monitor, screen, LCD, LED display, or other suitable electronic display capable of graphically displaying data and/or information under control of the processing system 224.

[0033] The processing system 224 generally represents the hardware, circuitry, logic, firmware and/or other components of the MCDU 208 configured to perform the various

tasks, operations, functions and/or operations described herein. Depending on the embodiment, the processing system 224 may be implemented or realized with a general purpose processor, a microprocessor, a controller, a micro-controller, a state machine, an application specific integrated circuit, a field programmable gate array, any suitable programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof, designed to perform the functions described herein. Furthermore, the steps of a method or algorithm described in connection with the embodiments disclosed herein may be embodied directly in hardware, in firmware, in a software module executed by the processing system 224, or in any practical combination thereof. In this regard, the processing system 224 includes or accesses a data storage element (or memory), which may be realized using any sort of non-transitory short or long term storage media, and which is capable of storing code or other programming instructions for execution by the processing system 224. In exemplary embodiments described herein, the code or other computer-executable programming instructions, when read and executed by the processing system 224, cause the processing system 224 to implement with an FMS 230 (shown previously as 104 in FIG.1) additional tasks, operations, functions, and processes described herein.

[0034] The communications module 226 generally represents the hardware, module, circuitry, software, firmware and/or combination thereof that is coupled between the processing system 224 and a communications interface 228 of the MCDU 208 and configured to support communications between the MCDU 208 and the FMC 202 via an electrical connection 229 between the MCDU communications interface 228 and the FMC communications interface 214. For example, in one embodiment, the communications module 226 is realized as an Ethernet card or adapter configured to support communications between the FMC 202 and the MCDU 208 via an Ethernet cable 229 provided between Ethernet ports 214, 228. In other embodiments, the communications module 226 is configured to support communications between the FMC 202 and the MCDU 208 in accordance with the ARINC 429 (A429) standard via an A429 data bus 229 provided between A429 ports 214, 228 of the respective modules 202, 208. In yet other embodiments, the communications module 226 is configured to support communications between the FMC 202 and the MCDU 208 in accordance with the ARINC 422 (A422) standard via an A422 data bus 229 provided between A422 ports 214, 228 of the respective modules 202, 208. In yet other embodiments, the communications module 226 is configured to support communications between the FMC 202 and the MCDU 208 in accordance with the ARINC 739 (A739) standard via an A739 data bus 229 provided between A739 ports 214, 228 of the respective modules 202, 208.

[0035] In various embodiments, the FMC 202 and MCDU 208 communicate using a different communications protocol or standard than one or more of the avionics LRUs 204 and/or the display devices 206. In such embodiments, to support communications of data between the MCDU 208 and those LRUs 204 and/or display devices 206, the data concentrator application 216 at the FMC 202 converts data from one format to another before retransmitting or relaying that data to its destination. For example, the data concentrator application 216 may convert data received from an avionics LRU 204 to the A429 or Ethernet format before

providing the data to the MCDU 208, and vice versa. Additionally, in exemplary embodiments, the FMC 202 validates the data received from an avionics LRU 204 before transmitting the data to the MCDU 208. For example, the FMC 202 may perform debouncing, filtering, and range checking, and/or the like prior to converting and retransmitting data from an avionics LRU 204.

[0036] It should be noted that although the subject matter may be described herein in the context of the multifunction computing module 208 being realized as an MCDU, in alternative embodiments, the multifunction computing module 208 could be realized as an electronic flight bag (EFB) or other mobile or portable electronic device. In such embodiments, an EFB capable of supporting an FMS 230 application may be connected to an onboard FMC 202 using an Ethernet cable 229 to support flight management functionality from the EFB in an equivalent manner as described herein in the context of the MCDU.

[0037] In one or more embodiments, the MCDU 208 stores or otherwise maintains programming instructions, code, or other data for programming the FMC 202 and transmits or otherwise provides the programming instructions to the FMC 202 to update or otherwise modify the FMC 202 to implement the data concentrator application 216. For example, in some embodiments, upon establishment of the connection 229 between modules 202, 208, the MCDU 208 may automatically interact with the FMC 202 and transmit or otherwise provide the programming instructions to the FMC 202, which, in turn, executes the instructions to implement the data concentrator application 216. In some embodiments, the data concentrator application 216 may be implemented in lieu of flight management functionality by the MCDU 208 reprogramming the FMC 202. In other embodiments, the FMC 202 may support the data concentrator application 216 in parallel with flight management functions. In this regard, the FMC 202 may perform flight management functions, while the FMS 230 application on the MCDU 208 supplements the flight management functions to provide upgraded flight management functionality within the aircraft system 200.

[0038] The FMS systems described above may be periodically updated and replaced. Thus, as previously mentioned, aircraft operators (e.g., pilots) may need to undergo training on the new and/or updated systems. To facilitate such training, a system and method for cloud-based flight management system (FMS) familiarization training has been developed. The system includes a cloud-based training module that uses the same avionics software assets as present on board the aircraft in question. This ensures a complete match of behavior between training module and actual onboard avionics. Ideally, pilots may receive hands-on experience and training on real avionics software along with ability to create scenarios and practice on desktop or hand-held devices.

[0039] Turning now to FIG. 3, a block diagram is shown of a system 300 for a cloud-based FMS familiarization training in accordance with one embodiment. The depicted system 300 includes cloud-based infrastructure 302, which includes a flight management engine (FME) 310 that is used for implementing the software for training on the FMS (shown as 104 in FIG. 1). The FME 310 controls a database configuration module 408 that retrieves loadable data from such sources such as a navigation database (NAVDB) 312 and an aero-engine database (AEDB) 314. These databases

312, 314 provide technical details and aircraft performance data for various training scenarios. The retrieved data is loaded into a cloud-based server **316** that also retrieves historical events customized to a specific pilot's personal performance data or "persona" from the aircraft **306** operated by the pilot. The server **316** also retrieves third-party data from third-party analytics services **318** through an external application programming interface (API). Examples of such third-party data include current weather conditions, weather forecasts and airport conditions, just to name a few.

[0040] The FME **310** receives a request for training through an API connected to a computer **304** that is used by the pilots **308** for training. The computer **304** may be a desktop, laptop or a mobile device **310** such as a tablet or smartphone that is connected to the computer **304** via a wired or wireless communication link. In response to the request, the FME **310** provides the training scenario to the user's computer **304** through the API. The user's computer **304** may include additional plug-in equipment, such as a joystick controller **320**, (non-illustrated) pedals and/or other equipment necessary to simulate required actions during the training scenario. Additionally, supporting casting (e.g., air traffic controllers) for the scenario can be supplied to the laptop computer **304** through wired or wireless connections to mobile devices **310** such as smart phones or tablets.

[0041] Turning now to FIG. 4, a block diagram **400** is shown of the cloud-based infrastructure **402** for a system for FMS familiarization training in accordance with one embodiment. The depicted cloud-based infrastructure **402** (which may be used to implement the cloud-based infrastructure **302** of FIG. 3) includes an API interface layer **406a, 406b** that comprises an API request interface **406a** and an API response interface **406b**. The API request interface **406a** and the API response interface **406b** handle requests and responses, respectively, to and from the system user. The API interface layer **406a, 406b** is the sole entry point into the cloud services **402** for the user of the cloud-based flight simulator. The API interface layer **406a, 406b** typically uses the hypertext transfer protocol (HTTP). The cloud-based infrastructure **402** also includes an FME **408** (shown as **310** in FIG. 3) which is a configurable aeronautical flight planning and mission predictions engine that is built out identically to the FMS onboard the simulated aircraft (e.g., aircraft **306**). The engine uses the same algorithms as that of the FMS to take flight based planning data and: determine the flight plan; calculate the lateral plan; calculate time and fuel information; optimize the flight plan for the optimum lateral and vertical flight paths; generate time point calculations for trans-oceanic flight; and simulate "what if" situations such as depressurization and engine emergencies. The FME **408** also has a mechanism to model guidance and navigation for simulating the onboard FMS.

[0042] Also included in the cloud infrastructure **402** is a register manager **410** that is responsible for controlling user access and authentication at any point during the training. A configuration manager **412** is included that is responsible for adapting the flight simulator for a given user, aircraft and user interface (UI). The configuration manager **412** also ensures that the FME **408** and aircraft model databases are set up correctly for each individual user. A historical data processor **416** is included that is responsible for collecting, storing and processing the historical data based on past performance of a user. It is able to create a persona based on

past performance. The persona can be loaded with the other data to provide a personalized scenario-based training for the user. An aircraft UI modeling unit **414** is included to ensure a tailored UI is provided for the selected avionics based on the user needs. Also included is a session manager **420** and a learning manager **418** which monitors the performance and progress of the user during the flight simulator training.

[0043] The cloud-based infrastructure **402** also includes a database configuration module **422**. The database configuration module **422** retrieves and manages data from multiple databases for the flight simulator, including: navigational **424**; terrain **426**; aircraft performance modeling **428**; evident/user persona **430**; and aircraft configuration **432**. Finally, an external API interface **434** is included that connects to external third-party components **404** such as a tablet UI or other mobile device. This UI is delivered to tablet users to allow utilization of the flight simulator and may be configured for use on various mobile device operating systems such as Android or iOS. Other external components **404** could include an augmented reality/virtual reality (AR/VR) kit, commercial off-the-shelf (COTS) simulators. There's also the capacity for external plug-ins and third-party external interfaces to provide additional data and features for use in the training scenario.

[0044] Turning now to FIG. 5, diagrams of display views **500** are shown for training scenarios for a system for a cloud-based FMS familiarization training (such as system **300** of FIG. 3) in accordance with one embodiment. These depictions include: guided lessons **502**; scenario previews **504**; display views in various formats **506**; and simulation views of navigation and aircraft performance indicators **508**.

[0045] In an example of a training scenario, the pilot will take a snapshot of all onboard avionics states and send the information to the secure cloud infrastructure **402**. This can be done when the pilot is not sure about how to optimally use the avionics systems. When the pilot is ready to replay the training scenario, the cloud-based simulator **400** will load the data and suggest the optimal method to interact with the avionics components such as an FMS. The scenario may be based on learning models that are created over time by the simulator **402** and standard operating procedures (SOPs). The learning models may also develop and suggest personalized improvement plans for each pilot based on their performance during the scenario.

[0046] Turning now to FIG. 6, a flowchart **600** is shown for a method of generating a training scenario for FMS familiarization training in accordance with one embodiment. Initially, a training scenario is requested for a user through an API **602**. The request is received by an FME which proceeds to generate the training scenario **604**. The FME retrieves performance and condition data from a database configuration model **606** as well as external third-party data from an external API **608** for use in generating the training scenario. The training scenario is then transmitted to a computer system for execution by a user **610**.

[0047] Techniques and technologies may be described herein in terms of functional and/or logical block components, and with reference to symbolic representations of operations, processing tasks, and functions that may be performed by various computing components or devices. Such operations, tasks, and functions are sometimes referred to as being computer-executed, computerized, software-implemented, or computer-implemented. In practice, one or

more processor devices can carry out the described operations, tasks, and functions by manipulating electrical signals representing data bits at memory locations in the system memory, as well as other processing of signals. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, optical, or organic properties corresponding to the data bits. It should be appreciated that the various block components shown in the figures may be realized by any number of hardware, software, and/or firmware components configured to perform the specified functions. For example, an embodiment of a system or a component may employ various integrated circuit components, e.g., memory elements, digital signal processing elements, logic elements, look-up tables, or the like, which may carry out a variety of functions under the control of one or more microprocessors or other control devices.

[0048] When implemented in software or firmware, various elements of the systems described herein are essentially the code segments or instructions that perform the various tasks. The program or code segments can be stored in a processor-readable medium or transmitted by a computer data signal embodied in a carrier wave over a transmission medium or communication path. The “computer-readable medium”, “processor-readable medium”, or “machine-readable medium” may include any medium that can store or transfer information. Examples of the processor-readable medium include an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable ROM (EROM), a floppy diskette, a CD-ROM, an optical disk, a hard disk, a fiber optic medium, a radio frequency (RF) link, or the like. The computer data signal may include any signal that can propagate over a transmission medium such as electronic network channels, optical fibers, air, electromagnetic paths, or RF links. The code segments may be downloaded via computer networks such as the Internet, an intranet, a LAN, or the like.

[0049] The following description refers to elements or nodes or features being “connected” or “coupled” together. As used herein, unless expressly stated otherwise, “coupled” means that one element/node/feature is directly or indirectly joined to (or directly or indirectly communicates with) another element/node/feature, and not necessarily mechanically. Likewise, unless expressly stated otherwise, “connected” means that one element/node/feature is directly joined to (or directly communicates with) another element/node/feature, and not necessarily mechanically. Thus, additional intervening elements, devices, features, or components may be present in an embodiment of the depicted subject matter.

[0050] In addition, certain terminology may also be used in the following description for the purpose of reference only, and thus are not intended to be limiting. For example, terms such as “upper”, “lower”, “above”, and “below” refer to directions in the drawings to which reference is made. Terms such as “front”, “back”, “rear”, “side”, “outboard”, and “inboard” describe the orientation and/or location of portions of the component within a consistent but arbitrary frame of reference which is made clear by reference to the text and the associated drawings describing the component under discussion. Such terminology may include the words specifically mentioned above, derivatives thereof, and words of similar import. Similarly, the terms “first”, “second”, and

other such numerical terms referring to structures do not imply a sequence or order unless clearly indicated by the context.

[0051] For the sake of brevity, conventional techniques related to signal processing, data transmission, signaling, network control, and other functional aspects of the systems (and the individual operating components of the systems) may not be described in detail herein. Furthermore, the connecting lines shown in the various figures contained herein are intended to represent exemplary functional relationships and/or physical couplings between the various elements. It should be noted that many alternative or additional functional relationships or physical connections may be present in an embodiment of the subject matter.

[0052] Some of the functional units described in this specification have been referred to as “modules” in order to more particularly emphasize their implementation independence. For example, functionality referred to herein as a module may be implemented wholly, or partially, as a hardware circuit comprising custom VLSI circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices, or the like. Modules may also be implemented in software for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical modules of computer instructions that may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together but may comprise disparate instructions stored in different locations that, when joined logically together, comprise the module and achieve the stated purpose for the module. Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

[0053] While at least one exemplary embodiment has been presented in the foregoing detailed description, it should be appreciated that a vast number of variations exist. It should also be appreciated that the exemplary embodiment or embodiments described herein are not intended to limit the scope, applicability, or configuration of the claimed subject matter in any way. Rather, the foregoing detailed description will provide those skilled in the art with a convenient road map for implementing the described embodiment or embodiments. It should be understood that various changes can be made in the function and arrangement of elements without departing from the scope defined by the claims, which includes known equivalents and foreseeable equivalents at the time of filing this patent application.

What is claimed is:

1. A system for flight management system (FMS) familiarization training, comprising:
 - a user application programming interface (API) that receives a request for a training scenario from a user;

- a flight management engine (FME) in operable communication with the user API, where the FME generates the training scenario that simulates the flight mission of an FMS onboard a designated aircraft, and where the training scenario is generated using identical algorithms as used by the FMS;
- a database configuration module that retrieves performance and condition data that is provided to the FME for use in generating the training scenario; and
- an external API that retrieves external third-party data that is provided to the FME for use in generating the training scenario.
2. The system of claim 1, where the database configuration module retrieves performance and condition data from an aero-engine database (AEDB).
3. The system of claim 1, where the database configuration module retrieves performance and condition data from a navigational database (NAVDB).
4. The system of claim 1, where the database configuration module retrieves performance and condition data from a terrain database.
5. The system of claim 1, where the database configuration module retrieves performance and condition data from an aircraft performance model database.
6. The system of claim 1, where the database configuration module retrieves performance and condition data from a persona database.
7. The system of claim 1, where the database configuration module retrieves performance and condition data from an aircraft configuration database.
8. The system of claim 1, where the external third-party data comprises present weather conditions.
9. The system of claim 1, where the external third-party data comprises weather forecasts.
10. The system of claim 1, where the external third-party data comprises airport conditions.
11. The system of claim 1, further comprising:
a processor-based system that receives the training scenario from the FME via a wireless communications link, where the user executes the training scenario on the processor-based system.
12. The system of claim 11, where the processor-based system comprises a personal computer.
13. The system of claim 11, where the processor-based system comprises a laptop computer.
14. The system of claim 11 where the processor-based system comprises a smart phone.
15. The system of claim 11, where the processor-based system comprises a mobile tablet computing device.
16. A method for generating a training scenario for flight management system (FMS) familiarization training, comprising:
requesting a training scenario for a user through an application programming interface (API);
generating a training scenario with a flight management engine (FME) that receives the request from the API, where the training scenario simulates the flight mission of an FMS for a designated aircraft using identical algorithms as used by the FMS;
retrieving performance and condition data with a database configuration module, where the performance and condition data is provided to the FME for use in generating the training scenario;
retrieving external third-party data from an external API, where the external third-party data is provided to the FME for use in generating the training scenario; and
transmitting the training scenario to a computer system via a wireless communications link, where the user executes the training scenario on the computer system.
17. The method of claim 16, where the database configuration module retrieves performance and condition data from an aero-engine database (AEDB).
18. The method of claim 16, where the database configuration module retrieves performance and condition data from a navigational database (NAVDB).
19. A system for flight management system (FMS) familiarization training, comprising:
a user application programming interface (API) that receives a request for a training scenario from a user;
a flight management engine (FME) in operable communication with the user API, the FME operable to generate the training scenario that simulates the flight mission of an FMS onboard a designated aircraft, where the training scenario is generated using identical algorithms as used by the FMS;
a database configuration module that retrieves performance and condition data from an aero-engine database (AEDB), from a navigational database (NAVDB), from a terrain database, from an aircraft performance model database, from an aircraft configuration database, and from a persona database, that is provided to the FME for use in generating the training scenario;
an external API that retrieves external third-party data that is provided to the FME for use in generating the training scenario, the third-party data comprising present weather conditions, weather forecasts, and airport conditions; and
a processor-based system that receives the training scenario from the FME via a wireless communications link, where the user executes the training scenario on the processor-based system.
20. The system of claim 19, wherein the processor-based system is selected from the group consisting of a personal computer, a laptop computer, a smartphone, and a mobile tablet computing device.

* * * * *