



# Service-Oriented Intelligent Avionics System Architecture

Zhao Yan<sup>1,2(✉)</sup>, Wen Pengcheng<sup>1,2</sup>, Bai Linting<sup>1,2</sup>, and Wu Na<sup>1,2</sup>

<sup>1</sup> Xi'an Aeronautics Computing Technique Research Institute, AVIC, Xi'an 710068, China  
zhaoyan@ftcl.hit.edu.cn

<sup>2</sup> Aviation Key Laboratory of Science and Technology On Airborne and Missile-Borne Computer, Xi'an 710065, China

**Abstract.** As the core component of the aircraft, the avionics system has gone through the independent, federated, and integrated architecture to deal with the increasingly complex air combat scenarios. With the development of artificial intelligence and integrated circuits, avionics systems should be prepared to equip a large number of intelligent applications, which contributes to more effectively completing the military missions. In addition, the next generation of avionics systems need to be more flexible, and thus can be conveniently updated and allow intelligent applications effectively communicate with each other. Therefore, to achieve a more intelligent and flexible avionics system, this paper proposes a novel Service-oriented Intelligent Avionics System Architecture, named as SIASA. With such a service-oriented architecture, SIASA provides a flexible software platform by treating intelligent applications as services. Specifically, the proposed SIASA consists of seven layers, i.e., airborne intelligent chip layer, abstraction layer, hashrate abstraction layer, equipment abstraction layer, airborne service capability layer, airborne intelligent application atomic service layer, and airborne intelligent application layer. With these layers, SIASA well supports intelligent applications and is able to make effective and efficient updates. Furthermore, SIASA also supports more advanced scenarios, including airborne OTA, drag-and-drop software development, and system intelligence.

**Keywords:** Avionics system · Service-oriented architecture · Artificial intelligence

## 1 Introduction

The concept of Artificial Intelligence (AI) was proposed at the Dartmouth Conference in 1956 [1]. After more than 60 years of continuous development, AI has become the core driver of a new round of industrial revolution [2]. In the defense industry, AI plays an important role in promoting the upgrade of combat equipment and the transformation of combat mode. However, general AI cannot fully support the needs of intelligence in the defense industry. The reason is that military scenarios contain the characteristics of highly complex environment, strong confrontation, and requiring real-time response. Therefore, multiple countries have put forward projects to support intelligence for military purposes,

e.g., the Intelligence and Data Acceleration (ADA) program from U.S. Department of Defense and the “Future Defense 2030 Technology Strategy: Defense AI Technology Roadmap” released by the Korea Institute for Defense Technology Promotion.

As a key component of combat equipment, military aviation equipment needs to be comprehensively upgraded for improving the degree of intelligent defense industry [3]. Applying AI to avionics contributes to extending and enhancing human capabilities and efficiency [4]. To achieve the goal of intelligent aviation equipment, the avionics system architecture needs to be enhanced. Specifically, avionics system is the sum of all electronic systems on the aircraft [5], including fire control, display and control, radar, communication and navigation, external management, and many other subsystems. Currently, most military aviation systems have their own independent requirements, and each of them is provided by a single supplier with customized platforms and services. This circumstance results in a lack of hardware and software reusability between various aircraft platforms, long delivery times, and makes the improvement processes complicated. Especially, after the introduction of AI technology, the avionics system may consist of tens or even hundreds of intelligent applications inside, and the applications often need to be upgraded or even reconfigured online to cope with different operational tasks. To confront these circumstances, the characteristics of openness, dynamism, integration, modularity, and generalization might be essential for the future avionics system architecture [6].

To more effectively manage the increasingly complex software system and provide a more flexible and dynamic platform, Service-Oriented Architecture (SOA) has been adopted by multiple industrial areas. Specifically, SOA is an architectural design method that abstracts the capabilities of a system into services with standard interfaces that can be accessed, thus an application can be implemented by invoking one service or composing multiple services [7]. By defining the logical service interfaces, the service-oriented architecture clarifies the division of functions and the agreement of information communication. Therefore, SOA can be utilized to shorten the development cycle of the intelligent OODA loop for each scenario, enhances the reusability of software, and reduces the cost of system integration.

To make the avionics system more intelligent and flexible, this paper proposes a service-oriented intelligent avionics system architecture, named as SIASA. Specifically, SIASA consists of seven layers, i.e., airborne intelligent chip layer, hashrate abstraction layer, airborne equipment abstraction layer, airborne service capability layer, airborne intelligent application atomic service layer, service intelligent governance layer, and airborne intelligent application layer. In the following, this paper first introduces the related work on avionics system architecture and service-oriented architecture, then presents the proposed SIASA, after that makes an outlook, and finally summarizes the paper.

## 2 Related Work

### 2.1 Avionics System Architecture

Since the 1960s, the avionics system architecture has gone through four stages, i.e., independent, federated, integrated, and advanced integrated avionics system architecture [8].

In the 1940s-60s, the avionics system architecture was independent and consisted of multiple control systems [9]. Each of these control systems consists of a number of independent controllers connected point-to-point to dedicated sensors and peripherals. In the independent avionics system architecture, no interconnection mechanism exists among controllers.

In the 1970s and 1980s, as a result of the development of digital avionics information systems, the federate avionics system architecture interconnected the avionics subsystems through the 1553 bus [10], which enhanced the information interaction capability and made the control systems loosely coupled with each other.

In the 1980s and 1990s, with the rapid development of integrated circuits and network technology, the integrated avionics system further enhanced the data processing capability by utilizing high-performance microprocessors over fiber-optic data buses [11]. Later, the advanced integrated avionics system architecture has been proposed based on the integrated avionics system architecture to further enhance the integration of various system processing and sensor signal processing capabilities. At the same time, a number of new technologies related to AI are introduced to assist the aviator to operate military operations.

Currently, with the application of emerging technologies such as deep learning and big data in aviation systems, the performance requirements of avionics systems continue to rise. To support the intelligent applications over big data, heterogeneous intelligent processing architectures are required to fuse multiple intelligent chips, such as GPUs, FPGAs, and ASICs [8]. In addition, with the further development of system decoupling and service-oriented concepts, the top-level applications may face a considerable number of atomic services, which require a service-oriented architecture to support flexible and efficient intelligent management and scheduling.

### 2.2 SOA Architecture Research and Application Status

Service-Oriented Architecture (SOA) [12] is an architectural design approach that abstracts the capabilities of a system into services with standard interfaces that can be invoked or composed to meet the specific needs of the users. It is recognized that SOA has the characteristics of high cohesion within services, loose coupling between services and standardization of service communication, etc. There is also a consensus that SOA can build a flexible and variable platform system, which contributes to effectively maintaining and extending the increasingly complex software systems.

SOA has been widely utilized in many fields, e.g., the automotive industry, IoT, and banking industry. Taking the automotive industry as an example, SOA software architecture technology for in-vehicle systems has received increasing attention from global

vehicle manufacturers due to the demand for intelligent and networked vehicles, especially for the development and application of autonomous driving systems. Specifically, Volkswagen, Audi, BMW, Ford, and other automotive giants have formed their own alliances to conduct research on the application of SOA technologies and specifications, which are expected to start being applied to mass production models around 2023.

### 3 Service-Oriented Intelligent Avionics System Architecture

#### 3.1 Overall Framework

Figure 1 demonstrates our proposed Service-oriented Intelligent Avionics System Architecture, named as SIIASA. Specifically, SIIASA contains 7 layers, i.e., airborne intelligent chip layer, abstraction layer, hashrate abstraction layer, equipment abstraction layer, airborne service capability layer, airborne intelligent application atomic service layer, and airborne intelligent application layer.

The above-mentioned layers are briefly introduced in this paragraph and more details are presented in Sects. 3.2–3.8. Airborne intelligent chip layer contains intelligent chips such as CPU, GPU, FPGA, ASIC, and neuromorphic chip, which are suitable to support the computational task from intelligent applications. These chips are abstracted by arithmetic abstraction layer through resource virtualization and heterogeneous chip difference shielding. Similarly, the equipment, including sensors and weapons, are abstracted by the equipment abstraction layer to provide a formalized interface. Then, the airborne service capability layer is provided to support service description, service registration, service discovery and service communication, which are the fundamental elements of a service-oriented architecture. After that, we devised the airborne intelligent application atomic service layer, which contains observe services, orientation services, decision services, and action services. Finally, based on these atomic services, the airborne intelligent applications including autonomous flight applications and air combat applications are implemented in the airborne intelligent application layer.

#### 3.2 Airborne Intelligent Chip Layer

Despite of high performance, intelligent applications require a large number of calculations and can hardly be fulfilled by the conventional CPU. Thus, intelligent chips have been studied and manufactured to fill this gap. Intelligent chips generally refer to all hardware employed to accelerate intelligent applications, especially applications based on deep learning techniques. Specifically, based on the technical architecture, intelligent chips can be classified into GPU, FPGA, ASIC, and neuromorphic chips. Different classification methods exist for intelligent chips. For example, based on the location, intelligent chips can be classified into cloud chips, edge chips, and terminal chips; and based on the usage stage, intelligent chips can be classified into training chips and inference chips. In the airborne scenario, the intelligent chip is usually used as inference chips at the edge end or terminal end, while the training process is executed in the ground-based cloud end.

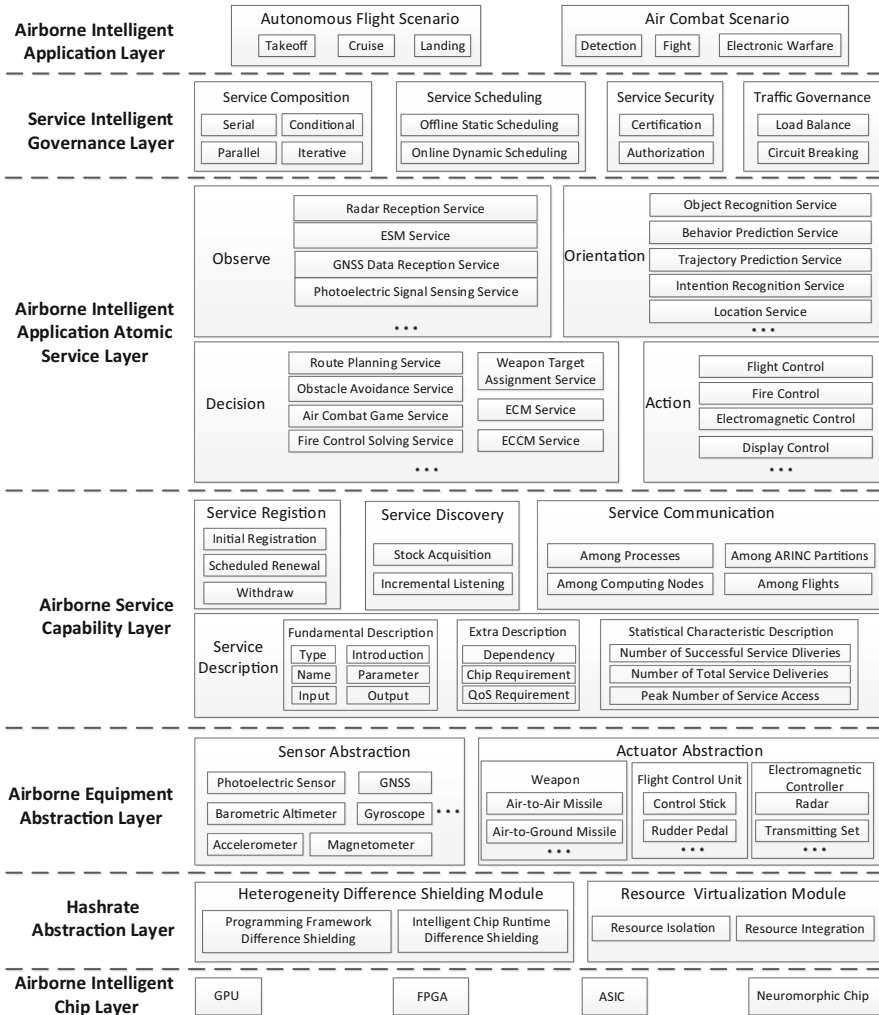


Fig. 1. The overall framework of the service-oriented intelligent avionics system architecture

### 3.3 Hashrate Abstraction Layer

The demand for intelligent applications in airborne scenarios is various, and a single type of chip cannot support intelligent applications in all airborne scenarios. Therefore, multiple types of chips are essential for the airborne computing platform to improve the performance of intelligent applications. In order to achieve flexible deployment of intelligent applications across multiple chip types and improve the utilization rate of computing resources, it is necessary to mask the differences in runtime and programming frameworks between different chip types. Moreover, in order to increase the utilization ratio, the virtualization technique is also essential to allow one chip to be utilized by multiple applications. Therefore, this paper devises a heterogeneous difference masking module

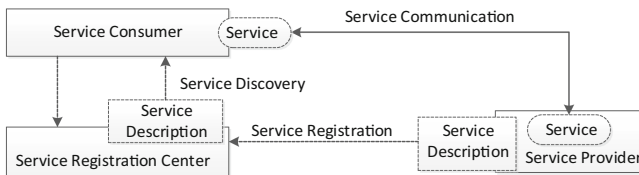
and a resource virtualization module. The heterogeneous difference masking module includes deep learning programming framework difference masking and intelligent chip runtime difference masking, and the resource virtualization module includes resource isolation and resource integration.

### 3.4 Airborne Equipment Abstraction Layer

This layer is dedicated to the abstraction of the access control interface of relevant sensors, actuators, I/O and other physical devices to shield their physical differences, and thus realize the decoupling of hardware and software. Through this way, SIASA shortens the research and development cycle, enhances portability, and achieves the purpose of plug-and-play. Focusing on intelligent avionics systems, this paper abstracts the sensors and actuators related to intelligent applications. Specifically, typical sensors include photoelectric sensors, barometric altimeters, acceleration instruments, GNSS, gyroscopes and magnetometers, etc.; and typical actuators include weapons, e.g., air-to-air missiles and air-to-ground missiles, flight controllers, e.g., control sticks and rudder pedals, and electromagnetic controllers, e.g., radars and radio transmitters.

### 3.5 Airborne Service Capability Layer

This layer provides the essential components of a service-oriented architecture, and is a key role of SIASA. As presented in Fig. 2, the service-oriented architecture contains three key roles, i.e., service consumer, service provider, and service registry; and contains four main functions, i.e., service description, service registration, service discovery, and service communication.



**Fig. 2.** The key roles and main functions of the service-oriented intelligent avionics

Brief descriptions of service provider, service consumer, and service registry are as follows.

- Service provider. The entity that provides services. Typical functions needed to be implemented include service development, service description (i.e., the standardized definition of the service interface, which will be introduced in the following), service registration, and other functions.
- Service consumer. The entity that consumes the service. Typical functions needed to be implemented includes querying the service description and communicating with the service.

- **Service registry.** The entity that supports the registration, publication, search, and evaluation of services. The typical functions needed to be implemented include publishing, adding, deleting, modifying, querying and classifying service description, verifying whether the service meets the standards, and notifying service consumers when the registration information updates.

A detailed functional description of the service-oriented architecture is as follows.

**Service Description.** Service description refers to a standardized definition of the service interface with a unified syntax that describes the basic information, functions, requirements, and statistical information of services. According to the characteristics of airborne services, the service description defined in this paper contains three parts, i.e., basic description, additional description, and statistical characteristics description. Specifically, the basic description includes attributes such as type, name, parameters, input format, output format, and service introduction; additional description includes attributes such as service dependency, intelligent chip requirements and QoS requirements; and the statistical characteristic description includes attributes including the number of successful service deliveries, and the number of total service deliveries.

**Service Registration.** Service registration means that the service provider registers the description information of the service to the service registry. After the initial registration, which allows service consumers to query the service description. Service registration is divided into three stages, i.e., initial registration, scheduled renewal, and withdrawal.

- **Initial registration.** The service needs to be registered to the registration center once it starts up and is ready to accept requests, which allows it to be discovered by other services or programs.
- **Scheduled renewal.** Scheduled renewal is equivalent to keeping alive, which periodically notifies the service registry that the service is still available and can continue to provide the service.
- **Withdrawal:** The registration information should be actively withdrawn when the service stops.

**Service Discovery.** Service discovery refers to the process by which a service gets access to services that can meet specific functional and non-functional requirements. This process involves the matching between the service requirement from the service consumer and the service description from the service provider. Service discovery is generally divided into two phases, i.e., stock acquisition and incremental listening.

- **Stock acquisition:** The service registry should provide a list of registered services to the new service or program.
- **Incremental Listening:** When a new service is registered to the service registry, the service registry shall notify this information to existing services.

**Service Communication.** Service communication refers to the cross-platform and cross-programming language data transfer between services through a unified and standardized communication mechanism. In order to improve the flexibility and efficiency

of communication between services in the airborne environment, it is necessary to realize multi-channel communication, such as communication between processes, ARINC partitions, computing nodes, between flights, etc.

### 3.6 Airborne Intelligent Application Atomic Service Layer

Atomic services are the services with the smallest granularity and are generally designed as stand-alone logic modules and algorithms. Note that device-oriented functional atomic services are based on the device abstraction layer, and provide input or executed the instructions from logic or algorithmic services. Mapped to the field of airborne intelligence applications, atomic services can be classified into two types, i.e., (1) stand-alone intelligent algorithms, and (2) sensor-oriented or actuator-oriented functions associated with intelligent algorithms. As demonstrated in Fig. 3, this paper splits intelligent applications into four types of atomic services, i.e., observe services, orientation services, decision services, and action services based on the OODA loop theory.

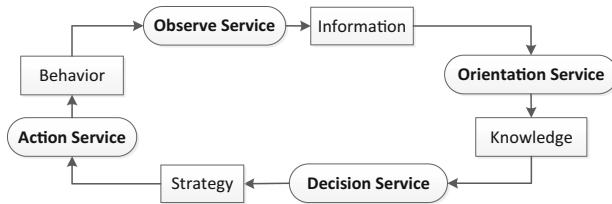


Fig. 3. Illustration of airborne atomic services in an OODA loop

In this paper, atomic services are devised for the autonomous flight scenario and air combat scenario. Specifically, the typical applications for the autonomous flight scenario include takeoff, cruise, and landing; and the typical applications for the combat scenario include detection, fight, and electronic warfare. To compose the applications mentioned above, a total of 18 atomic services have been devised. These atomic services contain 4 types of observe services, i.e., radar reception service, GNSS data reception service, photoelectric signal sensing service, ESM service; 5 types of orientation services, i.e., object recognition service, behavior prediction service, trajectory prediction service, intention recognition service, and location service; 7 types of decision services, i.e., route planning service, autonomous obstacle avoidance service, air combat game service, fire control solving service, weapon target assignment service, ECM service, and ECCM service; and 4 types of action services, i.e., flight control service, fire control service, electromagnetic control service, and display control service.

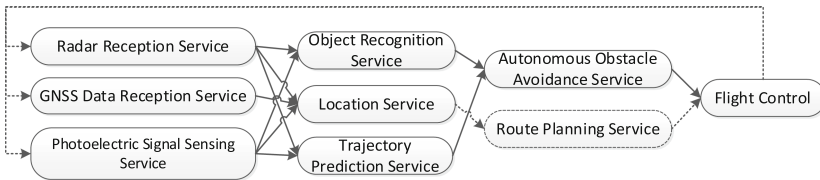
### 3.7 Service Intelligent Governance Layer

This layer focuses on the intelligent governance of services and is responsible for the intelligent composition of atomic services and their effectivity, security and reliability. Specifically, this layer consists of four modules, i.e., service composition, service



scheduling, service security, and traffic governance. More details about these modules are introduced in the following.

**Service Composition Module.** The service composition module defines the logical execution order of atomic services and is devised specifically for composing intelligent applications. In this paper, 4 types of execution order are defined, i.e., serial order, parallel order, conditional order, and iterative order. The service composition design for the typical intelligent application, i.e., autonomous cruise, is presented in Fig. 4. In this case, serial order exists between radar reception service and location service, parallel order exists between radar reception service and GNSS data reception service, conditional order exists between route planning service and location service as route planning service is not always called, and the whole OODA loop is an iterative order.



**Fig. 4.** Atomic service design and combination for autonomous cruise

**Service Scheduling Module.** This module is responsible for the mapping from atomic services to the computational resources by intelligent techniques. Such a mapping process should consider the matching degree between atomic services and resources as well as the matching degree between communication requirements among atomic services and the communication efficiency. Generally, service scheduling can be divided into two stages, i.e., offline static scheduling and online dynamic scheduling.

- **Offline Static Scheduling** refers to the scheduling process before taking off and is generally conducted on the ground center. The goal of this scheduling process is to provide an optimal or near-optimal accurate mapping relation from atomic services to the computational resources given enough scheduling time. The typical techniques suitable for this scheduling process are generally from the field of operation research, e.g., mixed-integer programming.
- **Online Dynamic Scheduling** refers to the scheduling process when the flights are in the air. For example, such online dynamic scheduling is essential when some computational resources are not available because of malfunction or being destroyed. The goal of this scheduling process is to provide a feasible solution given limited scheduling time. The typical techniques suitable for this scheduling process are evolutionary algorithms, e.g., the ant colony algorithm.

**Service Security Module.** This module is responsible for the security of service communication. To achieve this goal, the generally employed techniques are authentication and authorization.

- Authentication verifies whether a request has the permission to the called service, and thus enforces access control to the services.
- Authorization is the process that asserts the identity of the service. Its typical application is to secure service-to-service communication from the man-in-the-middle attack.

**Traffic Governance Module.** This module is responsible for the governance of traffic-related issues. Traffic governance contributes to the stability and reliability of service-oriented architecture. Specifically, the typical techniques employed for traffic governance include load balance and circuit breaking.

- Load Balance is a mechanism that utilizes multiple replicas of services on different computing nodes, and then intelligently distributes the requests to different service replicas. This mechanism effectively avoids the single-point failure and increases the availability of the system. Typical load balance algorithms include round-robin, random, and least requests.
- Circuit Breaking is a mechanism to limit calls to a service, such as the number of concurrent connections or the number of failed calls to record the number of call errors. Once the limit has been reached within a certain service, calling this service returns the error code immediately. Such a process allows users to limit the impact of failures, latency spikes, and other undesirable effects of network peculiarities.

### 3.8 Airborne Intelligent Application Layer

Focusing on two main airborne scenarios, i.e., autonomous flight scenario and air combat scenario, this paper lists six typical airborne intelligent applications. Specifically, for the autonomous flight scenario, three typical intelligent applications have been considered, i.e., takeoff, cruise, and landing. As for the air combat scenario, this paper lists detection, fight, and electronic warfare as three typical intelligent applications.

## 4 Outlook

### 4.1 Airborne OTA

Over-The-Air (OTA) [13] is a remote wireless upgrade technology, which has been widely employed in multiple fields, e.g., automotive and Internet of Things. Applying OTA technology to avionics contribute to rapid updates and upgrade of intelligent applications while aircraft are flying, which are of vital importance in the highly dynamic and competitive military environment. With the service-oriented avionics system architecture, based on its unified and formatted service interface, the intelligent applications can be more easily updated or upgraded with the OTA process.

### 4.2 Drag-and-Drop Software Development

The drag-and-drop application development platform [14] enables developers to quickly complete simple application development for new scenarios through drag-and-drop operations or a small amount of code. Such a platform is especially helpful when developing

avionics intelligent applications, which are commonly composed of several functions and might be updated frequently. The drag-and-drop application development platform could be conveniently implemented with a service-oriented architecture. Specifically, first devise each of the required functions as an atomic services, then define the unified and formatted service interfaces, and finally implement the application by service composition with a drag-and-drop user interface.

### 4.3 System Intelligence

System intelligence [3] is an advanced stage in the development of military intelligence, which aims to intelligently integrate all combat units for taking effective military operations. Specifically, system intelligence requires intelligent control, on-demand combat resource extraction and scheduling. The information communication and sharing as well as interoperability are the essence of system intelligence. The service-oriented architecture is a potential solution to realize the system intelligence with the unified service interface, service registration, and service discovery.

## 5 Conclusions

In order to provide a more intelligent and flexible avionics system, a Service-oriented Intelligent Avionics System Architecture, named as SIASA, has been proposed. In this paper, firstly, the development trend of avionics system architecture and the related work on service-oriented architecture is introduced. Then, the proposed SIASA is presented. Specifically, SIASA consists of seven services, i.e., airborne intelligent chip layer, abstraction layer, hashrate abstraction layer, equipment abstraction layer, airborne service capability layer, airborne intelligent application atomic service layer, and airborne intelligent application layer. With these layers, SIASA is able to make effective and efficient updates and well support the intelligent applications. After that, some advanced scenarios supported by SIASA have been presented, including airborne OTA, drag-and-drop software development, and system intelligence.

**Acknowledgement.** This work was partially supported by Aeronautical Science Foundation of China—Aerospace Artificial Intelligence Special (Grant No. 2018ZC31003).

## References

1. Moor, J.: The dartmouth college artificial intelligence conference: the next fifty years. *AI Mag.* **27**(4), 87–91 (2006)
2. Zhou, J., Li, P., Zhou, Y., Wang, B., Zang, J., Meng, L.: Toward new-generation intelligent manufacturing. *Engineering* **4**(1), 11–20 (2018)
3. Lei, H., Yao, C.: Technical architecture of aviation artificial intelligence for military application. *Navig. Position. Timing* **7**(1), 1–11 (2020)
4. Zhou, Q., Li, B., Chen C., Zheng, X.: Research on avionics system architecture design technology of integrated AI. In: *Proceedings of the Chinese Society of Aeronautics and Astronautics Youth Science and Technology Forum'2020*, pp. 1054–1061 (2020)

5. Collinson, R.P: Introduction to Avionics Systems, 3rd edn. Springer (2013)
6. Jin, Y., et al.: The Characteristics and technical approaches of avionics system in the future from the new concept of combat. *Aircr. Des.* **41**(3), 49–52 (2021)
7. Perrey, R., Lycett, M.: Service-oriented architecture. In: Proceedings on Symposium on Applications and the Internet Workshops'2003, pp. 116–119 (2003)
8. Yan, L., Zhu, G., Qu, G., Peng, Q.: Research on the architecture development of avionics system core processing platform. *Avion. Technol.* **52**(3), 23–29 (2021)
9. Blasch, E., Kostek, P., Pačes, P., Kramer, K.: Summary of avionics technologies. *IEEE Aerosp. Electron. Syst. Mag.* **30**(9), 6–11 (2015)
10. Watkins, C.B., Walter, R.: Transitioning from federated avionics system architectures to integrated modular avionics. In: Proceedings on Digital Avionics Systems Conference, pp. 1–10 (2007)
11. Gaska, T., Watkin, C., Chen, Y.: Integrated modular avionics-past, present, and future. *IEEE Aerosp. Electron. Syst. Mag.* **30**(9), 12–23 (2015)
12. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: service-Oriented Architecture Best Practices, 1st edn. Prentice Hall (2004)
13. Halder, S., Ghosal, A., Conti, M.: Secure over-the-air software updates in connected vehicles: a survey. *Comput. Netw.* **178**, 107343 (2020)
14. Joy, J.: Review on different types of drag and drop mobile app development platforms. *Int. J. Comput. Sci. Eng.* (2018)