IOSR Journal of Business and Management (IOSR-JBM)
e-ISSN: 2278-487X, p-ISSN: 2319-7668. Volume 25, Issue 3. Ser. III (March. 2023), PP 35-43
www.iosrjournals.org

A Study of an Organizational Structure for Flight Simulators Design: A System Design Approach to Artificial Intelligence

Hung Q. Tran, Mike Pacitti
CAE USA, FL, USA

Abstract: *Traditionally, flight simulation engineers designed their software by reductionism, i.e. functional decomposition. Software developers assumed that a system is bounded and can be decomposed into smaller components and subsequently, they can break down into several processes. For this type of system architecture, the system components communicate with each other via data and control flows using pre-defined interfaces. Generally, this system architecture does not require a central processor because the smart properties of the system reside within each component of the system.*

Background*: Flight Simulator (FS) engineers develop software to simulate the functionalities of aircraft equipment. To simulate an aircraft equipment system, whether it an avionic box or a sensor system, software developers normally model the system by reductionism, aka "Functional Decomposition". This approach involves breaking down complex software systems into smaller, more manageable components, and focusing on the essential functionality that is needed to meet the requirements of the system. It involves eliminating unnecessary features, reducing dependencies between components, and simplifying algorithms and data structures.Functional decomposition methodology is not optimal to design simulation software with intelligent properties.*

Objective*: An artificial intelligence system possesses the property of mimicking human thinking and behavior. Most importantly, it designed to take the same decisions that a human expert would have reached if he/she had time to process all the data available and assess the situation. A traditional functional decomposition system architecture is not appropriate for this type of system design, because, in order to obtain an emergent behavior (e.g. artificial intelligence), a central processor is essential to fuse and process the data produced from the system components. Therefore, there is a need to adapt traditional engineering organizational structure to effectively implement this type of simulation systems.*

Conclusion*: In this study, we propose a collaboration team structure to enable effective design and implementation of simulation software with AI property. Comparing to the legacy functional teams organization structure, the proposal organization structure should help to address the challenging situation of designing of simulation software of today-advanced modern military aircraft.*

Key Word*: Flight Simulation, Engineering Organizational, System Design, Artificial Intelligent*

Date of Submission: 10-03-2023 Date of Acceptance: 23-03-2023

I. Introduction

In advanced modern military aircraft, the number of data and information that the flight crew must absorb increase constantly, due to the expanding complexity and number of avionic equipment in the flight deck[1]. As these systems become increasingly complex, pilots' mental and physical workload will also increase beyond realistic limits. Therefore, expert systems with artificial intelligent properties are designed to assist the pilots in critical decision-making processes. Recently, a number of AI-based applications are designed for use in military combat aircraft, including weapon delivery, smart counter-measure or threats avoidance. Tactical displays on today-military aircraft are not solely used to display situation awareness, but it works in conjunction with a number of systems such as Navigation Support System (NSS), Threat Avoidance (TA), Electro-Optical Infra-Red (EO/IR) or Weapon Delivery System (WDS). This type of tactical display system is designed not only to assisting pilots in their decision-making process, but it performs the tasks intelligently. The system sees the world through it EO/IR camera, it understands and categorizes the threats using its internal database, it computes the most optimal route, by considering the environment constraints (e.g. weather, terrain, etc.), to engage the threats and uses the most appropriate available weapon on-board to destroy the targets. In order to perform this task efficiently, the system must incorporate a central processor to gather, fuse, and process the data from various sensors and avionics equipment. System engineers describe this type of system architecture as a “Systems of Systems” (SoS).

Flight Simulator (FS) engineers develop software to simulate the functionalities of aircraft equipment. To simulate an aircraft equipment system, whether it an avionic box or a sensor system, software developers normally model the system by reductionism. They assume that a system is bounded and can be decomposed into smaller components, and then each component, i.e. smaller systems, can be simulated independently with detailed processes and are interfacing with each other via data and pre-defined interfaces. For instance, a tactical mission planning system can be composed of a navigation system, a communication system, sensors, terrain database, electronic warfare, and displays. Each of those systems can breaking down further to more smaller systems and simulate independently. Generally, this functional decomposition approach had a strong relationship to the organization/team structure. Thus, it is not a coincidence that most flight-simulation companies organized their software engineering department by functional groups. A Work Break-Down Structure (WBS) is defined to organize the overall work and define deliverable products. Product specifications and software requirements are reviewed, analyzed, and assigned to each functional group for implementation. Software products will be integrated later in the development cycle via data and pre-defined interfaces. For this reason, functional groups normally employed developers with expertise pertinent to each discipline. This type of organization is recognized as a “top-down” hierarchy and is characterized by command-and-control systems of authority from-the-top to-the-down.

Nowadays, functional decomposition design still represents a popular approach for flight simulation engineering, because it still provides a number of benefits, such as modular design, software reuse, easy to inspect, etc. However, for system with intelligent properties, functional decomposition methodology is not optimal. Since this design approach uses a data-driven approach to interface with each other, and because the smart property of each component of the systems is designed and embedded within each component, intelligent properties are difficult to transfer from one component to another. While it is possible to create a more complex interface to communicate not only the data but also the intelligent properties. Such design is difficult to define and subsequently assign system requirements to the traditional functional groups. A better approach to design systems with intelligent properties is to mimicking a natural system by designing a central processor, i.e. the “brain” of the system. In this design methodology, the intelligence is not within the components, but at the central processing. This processor collects the data of the “unintelligent” components, fuse and process. The central processor may also issue commands to components to execute an operation and send back the results. This design approach is now employed in many modern aircraft, for example, a Common Intelligent Display (CID), or an Advanced Counter-measure Dispenser System (ACDS), or an Intelligent Threat Avoidance System (ITAS) are already in-service on many military aircraft. An intelligent system can be characterized by the following abilities:

- learn or understand from experience
- acquire and retain knowledge
- respond quickly and successfully to a new situation
- use of the faculty of reason in solving problems

A system with artificial intelligent properties is characterized as a system with emergent behavior, i.e. unbounded. Because each system is a SoS with emergent behavior, the resulting system is not equal to the sum of its components. For instance, when observing the system at a macro scale, the intelligent properties of the system cannot be identified at the level of its components. The intelligent property emerges only when these components are interface together. For this reason, if the system is broken down into smaller systems and processes, it not possible to observe its capabilities as a whole. Unfortunately, system functional decomposition just does that: it uses the WBS and decomposes a system into smaller systems until the system requirement engineer can assign each requirement to a corresponding functional group.

With the above attributes of an intelligent system, it very difficult to map the system requirements derived from these attributes to the WBS and assigned to traditional functional groups. Therefore, to model and simulate systems with AI, it will require a different organization structure than the traditional functional groups. This study will examine a system architecture that mimics the structure of natural system and its ability to embed artificial intelligent properties. Subsequently, the study will define how an organization must look like to effectively develop and implement this type of system.

II. Background

2.1. Emergent Behavior in System of Systems

System of Systems (SoS) represents a collection of systems that are interconnected together to joint theirs resources and capabilities to produce a new and more complex system. The resulting system enhances the overall performance by providing new capabilities. SoS architecture is mainly focused on the integration of existing systems to produce new operational capabilities that are not normally available from each individual system if operating separately (e.g. emergent property). This characteristic of a complex system is commonly though as "the whole is not equal of the sum of the constituents". As the development of SoS gains traction in the System Engineering field, the issue of "emergent behavior", whether they are desirable or not, is becoming critical and require a significant effort from system engineers to analyze the emergent behaviors of a SoS. Complex systems are often characterized by the fact that they are "hard to design and understand". This characterization is reflected by the fact that a complex system is normally designed by inter-connecting several systems together, therefore the resulting complex system contains many interactions through several layers of interfaces make it difficult to analyze and predict its general behavior. However, the notion of complexity does not only mean that the architecture is complex and difficult to understand, but it also represents by several additional attributes. The most relevant attribute of a complex system is its emergent behavior. Emergent properties are often used to distinguish between "complex systems" and "systems that are complicated". There are many definitions of emergent behavior of a SoS. Emergent behavior was defined as the consequent of the action of combining simple systems to produce a complex system [2]. Emergent behavior is sometime also defined as "something that cannot be predicted through analysis at any level simpler than that system as the whole [3]. Fisher (2006) defined emergent behavior as a cumulative effect of the actions or interactions amount components of a complex system [4]. Nonetheless, studies on emergent behavior are disagreed on whether it can be always be understand, predictable and at some degree, controllable. Fromm (2006) identifies four type of emergence behavior, of which two are predictable and two are not [5]. Therefore, properties that emerged from a SoS can be desirable and intentional, but they can be also harmful, for example, if they undermine important safety requirements when these emergent properties are completely unexpected. One of the predictable and desired emergent property of a SoS is Artificial Intelligent (AI). Generally, a SoS that designed with AI capability represents a system that when viewed from a macro-scale, the AI behavior of the SoS cannot be observed from the individual systems that composed it, because AI can only emerge through the interaction of systems and a clearly defined interface between these systems.

2.2. System Approach to Artificial Intelligence

The data-driven system approach assumes that a system can recursively break down into independent components until each component is simple enough to understand and develop a solution. The system architecture where sub-systems interface with each other via data and control flows is illustrated in Figure 1. Depending on the simulation environment, these systems interface with each other via a shared memory, common database or data bus. This type of system approach is normally employed by most flight simulator manufacture.The data-driven system approach assumes that the "smart" is contained in each sub-system, so only data and sometime commands are sent from one sub-system to another one. Consequently, no major processing is performed outside of each node. A typical application that such system approach is suitable would be a missile threat is detected and processed by a Missile Warning System (MWS). The advantage of a data-driven system approach is it very straightforward to assign WBS elements to the corresponding functional groups. However, for applications that must contain an element of artificial intelligent, the data-driven design approach is not sufficient, because there is a lack of central processing which is essential to fuse and process data from a various of sub-systems.

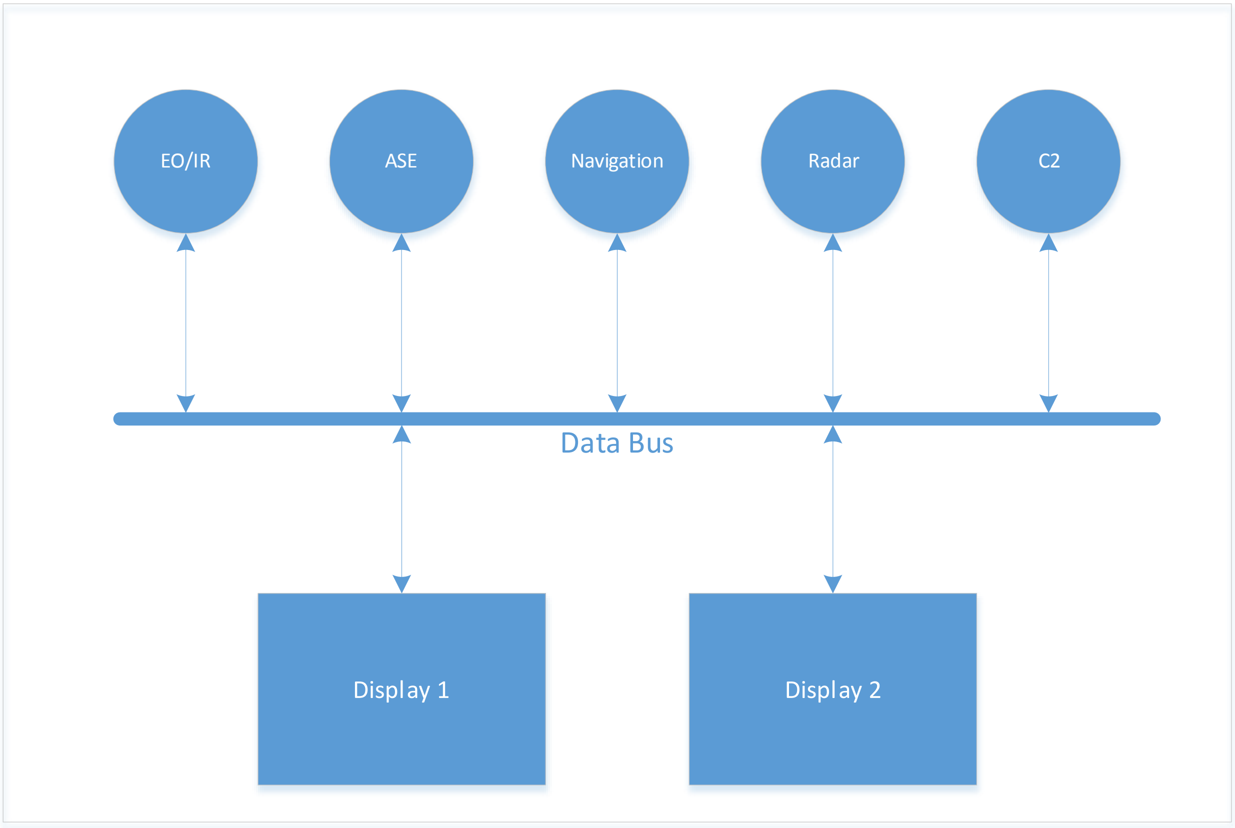

Figure 1. Data-driven System Design Approach

No system is useful in its isolation, but instead, from the perspective of how it interacts with others systems and most importantly with its environment. Therefore, the effectiveness of this system is depending on how it understands the environment in which it operates, i.e., how the environment data is acquired and manipulated by this system [7]. An example of this system approach is in the field of modern aircraft cockpit control and flight deck management. In this case, these systems utilized an array of sensors (e.g., Multi-Sensors Fusion Technology) to dynamically interact with its environment. Through a central processor, fused information is processed to assist the crew to detect, track and identify hostile targets more rapidly and accurately. Additionally, it can provide to the crew essential information to analyze the situation and direct the crew to the most optimal route to reach the destination without having to confronting the hostile threats. This represents a characteristic of a smart cockpit management system. Nowadays, most smart aircraft development involves multiple sensor systems. Sensor system such as Radar, Electro-Optical Infra-Red (EO/IR), Radar Warning Receiver, Electronic Measuring Support (ESM), Identification Friend and Foe (IFF) provide useful tactical information to a central processor for processing. The outcomes of such processing assist the crew to analyze the current and future combat theater situation. To be useful, tactical data from sensor systems must be integrated and processed by an intelligent central processor. Multi-Sensor Fusion (MSF) fused sensors data on different levels of information, such as data layer, feature layer, and decision layer [8]. For this reason, this approach is very effective to optimize large volumes of data and is implemented by combining information from multiple sensors to achieve inferences that are not feasible from a single sensor. For instance, a Tactical Assistant and Smart Display (TASD) system on today-military aircraft represents a category of system that was designed with artificial intelligent properties. This type of system works in conjunction with a number of sub-systems and is designed not only to assisting the crew in their decision-making process, but it performs the tasks intelligently. A tactical assistant and support system can aid the crew to analyze the situation and make the correct decision. It does that by displaying the most critical information on the cockpit displays, predict the outcomes, and recommend the crew suitable actions. To achieve this result, the system possess a central processing unit which is the “brain” of the system. The central processor interfaces directly with the processing nodes (e.g. sub-systems) to collect and process data. Additionally, it may also issue commands to the processing nodes. Each subsystem, has its own data processing capability, acquires environment data through theirs associated sensors. In addition to this capability, these systems also delivered data to an intelligent node (e.g. fusion node) for processing. Fusion processing is a technique used in artificial intelligence (AI) systems to combine information from multiple sources or sensors into a single representation. The role of fusion processing in an AI system is to improve the accuracy and reliability of the system by reducing the effects of noise, uncertainty, and incomplete information.Fusion processing can be performed at different levels in an AI system, including:

- Data fusion: Data fusion involves combining raw data from multiple sources into a single representation. This may involve preprocessing the data to remove noise or inconsistencies, and then combining the data using techniques such as averaging or weighting.
- Feature fusion: Feature fusion involves combining features or attributes extracted from the data into a single representation. This may involve using techniques such as principal component analysis (PCA) or independent component analysis (ICA) to identify the most relevant features, and then combining these features using techniques such as max or min fusion.
- Decision fusion: Decision fusion involves combining decisions or outputs from multiple sources into a single decision or output. This may involve using techniques such as majority voting or Bayesian inference to combine the decisions or outputs and produce a final decision or output.

The design purpose of the intelligent node is to communicate and coordinate with other systems of the network. Figure 2 presents a system architecture of that concept.

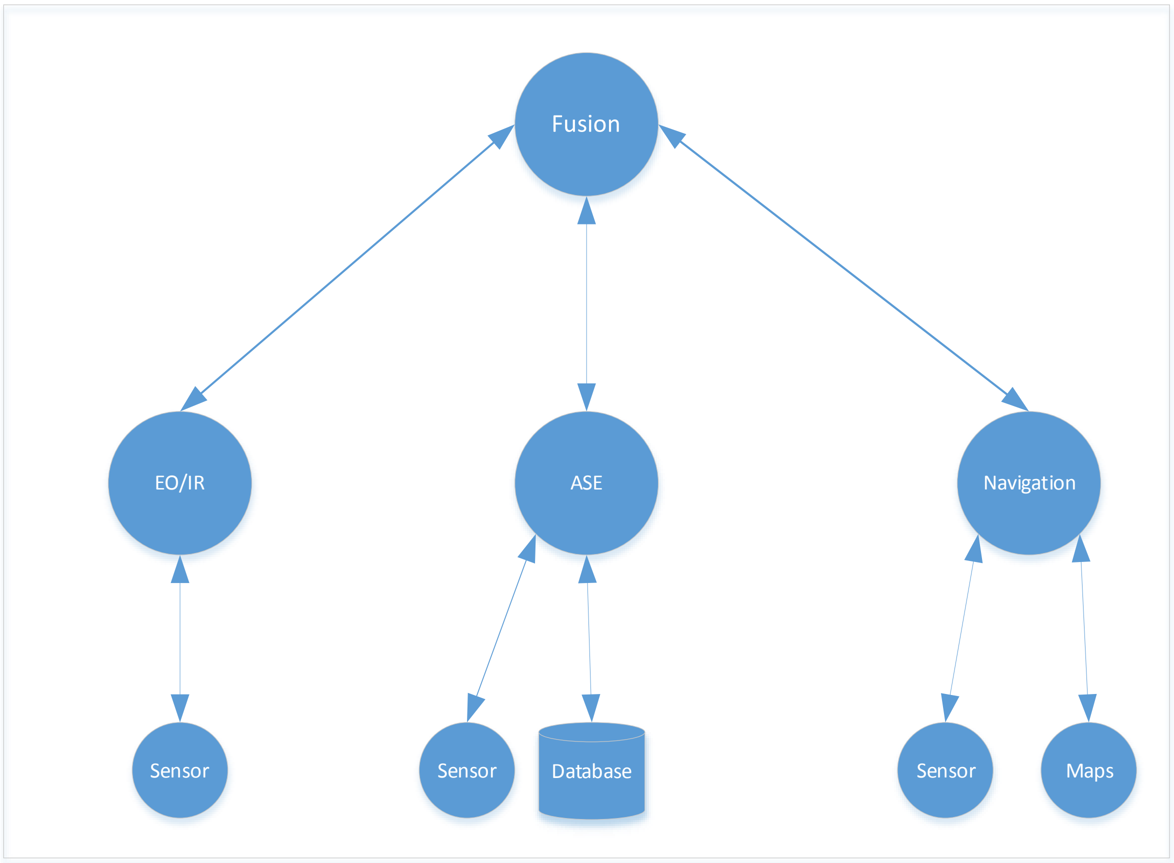

Figure 2. System Design Approach with a Data Fusion Node

A typical integration of this system approach is illustrated at Figure 3. Each sensor detects, tracks, and process the information independently, then send the end-results to the central processor. The artificial intelligent property of the system emerged at this stage of processing. For example, the intelligent central processor may try to determine if the radar threat detected by the RWR is the same threat that also detected by the ESM, and then combined them into a single threat. It may command the EO/IR to determine the range of the radar hostile threat. Additionally, the central processor may also use the GPS data to determine the exact location of the threat. Finally, the information of the threat is send to the cockpit for display. Without this type of central processing, the crew must perform mentally the same kind of analysis. For example, the cockpit may display one radar threat detected by the RWR and one radar threat detected by the ESM. The crew must mentally analyze the available data and decide if they represented a same threat. Additional the crew must mentally determine what the best available on-board weapon system to be used to engage the target, or if the hostile threat engaged the aircraft, what will be the best countermeasure method to use, etc.

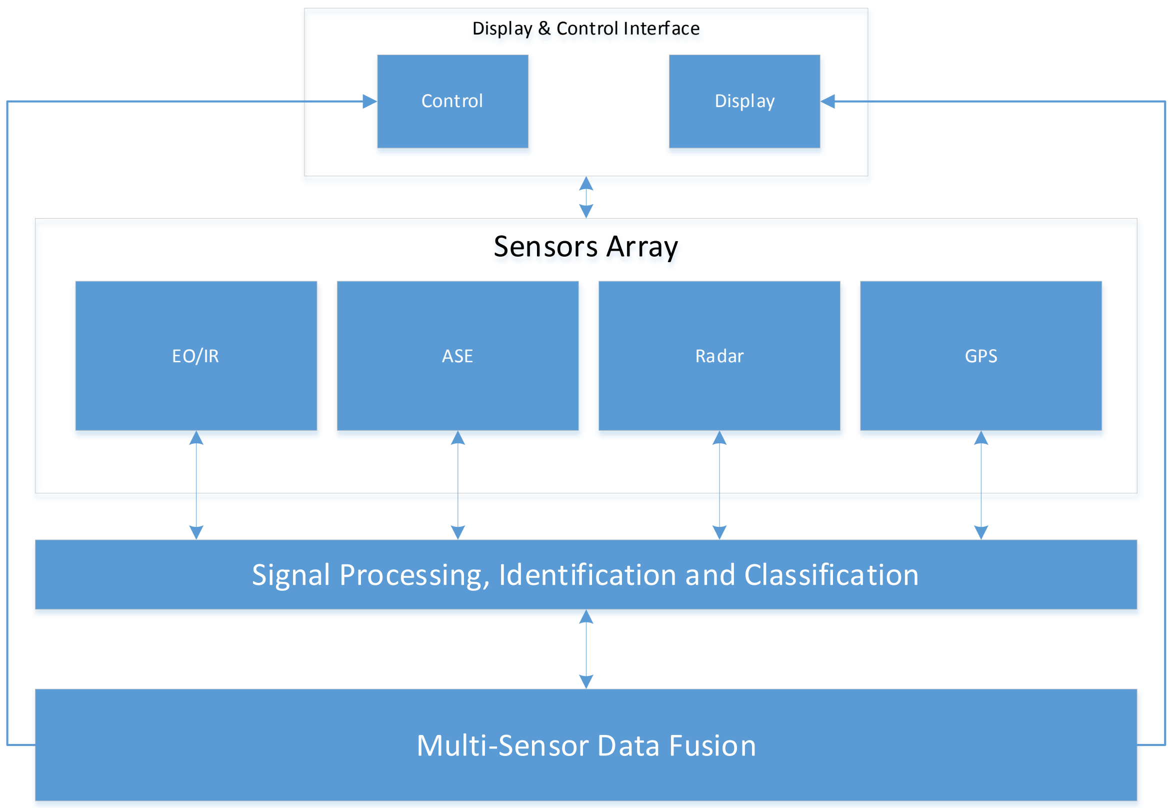
Display & Control Interface
Control
Display
Sensors Array
EO/IR
ASE
Radar
GPS
Signal Processing, Identification and Classification
Multi-Sensor Data Fusion

Figure 3. System-of-Systems Design Approach

This SoS design approach differs from the data-driven system approach in the way that rather than having each sensor on the platform operated as an independent system, an intelligent central processor dynamically manages and process data and information from various sensors. The system architecture as illustrated in figure 2 exhibits a fractal-like pattern, i.e. the basic structure remains the same regardless of the scale [9]. This characteristic is commonly known as “self-similarity”, i.e., exhibit similar patterns or structures at different scales or levels of magnification.For instance, the system as illustrated in Figure 2 can be nested to a more complex system. As seen in Figure 4, the fusion node in Figure 2 can just be another node of a more complex system. Consequently, a SoS can also be decomposed into smaller SoS. However, as oppose to the functional decomposition where the system is decomposed based on “discipline”, multi-sensor fusion system is decomposed based on ”service” as each fusion node in the system is designed to provide a very specific purpose. Additionally, the act of decomposing a complex system with fractal-like pattern does not necessary simplify it because the structure remains the same and appears equivalently complex. However, the decomposition does help in the understanding and characterization the system.

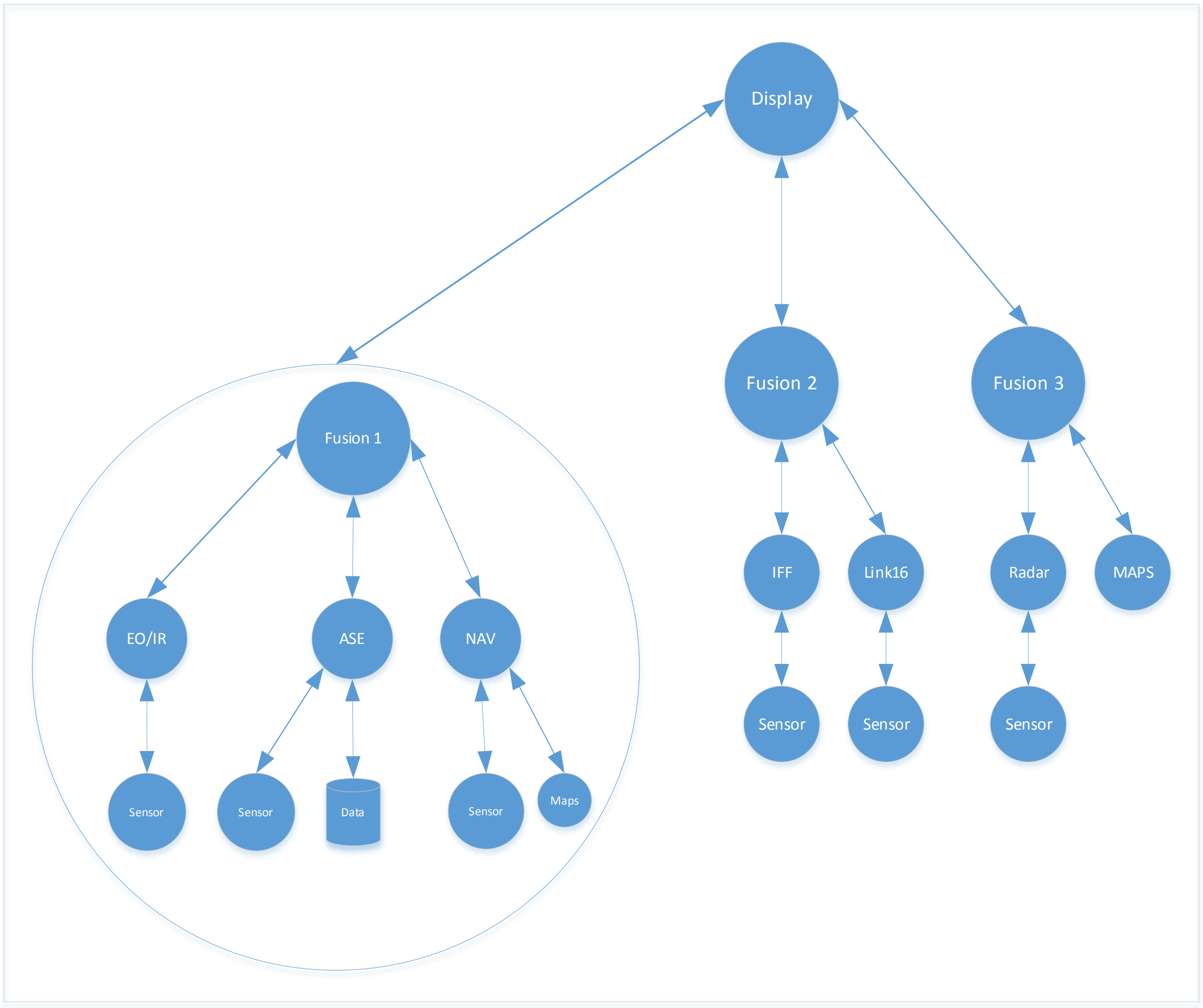

Figure 4. Fractal-Like System Design Approach

III. Flight Simulator Development Team Structure

As mentioning above, flight simulator organization builds the engineering team's structure with functional groups to reflect the WBS. Engineering functional group consists of engineers from the same functional area of the company (e.g. Avionic Systems, Aircraft System, etc.). In this type organization, system engineers decomposed recursively the specifications of the project into smaller specifications until each specification can be assigned to one or more functional groups. The purpose of this task is to be able to decompose the total work into manageable work packages. The components/systems that are defined at the lowest level of the WBS are then assigned to corresponding functional groups for development. This organization structure minimizes the interdependency across functional groups by assuming that each functional group had the required expertise to develop and implement the WBS elements that were assigned to the group. Such organization structure works well with the data-driven system approach, however it lost the effectiveness when dealing with systems that involved emergent property, or more specifically with systems with intelligent property. Therefore, there is a need to adapt the organization structure to effectively address the modeling and simulation of SoS that go beyond the traditional data-driven and control flow architecture.An engineering organization can be structured like a fractal system by adopting a hierarchical structure that is self-similar and exhibits similar patterns or structures at different levels of the organization [9].

3.1. Interdisciplinary Collaboration

There's a lot of debate about where AI capability should reside within a flight simulation engineering organizations. However, an acceptable consensus is a system design approach that enabling AI capability is much more optimal if developed by a cross-functional teams with a mix of skills and perspectives. In fact, consolidate the majority of AI and analytics capabilities within a cross-functional group (e.g. central "hub") will mimic the natural principle of AI system where a central processor is essential. This type of organization is known as an "Inter-disciplinary Development Team", which composed of individuals from different fields or areas of expertise who work together to develop a product, service, or solution. The team members have different backgrounds, knowledge, skills, and perspectives, which they bring together to solve complex problems and create innovative solutions.They are structuring by two different engineering development groups: Cross-functional and Functional teams. This type of organization approach is illustrated in figure 5.

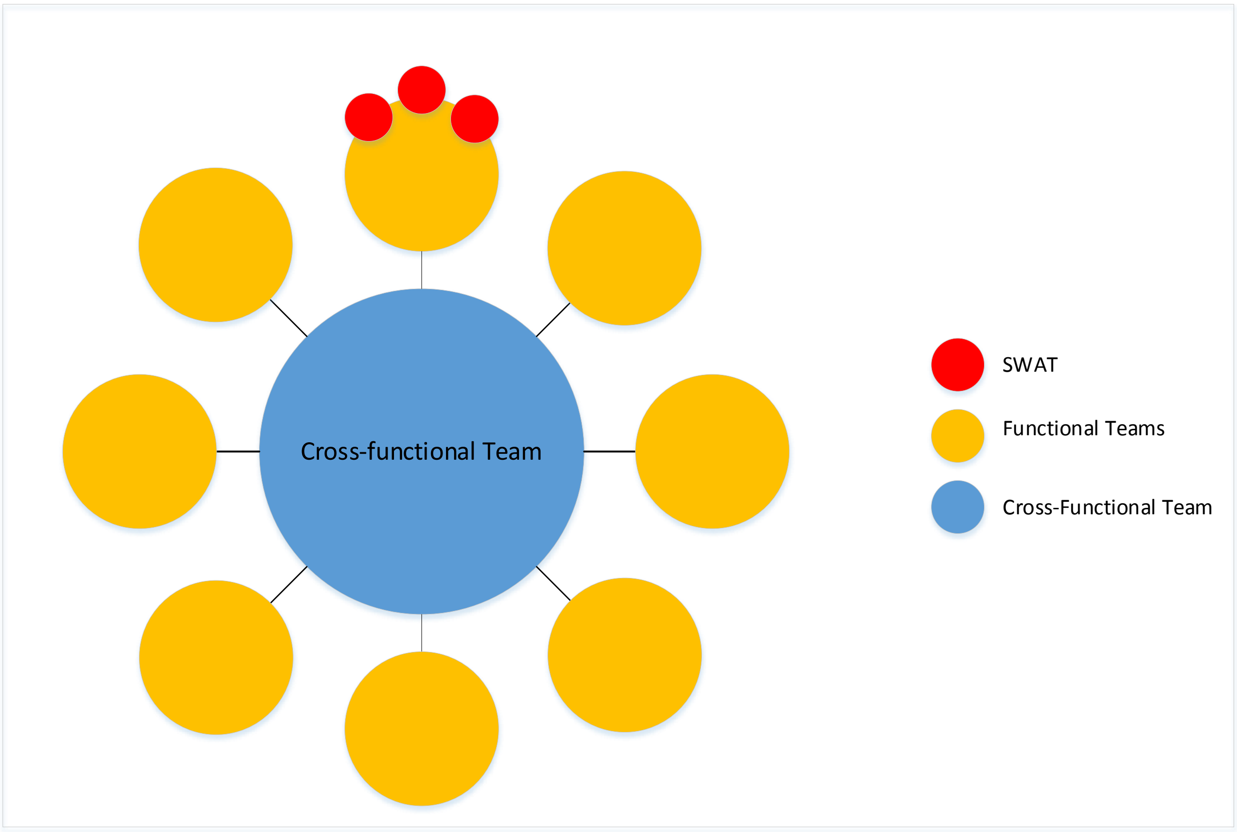

Figure 5. Engineering Organization with Interdisciplinary Collaboration

3.1.1. Cross-functional Team

Cross-functional team members have different skill sets, which they each leverage toward a common goal: in this case, the development of a software application with AI property. A few key representatives with appropriate skills and experience from different functional teams should be selected to compose the cross-functional team. Cross-functional team should be responsible for systems and software design related to AI. This team should nurture AI talent, create communities where AI experts can share best practices, and lay out processes for AI development across the engineering organization.The goal of a cross-functional engineering team is to bring together individuals with different perspectives and expertise to work collaboratively on a project. This approach can help to identify and solve problems more effectively and efficiently than if the work were done by individuals working independently.

3.1.2. Functional Teams

Legacy functional team members are developers with expertise pertinent to each discipline (e.g. avionic, or navigation, etc.). Functional teams represent specialized teams and do most of their work independently. A handful of responsibilities should always be owned by the functional teams, because they're closest to those who will be using the AI systems. Among them are tasks related to adoption and integration of AI software produced by the cross-functional team with their specific simulation systems.The goal of a functional engineering team is to bring together individuals with a deep understanding of a particular subject matter to work collaboratively on a project. This approach can help to ensure that work is done efficiently and effectively, as team members are already familiar with the processes and best practices in their respective fields. Functional team may also compose of "SWAT teams" that are specialized of different simulation platforms (e.g., C-130J Aircraft).

IV. Conclusion

Functional decomposition methodology is not optimal to design simulation software with intelligent properties. In this study, we propose an organization team structure to enable effective design and implementation of simulation software with AI property. Comparing to the legacy functional teams organization structure, the proposal organization structure should help to address the challenging situation of designing of simulation software of today-advanced modern military aircraft. However, there are tasks needed to be completed prior to moving forward with the present proposal:

- Review and update the WBS to add system components that contain AI property (e.g. central processing, data fusion, etc.).
- Build the cross-functional team, i.e., recruit and allocate appropriate personnel to this team. Requirements pertinent to AI must be assigned to this team during the requirement analysis phase.
- Develop the mechanism that will allow the collaboration between the cross-functional team and the functional teams.

Finally, we are still at the early stages of the AI era and it will continue to evolve, therefore by adopting the above strategies, an engineering organization can be structured like a fractal system [10], with self-similar units and patterns of behavior at different levels of the organization. This can promote greater efficiency, flexibility, and adaptability, allowing the organization to respond more effectively to changing conditions and challenges.

References

[1]. Steven J. Thatcher. The use of artificial intelligence in the learning of flight crew situation awareness in an undergraduate aviation programme. World Transactions on Engineering and Technology Education.Vol.12, No.4, 2014.
[2]. Rolling. A., Ernest. A. On game Design. New Riders Publishing, Indianapolis, IN 2003.
[3]. Dyson. G. Darwin among the machine. Addison-Wesley, Reading, MA 1997
[4]. Fisher. D. An emergent perspective on interoperation is System of Systems. CMU/SEI-2006-TR-003. Pittsburg,
[5]. J. Fromm. Types and forms of emergence. arXiv preprint nlin/0506028, 2005
[6]. Tagarev, T., Ivanov, P., "Computational Intelligence in Multi-Source Data and Information Fusion," Issue: Information and Security, vol. 2, 1999.
[7]. RampogalKashyap. Artificial Intelligence Systems in Aviation. Cases on Modern Computer Systems in Aviation (pp.1-26). 2019.
[8]. Bahador, K., Alaa, K., Fakhreddine O, K., Saeodeh N., R. Multisensor Data Fusion: A review of the state-of-the-art. Information Fusion, Vol 14, Issue 1, Page 28-44. 2013.
[9]. Mandelbrot, B. B. (1983). The fractal geometry of nature. New York: W.H. Freeman.
[10]. Janna Ray. Fractal Organisation Theory. Journal of Organisational Transformation & Social Change, Vol. 11 No. 1, 50–68. 2014.

Hung Q. Tran, et. al. "A Study of an Organizational Structure for Flight Simulators Design: A System Design Approach to Artificial Intelligence." *IOSR Journal of Business and Management (IOSR-JBM),* Vol.25, No. 03, 2023, pp. 35-43.