iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61836

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

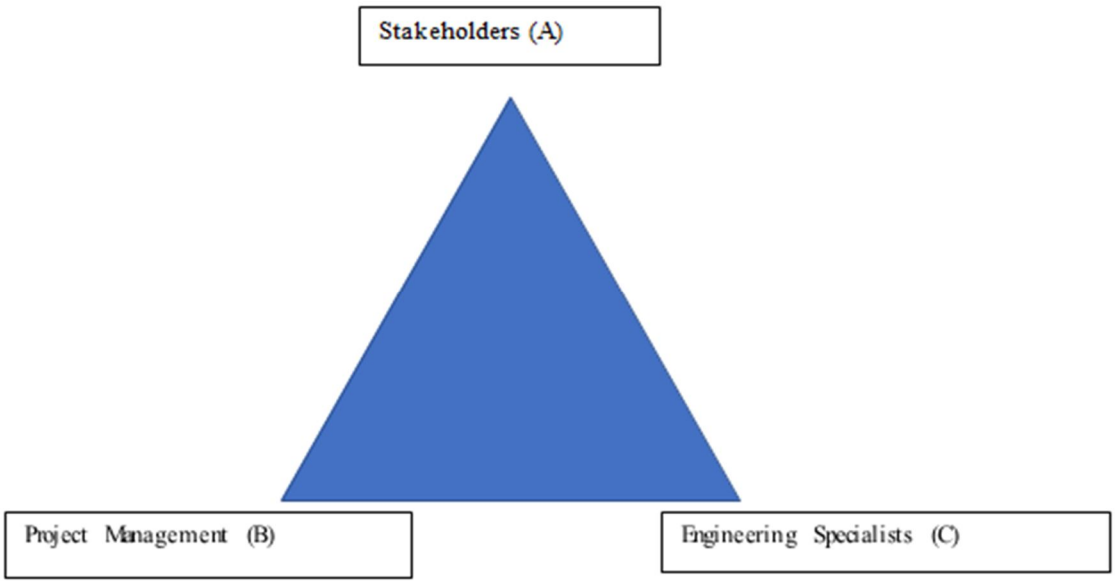
Avionics System Design using MBSE Methodology

Aryan Rahaman[1], Abhinav Sharma[2]
Amity Institute of Technology Aeronautical Engineering, Amity University Noida

Abstract: ***The West Air Sweden Flight 294 serves as a reminder of tragic crashes of aircraftdue to Avionics System failure. Due to malfunction of the Inertial Reference Unit (IRU), the aircraft lost its control resulting in tragic loss of life of the peopleon board. The Avionics System is a complex set of instruments, logical and physical links which must comply with the guidelines of Dob178c. The design of the system is complicated. This paper aims to understand the behavior of the Avionics System using the state-of-the-art Model Based System Engineering (MBSE) technology. This includes the two widely used tools – Capella and Simulink. We aim to understand the dynamic nature of avionics through these tools. Model Based System Engineering can be specifically beneficial because it will help the engineers to detect the flaws in the avionics system much earlier. This will potentially contribute to helping to prevent aircraft crashes due to avionics system failure. The idea of model-based system engineering includes the concept of taking a model as a reference to systematically solve the complex designs of any engineering stage. Before the concept of Model Based Systems Engineering came to be, Documents Based Systems Engineering was the trend in the system engineering department. However, it would take a lot of time, effort and accuracy to correctly pinpoint the information and accordingly design the system from a bunch of documents. MBSE, however, eases such kind of trouble by providing a model as a reference for the systems design.***

I. INTRODUCTION

Avionics system is a complex set of logical and physical links connected to different instruments displaying information on the screen for the pilot. This information is transmitted and processed through the navigation computer. Document Based System Engineering (DBSE) was introduced as method to aid in design of a product by utilizing documents and their written data. It majorly relies on the paperwork and management of the written data. However System Engineering is more complex than meets the eye. It is an interconnected and an iterative process of design between 3 different parties:



- Stake Holders: start the project, customers, assign the value to the product.
- Project Management: scheduling, budget
- Engineering Specialists: core, domain, knowledge

©IJRASET: All Rights are Reserved

A, B,C – The whole process is guided by Systems Engineering.
What is System Engineering?
It is a concept and a method of designing complex entities which cannot bedesigned as a single entity.
Example?
Supposedly there is an idea to make a drone. System Engineering helps toguide the engineering process and ensure that the project needs are being met.
Breaking down system engineering's function systematically:Project Objectives -> Different ideas to achieve this project Let these ideas be:

- *
- !
- .
- , ,

We select one of these after narrowing down the options through analysis.In our case * is selected.
However, * is complex and big.
The obvious approach is to break down "*" into smaller and simpler componentswhich can be easily designed and engineered. Lets say these smaller sub components are:

- |
- \
- /
- -

System Engineering will ensure each component above will do what its supposed to do. Then the proceeding integration of the components must perform the function. This will theoretically lead to the formation of a proper product. However, in reality, the process involves coordination between all three stakeholders, project management and the engineers in their own respective ways.
As we study this report, chapter 2 will talk about the Model Based System Engineering Technology and how the avionics system works. Chapter 3 will talkabout the software Capella and its different analysis layers. We will study aboutthe proposed work in capella and how this capella model can be further simulatedusing MATLAB. Chapter 5 will discuss the overall results and finally chapter 6 will cite references which inspired this work and report.

II. MODEL BASED SYSTEM ENGINEERING AND AVIONICS

System Engineering is additional work. To optimize the engineering process of acomplex product. It is, however, not a direct part of core engineering of the product. It acts more like a catalyst in an engineering process.

- Stakeholder Needs
- Discussing the requirements
- Planning a design
- Approving the design
- Actually, building the product based on the design
- Testing
- Market

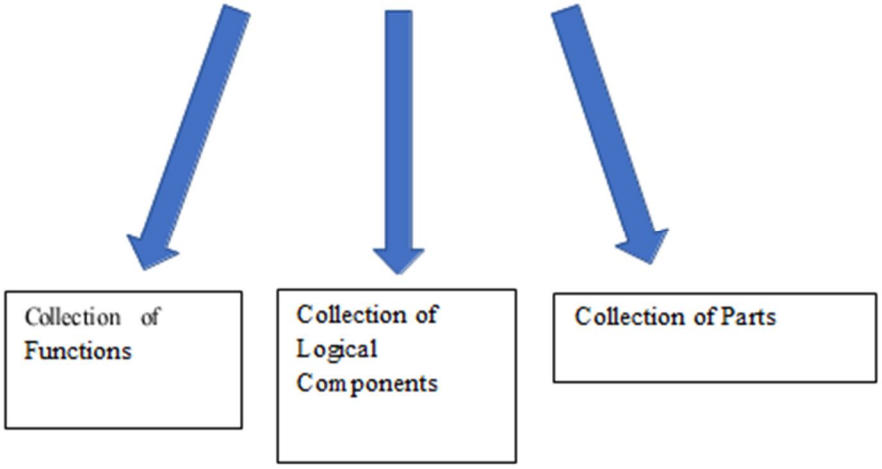
(not linear, iterative in nature)



Misconception: System Engineering helps optimize stakeholders, management and engineering for best outcome.
Reality: System Engineering helps optimize to get a suitable working solution.
To obtain the best outcome, you would have to gather knowledge over every single outcome which is not feasible. Most of these possibilities will have a deadend anyway.
Model -> Approximation of reality (physical, mathematical, behavior, architecture)
MBSE -> Using models for system engineering (making system level decisions)
For example: Model of an airplane is used for aerodynamics analysis. After that,other aspects are considered as well such as wing tip and airfoil design. These models can be put into simulation and testing.
Lets try to understand the layout of architecture of a complex system such as avionics:

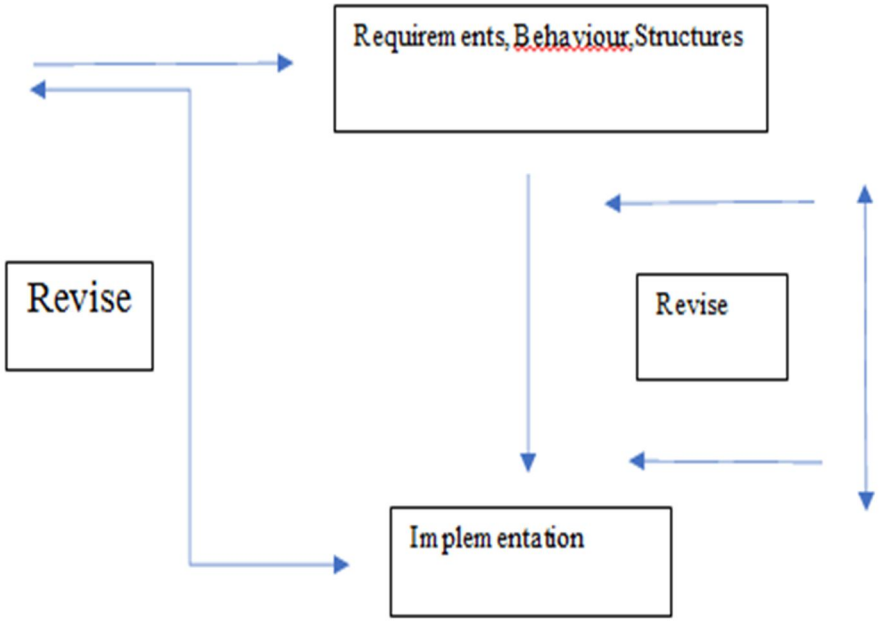
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

III. SYSTEM



This simple layout of architecture captures the relationship between elements. It's a mapping of the elements and how they interact with each other.

Stakeholder Needs



Requirement consists of:

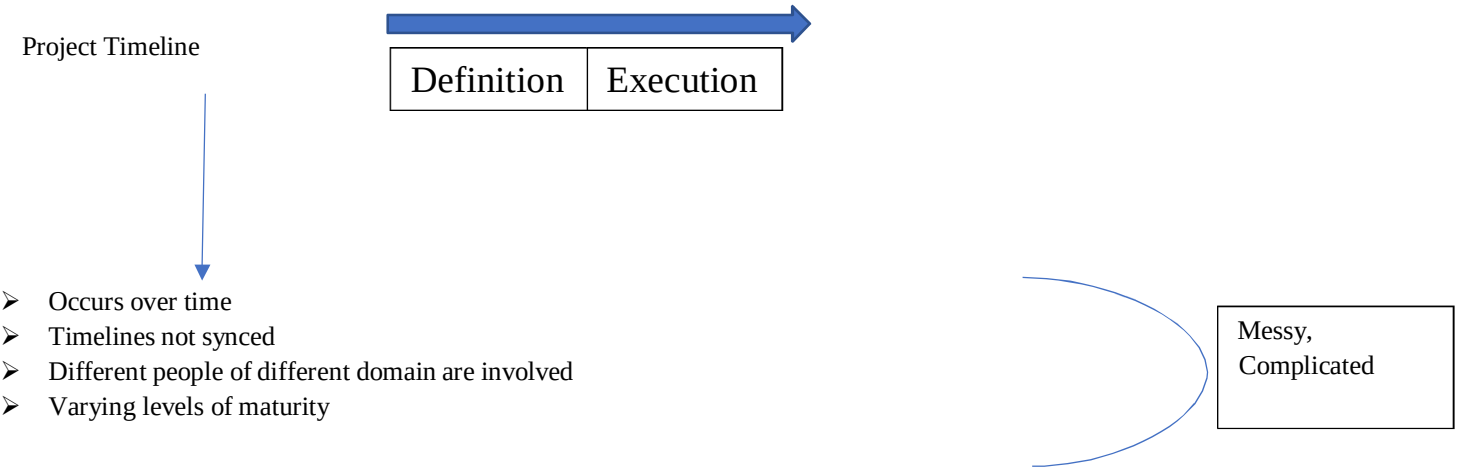
- A description of a particular need
- A rationale for why its valid
- A way to verify

Type of requirement are as follows:

- Functional (what functions need to be performed)
- Performance (how well the performance is)
- Constraint
- Environmental

©IJRASET: All Rights are Reserved

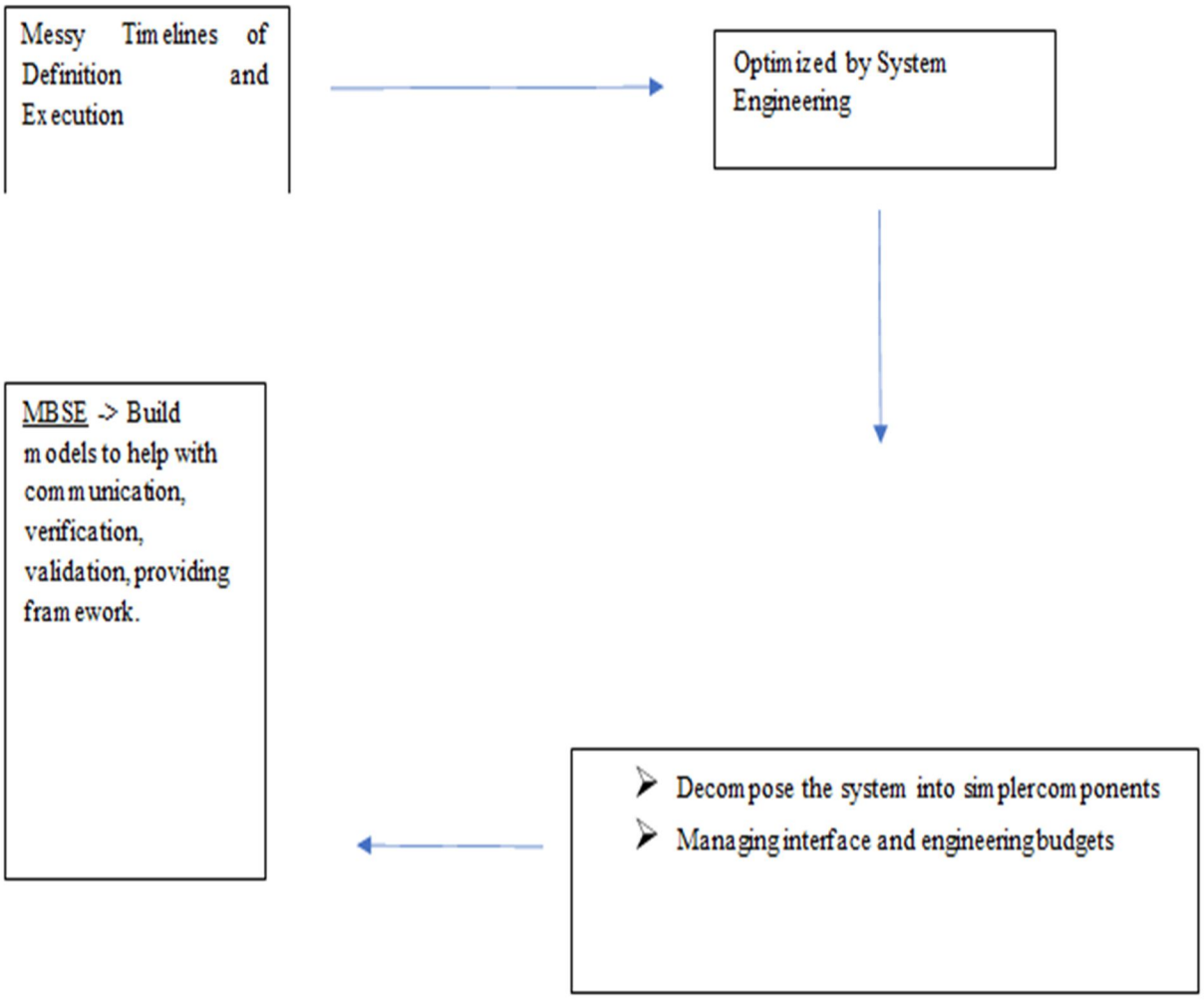
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com



Solution -> Model Based System Engineering

A model helps to provide a sort of reference to the domain experts for working on a problem. They can model the physics of a problem. Model provides the definition of the interface. The requirements go conveniently into the model. Model is the starting point of the implementation.

IV. COMPLEX ENGINEERING PROJECTS



©IJRASET: All Rights are Reserved

V. MODEL

VERIFICATION (IF I AMDOING IT RIGHT)

VALIDATION (IF I DID ITRIGHT)

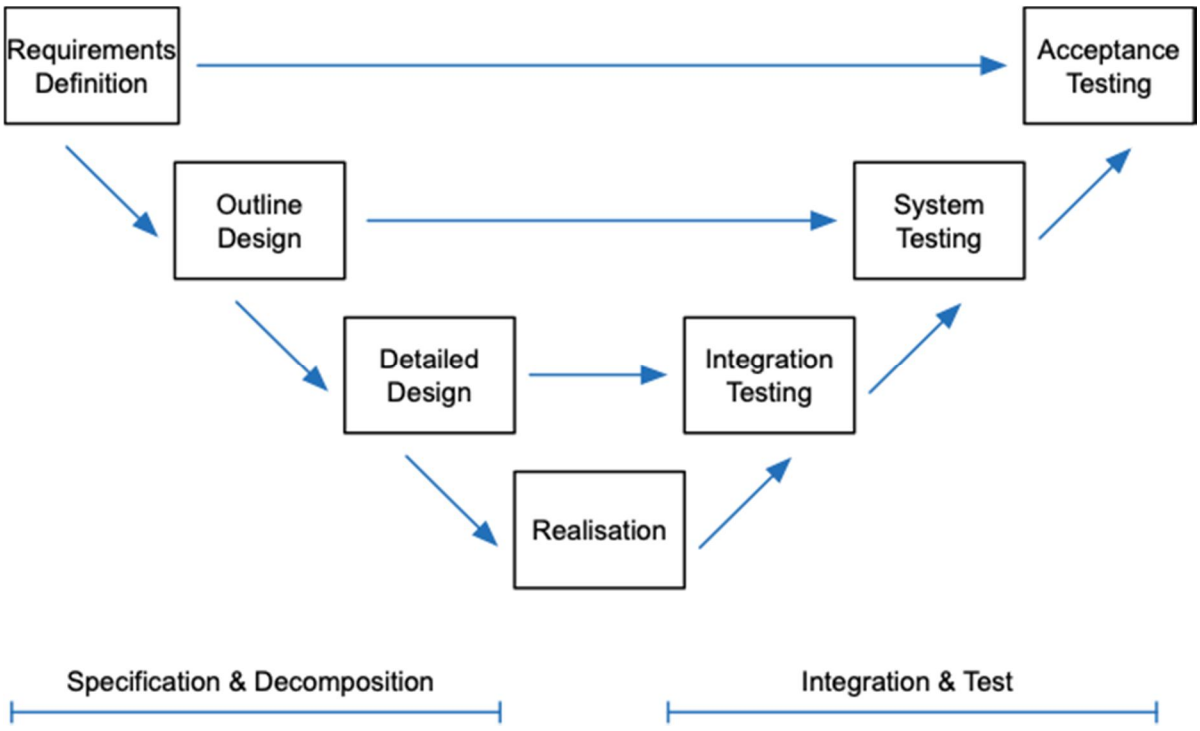
Testing is associated with every single phase of the life cycle of a product. Each component or stage is associated with testing. Testing always runs parallel to theprocess.

Verification Phase

- Requirements Analysis, System Design, Architecture Design, ModuleDesign.

Validation Phase

- Unit Testing, Integration Operations.



To further understand how well designed an avionics system should be, we must also understand the standards it must uphold according to the guidelines of Do178 and Do254.

Design of avionics and their approval is complicated because of complex software system and the guidelines set by Do178 and Do254. It is an iterative process and hence different domain people are needed.

MBSE will provide models as framework to improve the communication between the different elements and domains, help to figure out the defects earlier, any faultin the coding of the software. This helps us to understand the complexity of avionics.

Do178B

- Software will perform reliably in airborne environments.
- Worldwide avionics software guidelines to which all airborne software isrequired to comply with.

Software Levels (DAL): Levels E-A

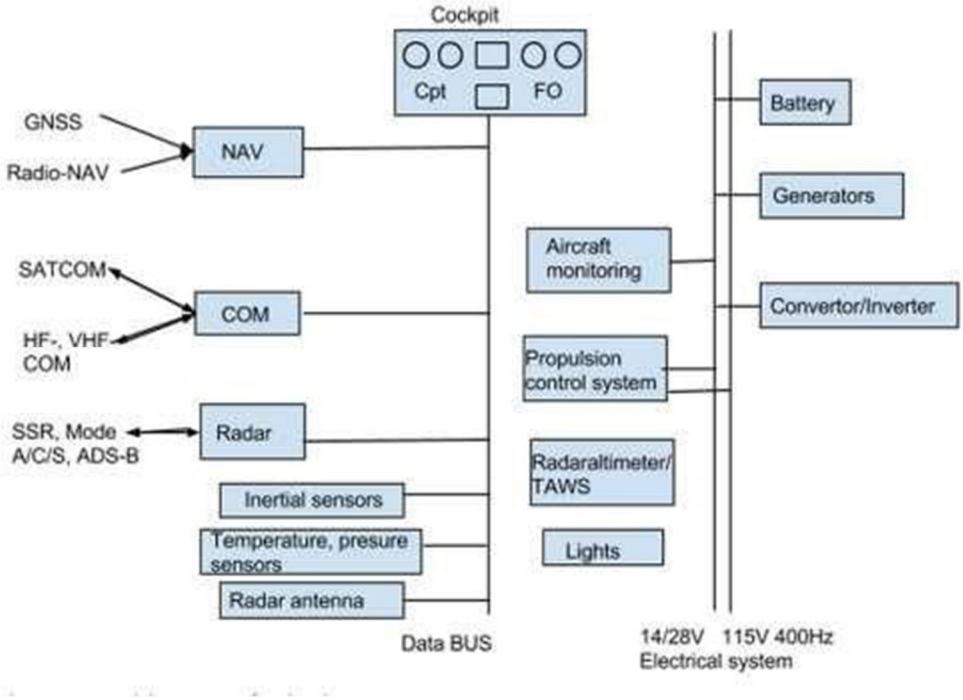
For the note, if the software is for Design Assurance Level A, no need tocomply with guidelines because there wont be any safety risk.

A -> lowest level

AVIONICS SYSTEM:

Avionics components are as follows:

- Communication System
- Display System
- Flight Control System (FLY BY WIRE)
- Navigation System
- Radar System



The figure above demonstrates the basic architecture of an avionics system.Understanding how avionics works:

Avionics -> Aviation Electronics

- Navigation
- Communication
- Surveillance

NAVIGATION -> Monitoring and controlling the movement of a craft orvehicle from A to B.

COMMUNICATION -> Forms of exchanging information from aircraft to Air Traffic Control and vice versa.

SURVEILLANCE -> Monitoring or close observation of behavior, activities or information for the purpose of influencing, managing or directing.

Avionics Devices:

1) NDB -> Non Directional Beacon

2) VOR -> VHF Omni Directional Range

3) DME -> Distance Measuring Equipment

4) TACAN -> Tactical Air Navigation System
5) LORAN-C -> Hyperbolic Radio Navigation System
6) ILS -> Instrument Landing System
7) GPS -> Global Positioning System
8) Auto Pilot Gyroscope -> For stability and correcting the orientation

VI. CAPELLA AND ITS MODEL

Capella is one of the most known choices for the purpose of system engineeringprocess in the avionics department. Capella works as the gateway between the requirements and the model of the design since you can map the requirements ofthe design into the capella model and treat it as the reference.

Models are a single source of truth. Queries are made on them and design iterations can be further made on them to properly define the problem and the requirements.

Capella model acts as a reference in the engineering process. Its use brings consistency. It enables to have modifications with more flexibility.

There are two main layers in a capella model.

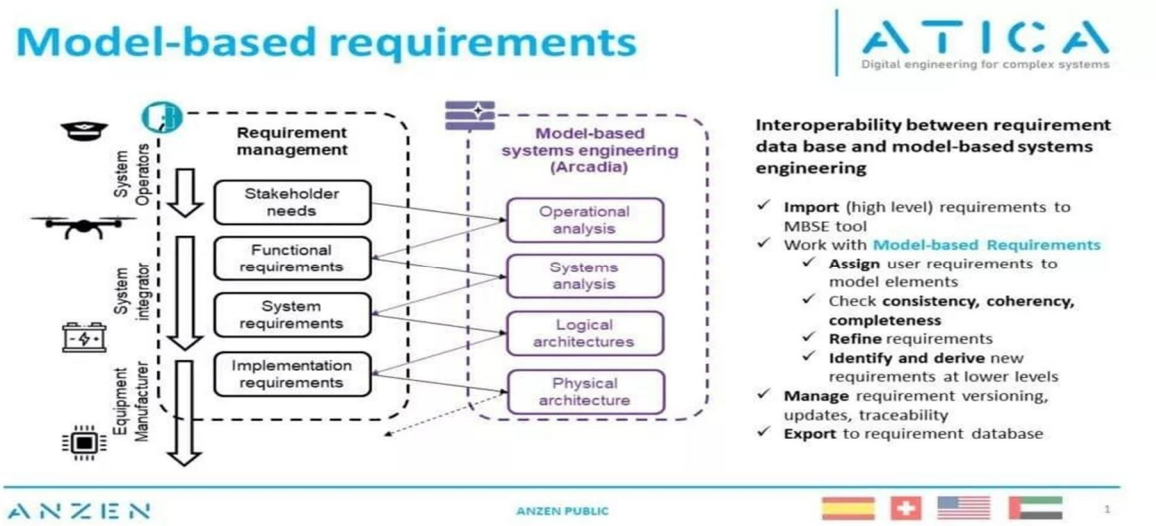
- Logical Architecture
- Physical Architecture

Several properties can be modeled in capella which can be further linked with other tools. In our case, this other tool will be MATLAB Simulink. We will use it for the simulation purpose of the capella model. However our work doesn't only stop with linking the capella model with other tools. We also need to build a solid link with the corresponding functions for the model which will help to achieve the requirements which were previously mapped into the capella model.

Arcadia layer	Requirements	Capability	Capability description	Functional	Structure	Modes and States	Data	Interfaces
	R-OA	OA1	OA2	OA3	OA4	M&S-OA5	D-OA6	I-OA7
Operational Analysis	Capture stakeholder requirements	Define Operational Capabilities	Define processes and scenarios	Define Operational Activities and interactions	Capture Operational Entities and Actors. Allocate Operational Activities to Operational Actors, Entities	Define operational modes and states	Define operational data model	Define interfaces and describe interfaces scenarios
	R-SA	SA1	SA2	SA3	SA4	M&S-SA5	D-SA6	I-SA7
System Analysis	Derive Stakeholder requirements and capture System requirements	Define System Missions and System Capabilities	Define Functional Chains and Scenarios.	Define System Functions. Define Functional Exchanges and components	Allocate System Functions to System and Actors	Define system modes and states	Define system data model	Define interfaces and describe interfaces scenarios Enrich Logical Scenarios.
	R-LA	LA1	LA2	LA3	LA4	M&S-LA5	D-LA6	I-LA7
Logical Architecture	Derive system requirements and Capture components requirements	Transition Capabilities Realization from system layer	Define Functional Chains and scenarios	Derive System Functions and define Logical Functions. Define Functional Exchanges and components.	Allocate Logical Functions to Logical Components	Define logical components modes and states	Define logical data model	Delegate System Interfaces and create Logical Interfaces. Enrich Logical Scenarios.
	R-PA	PA1	PA2	PA3	PA4	M&S-PA5	D-PA6	I-PA7
Physical Architecture	Derive logical requirements and capture physical requirements	Transition Capabilities Realization from logical layer	Define Functional Chains, Scenarios, and Physical Path	Derive Logical Functions and define Physical Functions. Define Functional Exchanges and components.	Define Physical Nodes and refine Behavioural Physical Components. Allocate Behavioural Components.	Define physical nodes modes and states	Define physical data model	Delegate Logical Interfaces and create Physical Interface. Enrich Physical Scenarios.

The figure above elaborates on the primary layers of capella. However for avionics department, our focus will lie on the logical architecture which would gather all the logical components and functions and the physical architecture which will implement these functions into reality.

The following image gives a clear pathway to how the layers of capella correspond to the requirements of the project's design.



A. Model Based FDIR Design:

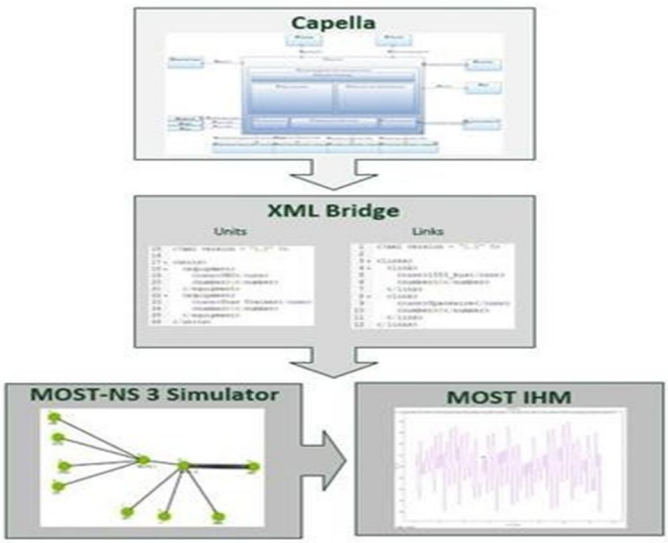
FDIR stands for Fault, Detection, Isolation and Recovery. This design aims to detect any fault in the model and provide a feedback while trying to minimize the errors and reduce the window of any fault in a component. As good as this sounds, every stage in engineering comes with its own risks and functions. FDIR carries a huge risk which sometimes is not even worth the function it provides and its drawback cannot be compensated enough.

In order for FDIR to be operational, the existence of aircraft/spacecraft database is necessary. However this is only possible during the testing and validation phase. This poses a huge problem since this directly leads to the lack of any early verification analysis and produces a lot of risk. Without the key stage of early verification phase, we cannot de-risk the testing phase to optimum level. This is a widely known problem amongst the FDIR engineers and needs to be constantly worked on. For now, a good solution would be GSTP design of FDIR. This design is obviously model based as well but follows more of an end to end MBSE solution.

After the discussion on FDIR it is agreeable to now proceed to RAMS model and its analysis.

RAMS stands for Reliability, Availability, Maintainability, Safety. While it does not directly contribute to the very design of the system, it holds the testament to the proper working properties of the design. It is kind of a safe proof of the model.

Now how do you transition from the capella model to its simulation? After all, its important to simulate the air/space traffic and dynamically study the model behavior as well.



B. From Modeling To Simulation

Thankfully there are quite a number of tools for this purpose. However MATLAB Simulink makes things much easier and simpler for us in a bigger picture.

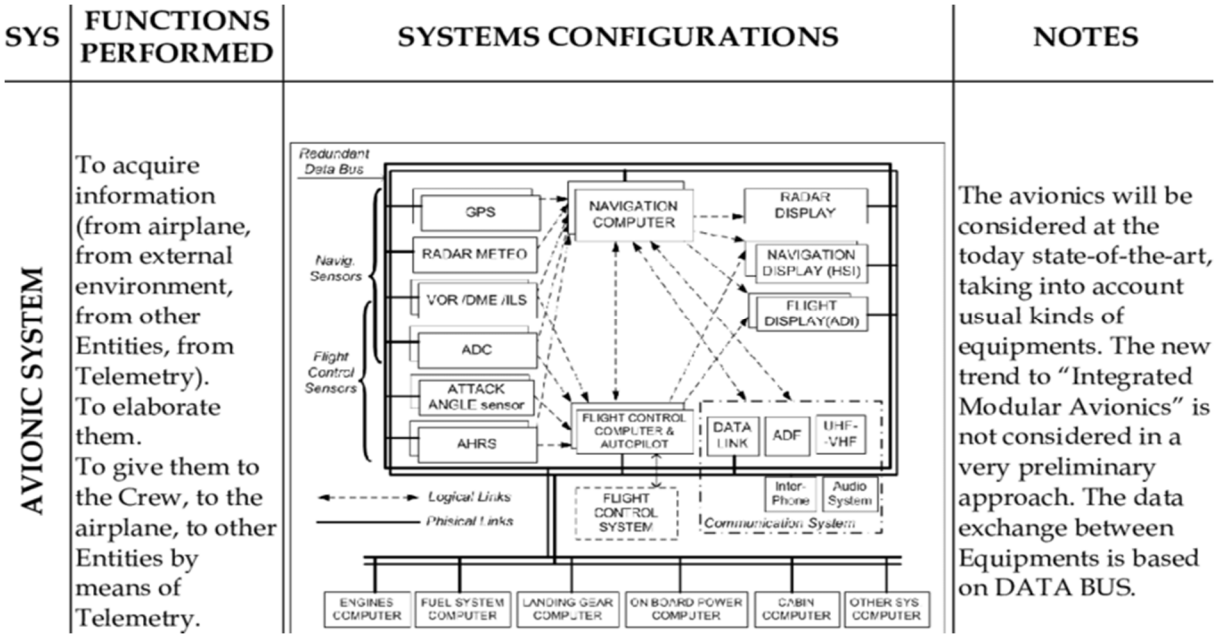
The model designed can be directly plugged into the Simulink and its traffic specifications can be mapped into the software which can be then dynamically processed giving us a good insight into the simulated product.



REQUIREMENTS MAPPED INTO THE MODEL

VII. PROPOSED WORK AND RESULT

To begin with the design of an avionics system we need to understand the fundamentals of avionics system layout and the tools aimed for MBSE breakdown.



A. Tool for MBSE

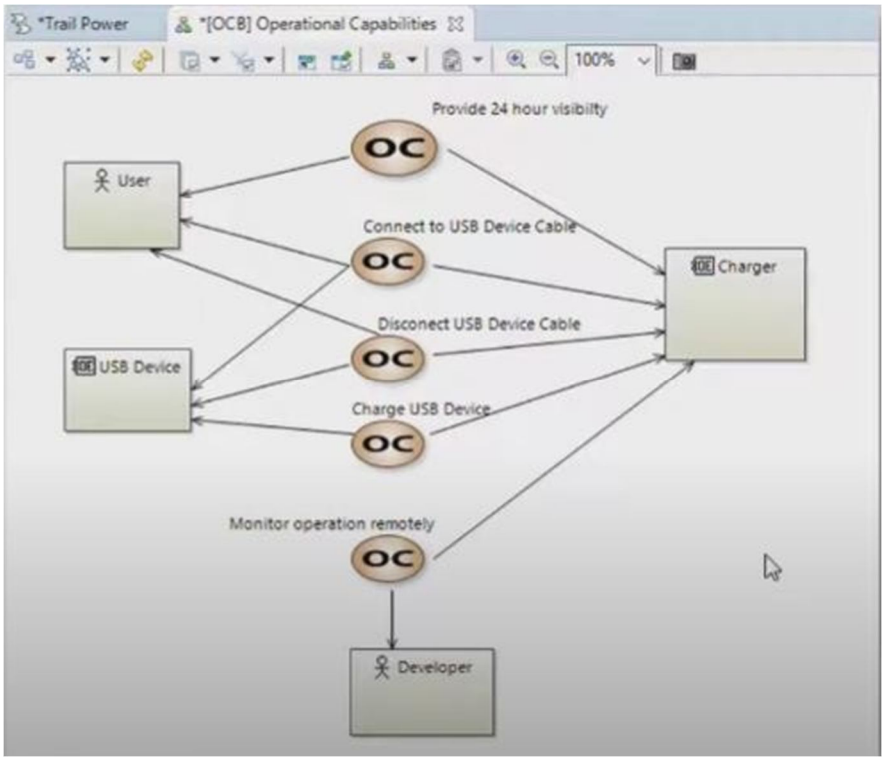
1) The main tool used for system engineering activities is Capella. Even though models serve for different goals, developing a reference such as a capella models helps to understand the mapping ofd the problem and the requirements and also modify the properties in the same model.

2) The primary highlight of such a model is that manual requirements can beeasily loaded into the capella model. Two Capella layers are mainly used for the avionics field: the Logical Architecture which gathers logical functions answering to the need, and the Physical Architecture representing the implementation including the physical nodes and physicalfunctions.

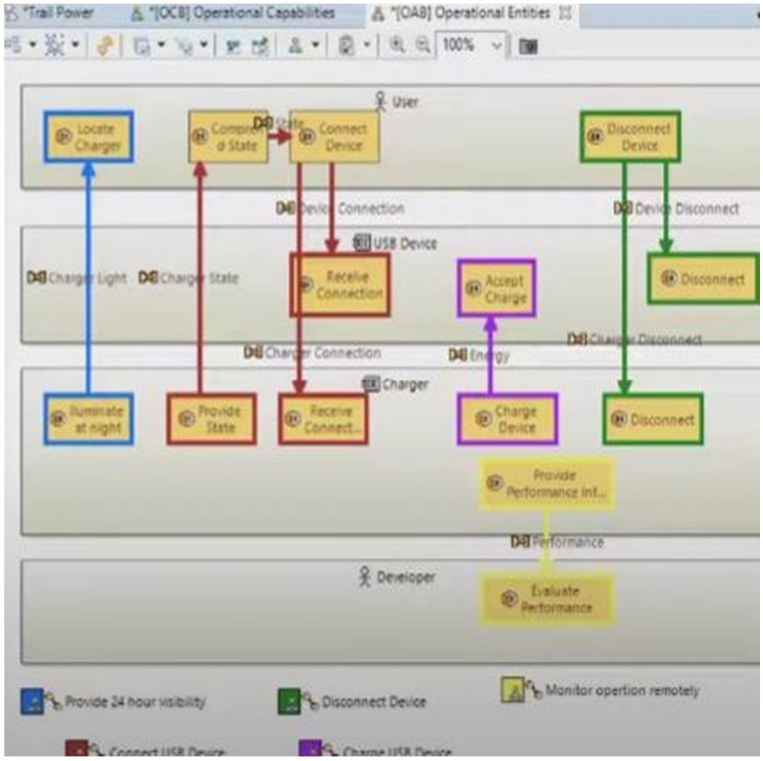
3) This same model can be later used for simulation purposes in a tool like Simulink.

B. Operational Analysis

Operational Analysis helps to understand the problem we wish to solve bymodelling the same exact problem.



Our first operational analysis model for the purpose of creating:



Operational capability blank diagram depicts the capability a user canexpect.
Operational Processes are then created by mapping a series of Activities toActors and Entities

Operational Activity Blank Diagram will deliver the capabilities withOperational Process.

#Operational Analysis

- Helps you understand the problem you wish to solve by modelling theproblem.
- Operational capability blank diagram depicts the capabilities the usersexpect.
- Operational activity blank diagram delivers the capabilities withoperational process.

Capella helps define and understand the problems with two types of models.

After defining the requirements to the operational analysis panel, we need a gateway to make sense of these requirements. We need to "load" these requirements into a sub space.

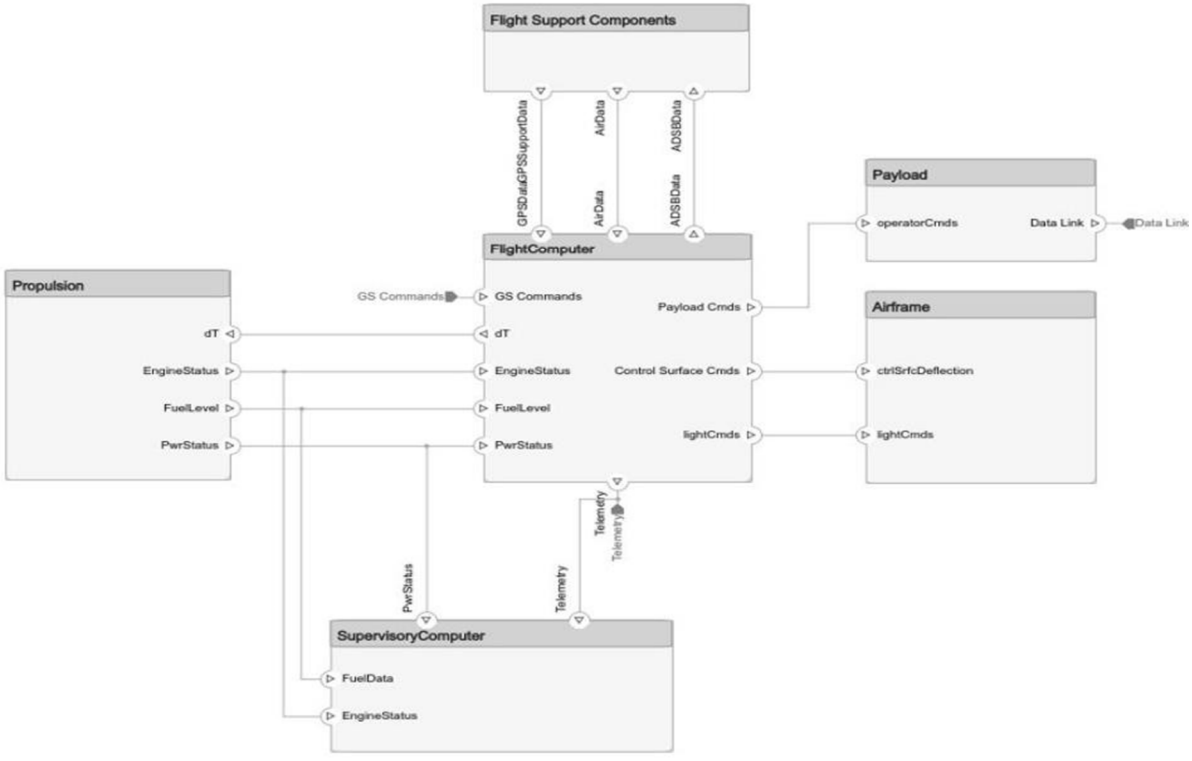
Lets call this space as a model. In one sense, we are basically mapping the requirements and their physics into the model.



Here, in the given figure we have mapped the requirements into our capella model.

For the note, we have only assigned the function of receive and transmission intoour avionics block, since we are only focusing on the communication devices. However more classes of different functions can be linked to the same block.

Having linked all the other blocks as well, the figure below shows our final functional block designed in capella.

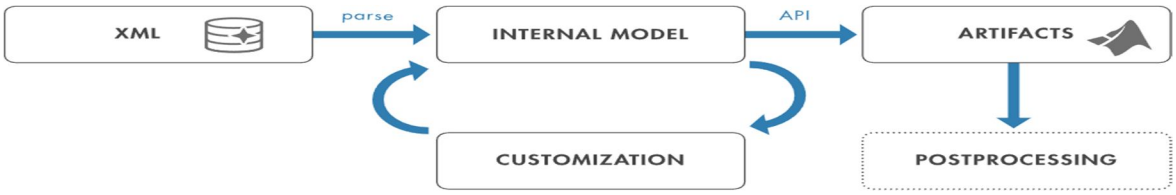


C. Aircraft System Architecture

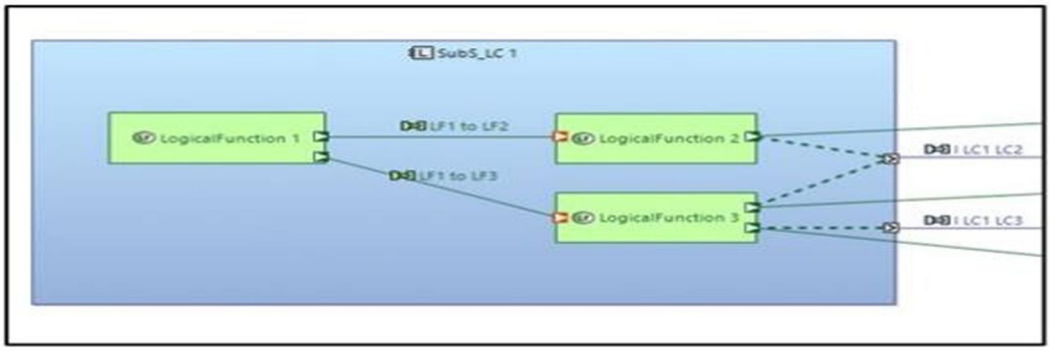
Now our next question is: how do we determine which information we need toimport from capella layers to the Simulink?
The following table gives a brief elaboration:

Capella	MathWorks
LogicalArchitecture	System Composer architecture
LogicalComponent	System Composer architecture, Simulink model
LogicalFunction	Simulink model, subsystem reference, subsystem
ComponentExchange	Root-level port with bus object
FunctionalExchange	Simulink signal or line
ExchangeItem	Simulink data type (Simulink bus, numeric data type)
Datatype	Numeric Simulink data type
Class	Simulink bus

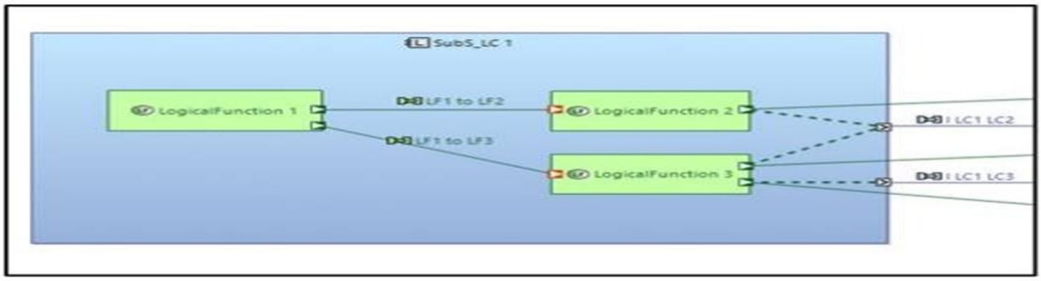
This table solely focuses on the logical layer. And logical layer is indeed our primary choice.
All the information is registered in a XML file which is then parsed into an internal model. This model is then converted into ARTIFACT, which is an entityused to post process the data into Simulink.



Of course, customization can also be made to this model if need be. (not strictly necessary)
We converted the logical functions into the subsystem which will act as a data storage point (a reference) for the Simulink.



D. First Logical Function Block



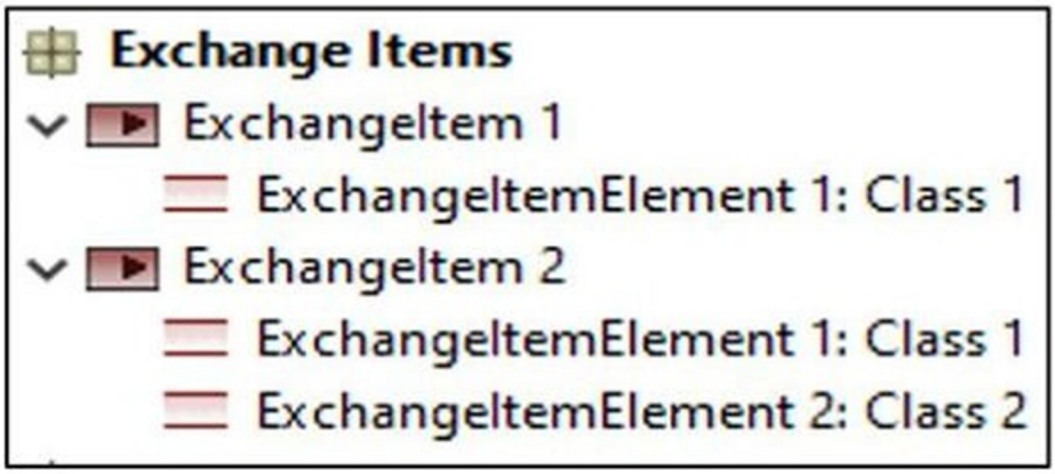
E. Simulink Subsystem

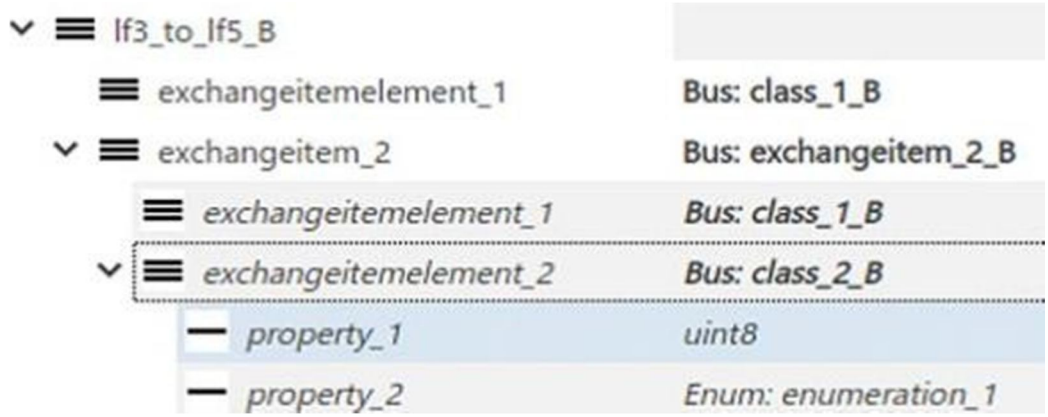
Problem – operational analysis and system analysis (what the system needs to accomplish for the user)

Logical Architecture depicts how the system will work to fulfil the expectations of the user. Logical Architecture Blank Diagram will depict the abstract components and function of the solution.

On the other hand, Physical Architecture depicts how the system will be developed and built.

End Product Breakdown Structure – Formalizes Component Requirements

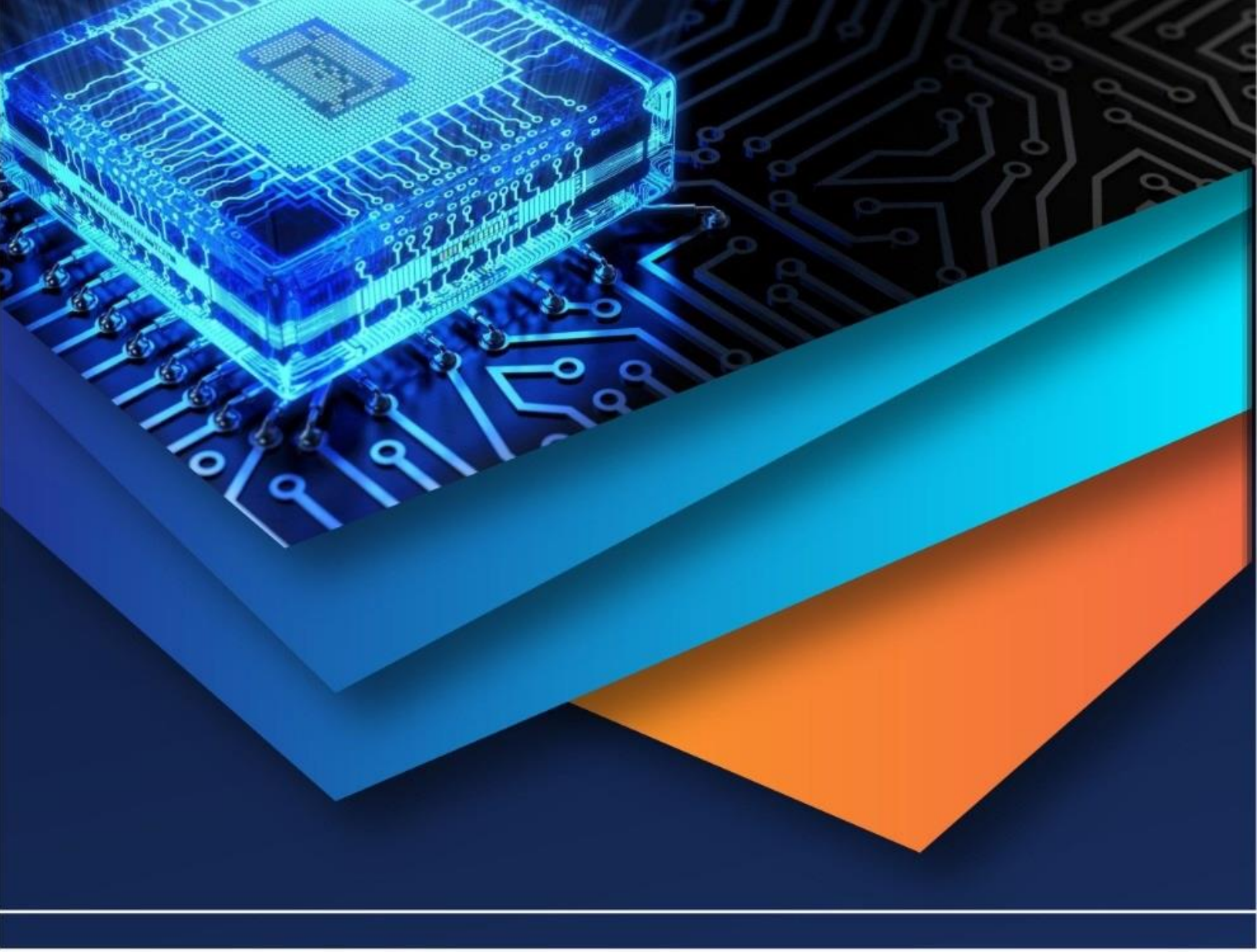




CONVERSION OF CAPELLA CLASSES INTO SIMULINK DATA

REFERENCES

[1] Barthélémy Attanasio, Délia Cellarier, Regis De Ferluc. MODEL-BASED SYSTEM ENGINEERING FOR AVIONICS PROCESSES- 2017
[2] D. Perillo, C. S. Malavenda. Adoption of model based solution for MBSE transition , MBSE-2020 workshop
[3] Régis De Ferluc, Marco Panunzio. Experience Report: History and State of the Practice of Model-Based Software Engineering in Thales Alenia Space in France, MBSE-2020 workshop
[4] P Roques. MBSE and ARCADIA method. 2016 European Congress on Embedded Time
[5] U Sukhatme, JTT Van, Cl Tan. Dual Parton Model. 1994 – Elsevier
[6] Y Chen. Simulation Technique with MATLAB. 2013
[7] LA Dessaint, K AI-Haddad. Simulation tool based on Simulink. 1999 – IEEE Transactions
[8] H Klee, R Allen.Simulation of Dynamic System. 2018 – taylorfrancis
[9] BS Blanchard. System Engineering Management. 2004
[10] K Forsberg, H Mooz. Relationship of System Engineering with project cycle. 1991.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)