From automation to autonomy – designing a complete ship control system

Colin John Field[1], CEng MIET

Working for Rolls-Royce plc
Email: colin.field@rolls-royce.com

Synopsis

This paper describes the way in which Systems Engineering has been used to map out and address the technical, operational and regulatory considerations necessary for autonomous platform management of Unmanned Surface Vehicles. Building on an approach originally developed for Unmanned Aerial Vehicles, Model-Based Systems Engineering has been used to derive the context and requirements for this high-level ship control system to ensure that it is properly structured, adaptable and re-useable. Mapping out use cases of the platform systems of a large, complex unmanned ship has allowed the functional requirements to be derived rigorously and therefore informs the selection of the most efficient architecture and interfaces ahead of software creation. This practical application of Systems Engineering has paved the way to the creation of robust, open-architecture control of platform systems which enables vessel autonomy in the Naval domain.

Keywords: Autonomy; USV; Machinery; Platform systems; Systems engineering; MBSE

1 Introduction

The operation of any ship requires many diverse platform systems, including electrical, propulsion, fuel and stability systems, to be effectively, efficiently and safely operated. As Global Navies discover applications for autonomous ships with greater speed, endurance and autonomous capability than today's commercially available Unmanned Surface Vessels (USVs), larger vessels with more complex platform systems are required. The increased expense and difficultly of on-shore personnel remotely controlling these complex platform systems, particularly during periods of restricted communication, means that higher levels of autonomy will soon be required in the operation and management of the power, propulsion and auxiliary systems.

One solution to this problem is to implement an on-board decision-making and control layer in the USV system architecture which will autonomously operate the platform systems in support of the mission, performing the role of the engineering department on a manned ship. Such a control system would communicate with the mission management system, sense the status and health of machinery on board, make decisions about how best to operate the systems and then provide control accordingly. Its aim is to provide the optimum balance economy, performance or life, according to the priorities of the mission. It would not be part of the Integrated Platform Management System (IPMS) but provides the judgement about how it should be operated, as illustrated in **Figure 1**.

Rolls-Royce Defence is therefore developing a prototype of a control system able to perform this platform management role which could be demonstrated at sea and become a high-integrity product to support future large Naval USVs. The control system is referred to in this paper as the Autonomous Vessel Equipment Controller (AVEC) and has been the focus of the author since the project launch in July 2017.

[1] **Author's Biography:**
Colin Field is a Systems Engineer in Rolls-Royce Defence, working on autonomy projects spanning both Commercial and Naval maritime domains. His specialisms are systems engineering and complex systems design for remote and autonomous vehicles, previously having worked for the Ministry of Defence (DE&S and Dstl) and Rolls-Royce Defence on a number of Unmanned Aircraft research and design programmes.

 http://doi.org/10.24868/issn.2515-818X.2018.008

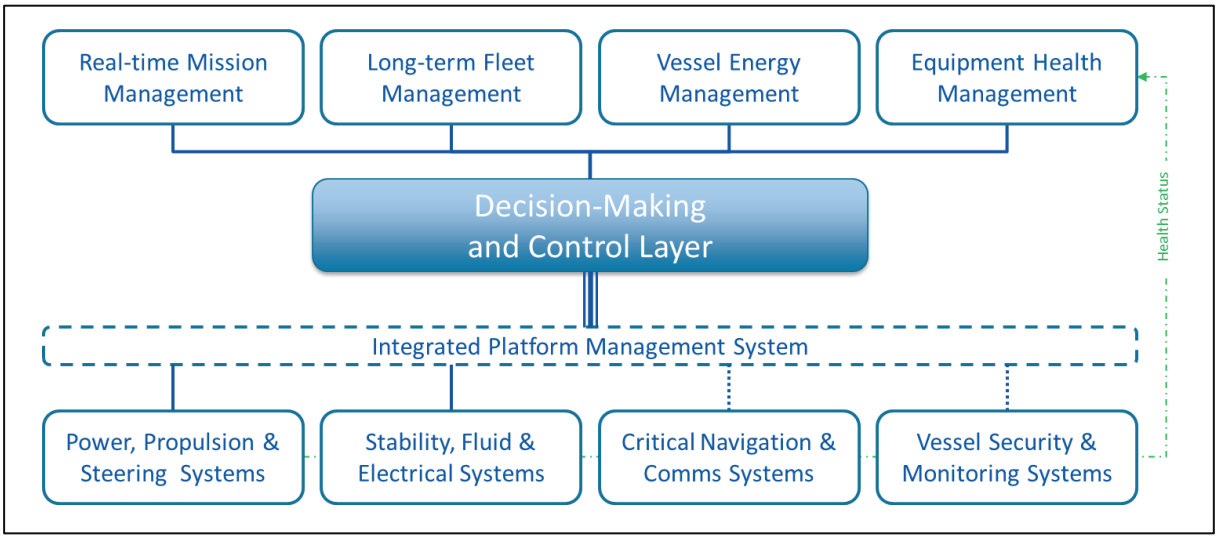

Figure 1: High-level context of AVEC decision-making layer

In common with many novel and conceptual technology development projects, this was launched in advance of having a rigid context and starting requirements set. Progress was therefore wholly reliant on being able to form a highly structured approach to this problem. This paper describes the Model-Based Systems Engineering (MBSE) approach that was crafted to address this problem and also describes the activities to scope out the context, requirements and architecture of this equipment controller from the launch of the project until June 2018.

2 Selection of Systems Engineering principles

Many of the Systems Engineering (SE) techniques were a natural fit to this problem because the tools and perspectives allow complex problem to be addressed from many different angles. Knowing that the product itself will be a system of software components which must work together in a co-ordinated manner, SE provides a framework to determine requirements, design the software architecture and its functionality using a systematic approach which enforces consistency and reducing re-work.

The approach to designing the AVEC was informed by the prior development of an Unmanned Aerial Vehicle (UAV) power management system developed by Rolls-Royce during ASTRAEA. This programme developed and then demonstrated the key technologies for autonomous control of propulsive, electrical and thermal power and the integration of system health management with these. It is also intended that some of the core system functionality of the AVEC will be derived from ASTRAEA power management system components.

Four challenges were evident at the start of the project which shaped the how Systems Engineering was applied:

- How can a robust requirements set be derived in the absence of a specific vessel application?
- How should this complex system design task be approached to prioritise functionality of the demonstrator product in these rapid timescales?
- How to maximise the re-use of ASTRAEA components in a new maritime domain?
- How could a ‘generic’ product be developed that can be readily adapted for both Naval and Commercial applications?

One key SE principle for the clean-sheet design of complex systems is working with the business stakeholders to define the boundary of the system of interest, treating the system as a ‘black box’ whose functions are known but whose internal mechanisms are (at this point) undefined. This allows stakeholders can agree on the purpose of the system without attempting to solve the technical problem.

 http://doi.org/10.24868/issn.2515-818X.2018.008

Since no industry-standard architecture for a generic USV exists which could define the boundary for the AVEC, a period of effort was planned to develop and validate a rich context by establishing the architecture of a generic USV, from which the controller functions and other requirements could be derived.

Once the context and the requirements of the system are defined, work can begin to identify the processes that it must perform internally to exhibit the desired behaviours externally. These processes depend on a structure of software sub-systems working together which, if properly designed, gives an architecture which is an enabler of an effective, efficient and safe system which is also easy to implement and upgrade.

Another key SE principle deemed vital for the AVEC project is to invest creative effort in determining what the purpose of the system should be (modelling the 'problem'), then equally investing creative effort in determining the best architectural and functional design to achieve that purpose (modelling the 'solution').

In both instances, this creative effort is structured by having periods of divergent then convergent thinking. In the divergent thinking phases, the concepts are generated from many different perspectives to create diversity – then in periods of convergent thinking, the concepts are compared then progressed, retired or combined based on their relative merits (Burge et al, 2011).

However, for clean-sheet designs, it is risky and inefficient to attempt to generate a complete model of the problem before starting any work on the solution; it is inevitable that many more problems will be revealed as the solution takes shape. Therefore the work of the team must be planned to include discrete periods of work on both problem and solution models, followed by review, which shape the direction and priorities of the next period of work.

As a software project in Rolls-Royce, the creation of the AVEC has followed an approach using the Agile Scrum framework, inseparable from this plan-do-check-act cycle which iterates every fortnight in this project. This framework is traditionally introduced when the software requirements and architecture have already been set and software features can be demonstrated as they are implemented. However, at this stage of the project, no code had yet been written; so an approach was taken to treat the problem and solution models as products themselves and demonstrate their features instead.

These three concepts: problem and solution models, plan-do-check-act and divergent-convergent thinking are illustrated as part of the Software Engineering V in **Figure 2**.

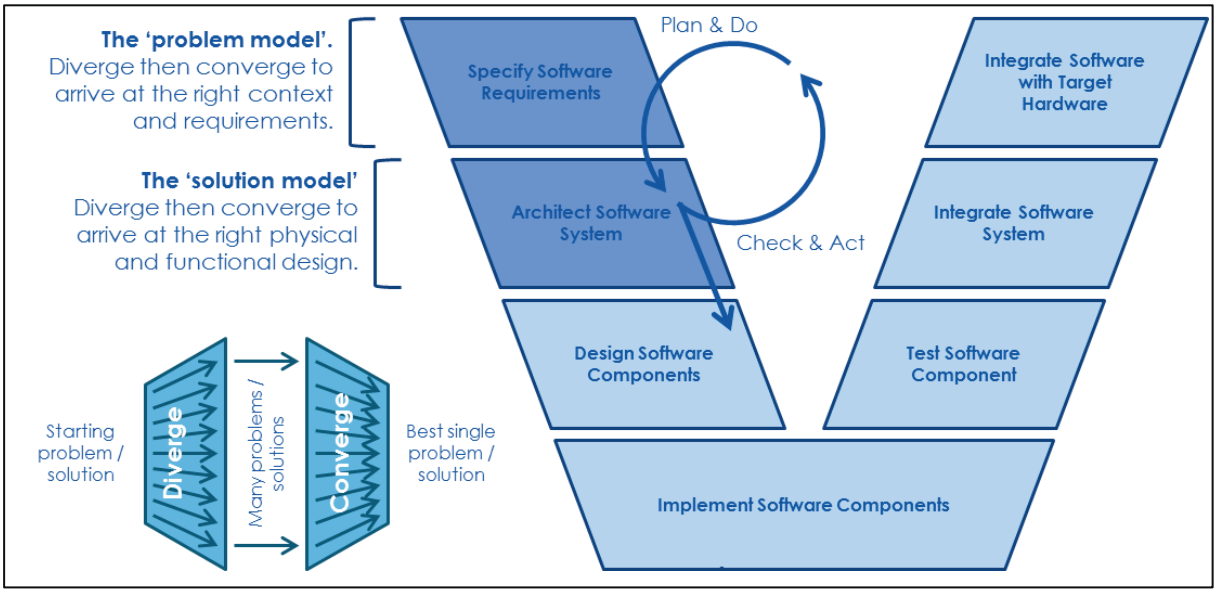

Figure 2: Problem model, solution model and plan-do-check-act cycle in the Software Engineering 'V'.

The parallel development of a problem model and a solution model is much easier if they are created and stored together in a computer. This is known as a Model-Based Systems Engineering (MBSE) approach, which stores model requirements, structure and behaviour in a database which maintains consistency as changes are made during the evolution of the models. The model elements are described using plain text and hyperlinks, and their relationships are illustrated using diagrams, making models easy to create and review by humans.

 http://doi.org/10.24868/issn.2515-818X.2018.008

The preliminary design of the AVEC therefore follows this approach:

1. Set the context by modelling the equipment and control architecture of a generic USV,
2. Elicit the basic functional requirements for the controller by analysing how the different parts of the generic USV control system interact to allow it complete missions,
3. Design a software architecture which enables those functions to be efficiently performed,
4. Allocate functions to parts of the software architecture and commence software creation.

The following sections describe how aspects of MBSE have been used to enact this approach, using diagrams from the problem and solution models as illustration.

3 Design process of Autonomous Vessel Equipment Controller

3.1 System context from a generic large USV

An 'autonomous' system is defined as having the following characteristics:

"The condition or quality of being self-governing. An unmanned system's own ability of sensing, perceiving, analyzing, communicating, planning, decision-making, and acting, to achieve its goals as assigned by its human operator(s) through designed human-robot interface." (NIST, 2014)

This definition of autonomous system identifies that there are three main functions of an autonomous system: sense, think, act. Using these three functions, the operation of an autonomous Naval ship was analysed and structured. This recognises three nested processes of sense, think, act at the battlefield system level, the USV system level and the vehicle system level.

For each system level, information is taken in about its objectives, the environment which it exists in and its present capabilities. It must then decide how best to achieve those objectives by creating a plan then enact that plan through its effectors which can either change the environment which it sits in (eg by moving through the sea) or change its capabilities (eg by reconfiguring the propulsion equipment). This is then reported back up to the system which it is part of, as shown in **Figure 3.**

At each of these levels, there is a part of the system which performs a 'think' function (which includes planning and decision-making). These were defined as the key actors in this autonomous architecture and correspond approximately with the Naval command structure as shown below:

- The Command, which is responsible for operating the USV system as a whole to achieve the objectives of the warfighting system (Captain),
- The Navigator, which is responsible for operating the manoeuvring systems of the USV to achieve the objectives of the Command (Officer Of the Watch / Navigator),
- The Operator, which is responsible for operating the mission systems of the USV to achieve the objectives of the Command (Principal Warfare Officer),
- The Engineer, which is responsible for operating the USV's propulsion, electrical power, cooling and other internal systems to achieve the objectives of the Navigator and the Operator (Marine Engineering Officer / Weapons Engineering Officer).

 http://doi.org/10.24868/issn.2515-818X.2018.008

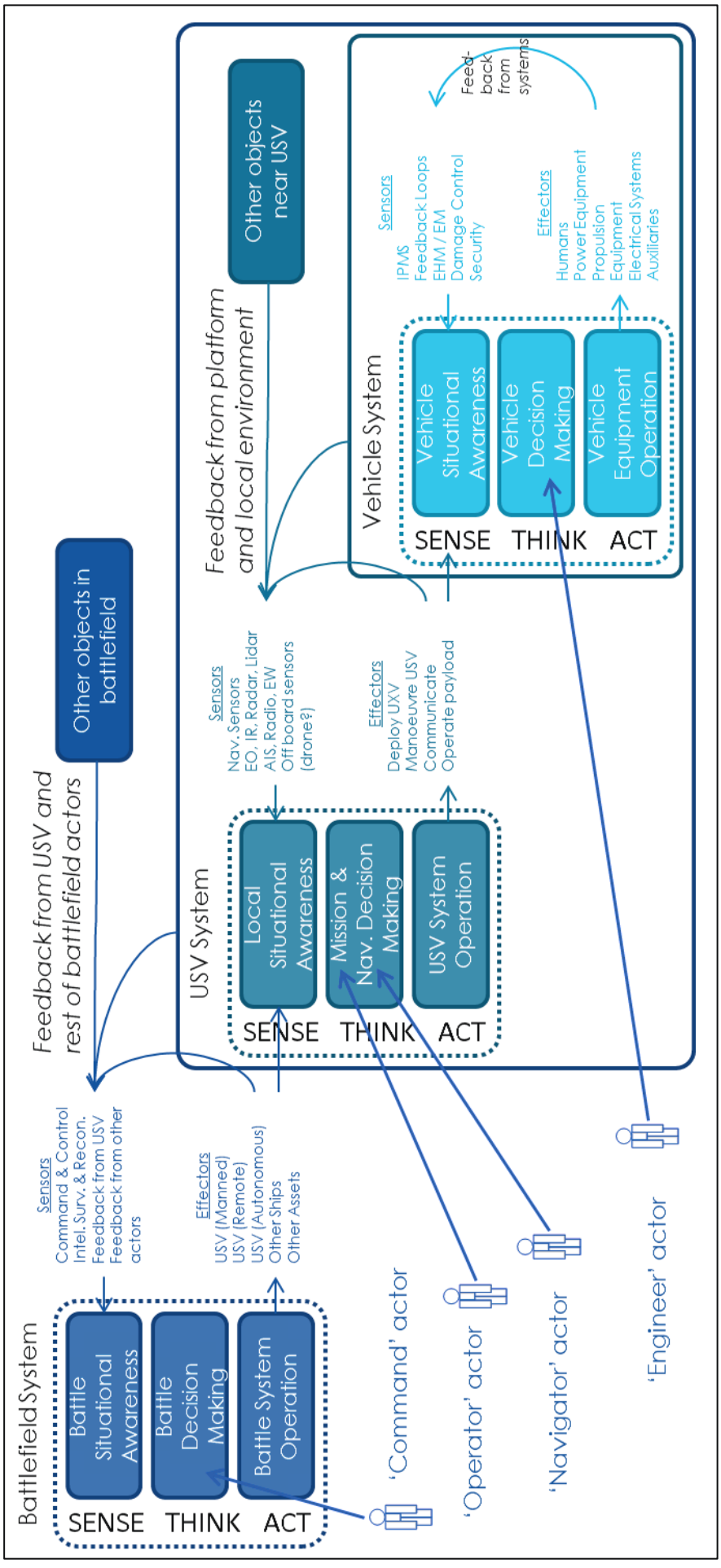

Figure 3: Application of sense-think-act to the Naval maritime architecture

 http://doi.org/10.24868/issn.2515-818X.2018.008

The AVEC is part of the Engineer actor, but may not fulfil the entire role – only functional analysis of how these actors interact can actually determine the Engineer functions and which of these can be performed by a digital control system.

To allow this deeper analysis, a team including Naval architects prepared a preliminary design of a generic single-role large USV (**Figure 4**), which:

- Is a conventional displacement ship,
- Is powered by engines and has common machinery,
- Goes to sea and comes back again having performed a single mission,
- Relies entirely on its own resources,
- For the purpose of this analysis, has no specific mission or in-built mission equipment,
- Has no need for physical human interaction whilst at sea,
- Allows for humans to prepare and maintain it.

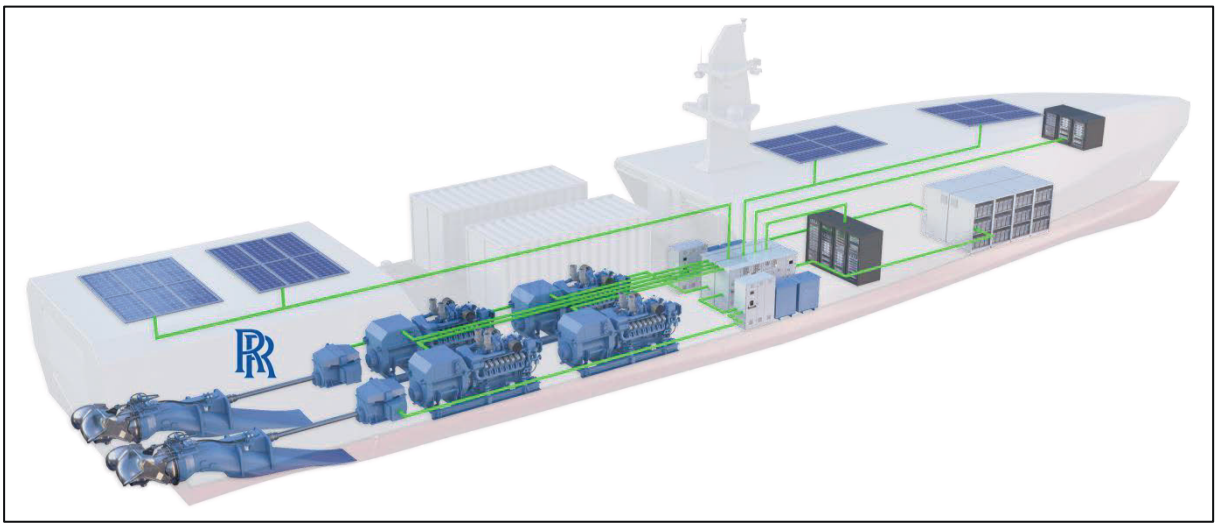

Figure 4: Conceptual design of a 53m generic large USV

A reference architecture of this large USV was put together which includes the necessary hull, mechanical and electrical systems of a USV as well as the physical embodiment of the control systems that are required, shown in **Figure 5**. This architecture begins to describe where the AVEC sits relative to other systems – which systems it is necessarily connected to for its primary functions (the mission management system, for instance) and which systems it is intentionally distinct from (such as the IPMS).

 http://doi.org/10.24868/issn.2515-818X.2018.008

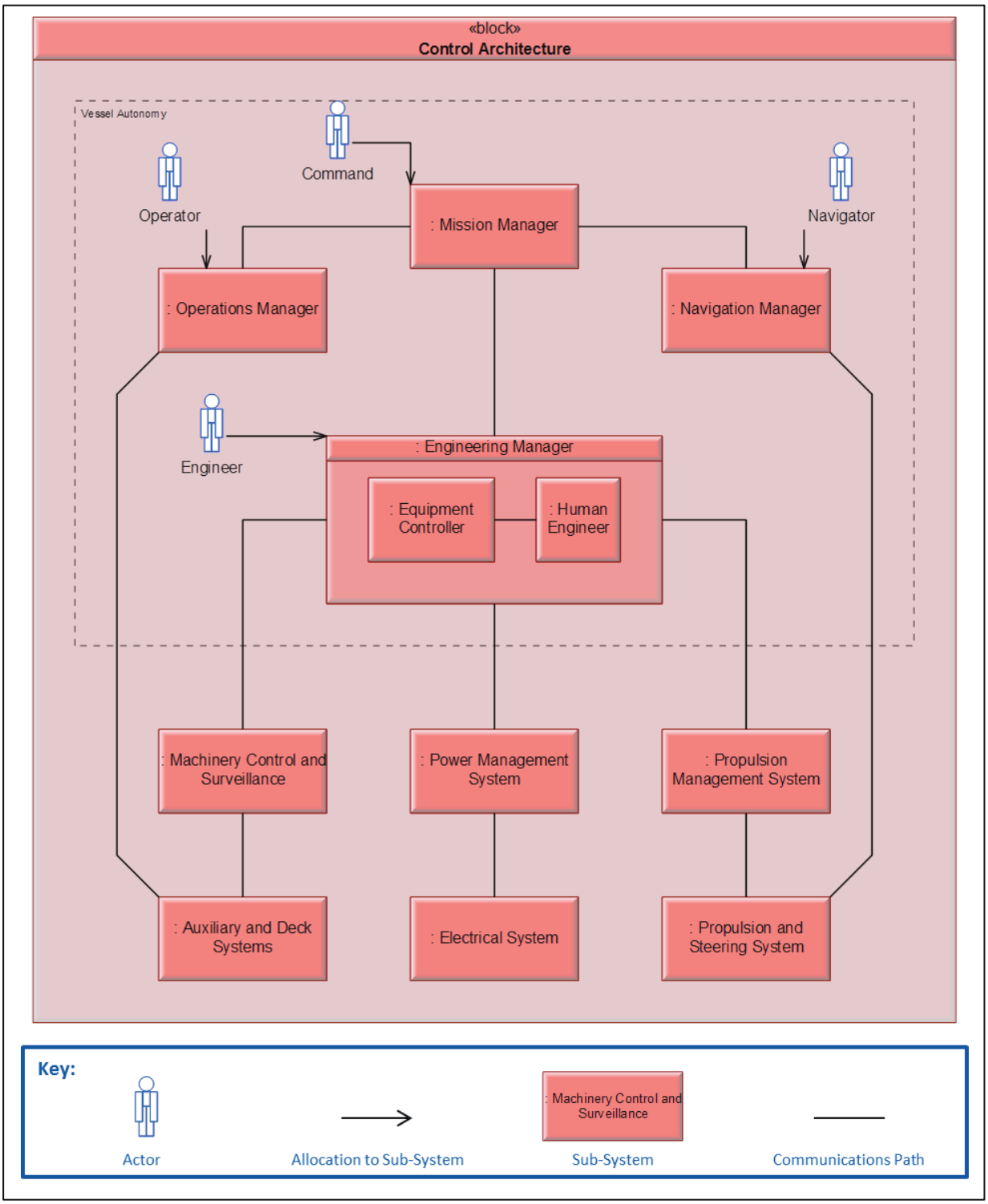

Figure 5: Simplified model of USV control architecture

Non-conventional systems such as Integrated Vehicle Health Monitoring which are necessary enablers of a large USV were included in the architecture model – allowing assumptions about these systems to be written down and even influence the design of these systems in future.

The USV has different phases of its operation, known as mission modes, and a mission plan is assembled by the Command which links these together to achieve the required objective. However, some of these mission modes must follow a logical sequence – for instance, a ship must prepare to go to sea before it can actually proceed to sea. A State Machine (**Figure 6**), which is a type of flow chart, was constructed which provides a model for the ship and its mission modes:

 http://doi.org/10.24868/issn.2515-818X.2018.008

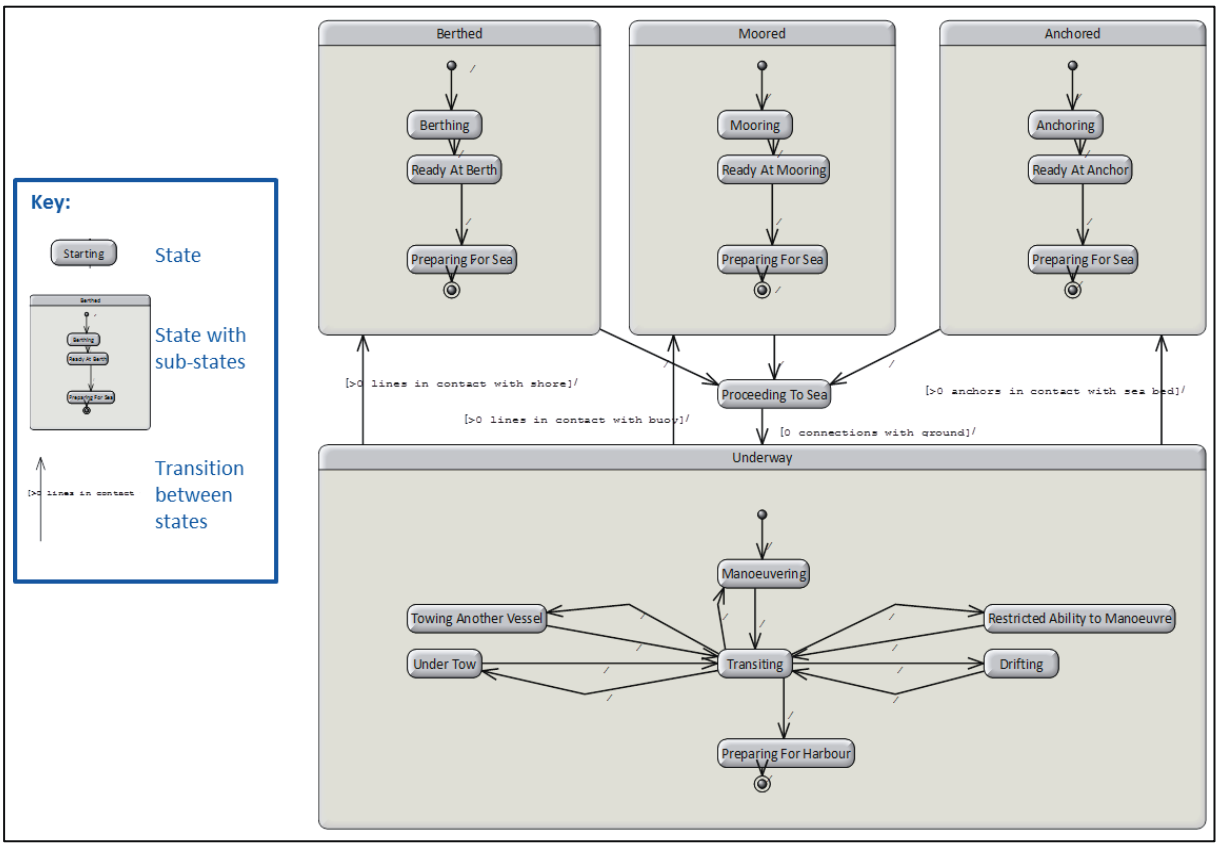

Figure 6: Modes of Operation of a Large USV

The context of the AVEC is more than simply the technical and operational considerations, although these are generally the most stimulating and straightforward requirements to derive for a project early in its lifecycle. The project's Business Requirements Document was written to identify the key principles which would lead to safety, regulation, maintenance, lifecycle and financial requirements. All these viewpoints must be considered too in order to come up with a complete requirements set.

By having a context for the USV, it is much easier to engage the experts in these viewpoints – for instance, safety – and decompose the non-functional requirements. The effect of using an Agile SE approach, which supports developing a functional prototype, steers the systems engineer away from attempting to capture all non-functional requirements too early in the lifecycle and instead rewards engagement with these experts when there is already a prototype product to show them.

3.2 System requirements development

Behind each mission mode of the USV is behaviour of the key actors, either sequential or parallel, that need to be carried out to support the evolution on towards the next mission mode. Functional analysis of these will elicit what the AVEC needs to do to support the USV in its mission and what interactions it needs to have with other systems in support. This is still part of the 'problem model', treating the system of interest as a black box but specifying what external behaviours are required.

Each mission mode in the model was described in detail using a use case diagram (**Figure 7**) with rich textual descriptions of what happens in each case, what prerequisites need to be met, what the starting state and the end state are and the roles of the four actors in each. These were constructed by a team of experienced seafarers including a Naval command systems engineer, a Naval marine engineer, a commercial captain and a commercial engineer who were inspired by Naval ship operations but not constrained by the established command structure.

Because the AVEC is the system of interest for this problem model, only the use cases which are assigned to the Engineer actor are analysed further. These use cases were then further described as functional models (**Figure**

 http://doi.org/10.24868/issn.2515-818X.2018.008

8), with information flowing from the sensing systems to the thinking systems, internal processes in the thinking systems, and orders being transmitted to the acting systems.

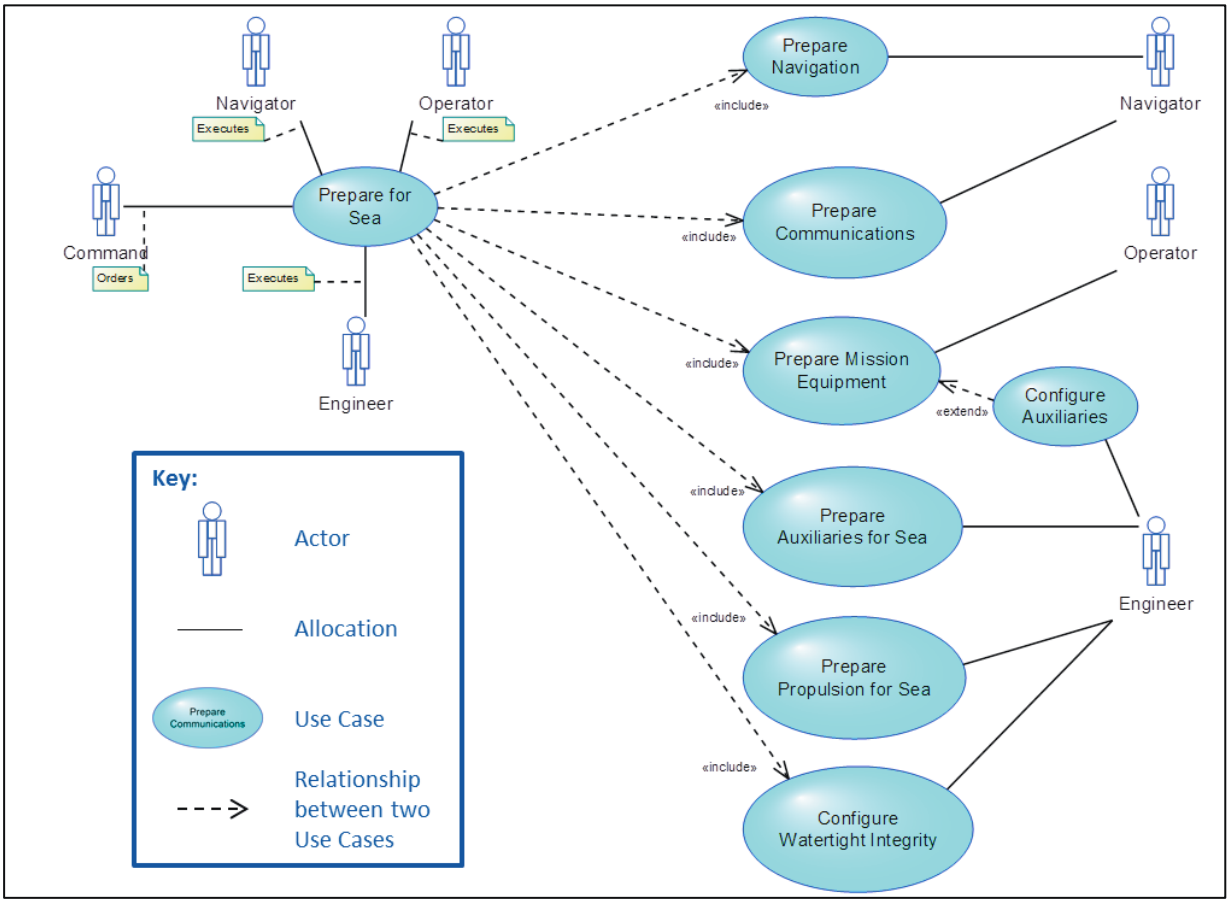

Figure 7: Each use case diagram describes a single mission mode

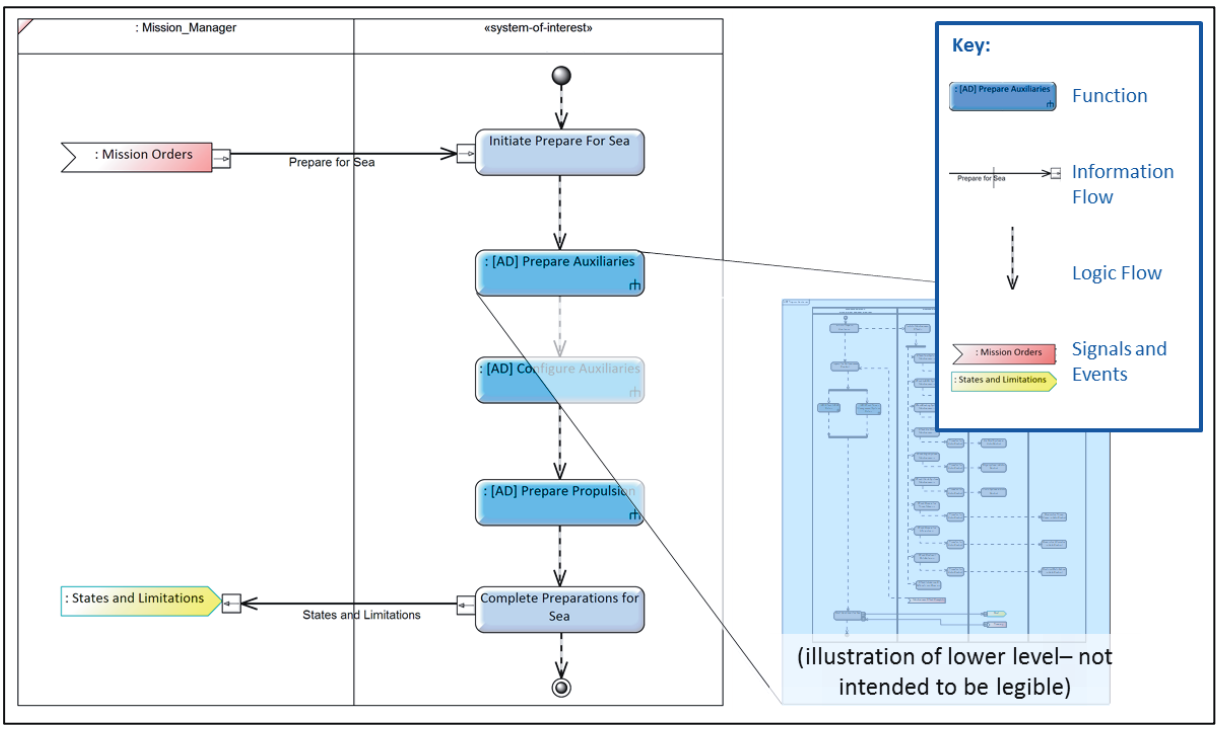

Figure 8: A functional model supports each use case (illustrative)

 http://doi.org/10.24868/issn.2515-818X.2018.008

Any functions in the model that could not be performed by a digital control system – such as checking that maintenance is complete, that manual valves are correctly operated, or actually carrying out any physical tasks – were assigned to other systems or actors as appropriate. In the absence of highly automated solutions to perform these physical tasks, there would still be a need for some sort of human chief engineer to be on board the ship during preparation for sea – this approach allows their job description to be created as well.

This functional approach leads to the generation of specific functions that the AVEC must perform, which are its functional requirements. Using this MBSE approach means that the requirements can be generated directly in the relevant diagram and interrogated all the way up through the model to see why they are required and on what assumptions they are built.

The non-functional system requirements stemming from outside the operational context of the large USV were decomposed directly using requirements diagrams. This allows the tracing and rationale to be captured explicitly and these diagrams are also easier to review and expand with the help of other experts.

All of the requirements of the AVEC are derived this way and extracted to a formal requirements table. The model enriches these requirements by showing their context, flow and dependencies, which would not be available in a standard requirements table.

3.3 System architecture and functional design

The Agile approach encourages the incremental development and testing of product 'features' which corresponds functions of the system. Although the architecture of the software product doesn't have any functions itself and could be de-prioritised by Agile Scrum, the SE approach here meant that the architecture was designed as soon as the main functions and non-functional requirements are known.

To select an appropriate architecture, a divergent-convergent approach was used again. Two team members created a total of five concepts between them, each using a different creative technique to try to encourage a diversity of solutions. Two concepts were created from based on knowledge of the problem model and the necessary processes that would need to take place inside the equipment controller. Another concept was created based on the ASTRAEA power management system architecture that would be donating some of its software functionality to the AVEC. Two more concepts were created by using an adapted N-squared approach.

N-squared analysis is a technique to identify the degree of interconnectivity within any system (Goldberg et al, 1994) and can show how the functions of a system depend on one another. When a group of functions is identified that are interdependent on each other and not very dependent on other functions, these can be logically clustered together as one sub-system, or software module, that is cohesive and only loosely coupled to the rest of the architecture.

Two conceptually different architectures were arrived at this way – one being a service-based architecture with many modules at the same level, the other being a hierarchical architecture which has nested sub-systems and is relatively simple at each level.

These five resulting concepts were scored against each other using a Pugh matrix (Goldberg et al, 1994) for their ability to meet the functional as well as non-functional requirements – such as ability to be configured and upgraded in future, ease of security, how open the architecture is and how quickly the software could be built from the readily-available functionality. As a team decision, a single architecture (**Figure 9)** was taking forward which combined some of the best features from three of the five concepts and scored the highest using this method.

 http://doi.org/10.24868/issn.2515-818X.2018.008

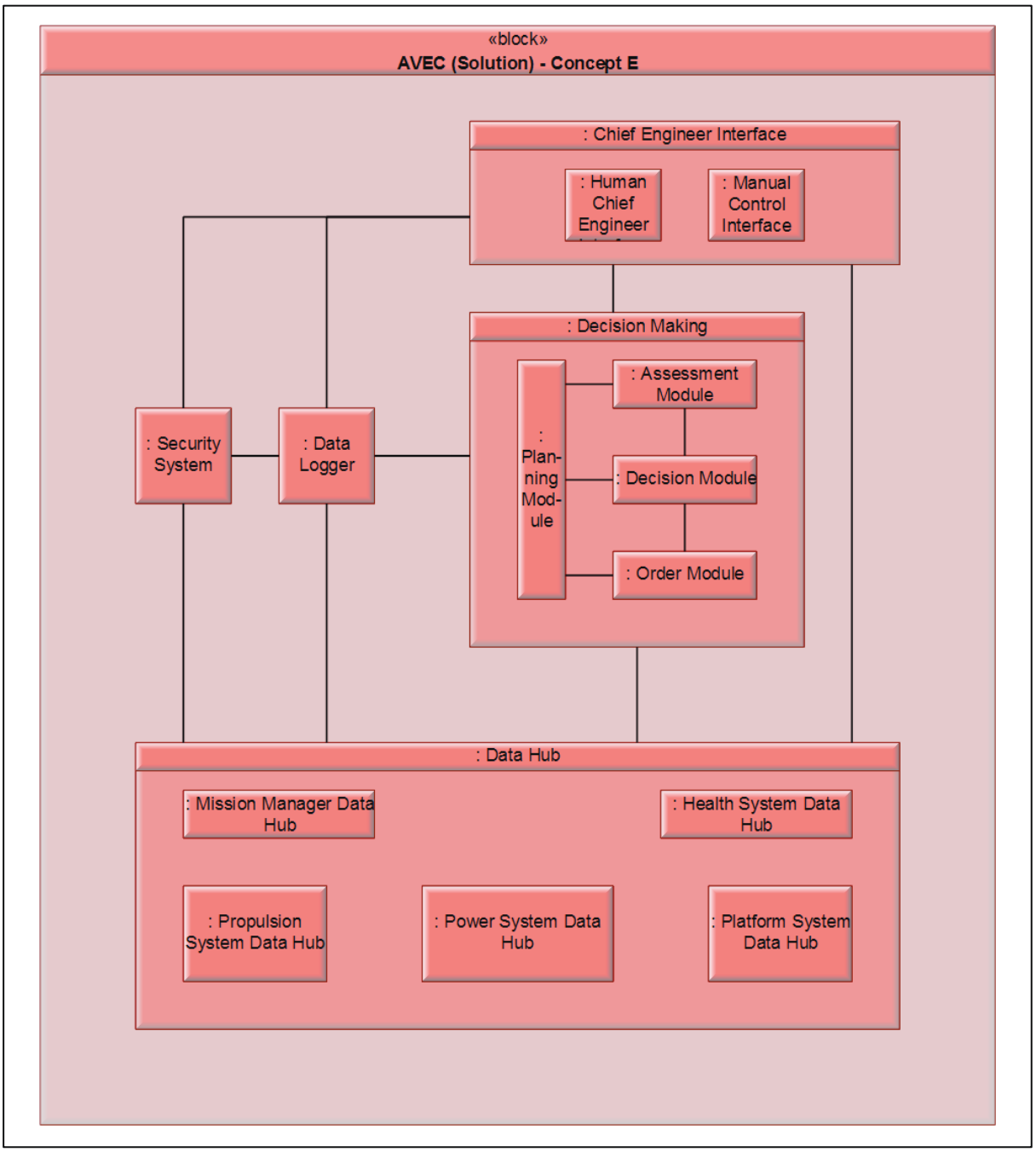

Figure 9: Simplified software architecture of the AVEC

At this stage, the software is no longer treated as a 'black box' and the design of the solution model begins. Using MBSE means that functionality could be assigned directly to the different software modules and the functional model could be built and linked, at each stage, to both the requirement that is being fulfilled and the verification method that will be used to test it.

The features of the problem model were prioritised for implementation and, for each one, the black box functional model was recreated as a white box functional model to show what sub-processes need to be performed by each software module and in what order to get the required effects. This included modelling the data structures that are needed to pass information around the software product and the logical decisions and gates that form the decision-making processes of the AVEC.

Much of this functionality is inspired by the ASTRAEA power management system, and by being explicit about the architecture and which parts correspond to ASTRAEA modules, the functions could be transplanted as easily as possible, maximising this re-use.

 http://doi.org/10.24868/issn.2515-818X.2018.008

Each software sub-system consists of further sub-systems and this fractal architecture means that there is no natural place for the system design to end and the component design to being. The approach taken for the AVEC is that the top two layers of architecture are modelled in MBSE and any additional complexity is designed freely by the software engineers to meet their emerging needs. The requirements for each software module are articulated to the software component designers through the solution model, both textually and diagrammatically in the functional models – as illustrated in **Figure 10**:

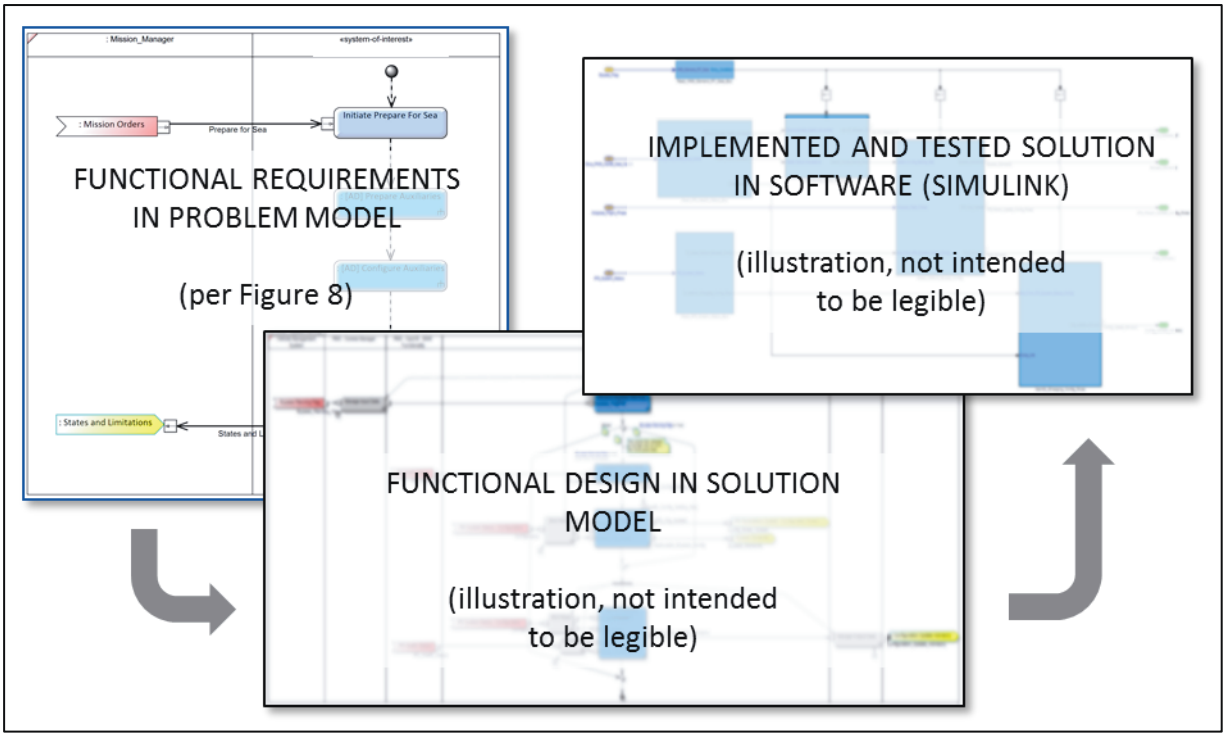

Figure 10: Transfer of information from problem model, to solution model, to software code (illustrative)

4 Conclusions

Following a structured Systems Engineering approach to this clean-sheet problem has yielded a budding software product that can now grow, feature by feature, to meet its requirements despite the initial ambiguity and lack of context. In that way, the SE approach used and promoted here has achieved its objectives and the team is now continuing to expand both the problem and solution model to cover the required functions in priority order.

At present, the code for the AVEC features is being written and tested , meanwhile real-time hardware is being selected. The team has defined interfaces with IPMS vendors and is implementing the concepts for communication between the mission manager and the AVEC. Through these discussions with the partners, the approach to creating open interface standards is being defined, paving the way for flexible and scalable autonomous system architectures in the future. None of this would have been possible without the rapid yet structured development of the software architecture and functionality that this MBSE approach has facilitated.

There have been challenges to set up the infrastructure to support this approach, particularly the networked computer systems needed to host an MBSE environment, but the team has broken new ground and found solutions to problems that the rest of the business can benefit from. Projects such as these help the company respond to emerging market demands in an agile and robust manner so that, in the case of future large USVs, Rolls-Royce is able to offer both product- and system-level solutions to meet the needs of customers.

 http://doi.org/10.24868/issn.2515-818X.2018.008

5 Acknowledgements

The author is grateful for guidance gratefully received from the corporate specialists in Systems Engineering, the team which developed the ASTRAEA Power Management System, and those working on autonomous control system technologies in Rolls-Royce Defence and beyond.

In particular, the author would like to thank Paul Ellison, Naval Combat Systems Engineer, for his extensive contributions to the problem model which progress on the control system has been entirely dependent on.

6 References and Bibliography

Autonomy Levels for Unmanned Systems (ALFUS) Framework, Volume I: Terminology, Version 1.1, NIST Special Publication 1011, September 2004.

Bridging the Gap Between Model-Based Systems Engineering Methodologies and Their Effecitve practice – A Case Study on Nuclear Power Plans Systems Engineering, J. Navas et al, INSIGHT Volume 21 Issue 1, March 2018

NASA Systems Engineering Handbook, Rev 2, NASA SP-2016-6105, November 2017

SysML for Systems Engineering, 2nd Edition: A Model-Based Approach, IET, J. Holt & S. Perry, 2013

System Engineering "Toolbox" for Design-Oriented Engineers, Issue 1, NASA Reference Publication 1358, B.E. Goldberg et al, December 1994

Systems Engineering Handbook, 4th Edition, INCOSE-TP-32003-002-04, D.D. Walden et al, 2015

Systems Engineering Short Course, Version 2.5, Burge Hughes Walsh Ltd, 2012

7 Glossary of terms

ASTRAEA	Autonomous Systems Technology-Related Airborne Evaluation and Assessment
AVEC	Autonomous Vessel Equipment Controller
IPMS	Integrated Platform Management System
MBSE	Model-Based Systems Engineering
SE	Systems Engineering
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicle

 http://doi.org/10.24868/issn.2515-818X.2018.008