Investigating the Flexibility of the MBSE Approach to the Biomass Mission

Joe Gregory, Lucy Berthoud, Theo Tryfonas, Antonio Prezzavento, and Ludovic Faure

***Abstract*—Model-based systems engineering (MBSE) represents a move away from the traditional approach of document-based systems engineering (DBSE), and is used to promote consistency, communication, clarity, and maintainability within systems engineering projects. MBSE offers approaches that can address issues associated with cost, complexity, and safety. One way that this can be achieved is by performing early functional validation of the high-level spacecraft functional avionics system. The use case discussed in this article focusses on the Biomass model, a systems modeling language-based representation of the Biomass Earth-observation mission. The MBSE approach is used to calculate the required size of the data handling unit onboard the Biomass spacecraft. The functional response of the system in terms of the onboard memory usage throughout the mission is simulated. Traditionally, this level of analysis would not be available at this early stage. The approach aims to replace ad hoc, spreadsheet-based calculations with a formal representation of the system that can be executed, interrogated and quantified. The flexibility of this MBSE approach is demonstrated by applying changes to the Biomass project and assessing the time required to implement these changes in the Biomass model and propagate them through to the results of the simulation. The changes have been made independently of each other and include: changes to the logical architecture, changes to the functional definition, changes to the mission profile, and changes to the requirements. Potential areas for improvement regarding the structure of the Biomass model are highlighted and discussed.**

***Index Terms*—Biomass, functional avionics, model-based systems engineering (MBSE), modeling, systems engineering.**

I. Introduction

THERE is increasing interest in model-based systems engineering (MBSE) over the traditional approach to systems engineering, document-based systems engineering (DBSE) [1], [2], whereby project and design information is stored in documents and must be manually maintained and transferred between domains [3], [4]. The traditional DBSE approach is labor-intensive and consists mostly of manual analysis, review, and inspection [5].

A core question that MBSE seeks to answer is how to integrate engineering models across tools and domain boundaries [6]. MBSE is the formalized application of modeling to support system requirements, design, analysis, optimization, verification, and validation [7]–[9]. By using interconnected models to store, represent, and relate this information and data, projects can expect improvements in consistency, communication, clarity, visibility, maintainability, etc.—thus addressing the growing issues associated with cost, complexity, and safety [10], [11].

Spacecraft represent an ideal candidate for the application of MBSE as they are complex systems with potential applications that are often limited by the high development costs they can incur [12], [13].

This article extends previous work [14] concerning the application of MBSE techniques to perform early validation on the payload data handling unit (PDHU) of Biomass, an Earth-observation spacecraft due to launch around 2021 [15], [16]. The process of validation confirms that the system satisfies the stakeholders' needs by providing objective evidence [17], [18]. Biomass is being designed by Airbus as the seventh Earth Explorer Mission for the European Space Agency (ESA). The work has been conducted during "Phase B" of the Biomass mission design process. The aim of Phase B is to establish a functionally complete preliminary design solution [19]. The focus of the work is on the spacecraft's high-level functional avionics system. MBSE is used to describe the high-level simulation of the design, thus contributing to the validation of the system behavior [20]. Specifically, the system response to a mission profile has been simulated to validate that the spacecraft has adequate memory for the science-collection phases of the mission.

The outcome of the work presented in [14] is the Biomass model—a mission-specific example of an MBSE approach to defining and analyzing the functional avionics of a Phase B spacecraft. It is based on an existing Airbus model template [21]. The aim of this article, therefore, is to investigate the flexibility of this MBSE approach. This is done by applying possible changes to the Biomass project and assessing the time and effort required to implement these changes in the Biomass model and propagate them through to the results of the analysis. The changes have been made independently of each other and include: changes to the logical architecture,

Manuscript received April 12, 2019; revised July 31, 2019; accepted November 23, 2019. Date of publication February 7, 2020; date of current version October 15, 2021. This work was supported by the Engineering and Physical Sciences Research Council Industrial Cooperative Awards in Science & Technology cosponsored by Airbus under Grant 16000151. This article was recommended by Associate Editor E. Tunstel. *(Corresponding author: Joe Gregory.)*

Joe Gregory and Lucy Berthoud are with the Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, U.K. (e-mail: joe.gregory@bristol.ac.uk; lucy.berthoud@bristol.ac.uk).

Theo Tryfonas is with the Department of Civil Engineering, University of Bristol, Bristol BS8 1TR, U.K. (e-mail: theo.tryfonas@bristol.ac.uk).

Antonio Prezzavento and Ludovic Faure are with Department of Functional Avionics, Airbus Defence and Space, Stevenage SG1 2AS, U.K.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2019.2958757.

Digital Object Identifier 10.1109/TSMC.2019.2958757

changes to the functional definition, changes to the mission profile, and changes to the requirements.

This article is necessary because changes are a vital part of the engineering of successful systems, and will realistically be part of any project lifecycle that looks to develop a complex system [22], [23]. Engineering change management poses a real challenge within systems engineering and can have a significant impact on the final cost of the product [24]. The ability of the Biomass model and the accompanying MBSE approach to effectively handle and implement these changes, therefore, is a crucial part of the assessment of the approach [25]. It is hoped that the demonstration of this MBSE approach will contribute to the progressive adoption of MBSE in industry by directly countering Motamedian's finding that the "lack of perceived value of MBSE" is one of the main inhibitors of MBSE adoption [26]–[28].

The research questions of the work presented in this article are the following.

1) How flexible is the Biomass model in terms of changes to the system specification (functional definition and logical architecture)?
2) How flexible is the Biomass model in terms of changes to the system requirements and the mission specification?

The novelty of this article is that it presents the definition of an MBSE approach and model structure that can be used to define the high-level functional avionics of a spacecraft—in terms of both structure and behavior—and simulate the response of this system to a given mission. The approach allows this high-level simulation to be achieved earlier in the system lifecycle than usual [19], [27]. The flexibility of the MBSE approach is also investigated and discussed.

Sections II and III provide the necessary background information on MBSE and the systems modeling language (SysML) used throughout this article. Section IV introduces the Biomass mission. Section V provides an overview of the Biomass model. The methodology adopted to assess the flexibility of the Biomass model is presented in Section VI, with the results of the investigation presented in Section VII. Section VIII discusses the outcomes of this approach and identifies potential areas for the future work discussed in Section IX. The investigation is concluded in Section X.

II. Model-Based Systems Engineering

Systems are characterized by complexity [29], and cannot be understood by reducing the system to the sum of its parts; the system is necessarily defined by the interactions between its components and the emergent behavior produced as a result [30]. This level of complexity requires a suitable approach to view the system as a whole and understand this behavior.

Systems engineering has evolved as a practice to help cope with the complexity inherent in systems, to help avoid omissions and invalid assumptions, manage real-world changing issues and produce the most efficient, economic, and robust solution [31].

MBSE is an approach to systems engineering that seeks to achieve the goals of systems engineering through the formal application of models. It has been summarized as follows [32].

MBSE is an approach to realizing successful systems that are driven by a model that comprises a coherent and consistent set of representations that reflect multiple viewpoints of the system.

A viewpoint describes the point of view of a set of stakeholders by framing the concerns of the stakeholders along with the method for producing a view that addresses those concerns [33].

While there have been significant efforts to develop the MBSE approach to the simulation and analysis of spacecraft [13], [21], [34], the focus has remained on the description of system designs, and overlooks the importance of using this information, present in the model, to automatically analyze and validate the system itself [35], [36]. MBSE makes this possible in the early phases of the design [27].

MBSE is often discussed in terms of the three MBSE pillars: 1) language; 2) tool; and 3) methodology [37]. The language used will be discussed in the next section. The tool is the software used to produce the model, which consists of model elements, tables, diagrams, etc., representing the appropriate modeling language. Cameo Systems Modeler (18.5), produced by No Magic, has been used as the central MBSE tool. An Airbus-specific methodology has been followed to produce the Biomass model [14], [21].

III. Systems Modeling Language

There are a number of languages that are available to the systems engineer looking to practice MBSE [38]. Of those available, the object management group's (OMG) SysML has become the de-facto modeling language for systems engineering [39]. SysML has been developed as an extension to the OMG unified modeling language (UML) in a joint effort by OMG and INCOSE.

Specializing in the definition and development of software, UML has been a popular choice of object-oriented software modeling language since the early 2000s. But while containing some common themes (such as ownership or activity flows), the difference between software engineering and systems engineering means that UML lacks the means to fully capture systems engineering concerns [39].

SysML [33] does have limitations. For example, some semantics have to be modified or used out of their originally intended context [40], and the notation does not use formal activity specifications [41]. It is, however, the graphical modeling language most suited to the description of the MBSE activities involved throughout this Phase B application—it is flexible and expansive, covering all aspects of systems engineering, and expressive, with numerous texts available for guidance [37], [42]. SysML also allows the integration of other, domain-specific languages if this is deemed necessary.

SysML consists of nine types of diagram [43] that together can describe the structure, behavior, requirements, and parametrics of a system.

IV. Biomass Mission

The Biomass mission is an Earth-observation mission due to be launched around 2021. The primary mission objectives are

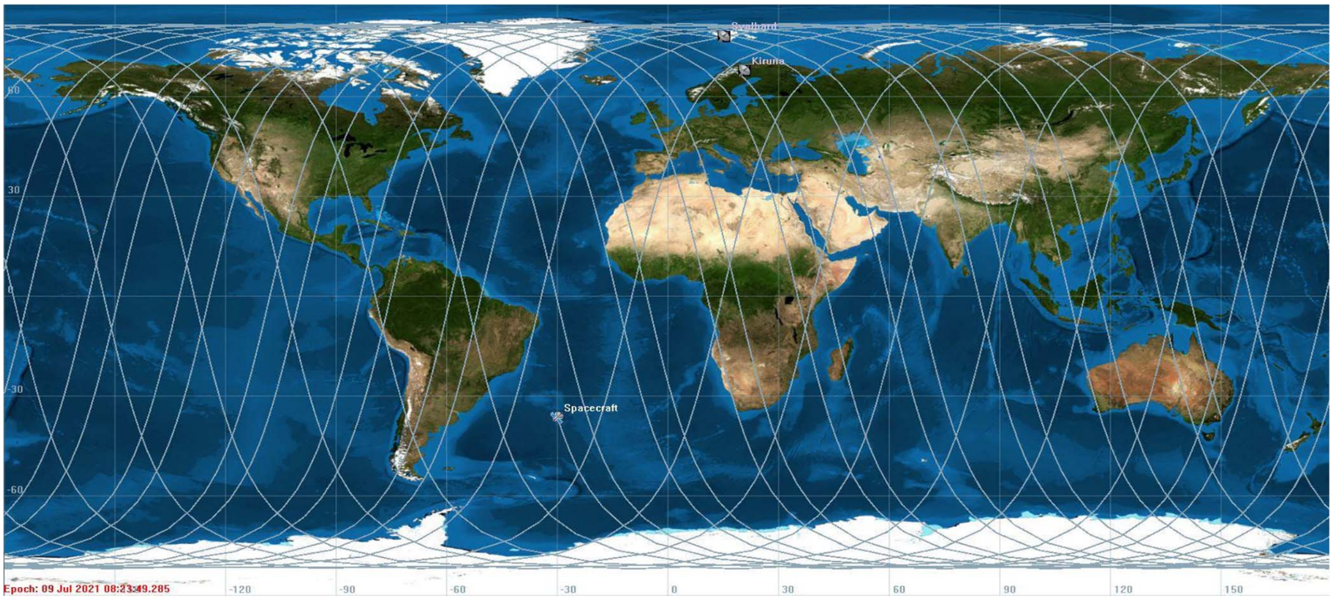

Fig. 1. Biomass ground track over 1.5 days.

to determine the distribution of above-ground biomass in the world forests and to measure annual changes in this stock over the period of the mission [15], [16]. To achieve these objectives, a P-band (435 MHz) Synthetic Aperture Radar (SAR) has been selected as the payload.

The Biomass space segment consists of a single low Earth orbit spacecraft (Biomass) carrying the SAR instrument. The mission will provide global coverage twice per year over the five-year mission. To achieve this, the spacecraft will be put into a sun-synchronous orbit during the nominal operations phase. A near-repeating ground track with a period of three days has been selected—a combination of controlled westward drift (of a small percentage of the instrument swath per orbit), rolling manoeuvres to position the SAR instrument and orbit drift phases are used to ensure that the global coverage requirement can be achieved [16]. The ground track of the spacecraft over a period of 1.5 days is displayed in Fig. 1 [44].

The SAR instrument itself can be in one of three modes: 1) *Off* (prelaunch); 2) *Ready* (on but dormant); and 3) *Measurement* (recording science data). Broadly speaking, the SAR instrument will be in measurement mode when over land and ready mode when over the ocean.

Biomass contains a PDHU with a proposed mass memory of size 960 Gb. All science and housekeeping data recorded over the full duration of the mission will be stored in the PDHU. The PDHU mass memory is divided into three directories, one for the housekeeping data and two for the science data. The housekeeping data will be recorded throughout the entire mission at a consistent rate of 7.5 kb/s and will be stored in Directory 1. The SAR data rate is much more substantial at 132.4 Mb/s, with an equivalent data rate of 66.2 Mb/s being stored in each of the two science directories when the SAR is recording: 1) horizontally polarized data in Directory 2 and 2) vertically polarized data in Directory 3.

During the nominal mission phases, the chosen orbit will allow Biomass to pass over the ESA ground station in Svalbard approximately once per orbit. During each pass, Biomass will downlink data from the spacecraft to the ground using its X-band downlink antenna (XDA). Data from the three directories are downlinked sequentially. The achievable data rate is assumed to be 467 Mb/s. The XDA can be in one of three modes: 1) *Off*; 2) *Ready* (on but dormant); and 3) *Downlink* (downlinking data to ground station).

The PDHU can be in one of five modes: 1) *Off*; 2) *Ready* (on but dormant); 3) *Write* (when the SAR is recording); 4) *Delete* (following a successful downlink of data); and 5) *WriteDelete* (whereby the SAR is recording and the XDA downlinking simultaneously).

The mission analysis has been done using systems tool kit (STK) [45]. This orbit modeling software has been used to produce a Biomass mission profile detailing the exact times (± 2 s) that the Biomass spacecraft will/will not be over land, and will/will not be over the ground station, for the full duration of the nominal mission phase.

V. Biomass Model

The production of the Biomass model, the reasons why it is necessary and the decisions regarding its structure to enable model execution have been discussed in detail in previous work by the authors [14]. In this article, the flexibility of the Biomass model to changes is assessed. In this section, specific aspects of the model relevant to this investigation are defined. For the purposes of this use case, the system under consideration is limited solely to the SAR-PDHU-XDA subsystem on board the Biomass spacecraft.

The overall structure of the Biomass model is presented in Fig. 2. The Biomass model centers on an SysML-based model (produced in Cameo Systems Modeler) which takes inputs from and feeds outputs to other tools: MATLAB, Microsoft Excel, and STK (via Microsoft Excel). The SysML-based model produced here can be further decomposed

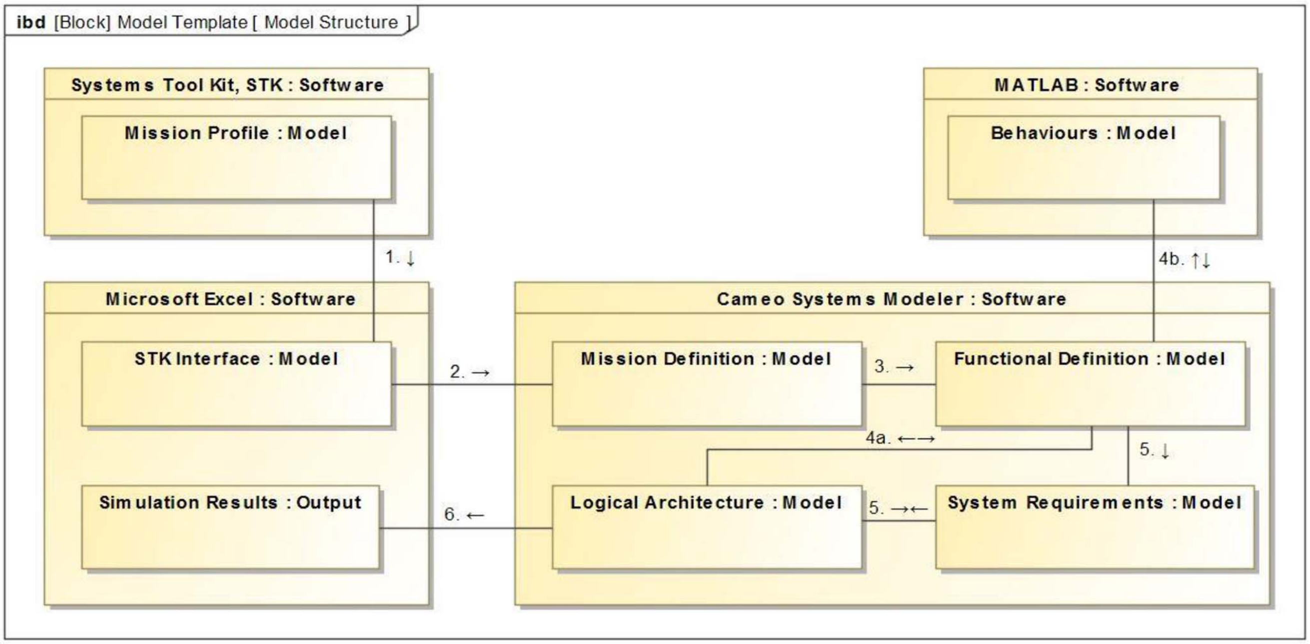

Fig. 2. Internal block diagram—biomass model structure.

TABLE I
FIRST 2.5 H OF NOMINAL MISSION PROFILE—PRODUCED BY ORBIT MODELING TOOL STK

ExcelMissionTime	ExcelDuration	ExcelPhase	ExcelSignal
0	412	Over Land	2
412	42	Over Ocean	1
454	126	Over Land	2
580	538	Over Ocean	1
1118	54	Over Land	2
1172	92	Over Ocean	1
1264	566	Over Land	2
1830	68	Over Both	4
1898	602	Over G. Station	3
2500	1034	Over Ocean	1
3534	570	Over Land	2
4104	1834	Over Ocean	1
5938	108	Over Land	2
6046	384	Over Ocean	1
6430	140	Over Land	2
6570	276	Over Ocean	1
6846	232	Over Land	2
7078	174	Over Ocean	1
7252	396	Over Land	2
7648	78	Over Both	4
7726	584	Over G. Station	3
8310	218	Over Ocean	1
8528	46	Over Land	2
8574	426	Over Ocean	1

into four separate models: 1) *System Requirements*; 2) *Mission Definition*; 3) *Functional Definition*; and 4) *Logical Architecture*. Together, the *Functional Definition* model (described using structural block definition and internal block diagrams) and the *Logical Architecture* model (described using behavioral activity and state machine diagrams) provide a complete description of the system under design.

The connectors in Fig. 2 represent the direction and order in which the information flows through the Biomass model during a simulation. A description of each of these connectors follows.

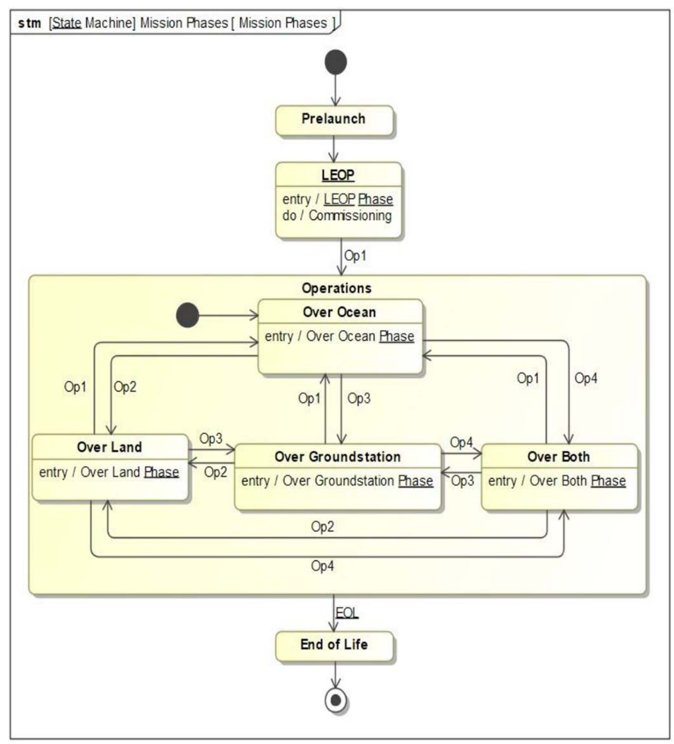

Fig. 3. State machine diagram—mission phases.

The mission analysis for the chosen orbit is performed in STK. *Connector 1* illustrates that the output of this analysis is stored in an Excel file. This output is the nominal mission profile, which details the times at which the spacecraft transitions between mission phases. The first 2.5 h of the nominal mission profile is presented in Table I.

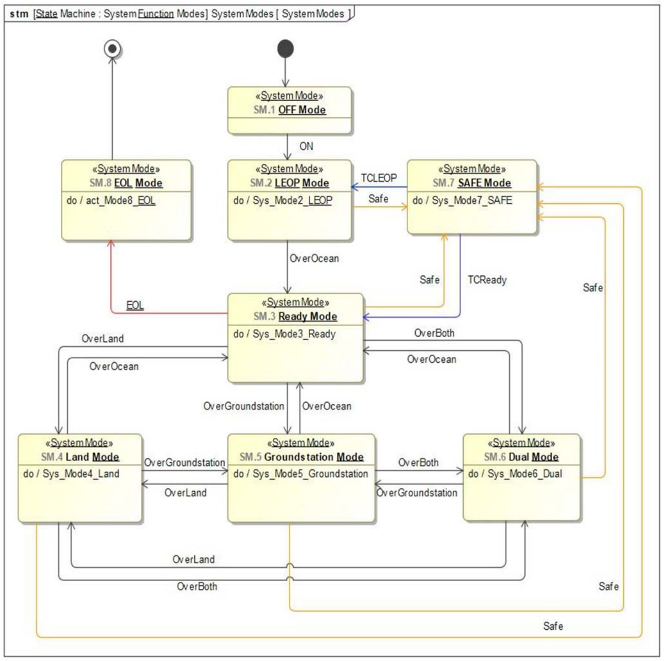

Fig. 4. State machine diagram—system modes.

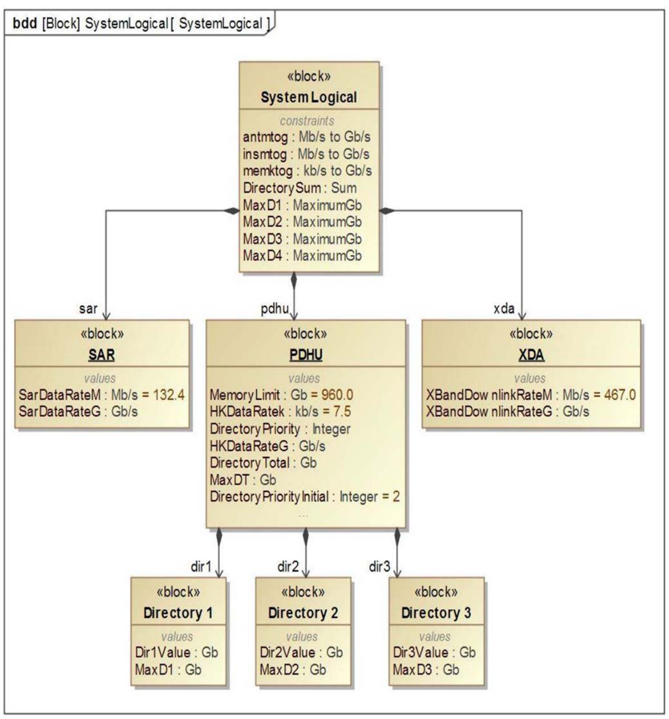

Fig. 5. Block definition diagram—logical architecture.

The *Mission Definition* model defines the phases that the mission will transition through over its life. The mission phases for this system are defined in a state machine diagram, presented in Fig. 3. After the completion of the launch and early operations phases (LEOPs), the mission transitions between the four operational mission phases—*Over Ocean, Over Land*, *Over Groundstation*, *Over Both* (land and ground station)—thousands of times throughout the mission as the spacecraft moves through its orbit around the Earth. The final

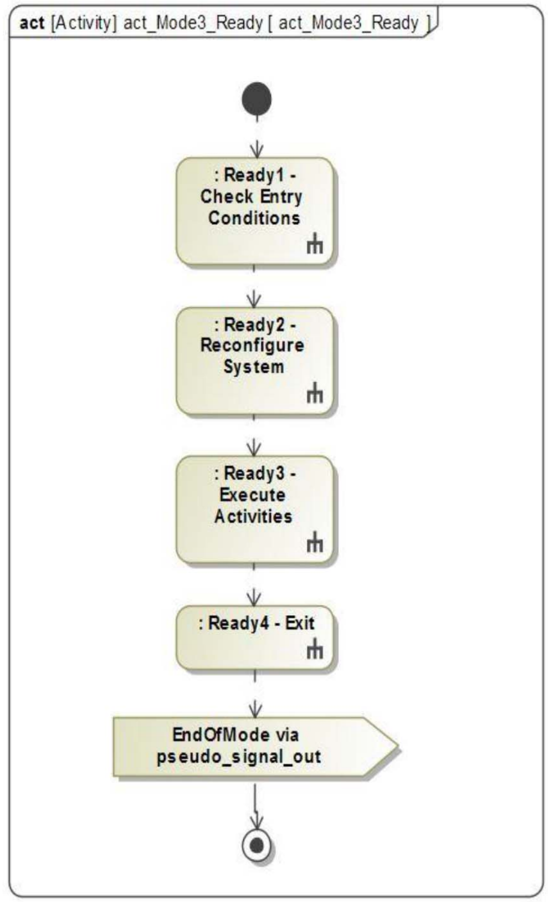

Fig. 6. Activity diagram—mode3 ready.

mission phase will be the end of life (EOL). Note that the *Mission Definition* model does not reflect the system design. The separation of the mission and the system is a crucial concept behind this approach [14].

On executing the Biomass model, the mission profile is read (via the excel-based STK interface) by the *Mission Definition* model, represented by *Connector 2* in Fig. 2. Essentially, this assigns a series of durations to the mission phases defined in Fig. 3. The *Mission Definition* model then begins stepping through these phases. This process is controlled by guards, visible in Figs. 3, 4, and 7—transitions to a new phase or mode can only be triggered by the receipt of an appropriate signal, which in turn can only be triggered by an appropriate activity. Whenever the mission transitions to a new phase, system-level functionality is triggered, represented by *Connector 3*, by requesting that the system transitions into a particular system mode.

The top-level system modes are defined in a state machine diagram, presented in Fig. 4. The functionality of the system will be different depending on which mode the system is in, and each mode has been defined in response to a particular phase of the mission. Table II displays this information.

For each mode, system behavior is defined in the form of an activity that must be performed on entry into the mode. This activity has the same structure for each mode, consisting of four nested activities, but the details are specific to each mode. The *Mode3_Ready* activity is presented as an

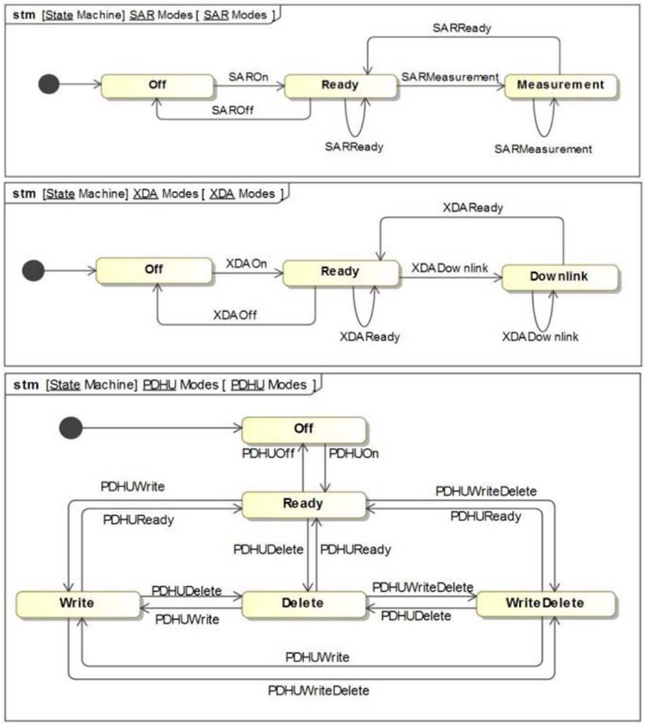

Fig. 7. SAR, XDA, and PDHU mode diagrams.

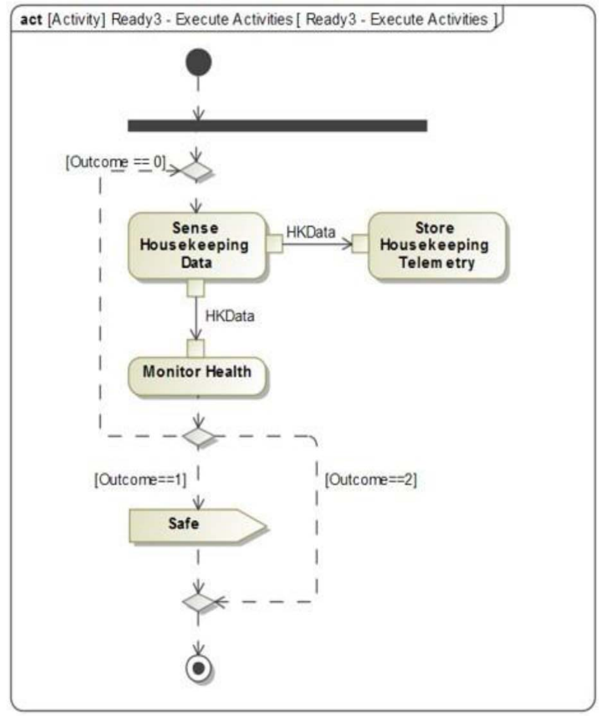

Fig. 8. Activity diagram—system ready mode execute activities.

TABLE II
MISSION PHASES AND SYSTEM MODES

Mission Phase	System Mode
Prelaunch	OFF
LEOP	LEOP / SAFE
Over Ocean	Ready / SAFE
Over Land	Land / SAFE
Over Groundstation	Groundstation / SAFE
Over Both	Dual / SAFE
EOL	EOL

TABLE III
SYSTEM AND SUBSYSTEM MODES

System Mode	SAR Mode	XDA Mode	PDHU Mode
OFF	OFF	OFF	OFF
LEOP	Ready	Ready	Ready
Ready	Ready	Ready	Ready
Land	Measurement	Ready	Write
Groundstation	Ready	Downlink	Delete
Dual	Measurement	Downlink	WriteDelete
SAFE	Ready	Ready	Ready
EOL	OFF	OFF	OFF

example in Fig. 6. This particular activity would be performed by the system when in ready mode, but the sequence of *Check Entry Conditions*, *Reconfigure System*, *Execute Activities*, *Exit* is identical for all modes. The *Functional Definition* model comprises the system-level mode diagram and these lower-level activities. Note that the *Check Entry Conditions* and *Exit* activities act as placeholders in the Biomass model—they have not been populated with further detailed activities.

Under *Reconfigure System*, the mode of each of the subsystems is updated to reflect the top-level system mode. The logical architecture diagram in Fig. 5 details the subsystems of interest. At this stage in the design process, this logical architecture contains very little detail. Its primary use at this stage is to act as a data repository for the proposed performance parameters of the system. The modes for each subsystem are presented as state machine diagrams in Fig. 7, and Table III defines their relationship with the system-level mode diagram.

Connector 4a in Fig. 2 illustrates the connection between the *Logical Architecture* and the *Functional Definition* models—the *Functional Definition* can call on values from the *Logical Architecture* (e.g., SAR data rate) to perform the analysis, and will then store the results (e.g., total memory used) back in the *Logical Architecture* model.

The *Execute Activities* activity, seen as the third action in Fig. 6, details the functions that the system must perform while in the current mode. These are very specific to each mode. When the mission is in the *Over Ocean* phase, the spacecraft will be in the *Ready Mode*—the SAR, PDHU, and XDA will all be *Ready* (on but dormant). Thus, the *Execute Activities* activity when in *Ready Mode*, presented in Fig. 8, is relatively simple—the system only needs to monitor housekeeping telemetry and determine whether a transition to SAFE mode is required. Thus, this function can be adequately represented by an activity diagram.

For *Land Mode*, *Groundstation Mode*, and *Dual Mode*, the functionality of the system is too mathematically complex to be clearly portrayed by an SysML activity diagram. In these cases, the functionality of the system has been defined mathematically in a MATLAB script, and is nested under the *Execute Activities* activity. These scripts mathematically

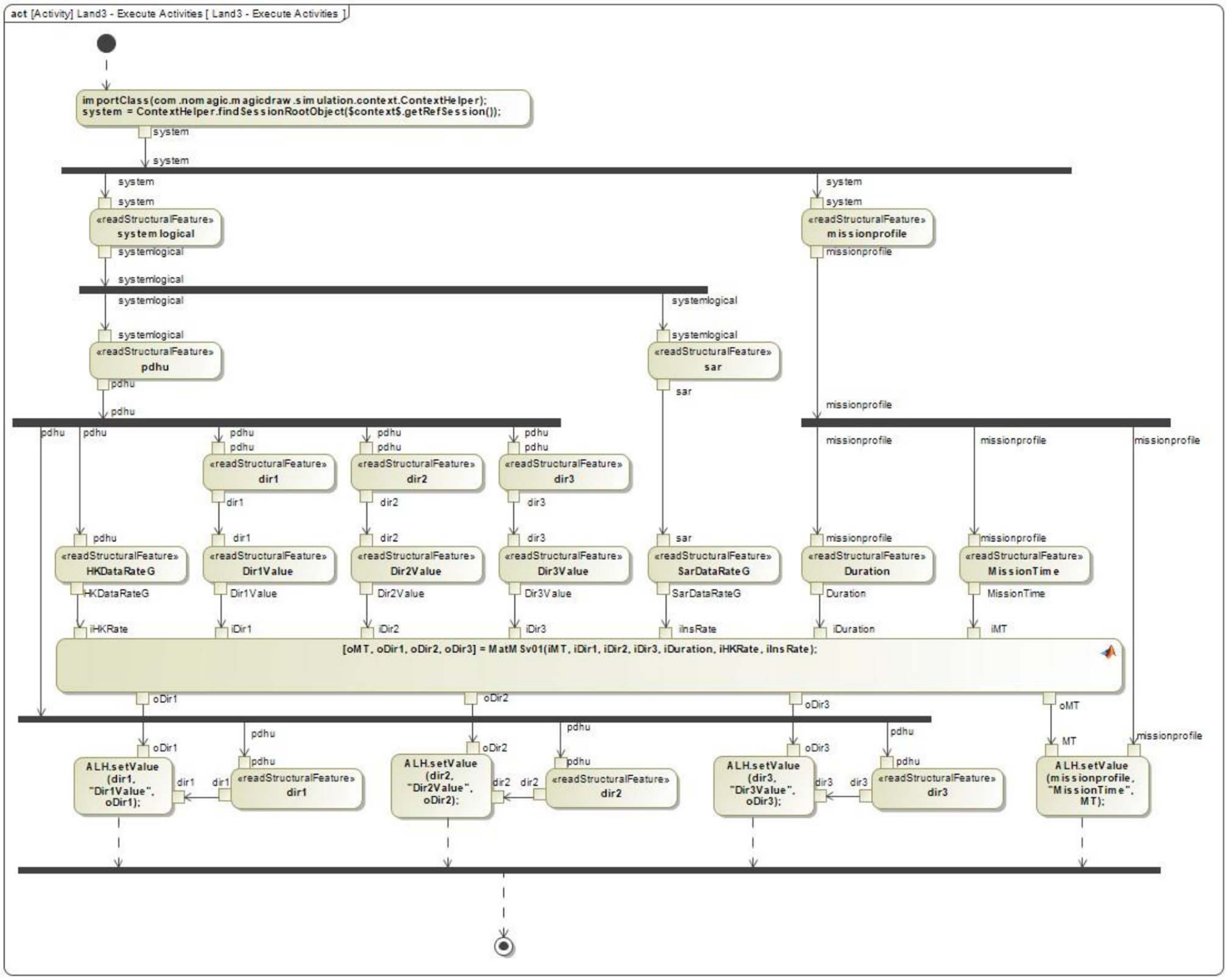

Fig. 9. Activity diagram—system land mode execute activities with MATLAB.

represent specific system behaviors. This is demonstrated in terms of the *Land Mode* in Fig. 9. In this case, the MATLAB script defines the increase in science data stored in the PDHU science directories as a function of the time spent in this mode and the science data rate. This relationship is represented by *Connector 4b* in Fig. 2. In this sense, these behaviors are extensions to the *Functional Definition* model.

Requirements relevant to the system under consideration are maintained in a *Requirements Model* and can be viewed as a Requirements Table in the model. Refined mathematical constraints are produced—these and all other references to requirements can be traced to the requirements in this table. When the model is executed and the system response to the mission profile is simulated, the quantitative system requirements are checked automatically and deemed either "satisfied" or "unsatisfied". This connection is represented by *Connector 5* in Fig. 2. Requirements can be satisfied by elements of the architecture, behaviors within the functionality, and by values calculated during the simulation.

Connector 6 in Fig. 2 represents the final stage of the analysis. A section of this output is presented in Table IV. On completion of the simulation, key information, including simulation properties (e.g., number of steps), performance

TABLE IV
SECTION OF NOMINAL SIMULATION OUTPUT

Req Ins Data Rate	Gb/s	0.36
Req Memory Limit	Gb	960
Downlink Priority		2
Sim Steps		1000
Mission Time	s	519670
Ins Data Rate	Gb/s	0.132
HK Data Rate	Gb/s	7.50E-06
Downlink Rate	Gb/s	0.467
D1 Peak	Gb	0.0453
D2 Peak	Gb	183.0603
D3 Peak	Gb	394.4795
Total Peak	Gb	503.5167

values (e.g., total memory used), and requirements checks (e.g., Requirement 1—pass), is formatted and output as the Excel-based *Simulation Output* file.

The Biomass model has, therefore, been structured in such a way as to enable the systematic execution of the design diagrams such that the system response to a mission profile can be analyzed. Checking these results against the requirements for a particular use case validates the mission and system design. For each simulation, key data is exported to an Excel file.

VI. Methodology

With respect to systems engineering, flexibility is defined as "enabling a system to change easily in the face of uncertainty considering technical and technological standpoints" [46], [47]. At this early stage in the system lifecycle, the Biomass system is solely represented by the Biomass model, and therefore the ease with which the system can change is represented by the ease with which the model can change.

Before any changes to the inputs to the Biomass model were made (and their effects assessed), the nominal analysis was completed. As with the detailed discussion regarding the justification and method behind the development of the Biomass model, this analysis is discussed in detail in [14].

Research Question 1) (defined in Section I) requires the investigation of the flexibility of the Biomass model in terms of changes to the system specification (*Logical Architecture* and *Functional Definition* models). To achieve this, the following changes were identified.

Case A: Update memory limit.

Case B: Update the number of directories.

Case C: Update system mode definition.

Case D: Add behavior.

Cases *A* and *B* represent possible changes to the logical architecture. Case *A* investigates how a simple value property can be updated and propagated through the model. Case *B* is more complex and represents a change to the structure of the logical architecture itself. Cases *C* and *D* represent changes to the system functionality. Case *C* investigates the addition of a new mode and its corresponding required behavior. Case *D* details the addition of new behavior to assess a particular aspect of the system.

Research Question 2) regards the flexibility of the Biomass model in terms of changes to the *Mission Definition* model and the *System Requirements* model (i.e., aspects of the project, other than the system design itself, that are represented in the model). Two cases were considered.

Case E: Update mission profile.

Case F: Update maximum SAR instrument data rate.

Case *E* represents a change to the mission (perhaps as a result of the selection of a new orbit, for example). Case *F* introduces a simple requirement and demonstrates the effect that changing this will have on the system model and simulation. The aim is to investigate whether modeling the requirements alongside the system will enable automatic requirements checking to be implemented, thus improving traceability and consistency, particularly when requirements are changed.

These six cases have been selected as they cover important aspects of the project that can reasonably be expected to experience late changes in a project and address the different ways in these aspects can change. The chosen cases cover changes to the system definition (*Logical Architecture* model and *Functional Definition* model), the mission profile and the requirements.

For all six of these cases, the changes necessary to represent these updates in the Biomass model were made by the same skilled SysML user that developed the Biomass model. The simulation was then rerun to produce a new set of results.

As a measure of the ease of a change, and thus the flexibility of the model, the time taken to implement the change and yield an updated simulation result has been used. This measure incorporates the ease with which the required model changes can be recognized, the navigability of the model, the scale of the change in terms of adding/removing model elements and the level of automation employed when updating the results to maintain consistency throughout the model.

VII. Results

The data acquired from each of the six defined cases is presented in this section, however, the implications of these results on the system design/mission goals will not be discussed in detail. The objective of this section is to investigate the flexibility of the model, not to consider the design itself. What will be discussed in this section, therefore, is how accommodating the model template is to these changes, and how easily and effectively the results can be generated.

The analysis was initially performed for the nominal case (no changes). The data stored in the PDHU over the first three days of the mission for the nominal case is presented in Fig. 10. This figure illustrates the variation of the data stored onboard the spacecraft (in each directory and in total) over time, and how these values compare to the nominal upper limit of 960 Gb.

A. Update Memory Limit

In the nominal case, the maximum allowable data to be stored on board the spacecraft at any point during the mission was defined as 960 Gb. Executing the nominal Biomass model demonstrated that the system design satisfied this requirement with 44% margin (with the total on-board memory usage peaking at 539 Gb) [14].

To investigate the flexibility of the model in this regard, this value was updated in Case *A*, and represents a change in the logical architecture model. The value of *Memory Limit*, a property of the *PDHU* block in the *Logical Architecture* model (see Fig. 5), was changed from 960 Gb to 400 Gb.

Changing this value does not affect the actual functionality of the spacecraft, but it does affect the outcomes of the simulation in that the requirement is no longer satisfied. The updated results for the first three days of the mission are presented in Fig. 11—it can be seen here that the total data stored on board the spacecraft exceeds the limit on multiple occasions.

Implementing this change in the model took 5 min, and the updated value was automatically used as an input to the simulation when the model was next executed. The graph produced and the Excel-based *Simulation Output* reflected this change, stating that the requirement was no longer satisfied. This is shown in Table V. The capability of the model to propagate a change in the logical architecture, through the simulation

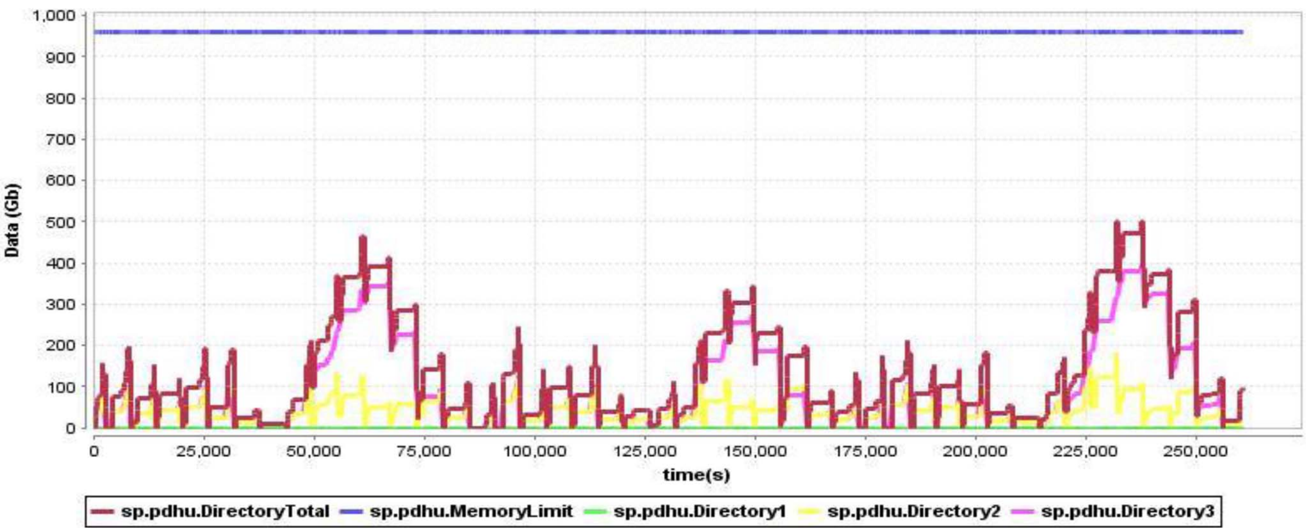

Fig. 10. Nominal case—data stored in PDHU over first three days of mission.

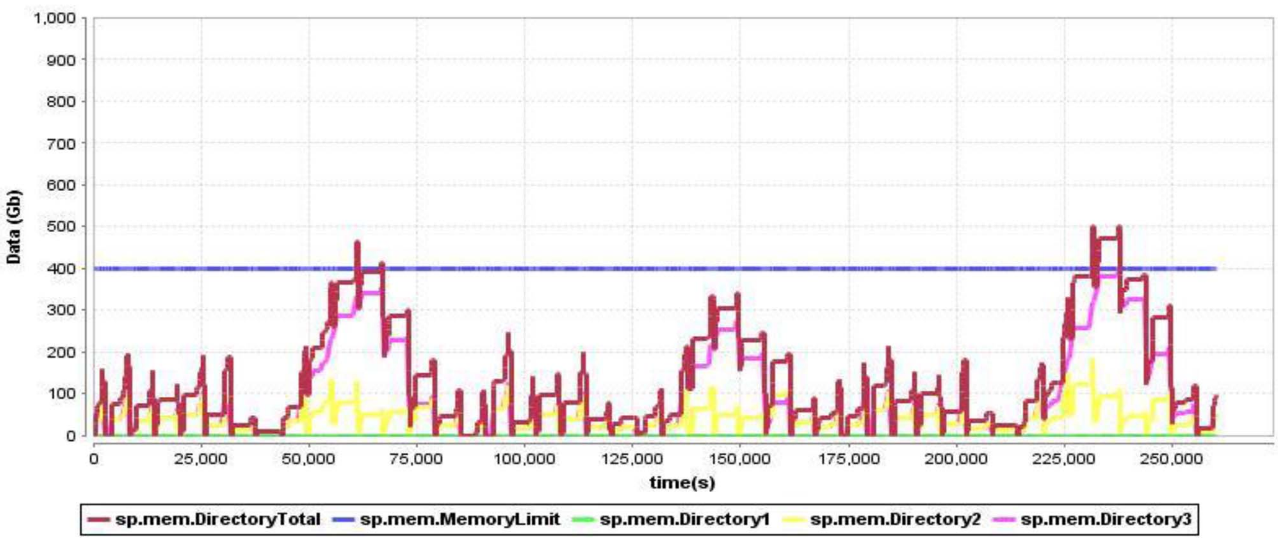

Fig. 11. Case *A*—data stored in PDHU over first three days of mission.

of the system and subsequent analysis, to the requirements checks and *Simulation Output* demonstrate the benefits of this approach in terms of flexibility, consistency, and traceability. This automatic propagation through the model, updating the status of the requirements, would not be possible in a traditional DBSE approach.

B. Update Number of Directories

Another, more complex, example of a change to the logical architecture is the addition of another directory into the PDHU. Case *B* investigates this change. Note that the science data from the SAR is stored into the two science directories in parallel, at an equivalent rate of 66.2 Mb/s (half of the nominal SAR data rate). Data from each of the three directories is then downlinked in series.

To represent this change, the *Logical Architecture* and the *Functional Definition* models must be updated. An additional block with the appropriate value properties is added to the *PDHU* to represent the addition of the fourth directory to the logical architecture—this change is presented in Fig. 15.

TABLE V
CASE *A*—SECTION OF SIMULATION OUTPUT

Req Ins Data Rate	Gb/s	0.36
Req Memory Limit	Gb	400
Downlink Priority		2
Sim Steps		1000
Mission Time	s	519670
Ins Data Rate	Gb/s	0.132
HK Data Rate	Gb/s	7.50E-06
Downlink Rate	Gb/s	0.467
D1 Peak	Gb	0.0453
D2 Peak	Gb	183.0603
D3 Peak	Gb	394.4795
Total Peak	Gb	503.5167

The MATLAB scripts have been updated to reflect the fact that incoming science data must be divided between three directories, and outgoing science data requires the downlink of data from three science directories. The *Execute Activity*

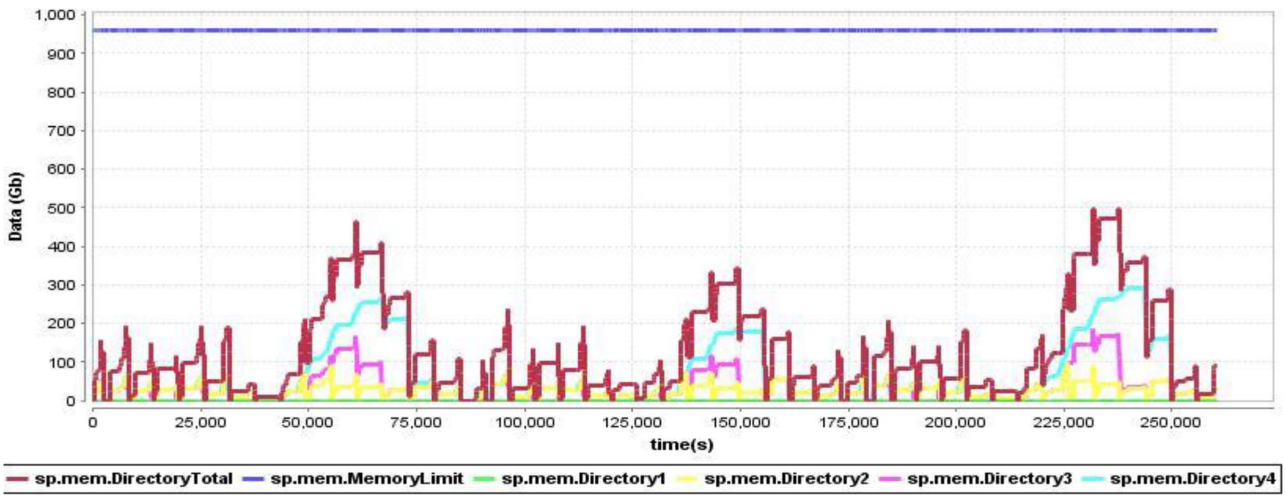

Fig. 12. Case *B*—data stored in PDHU over first three days of mission.

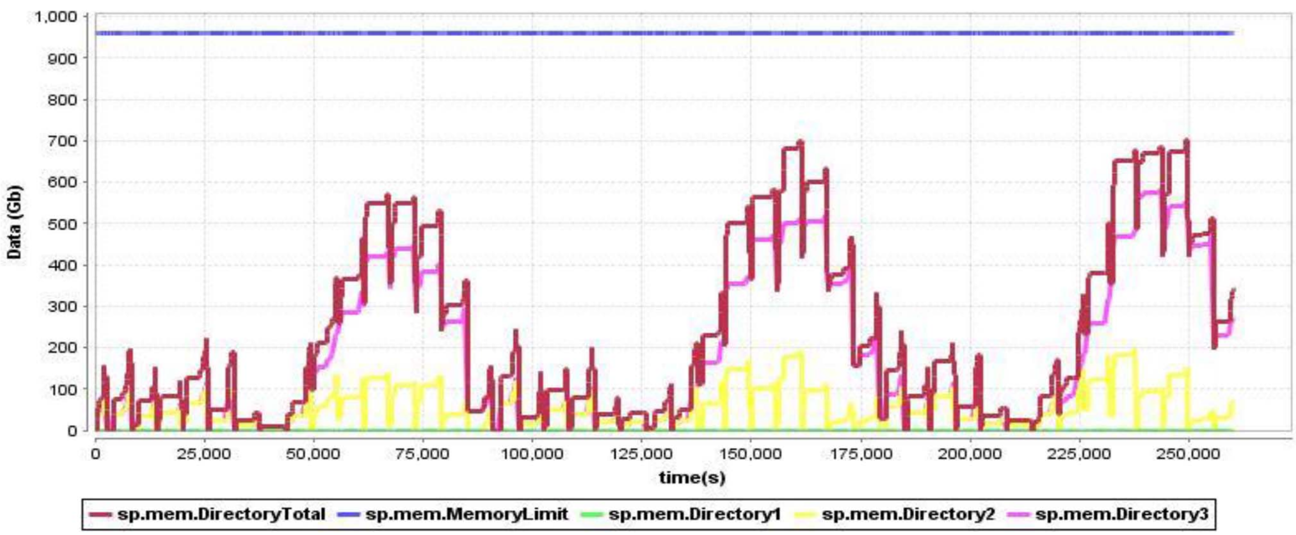

Fig. 13. Case *C*—data stored in PDHU over first three days of mission.

activity diagrams (see Fig. 9) have been updated to account for the extra input and output values required by the MATLAB script.

As can be seen in Fig. 12, the total memory stored on board the spacecraft throughout the mission remains unchanged. The difference is that this data is now stored across four directories, and this is reflected in the results.

Updating the logical architecture to include the fourth directory took 5 min. Updating the MATLAB scripts and the activity diagrams to correctly represent the new system behavior took 1 h. As with case *A*, this MBSE approach ensures that the system definition is directly linked to the system simulation and analysis, enabling automatic propagation through to the simulation results.

C. Update System Mode Definition

The flexibility of the model with regards to implementing a change to the system functionality has also been investigated. To represent this, it was assumed that a second science mode, *Land Mode 2*, was added at system level, and that this new mode had new behavior—the SAR data rate was four times greater than the nominal SAR data rate. This mode was added to the system mode diagram—this is presented (with the new mode highlighted) in Fig. 16. This can be compared to the original system mode diagram presented in Fig. 4. Like *Land Mode*, *Land Mode 2* was defined to set the SAR to *Measurement* mode, set the PDHU to *Write* mode and set the XDA to *Ready* (standby) mode.

To define this behavior in the model, the same functional structure was used as for all the other modes with complex behavior (see Figs. 6 and 9). A new MATLAB script was produced that used this increased data rate to calculate the data stored on board after a period of time in *Land Mode 2*. The results from the first three days of the mission can be seen in Fig. 13. While the effect of the increased data rate is evident, the maximum total memory stored on board the spacecraft remains under the limit, and this is reflected in the simulation results in which the requirements are deemed satisfied.

The time taken to add the new mode to the system mode diagram was 20 min. The time taken to define the behavior, however, will vary greatly depending on the complexity of

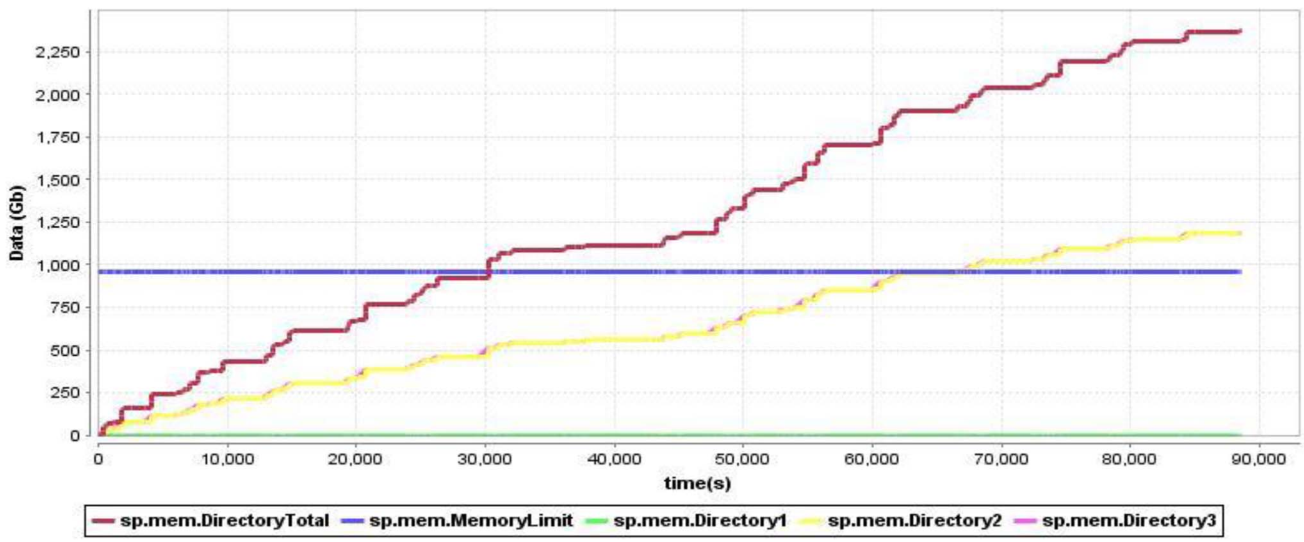

Fig. 14. Case *D*—data stored in PDHU over first day of mission.

the behavior. By following the existing Biomass model patterns, the user was able to produce the required structure for the behavior in approximately 1 h. The time taken to then produce an appropriate and accurate MATLAB script is dependent on the complexity of the behavior being modeled. In this example, the behavior was a slight modification of the behavior defined for *Land Mode*, and so it was a relatively quick process of 1 h. Therefore, the total time taken to define the behavior for the new mode, using the standard combination of activity diagrams and MATLAB scripts, was 2 h and 20 min.

This example demonstrates that new modes can be added relatively quickly and easily by following the standard template structure. Case *D* demonstrates how behavior can be added to the system definition without adding an extra mode.

D. Add Behavior

The structure of the model has been developed to allow for the addition of interrogative MATLAB script (see Fig. 9). For each mode, the standard MATLAB script produced and implemented in [14] represents the nominal operation of the spacecraft. The existing structure of the activity diagrams used to call these scripts allows the user to produce alternative MATLAB scripts and call them from the model.

As an example, alternative versions of the MATLAB scripts for *Groundstation Mode* and *Dual Mode* were produced. These were modified to assume the failure of Svalbard ground station at the very beginning of the mission, in order to get a better idea of how long it might take the spacecraft to completely exhaust its available memory without the ability to downlink data.

The nominal MATLAB script were replaced with the modified versions, and the model was executed. The results of this analysis are presented in Fig. 14. It can be observed that the memory on board the spacecraft exceeds the defined memory limit after approximately 8 h.

This case demonstrates the possibility of producing alternative spacecraft behaviors in the form of MATLAB scripts and plugging these into the existing system model. By following

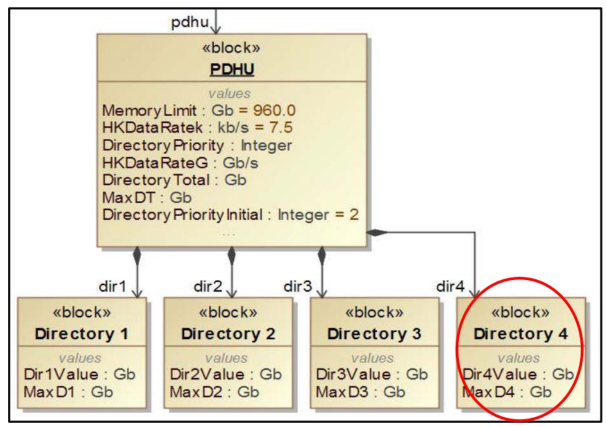

Fig. 15. Case *B*—updated PDHU structure.

the model template and utilizing an existing activity diagram like the one seen in Fig. 9, this can be achieved in approximately 1 h. Multiple behaviors can be defined ready to be called. This case provides a simple example of an alternative behavior script, but as with case *C* the time taken to produce an accurate and appropriate MATLAB script depends on the complexity of the functionality being modeled.

E. Update Mission Profile

As described in Section V, the mission profile was generated using dedicated orbit modeling software STK. The mission profile represents the orbit that the spacecraft will move through throughout its mission and is separated into distinct phases. A section of the nominal mission profile is presented in Table I.

The structure of the model is intended to maintain separation of the mission (*Mission Definition*) and system (*Logical Architecture* and *Functional Definition*) throughout the design process. In this case, then, no changes are being made to the system. The mission profile is updated, and the response of the nominal system design to this new mission profile can be assessed.

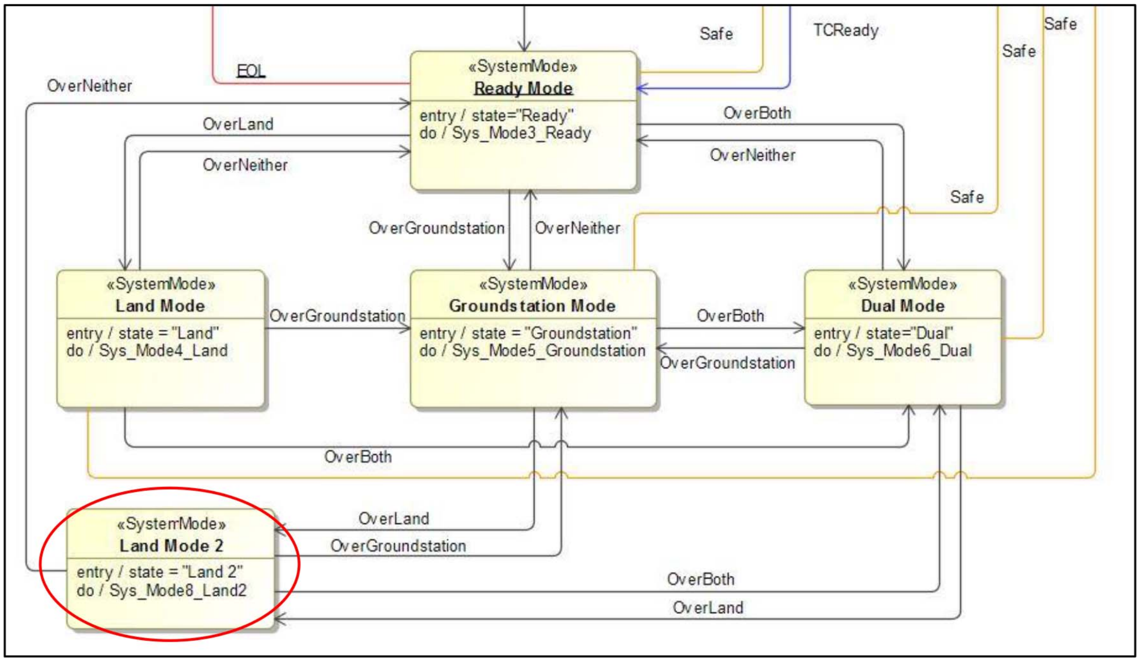

Fig. 16. Case *C*—system mode diagram with extra mode.

The mission profile, therefore, can only be updated externally by updating aspects of the mission and then using STK to regenerate the mission profile. Changing which ground stations are to be used, for example, will affect where the *Over Groundstation* phases of the mission are found, and thus the response of the system will be different. Similarly, updating the "area of interest" of the mission (i.e., when the SAR instrument must be in measurement mode, nominally when over land) will generate a new series of phases in which the spacecraft must switch to a measurement mode. Due to the separation of the mission and system in this process, these changes cannot be made as they are outside the scope of the system model.

Assuming the mission was updated, and the response of the nominal system needed to be analyzed, the link to the relevant mission profile could be updated in 5 min by the user. Executing the system model would then simulate the system's response to the updated mission profile and all results and outputs would reflect this change.

F. Update Maximum SAR Instrument Data Rate

The purpose of this case was to investigate how a requirement could be updated within the model, and how this change could be propagated through the model to the simulation automatically, thus providing an updated response to the set of results. The requirement in question was updated from: "the maximum total SAR instrument data rate shall be $\leq$ 360 Mb/s" to "the maximum total SAR instrument data rate shall be $\leq$ 120 Mb/s."

In this case, as opposed to case *A*, the requirement itself has been updated. The value capable of satisfying this requirement, *InsDataRateM*, is owned by the SAR block and is the actual value for the data rate of the instrument. It has a value of 132.4 Mb/s.

TABLE VI
CASE *F*—SECTION OF SIMULATION OUTPUT

Req Ins Data Rate	Gb/s	0.12
Req Memory Limit	Gb	960
Downlink Priority		2
Sim Steps		1000
Mission Time	s	519670
Ins Data Rate	Gb/s	0.132
HK Data Rate	Gb/s	7.5E-06
Downlink Rate	Gb/s	0.467
D1 Peak	Gb	0.0453
D2 Peak	Gb	183.0603
D3 Peak	Gb	394.4795
Total Peak	Gb	503.5167

The requirement has been purposely changed from one that would have been satisfied by the actual instrument data rate of 132.4 Mb/s to one that would not be. On beginning the simulation, the updated constraint is immediately flagged as being unsatisfied (the actual value of 132.4 Mb/s exceeds the maximum allowable data rate of 120 Mb/s). When the simulation is complete, this is recorded in the Excel-based *Simulation Output* detailing the simulation as a requirement that has not been satisfied. The updated output is presented in Table VI.

The time taken to implement this change in the Biomass model was 5 min. This change is then automatically fed into the simulation when the model is next executed, and the results (in this case stating that the requirement is not satisfied) are updated to reflect the new requirement automatically.

This example represents a trivial change to the requirements, as it is immediately observable that the requirement

TABLE VII
SUMMARY OF TIME TAKEN TO IMPLEMENT CHANGES AND COLLECT RESULTS

Case	Time Taken to Implement Change (mins)	Total Time to Collect Results (mins)
A	5	95
B	65	155
C	140	230
D	60	150
E	5	95
F	5	95

is not satisfied by the design. The same structure and process can be adopted, however, for more complex changes to the requirements, where the model must be executed to determine whether the new requirement is satisfied.

G. Results Summary

The flexibility of the Biomass model in terms of updating the requirements, the mission profile and the system definition has been assessed. The time required to implement the necessary changes to the model seen in this investigation varies from 5 min to a couple of hours depending on the complexity of the change being implemented. The total time to collect the results is this time plus the 1.5 h required to run the simulation. This information is presented in Table VII. A direct comparison to the time required using DBSE techniques is difficult to produce as these traditional techniques are not conducive to this level of simulation and analysis. What can be said, however, is that just implementing these changes with a DBSE approach would likely take longer as there is no intuitive model structure to help locate the relevant diagram/database. Without the central data repository that MBSE can offer, a DBSE approach might require the same change to be made in multiple models where data is replicated. Once the appropriate change (or changes) have been made, the major benefits of this MBSE approach become evident—the changes can be automatically propagated through the rest of the model and will be reflected in the results of any subsequent simulations and analyses. This capability sets this MBSE approach apart from the traditional DBSE methods, which would require manually updating disparate executable models based on the updated data in the documents.

VIII. DISCUSSION

In this section, the benefits and limitations of the approach presented in this article are discussed in more detail. The six cases presented in Section VII have investigated the flexibility of the Biomass model in terms of the ease with which changes to the requirements, system definition, and mission profile can be implemented.

While implementing the changes to the model (requirements, system definition or mission definition) can often be achieved in a matter of minutes, the effect that these changes have on the system response (in particular whether the requirements have been met) are often not observable until the model has been executed and the simulation run again. The propagation of the changes through the simulation to the outputs may be an automatic process, but work must be done to improve its running time. The method presented in this article checks the system functionality at every stage of the simulation. The time required to run the simulation (approximately 1.5 h for 11 500 rows of mission data, corresponding to 70 days of the mission) is a limiting factor of this approach.

This investigation has focussed on a specific aspect of the Biomass spacecraft functionality, namely the SAR-PDHU-XDA system, to demonstrate some of the benefits of the MBSE approach used. Thus, cases *A*–*F*, chosen here to represent the wide range of possible changes, have been limited by the application. For example, cases *A* and *B*, which investigate the flexibility of the model when applying changes to the logical architecture, are necessarily limited by the simplicity of the architecture. In addition to this, the applicability of the MBSE approach and the general structure of the Biomass model has only been investigated in the context of a single mission—the Biomass mission.

Cases *A* and *F* together highlighted that the way in which requirements are represented in the model must be carefully considered. In case *A*, for instance, the requirement itself does not include an upper value. The value of the upper memory limit is specified as part of the logical architecture. In case *F*, the upper limit for the SAR data rate is fixed within the requirement. Adjusting the requirements would not be possible without prior discussion with the customer. The MBSE approach adopted, therefore, encourages early discussion of requirements with customers by informing the systems engineers of any inconsistencies that may be present.

Case *A* also raised a question with regards to the general systems engineering process. As discussed in [14], the nominal Biomass model represents the project in Phase B, where the aim is to establish a functionally complete design solution [19]. Ideally, the logical architecture should not be constrained by physical implementation at this stage (although clearly in real industrial environments it often makes economic sense to deviate from this ideal). Thus, there is the question of imposing the 960 Gb limit on the PDHU at this early stage. Perhaps a more ideal process would be to specify the required margin, simulate the memory used during the mission, and derive from this a requirement for the minimum acceptable data capacity of the chosen solution.

Throughout the six cases, the time taken to implement the changes to the Biomass model has been noted. These changes have been made by the same skilled modeler that developed the nominal Biomass model, and thus there was a familiarity with the model structure that may have decreased the time required to complete any model updates. This can be justified to some degree by stating that all future versions of this model template, when applied to other use cases, will utilize similar, well-defined MBSE patterns that can be understood and updated relatively easily, without detailed knowledge of the model structure (which has been designed to enable simulation). To test this thoroughly, however, future investigations might use different modelers with a range of modeling experience.

The model developed and presented in this article has been constructed using SysML. While SysML is particularly suited to the description of the appearance of a system, there are limitations. For example, some semantics have to be modified or used out of their originally intended context [40], and the notation does not use formal activity specifications [41]. This can be addressed to some degree by refining SysML and developing a dedicated profile with application-specific semantics. For example, at this stage it is not always possible to distinguish between different types of SysML *state* or *action*. With the former, it would be useful to be able to distinguish between mission phases and subsystem modes, for example, and with the latter it is crucial to be able to differentiate between an action performed by the system itself (e.g., *monitor health*), and an action required to progress the simulation (e.g., *Call MATLAB*).

It must be noted, therefore, that the work presented in this article is part of a larger project to produce a Template for Early Functional Definition and Analysis. The Biomass model discussed throughout this article can be seen as the application of the first iteration of this template to the Biomass mission. This investigation has demonstrated some of the benefits of this MBSE approach when modeling the functional avionics of a spacecraft in Phase B, particularly the flexibility of such an approach when accommodating changes in the requirements, mission, and system. Further work is necessary, however, and is discussed in the next section.

IX. Future Work

The previous section discussed the benefits and limitations of this approach. Lessons have been learned and potential areas of improvement have been identified in terms of the continued development of the Biomass model. This section presents potential future work highlighted as a result of this discussion.

The core aim of the intended future work is to produce a Template for Early Functional Definition and Analysis. This template will incorporate MBSE patterns to aid in the definition and simulation of the functional avionics of a spacecraft, developed over multiple use cases—the use case presented in this article being the first. As mentioned in the previous section, this will involve the development of a suitable SysML profile to accompany the model template and address some of the limitations of the language.

The time required to run the simulation needs to be addressed. Future work will consider possible ways of reducing the repetitive nature of the analysis while maintaining its completeness, to improve the efficiency of future analyses.

The necessity of performing a similar investigation on a more complex mission model has also been highlighted in the previous section. While the same model structure and MBSE patterns would be adopted to describe and simulate more complex architectures, a more complex use case would provide the opportunity to explore a greater variety of engineering changes, and thus instil greater confidence in this MBSE approach. The approach could be expanded and demonstrated further by applying it to other use cases on different spacecraft missions and conducting similar work to assess the flexibility of the model in the context of these applications. The ultimate aim is to produce a system model template that can be applied to a variety of functional spacecraft use cases and enable the definition and simulation of the system. Only when the template has been shown to be applicable to numerous, varied use cases can it be said this has been achieved.

X. Conclusion

The Biomass model has been developed using an MBSE approach and represents the Biomass mission, currently in Phase B. The purpose of this model is to clearly and consistently describe the structural and functional aspects of the Biomass system, and to simulate the system behavior in response to a mission profile. In doing so, the Biomass model attempts to contribute to the validation of the initial high-level functional design of the Biomass PDHU against the mission needs. An initial high-level functional design of the spacecraft, focusing on the Synthetic Aperture Radar instrument, PDHU and XDA, has been defined using SysML. The structure of the spacecraft system has been represented by block definition and internal block diagrams, and the functionality by executable state machine and activity diagrams. High-level textual requirements, refined by mathematical constraints where appropriate, have been presented and maintained within the model and are formally linked to the logical and functional architecture. The functional definition of the system is extended and mathematically refined in MATLAB. The mission profile has been produced by orbit modeling software STK and is linked to the central SysML model via Microsoft Excel. Executing the Biomass model enables the validation of the system design against the requirements, with key results output to an Excel-based output file.

The flexibility of this MBSE approach has been investigated by applying changes to the Biomass project (requirements, mission definition, and system design), and noting the time required to implement these changes in the Biomass model and propagate them through to the results of the analysis. The changes have been made independently of each other and include: changes to logical architecture, changes to functional definition, changes to the mission profile, and changes to the requirements.

The investigation has highlighted some limitations of the Biomass model—including the time taken to run each simulation and the necessity of performing similar analyses for other applications on other missions. From these limitations, potential future work has been identified. The work identified will form the basis of the next phase of development of the Template for Early Functional Definition and Analysis, a continuation and generalization of the Biomass model.

The overall outcome of this investigation, however, is the demonstration that this model-based approach to defining and simulating the functional avionics of a spacecraft introduces flexibility into the design by providing standard diagram templates and reusable patterns for adding and updating system information.

Data Access Statement

Due to confidentiality agreements with industrial partners, supporting data cannot be made available without permission from the Head of department of Functional Avionics at Airbus, Stevenage. Please contact the corresponding author for further details if data is requested.

Acknowledgment

The authors would like to acknowledge support from Airbus employees A. Cortier, S. Estable, T. Fenal, J. O'Rourke, and A. Rossignol. The authors would also like to thank No Magic PLC for their provision of a Cameo Systems Modeler (18.5) Licence, under which this article was completed. The views expressed in this article do not necessarily represent those of Airbus.

References

[1] D. R. Wibben and R. Furfaro, "Model-based systems engineering approach for the development of the science processing and operations center of the NASA OSIRIS-REx asteroid sample return mission," *Acta Astronaut.*, vol. 115, pp. 147–159, Oct./Nov. 2015.

[2] K. M. Gough and N. Phojanamongkolkij, "Employing model-based systems engineering (MBSE) on a NASA aeronautics research project: A case study," in *Proc. Aviation Technol. Integr. Oper. Conf.*, Atlanta, GA, USA, 2018, pp. 3361–3374.

[3] R. S. Kalawsky *et al.*, "Bridging the gaps in a model-based system engineering workflow by encompassing hardware-in-the-loop simulation," *IEEE Syst. J.*, vol. 7, no. 4, pp. 593–605, Dec. 2013.

[4] B. London and P. Miotto, "Model-based requirement generation," in *Proc. IEEE Aerosp. Conf.*, 2014, pp. 1–8.

[5] M. Bozzano *et al.*, "Spacecraft early design validation using formal methods," *Rel. Eng. Syst. Safety*, vol. 132, pp. 20–35, Dec. 2014.

[6] T. Moser and S. Biffl, "Semantic integration of software and systems engineering environments," *IEEE Trans. Syst. Man, Cybern. Part C, Appl. Rev.*, vol. 42, no. 1, pp. 38–50, Jan. 2012.

[7] L. Anderson *et al.*, "Enterprise modeling for cubesats," in *Proc. IEEE Aerosp. Conf.*, 2014, pp. 1–16.

[8] F. Mhenni, N. Nguyen, and J. Y. Choley, "SafeSysE: A safety analysis integration in systems engineering approach," *IEEE Syst. J.*, vol. 12, no. 1, pp. 161–172, Mar. 2018.

[9] J. Holt, S. Perry, R. Payne, J. Bryans, S. Hallerstede, and F. O. Hansen, "A model-based approach for requirements engineering for systems of systems," *IEEE Syst. J.*, vol. 9, no. 1, pp. 252–262, Mar. 2015.

[10] R. Cloutier, B. Sauser, M. Bone, and A. Taylor, "Transitioning systems thinking to model-based systems engineering: Systemigrams to SysML models," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 45, no. 4, pp. 662–674, Apr. 2015.

[11] S. Chhaniyara, C. Saaj, M. Althoff-Kptzias, I. Ahrns, and B. Maediger, "Model based system engineering for space robotics systems," in *Proc. 11th Symp. Adv. Space Technol. Robot. Autom.*, Noordwijk, The Netherlands, 2011.

[12] Y. Jarraya, A. Soeanu, M. Debbabi, and F. Hassaïne, "Automatic verification and performance analysis of time-constrained SysML activity diagrams," in *Proc. Int. Symp. Workshop Eng. Comput. Based Syst.*, 2007, pp. 515–522.

[13] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo, "Integrated model-based systems engineering (MBSE) applied to the simulation of a CubeSat mission," in *Proc. IEEE Aerosp. Conf.*, 2014, pp. 1–14.

[14] J. Gregory, L. Berthoud, T. Tryfonas, and A. Prezzavento, "Early validation of the data handling unit of a spacecraft using MBSE," in *Proc. IEEE Aerosp. Conf.*, 2019, pp. 1–15.

[15] M. Arcioni *et al.*, "The biomass mission, status of the satellite system," in *Proc. IEEE Geosci. Remote Sensing Symp.*, 2014, pp. 1413–1416.

[16] *Report for Mission Selection: Biomass, ESA SP-1324/1 (3 Volume Series)*, Eur. Space Agency, Noordwijk, The Netherlands, 2012.

[17] D. Kaslow, "Developing a CubeSat MBSE reference model—Interim status #3," in *Proc. IEEE Aerosp. Conf.*, 2017, pp. 1–16.

[18] D. Kaslow and A. M. Madni, "Validation and verification of MBSE-compliant CubeSat reference model," in *Proc. 15th Annu. Conf. Syst. Eng. Res.*, 2017, pp. 1–10.

[19] S. J. Kapurch, *NASA Systems Engineering Handbook*, NASA, Washington, DC, USA, 2007.

[20] C. Gibson, M. Bonnici, and J.-F. Castet, "Model-based spacecraft fault management design & formal validation," in *Proc. IEEE Aerosp. Conf.*, 2015, pp. 1–12.

[21] S. Estable, "Application of the 'federated and executable models' MBSE process to airbus orbital servicing missions," in *Proc. Phoenix Integr. Int. Users' Conf.*, Annapolis, MD, USA, 2018.

[22] E. Fricke, B. Gebhard, H. Negele, and E. Igenbergs, "Coping with changes: Causes, findings, and strategies," *Syst. Eng.*, vol. 3, no. 4, pp. 169–179, 2000.

[23] J. R. Wertz and W. Larson, *Space Mission Analysis and Design*, 3rd ed. Dordecht, The Netherlands: Kluwer Academic, 1999.

[24] D. Habhouba, S. Cherkaoui, and A. Desrochers, "Decision-making assistance in engineering-change management process," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 3, pp. 344–349, May 2011.

[25] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Lithe: An agile methodology for human-centric model-based systems engineering," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 504–521, May 2013.

[26] B. Motamedian, "MBSE applicability analysis," *Int. J. Sci. Eng. Res.*, vol. 4, no. 2, pp. 1–7, 2013.

[27] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," *Syst. Eng.*, vol. 21, no. 3, pp. 172–190, 2018.

[28] C. Dutenhoffer and J. Tirona, "The value of SysML modeling during system operations: A case study," in *Proc. IEEE Aerosp.Conf.*, 2013, pp. 1–10.

[29] *Systems Engineering Vision 2020, Version 2.03*, document INCOSE-TP-2004-004-02, Int. Council Syst. Eng., Seattle, WA, USA, 2007.

[30] D. Hitchins, *Systems Engineering: A 21st Century Systems Methodology*. Chichester, U.K.: Wiley, 2007.

[31] S. Smith and D. Brown, *SE101: Why Do Systems Engineering*, Int. Council Syst. Eng., Seattle, WA, USA, 2014. [Online]. Available: https://www.incose.org/docs/default-source/default-document-library/twg-se101-v11-2014-01-20.pdf?sfvrsn=e6c882c6_4

[32] J. Holt and S. Perry, *SysML for Systems Engineering*, 2nd ed. Stevenage, U.K.: Inst. Eng. Technol., 2013.

[33] *An OMG Systems Modeling Language TM Publication OMG Systems Modeling Language v1.5*, Object Manage. Group, Needham, MA, USA, 2017.

[34] S. C. Spangelo *et al.*, "Model based systems engineering (MBSE) applied to radio aurora explorer (RAX) CubeSat mission operational scenarios," in *Proc. IEEE Aerosp. Conf.*, 2013, pp. 1–18.

[35] L. Lindblad, "Data-driven systems engineering: Turning MBSE into industrial reality," in *Proc. Int. Syst. Concurrent Eng. Space Appl. Workshop (SECESA)*, Glasgow, U.K., 2018.

[36] S. Jenkins, "Is systems engineering really engineering," in *Proc. Int. Syst. Concurrent Eng. Space Appl. Workshop (SECESA)*, Glasgow, U.K., 2018.

[37] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*. London, U.K.: Addison-Wesley, 2014.

[38] INCOSE. (2015). *What Is Model Based Systems Engineering? INCOSE U.K. zGuide*. [Online]. Available: https://incoseonline.org.uk/Documents/zGuides/Z9_model_based_WEB.pdf

[39] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 1, pp. 101–111, Jan. 2012.

[40] Y. Mordecai, O. Orhof, and D. Dori, "Model-based interoperability engineering in systems-of-systems and civil aviation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 637–648, Apr. 2018.

[41] A. Blekhman, J. P. Wachs, and D. Dori, "Model-based system specification with tesperanto: Readable text from formal graphics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 11, pp. 1448–1458, Nov. 2015.

[42] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, 2nd ed. Waltham, MA, USA: The MK/OMG Press, 2014.

[43] OMG. *What is SysML? |OMG SysML*. Accessed: Aug. 2, 2018. [Online]. Available: http://www.omgsysml.org/what-is-sysml.htm

[44] D. Conway and J. Parker, "Using the general mission analysis tool (GMAT)," in *Proc. AAS Guid. Control Conf.*, Breckenridge, CO, USA, 2017, pp. 139–151.

[45] AGI. *STK. Engineering Tools|AGI*. Accessed: Feb. 4, 2019. [Online]. Available: https://www.agi.com/products/engineering-tools

[46] E. Fricke and A. P. Schulz, "Design for changeability (DfC): Principles to enable changes in systems lifecycle," *Syst. Eng.*, vol. 8, no. 4, pp. 342–359, 2005.

[47] M.-A. Cardin, "Enabling flexibility in engineering systems: A taxonomy of procedures and a design framework," *J. Mech. Design*, vol. 136, no. 1, 2014, Art. no. 011005.

Joe Gregory received the M.Eng. degree in aeronautical engineering from the University of Bristol, Bristol, U.K., in 2014, where he is currently pursuing the Ph.D. degree investigating the application of model-based systems engineering (MBSE) to spacecraft functional avionics in collaboration with Airbus Defence and Space.

The focus of his current work is early functional validation and he is using MBSE to investigate possible methods of implementation within the context of Airbus Defence and Space.

Lucy Berthoud received received the M.Eng. degree in mechanical engineering from the University of Bristol, Bristol, U.K., in 1990, and the Ph.D. degree in space physics from École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, in 1993.

She did Postdoctoral Fellowships with the European Space Agency from 1993 to 1995 and the Royal Society Fellowship with Johnson Space Centre, NASA, Washington, DC, USA, from 1995 to 1996 before going to work for British Aerospace Space Systems. In 2009, she started working for the University of Bristol, where she specializes in systems engineering for spacecraft.

Theo Tryfonas received the B.Sc. degree in computer science from the University of Crete, Heraklion, Greece, in 1996, and the M.Sc. degree in information systems and the Ph.D. degree in informatics from the Athens University of Economics and Business, Athens, Greece, in 1998 and 2003, respectively.

He is a Reader in smart cities with the Department of Civil Engineering, University of Bristol, Bristol, U.K., with a background in systems engineering, software development, and cyber security. His research expertise includes secure and resilient operation of Internet of Things applications, privacy in mobile computing, energy efficient deployments of wireless sensor networks and open data architectures for smart buildings/infrastructure.

Antonio Prezzavento received the master's degree in aeronautical engineering from the Politecnico di Milano, Milan, Italy, in 1994, and the M.Sc. degree (Distinction Birkbeck) in environment and sustainability from the University of London, London, U.K., 2018.

He is a space professional working for the last 16 years in Airbus Defence and Space. He is currently the Avionics Technical Manager for the Biomass mission primed in U.K. He was lead SW and Research and Development Engineer for Meggitt Avionics, developing digital flight instrument for the commercial aircraft sector. He has been working for many of the Airbus Defence and Space Prime programs in U.K., such as Aeolus and Solar Orbiter in the capacity of Attitude and Orbit Control Systems lead Architect and ExoMars as an Avionics Manager. He has a keen interest in new design approaches and environmental and sustainability subjects.

Ludovic Faure received the engineering degree in networking and telecommunications from the Institut national des Sciences appliquees de Toulouse, Toulouse, France, in 2005, and the Erasmum Semester in computer science from the University of Limerick, Limerick, Ireland, in 2005.

He is a space professional who has worked for CNES, ESA, and currently Airbus Defence and Space. Starting as a Software Engineer, he has specialized in model-driven architecture using Eclipse EMF, and collaborated with the project Topcased (UML/SysML Editor) for the CNES. Passionate about the space domain, he became Simulation Officer for the Galileo Project, and was in charge of training the ESOC Mission Control Team for 7 Galileo launches. In 2017, he joined Airbus, Stevenage, U.K., where he is currently working as an Operations/FDIR Architect for the mission ExoMars. He also acts as the Coordinator for research and development activities for the Functional Avionics domain in Stevenage.