System-Level Design Tools Utilizing OPM and Modelica

Conference Paper · August 2016

5 authors, including:

Joshua Sutherland
The University of Tokyo

Kazuhiro Aoyama
The University of Tokyo

Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2016
August 21-24, 2016, Charlotte, North Carolina

DETC2016-60101

SYSTEM-LEVEL DESIGN TOOLS UTILIZING OPM AND MODELICA

Joshua Sutherland
The University of Tokyo
Tokyo, Japan

Kazuya Oizumi
The University of Tokyo
Tokyo, Japan

Kazuhiro Aoyama
The University of Tokyo
Tokyo, Japan

Takao Eguchi
Shinko Research Co. Ltd
Tokyo, Japan

Naoki Takahashi
Dassault Systèmes K.K.
Tokyo, Japan

ABSTRACT

This paper presents a methodology and tools to synthesize and assess different System-Level Designs. It utilizes the descriptive modeling language Object Process Methodology (OPM) to hierarchically describe the functionality of the System of Interest which is mapped by way of various intermediary models to an architecture in the Modelica numerical modeling language. The resulting Modelica architecture model is subsequently used as a framework for the rapid creation of alternative System of Interest designs by the variation of components.

To enable consistent assessment of the alternatives, Assessment Scenarios are created based on the functionality identified by OPM decomposition. By defining a hierarchy of Modelica models with the Assessment Scenarios at the top, all the System of Interest alternative models can be composed into the Assessment Scenarios and the resulting models simulated. With a combined score for each alternative across all the Assessment Scenarios being computed by way of Multi Objective Decision Analysis (MODA).

This paper demonstrates the approach with an actual student solar powered autonomous boat development project.

1. INTRODUCTION

Recently system design faces several new challenges, which are often characterized by large-scale complexity and rapid development. Modeling the system virtually prior to realizing it physically is seen by some [1] as playing a central role in addressing these challenges.

1.1. Solar-Boat Project Problems

As part of this research the identification of problems associated with real system development and potential solutions are explored by way of a case study of a student “design, build and race” solar powered boat competition (known herein as the Solar-Boat project). Where students are challenged to carry a small 64 gram payload over a 20km lake based course autonomously, powering their boat with a maximum of $2m^2$ of solar panels. An analysis of the 2014 University of Tokyo Solar-Boat project was conducted in [2]. Problems associated with the early lifecycle stages (LS) (i.e. where System-Level Design is conducted) are listed as: LS1 Clarify, slow time to acquire initial knowledge; LS2 Concept development, unclear as to what the design target was; LS3 System-Level Design, little exploration of alternatives and their predicted outcomes. As per [3] this research takes System-Level Design to “include the definition of the product architecture and the decomposition of the product into subsystems and components”.

As such the specific problems of the Solar-Boat project are found to be similar to the more general problems discussed in literature. When attempting to develop new systems, there is a need to quickly synthesize alternative designs and make assessments about them based on a prediction of their performance to an adequate level of accuracy.

1.2. Modeling Languages for Early Lifecycle Stages

Several languages have been developed to model systems based around two dominant approaches: descriptive and numerical.

Of Descriptive Modeling Languages System Modeling Language (SysML) is one of the most common. However, due to its wide coverage of multiple system aspects across multiple linked diagram types (structure on four diagrams, behavior on four diagrams and requirements on a single diagram), modelling requires much time and effort for even a single design alternative.

Object Process Methodology (OPM) [4] is an ISO standardized [5] conceptual modeling language which takes a significantly different approach to SysML in that all system aspects (i.e. structure and behavior) are displayed on a single diagram type rather than the nine linked diagrams of SysML. Multiple diagrams of the same system is encouraged in OPM but all use the same symbols. This simplicity when compared with SysML has made some suggest [6] that when compared to SysML OPM is more suitable for occasions where less detailed modeling is required. In OPM all systems consist of Objects (green rectangles) which are optionally stateful (yellow rounded rectangles inside), Processes (blue ovals) and relations between them. With an OPM Process defined in [5] as the transformation of one or more objects in the system and an OPM Object as a model element representing a thing that does or might exist physically or informatically [5]. In addition each diagram is accompanied by a textual language description (throughout this paper where space allows diagrams will include this text). FIGURE 1 provides a simple example for the Solar-Boat, where it is shown to consist of a Hull and Solar panels (structure) and enabling the process "Driving" (behavior). The "Driving" process affects the stateful attribute of the Solar-Boat "Speed" while consuming Solar Insolation. To manage the amount of detail displayed, hierarchy is employed, as such while a diagram such as FIGURE 1 does provide a compact description of a generic Solar-Boat, further detail can be revealed by the decomposition of the objects and processes in the model.

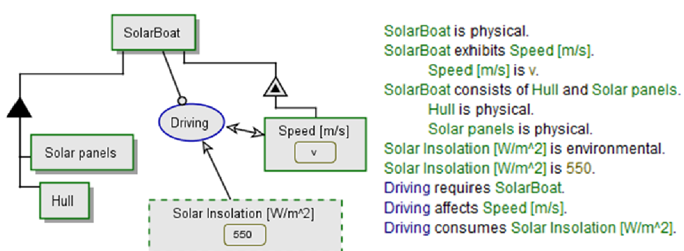

FIGURE 1. EXAMPLE OPM DIAGRAM (LEFT) AND OPM LANGUAGE (RIGHT) OF A SOLAR-BOAT.

OPM has been used to create executable simulations before in the form of stateful transitions whether it be based on petri nets [7] or the OPM model itself [8]. While past research with continuous numerical simulation [9,10] made use of MATLAB/Simulink to run the simulations for the purpose of gaining greater understanding of existing descriptive models.

Of Numerical Modelling Languages Modelica is gaining popularity. With Modelica behavior of individual components must be captured by equations. These components must subsequently be connected together to develop subsystems which ultimately form the system being attempted to be designed. Again this requires much time and effort as well. When considering rapid development projects, such time-consuming modelling needs to be avoided, but given ideally multiple alternative designs should be explored, the time and effort needed to create models becomes an even more important consideration.

1.3. Developing and Utilizing Early Stage Models

However the existence of the previously mentioned languages is not sufficient for them to be of value for early stage design, such languages must be supported by a methodology. Given System-Level Design is characterized by the need to answer a set of basic questions, the methodology should provide answers to them, as in: What functions does the system need to perform? How to assess the system performs these functions and to what level of performance? How to synthesize systems which can perform these functions? How to compare alternative systems which perform these functions?

1.4. Research Purpose

Based on the described problems this research's purpose is to develop tools and methodologies to enable:

- The synthesis of multiple alternative System-Level Designs.
- The synthesis of a common assessment criteria for the alternative System-Level Designs.
- The application of the common assessment criteria to the alternative System-Level Designs such that their ability to perform objectives can be assessed.

Specially to tackle these challenges, the methodology presented involves an attempt to integrate the descriptive modeling of Object Process Methodology (OPM) with the numerical modeling of Modelica, such that designs developed in Object Process Methodology (OPM) can be assessed numerically and Modelica model creation and simulation can follow a logical process. With the aim that both the descriptive and executable models can complement each other.

2. PROPOSED TOOLS AND METHODOLOGIES

At each of the Lifecycle Stages (LS) identified as problematic in Solar-Boat project a series of design questions are posed as shown in FIGURE 2. These are expanded as a more specific set of activities as shown in FIGURE 3 enabling the ability to move between the early lifecycle stages. The main focus of the LS1 Clarify is the identification of what value is derived from the system, how to measure this value and what resources are available to deliver it. LS2 Concept Development focuses on a decomposed functional description of what every system alternative must do and how to assess the alternatives. While LS3 System-Level Design aims to develop specific alternative designs predict their performance by numerical simulation and compare.

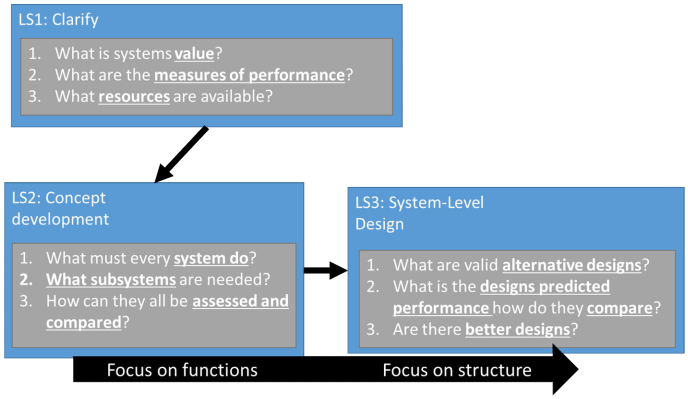

FIGURE 2. DESIGN QUESTIONS POSED AT DIFFERENT LIFECYCLE STAGES (LS).

The broad set of activities displayed in FIGURE 3 are used to define a framework enabling the logical creation of conceptual models of the system to be designed and how to assess the said system, in OPM. These conceptual models are subsequently and logically transferred to numerical models in Modelica by way of multiple model types (all of which have hierarchical decomposition) that are ultimately simulated. To achieve this, the proposed design methodology:

- Utilizes OPM to decompose the functionality required of the System of Interest and explore concepts.
- Defines a framework for mapping functional descriptions (in OPM) to a formalized common architectural forms which can deliver the functionality. Alternative designs can then be developed on the common architecture (and ultimately be simulated in Modelica).
- Utilizes by way of the INCOSE Decision Management Process [11,12], Multi Objective Decision Analysis (MODA) [13] such that all the alternative designs can be quickly compared to one another.

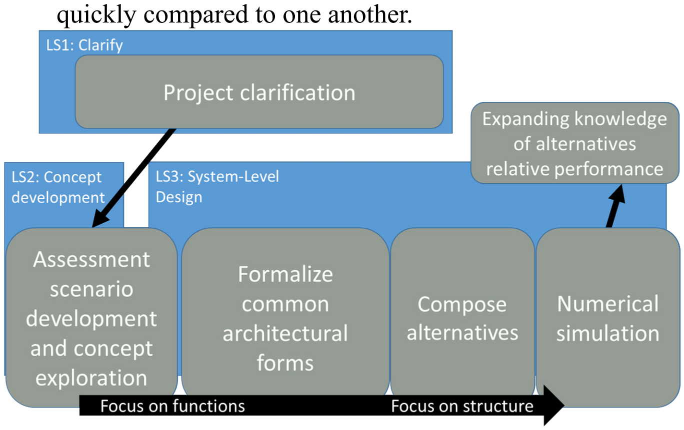

FIGURE 3. PROPOSED METHODOLOGY HIGH LEVEL FLOW. AT LIFECYCLE STAGES (LS IN BLUE), VARIOUS HIGH LEVEL ACTIVITIES ARE IDENTIFIED (IN GREY).

FIGURE 4 provides a breakdown of the broad activities of FIGURE 3 into a specific set of inputs/outputs (in green) which are linked together by specific transitions (labels on the links). The letters indicate a step in the process. To aid comprehension subsequent description of the methodology will make reference to the Solar-Boat project and many terms will be introduced on diagrams now to only be explained later in the paper.

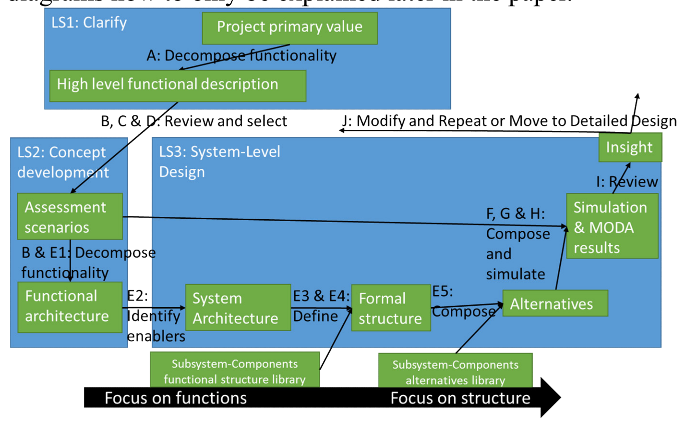

FIGURE 4. PROPOSED METHODOLOGY INPUTS/OUTPUTS (IN GREEN) AND ACTIVITIES (LABELS ON THE LINKS, (LETTERS REFERENCED LATER IN TEXT) AT VARIOUS LIFECYCLE STAGES (IN BLUE).

FIGURE 5 which depicts a representation of the various model types advocated in the proposed methodology at various levels of hierarchy makes reference to the Solar-Boat as an example (black triangle indicating the higher level object or process consists of those at the lower level) in addition FIGURE 6 provides a snapshot of examples of the models at each level of hierarchy. In subsequent sections the core concepts which enable this methodology are explained. These core concepts are:

- The use of Assessment Scenarios (hierarchy Level 1) to consistently compare Alternative Designs.
- Using a distinct set of Diagram Types to describe the System of Interest from various ways in both OPM and Modelica.
- Decomposition of the System of Interest into a distinct hierarchy and applying this hierarchy to each of the Model Types.

Names of the various transitions to complete decomposition to lower hierarchical levels, mapping between different model types and composition to higher hierarchical levels are provided in FIGURE 7 with letters indicating a step in the process which is consistent with FIGURE 4 and explained in the subsequent sections. The core transitions are listed as follows and explained in the subsequent sections:

- Converting between a process centric model to an object centric model
- Converting from OPM Formal Structure to Modelica Formal Structure
- Composing Alternatives rapidly by Modelica block replacement

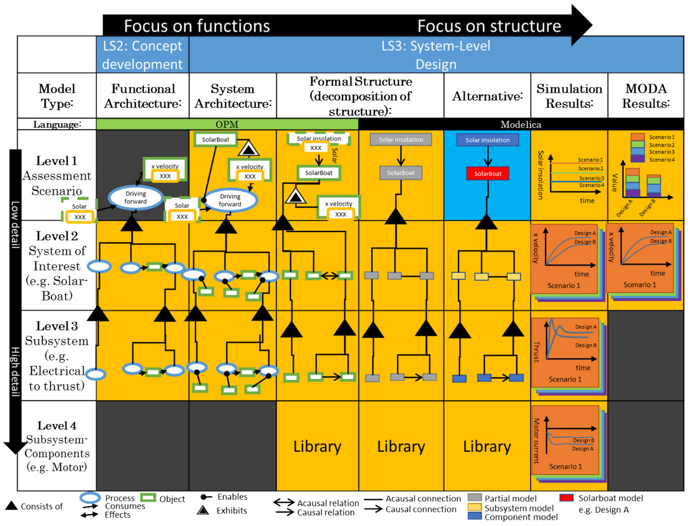

FIGURE 5. A REPRESENTATION OF DIFFERENT MODEL TYPES AT VARIOUS HIERARCHY LEVELS.

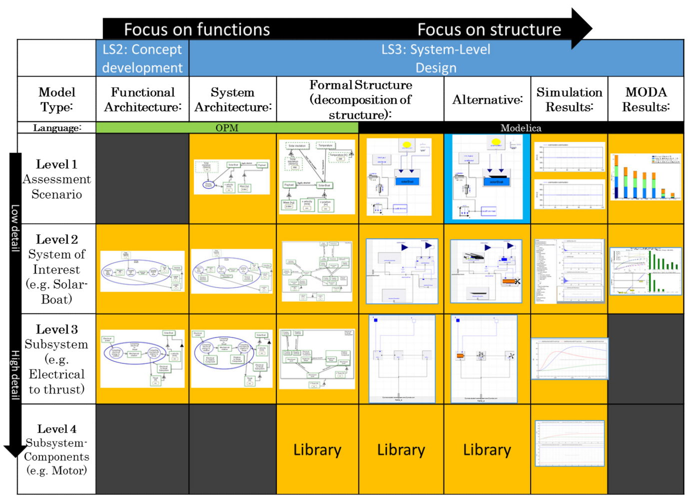

FIGURE 6. EXAMPLES OF THE DIFFERENT MODEL TYPES.

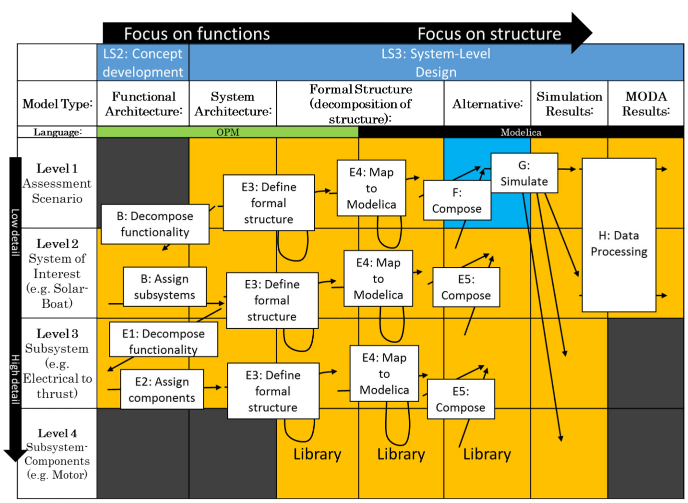

FIGURE 7. TRANSITIONS TO MOVE BETWEEN THE DIFFERENT MODEL TYPES AND THE HIERARCHY LEVELS.

2.1. Assessment Scenarios and Comparing Alternative Designs

Assessment Scenarios provide a consistent way to assess each alternative design's performance. For a Solar-Boat, examples could be "Floating" or "Straight line driving". To aid decision making the simulation results of multiple Assessment Scenarios are combined to give an overall assessment of performance, which can be compared for each alternative design.

To enable this two concepts are required: Models to describe the Assessment Scenarios (which design alternatives can be subject to) and a method to consolidate the results.

Assessment Scenarios are at the top of the hierarchy (Level 1) of the Modelica models as they represent the environment in which the design is placed for simulation (as per FIGURE 5). The system being designed sits below Assessment Scenario at Level 2. As such to simulate a particular design in a particular Assessment Scenario the models must be combined. This is depicted in FIGURE 8 where an alternative design "System of Interest A" is placed on the hierarchy below "Assessment Scenario A" such that the subsequent model when simulated represents "System of Interest A" subject to "Assessment Scenario A" is created, after which the resulting Modelica model is ready for simulation.

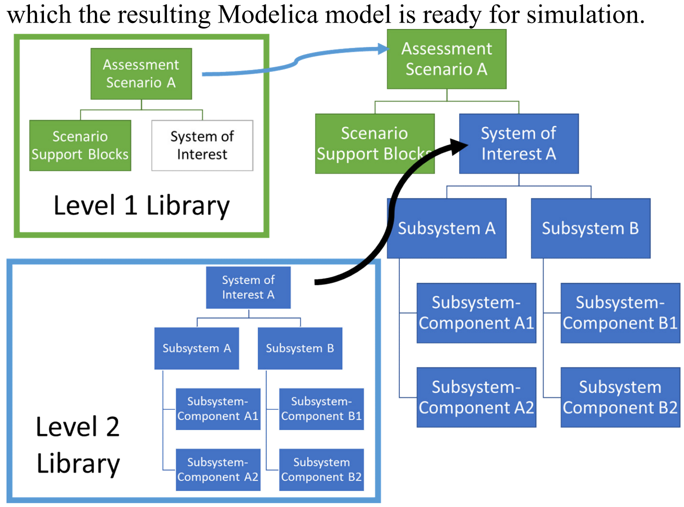

FIGURE 8. DEPICTION OF A LEVEL 2 MODEL BEING PLACED IN A LEVEL 1 ASSESSMENT SCENARIO.

Combining Simulation Results is undertaken to avoid information overwhelm and enable decision making between competing designs. Information overwhelm occurs from three sources: The simulation results from simulating a single Modelica model is a time series for every variable in every equation which describes the model; multiple alternative designs and multiple Assessment scenarios.

To combine these simulation results a simple weighted value sum is used as advocated by [13], the scheme is selected for its simplicity of implementation. And while such a scheme might not be appropriate for all decision making situations, for this research it was deemed adequate.

Implementation for this research involves extraction from each Assessment Scenario and design alternative pair simulated, a measure of the System of Interest's performance (e.g. max x_velocity) from the raw simulation results. The extracted variable is then used to compute an unweighted value by the utilization of the value function (for an example see [14] FIGURE 11). The unweighted value is then multiplied by the weight assigned for that scenario (weighted value). Summing all the weighted values for each scenario for each alternative results in the total weighted value. With the "ideal system" having a total weighted value of 1 as its performance is assumed to always be at the stretch goal. As such the total value can be presented broken down by Assessment Scenario. A representation of this is presented in Level 1 of FIGURE 5 while full examples are presented in the usage examples in FIGURE 30 and FIGURE 32 towards the end of this paper.

To accomplish this, each Assessment Scenario requires a definition of Value Function, Weight, Simulation Length and Data Extraction Method to enable the later MODA computation combining the results of multiple Assessment Scenarios for each Alternative. With the value function defining a mapping between the attribute of interest to value space (e.g. FIGURE 11 of [14]). Simulation Length indicates how long to simulate the Assessment Scenario. Data Extraction Method describes how to extract one value from the time series e.g. mean, maximum or minimum. The weight is used to combine the results of multiple Assessment Scenarios for one design Alternative.

2.2. Model Types

As stated earlier in this paper in the Research Purpose is to tackle the challenge of synthesizing and assessing new system designs by the combination of conceptual modeling (OPM) and numerical simulation (Modelica). However given both are rich descriptive languages it is inadequate not to provide a clear set of models types which the modeler should create such that a methodology linking the said model types can later be described. A caricature of each of the model types used is provided in FIGURE 5, each of which are explained in the subsequent subsections:

Functional Architecture based on that proposed by [15] aims to be a purely process and operand description of the item for modeling. This model type is used at the inception of the solution as it encourages the engineer to describe the intended functionality of the system of interest rather than immediately describing the form of the solution. As such when represented in OPM, objects are only those which are affected by a process, not those which enable a process; therefore there is no indication of what objects are needed to implement the system (form) and as such no Modelica model can exist. To provide an example, the caricature of FIGURE 5 Level 1 is expanded as FIGURE 9, it is possible to see that the processes "Driving forward" consumes "Solar insolation" and affects "x velocity" but there is no indication to the form which enables this function. Given the design target of a Solar-Boat is assumed throughout this paper this particular model does not strictly exist hence the dark background on FIGURE 5 Level 1, FIGURE 23 (top) provides an additional example at Level 2 hierarchy.

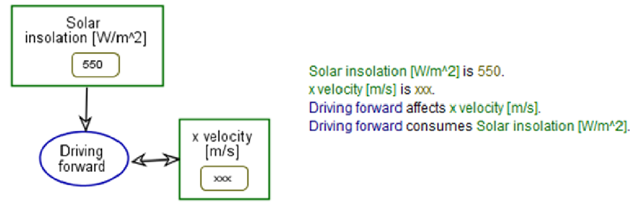

FIGURE 9. EXAMPLE FUNCTIONAL ARCHITECTURE (LEVEL 1 SOLAR-BOAT).

System Architecture as per that proposed by [15] enables the identification of form which will be required to deliver the functionality identified and described in the Functional Architecture. This occurs by the assignment of objects to enable the processes (functional requirements of the assigned objects is the Functional Architecture previously described). For this research an assumption is made that each process is enabled by an individual object forming a process and object pair in the System Architecture diagram. Further, for the proposed methodology, the name given to the enabling object is assumed to be used to categorize a library of components, with the name corresponding to the input, output and processing performed by the object. In FIGURE 5 Level 1 a representation is presented which is expanded as FIGURE 10. The "Driving forward" process is shown to be enabled by the "Solar-Boat" object, which is shown to be the exhibiter of the "x velocity" an additional example is provided in FIGURE 23 (bottom) for Level 2 hierarchy.

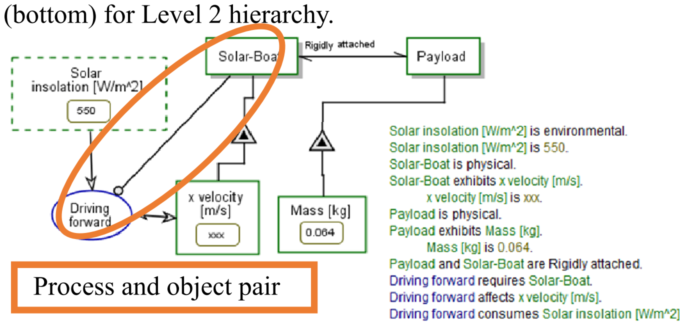

FIGURE 10. EXAMPLE SYSTEM ARCHITECTURE (LEVEL 1 SOLAR-BOAT PROCESS OBJECT PAIR HIGHLIGHTED).

Formal Structure is presented in this research in two separate modeling languages: OPM and Modelica. While the Formal Structure presented in [15] is System Architecture without the representation of processes, in this research the representation is somewhat more specific as its purpose is to enable the creation of a Modelica model. As such the OPM Formal Structure intends to map out the connections which will later be implemented in Modelica. For Modelica, Formal Structure is taken to mean a model which defines the interfaces exposed by the model components, the connections between such interfaces and the public accessibility of the measure of interest. By utilizing Modelica's object-oriented replaceability features such an interface based model can enable multiple alternative designs sharing common architectures. Such replacement is of Modelica replaceable partial components.

To aid the creation of the Formal Structures in Modelica, initially a Formal Structure is created in OPM which is subsequently mapped to Modelica. No processes are displayed in the model as processes are realized by the behavior of the components (which are realized by the equations of in Modelica blocks). However due to the focus on Modelica model creation in the Formal Structure OPM model, interfaces exposed by model components, the connections between such interfaces and the public accessibility of attributes of interest are explicitly modeled.

FIGURE 5 Level 1 presents a caricature of both OPM and Modelica Formal Structures which are expanded as FIGURE 11 and in [14] the left side of FIGURE 4 respectively. It should be noted that no longer is the "Driving forward" process displayed as this behavior is contained in the enabling object "Solar-Boat" as such the "Solar insolation" is now directly connected to the "Solar-Boat" enabling object (not the process). For the purposes of mapping to Modelica the relations between objects is as such: single headed relation between an object represents a causal connection and a double headed fishhook relation is an acausal connection.

The Modelica Formal Structure implements this same structure as the OPM model. Where the Solar-Boat block is a Modelica replaceable partial component. Attributes of interest "x velocity" and "z position" exist as variables of the Solar-Boat model. Further the named relations between objects in OPM are now specific connection types in Modelica. In [14] FIGURE 4 left an example is provided.

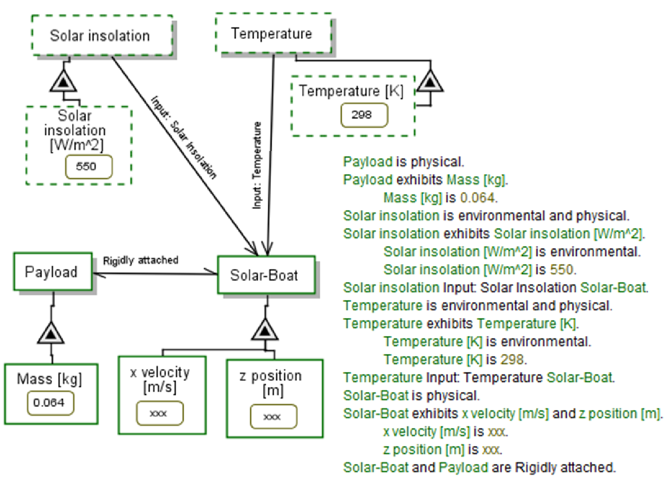

FIGURE 11. EXAMPLE FORMAL STRUCTURE – OPM (LEVEL 1 SOLAR-BOAT).

Alternative is for this research a fully implemented Modelica model (no use of partial). It is required such that simulation can actually be performed, as such at Level 1 this represents an Assessment Scenario and System of Interest Alternative Pair (depicted in FIGURE 8) while low level models represent subsystems and components which are integrated to form a model for simulation. Coloring is used in FIGURE 5 to represent the Formal Structure Modelica replaceable partial components (in grey) being replaced with Modelica components (colored) which can be viewed in more detail in [14] FIGURE 4 (right) where the Modelica replaceable partial component of a Solar-Boat has been replaced with an alternative design. In addition FIGURE 13 provides a representation of this composition.

2.3. Hierarchy

Hierarchy is used across all the model types (as shown in FIGURE 5) to manage complexity. It enables the grouping of information and thus the suppression of detail. How this manifests depends on the model type being considered, but to ensure consistency the same hierarchy levels on all the model types is used (as shown in FIGURE 5). To illustrate this more clearly an example hierarchy for the Solar-Boat is provided in TABLE 1, which shows the decomposition of objects and the processes they enable simultaneously.

TABLE 1. THE HIERARCHY USED.

Level	Name	Example object	Enabled processes
0	Functional Architecture – Primary Value	SolarBoat Race	
1	Assessment Scenario	Assessment scenario	
2	System of Interest	Solar-Boat	Driving forward, Floating
3	Subsystems	Electrical to Thrust subsystem	Converting Electrical to Thrust
4	Subsystem-Components	DC Motor	Converting Electrical to Rotation

For Functional Architecture (and Similarly System Architecture) Hierarchy given the models focus on the processes performed by the system, the hierarchy is based on the decomposition of these processes, enabling large monolithic processes to be decomposed into smaller processes which are more applicable for modeling in Modelica. An example of one step of such functional decomposition of a Functional Architecture is provided in FIGURE 12 where the large monolithic process of "Driving forward" is decomposed into "Displacing less dense volume", "Converting solar to electrical" and "Converting electrical to thrust".

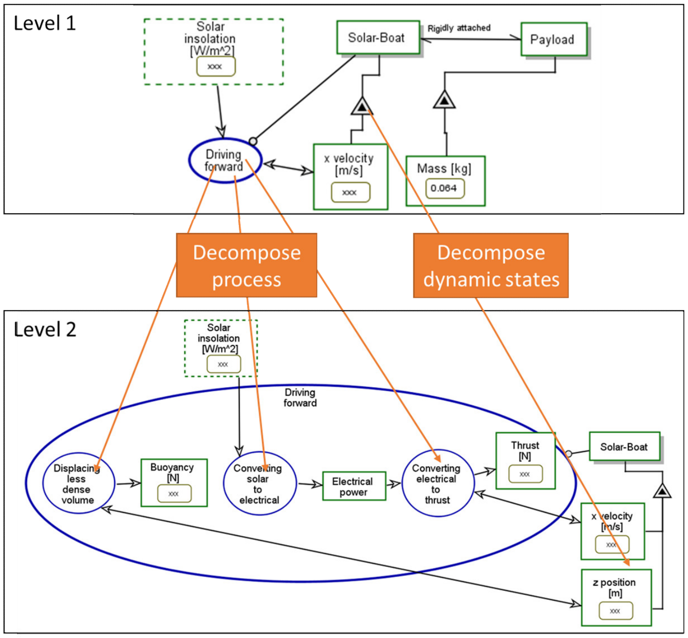

FIGURE 12. LEVEL 1 TO LEVEL 2: DECOMPOSING THE SYSTEM ARCHITECTURE OF "DRIVING FORWARD".

Formal Structure (and Similarly Alternative) Hierarchy focuses on the structure needed to perform the processes. As such the hierarchy is based on the decomposition of objects, enabling a system of interest to be described in terms of subsystems which intern are described as a collection of subsystem-components. FIGURE 13 provides a depiction of an Alternative being composed in Modelica by a bottom up approach on a Formal Structure.

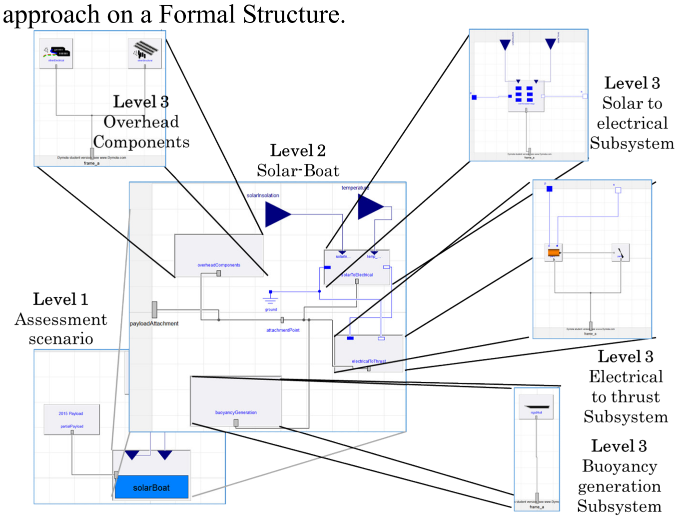

FIGURE 13. COMPOSING AN ALTERNATIVE SYSTEM OF INTEREST (LEVEL 2) WHICH IS THEN PLACED INTO AN ASSESSMENT SCENARIO (LEVEL 1).

2.4. Converting System Architectures to Formal Structures

To create a Modelica model which can later be simulated a Formal Structure (defined previously in this paper) must be created such that alternatives can be composed on to it. The Formal Structure is used to define the layout of the objects which are used to enable the functions and describe how objects interact with each other (i.e. what interfaces are to be used). Further there must be assurance of the exposure of the measure of interest for the Assessment Scenario such that performance can be assessed.

The Formal Structure is created from the System Architecture directly, by mapping between these two models the hierarchy remains intact and less knowledge is needed about creating a Formal Structure model as the modeler can be guided from the System Architecture representation.

An example System Architecture for a Level 3 Subsystem is provided in FIGURE 14. It is inadequate for direct Modelica modeling for the following reasons:

- OPM procedural links have been used to connect objects and processes implying causal relationships (e.g. "Mechanical rotation" is consumed by "Converting mechanical rotation to thrust"). A major benefit of the Modelica language is the handling of acausal relationships. As such a representation with these should be developed.
- The OPM processes although descriptive are manifested as equations within the object which enables them (as shown in the ringed processes and object pairs of FIGURE 14).

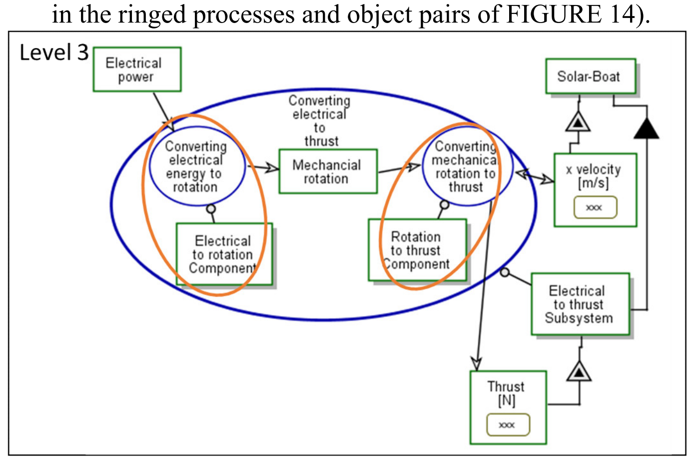

FIGURE 14. LEVEL 3: SYSTEM ARCHITECTURE OF "ELECTRICAL TO THRUST" SUBSYSTEM.

When compared to the Formal Structure representation in OPM (top right of FIGURE 15) it can be seen that the links between the objects are now labeled with the connection type, processes are no longer shown and the layout has been fixed. The focus has shifted from on the processes (System Architecture) to one of structural layout and interaction (Formal Structure).

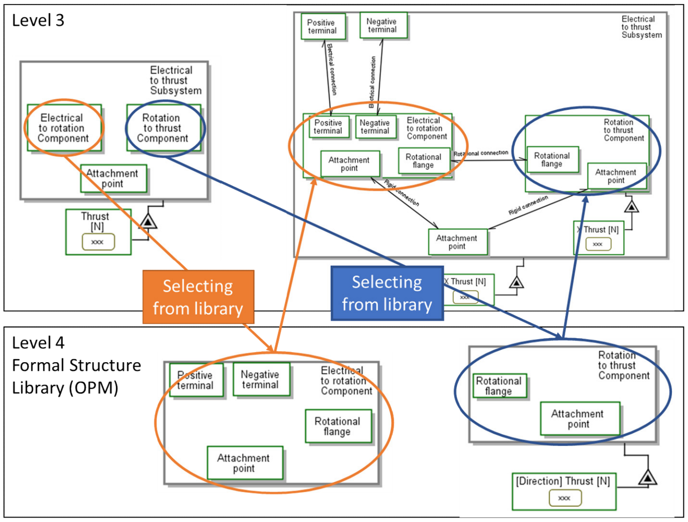

FIGURE 15. LEVEL 3: COMPOSING A FORMAL STRUCTURE OF THE “ELECTRICAL TO THRUST” SUBSYSTEM OUT OF LEVEL 4 SUBSYSTEM-COMPONENTS.

A procedure is now described of how a Formal Structure can be created from a System Architecture. As per FIGURE 7 (Step E3, System Architecture to Formal Structure) the process of converting any particular hierarchy level requires reference to the level below it, as such this procedure should be conducted from the bottom of the hierarchy to the top. In the following example the Level 3 System Architecture of FIGURE 14 is converted into an OPM Formal Structure.

Identification of Formal Structure of the Level Below initially occurs such that lower level models can be incorporated into the Formal Architecture being created. This occurs by selecting each OPM process and object pair of the System Architecture (ringed on FIGURE 14) and comparing to existing Formal Architectures in the hierarchy level below, keyed on the process and object pairs inputs and outputs. This lower level Formal Architecture is then selected and incorporated into the higher level model. As such for creation of Level 3 Formal Structures there is an assumption that there is a sufficient library of Subsystem-Components described as Formal Structure models.

Initial Placement of Level below Formal Structures is conducted by taking the object which enables the high level process decomposing it, with the lower level Formal Architectures added to the in-zoomed object (shown in FIGURE 16). In the case of Solar-Boat it is assumed it is a rigid body with all Subsystem-Components rigidly connected to a single position in the Subsystem (and similarly on the higher levels), therefore a rigid “Attachment point” is added.

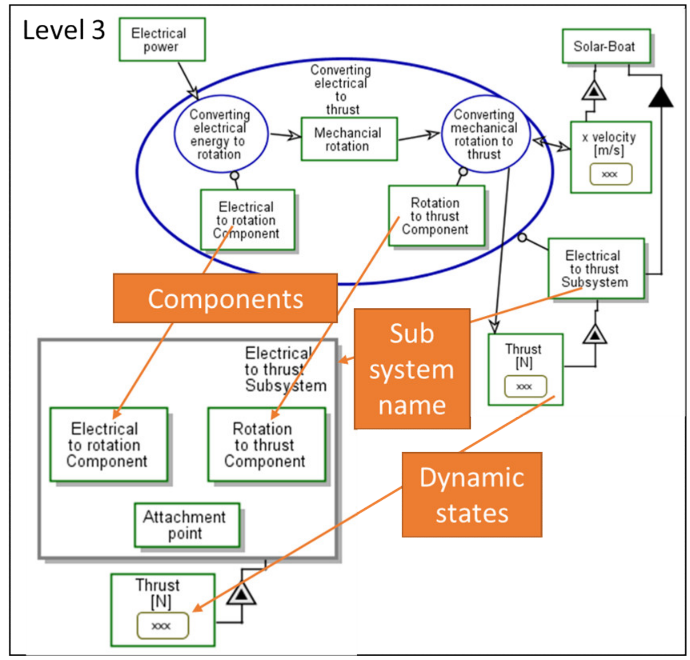

FIGURE 16. LEVEL 3: INITIAL FORMAL STRUCTURE FOR “ELECTRICAL TO THRUST” SUBSYSTEM.

Interfaces of the Level below Formal Structure must now be identified. Additional external interfaces of the process and object pairs must be compared to the library, as each pair is to be represented by a single object in the Formal Structure diagram. This is depicted in FIGURE 17 and FIGURE 18. In FIGURE 17 the process “Converting electrical energy to rotation” from FIGURE 14 is reviewed and Formal Structures from the Level 4 Subsystem-Component library is selected which define the interfaces of the Subsystem-Component (two electrical pins and a rotational flange). The “Converting mechanical rotation to thrust” process which is enabled by the “Rotation to thrust component” is different as thrust is a force which acts on the entire rigid body rather than being connected to a connection. As such the existing attachment point and a rotational flange is sufficient (with a configuration to which direction the thrust is acting is) as shown in FIGURE 18.

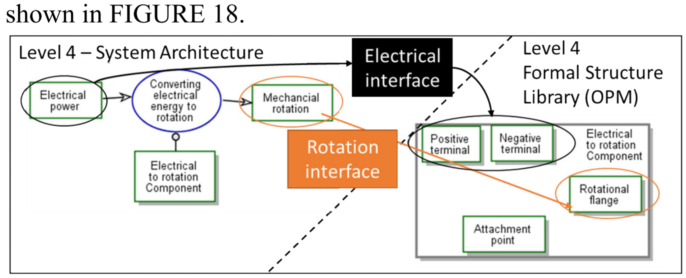

FIGURE 17. “CONVERTING ELECTRICAL ENERGY TO ROTATION” LEVEL 3 SYSTEM ARCHITECTURE OBJECT PROCESS PAIRS BEING COMPARED TO THE LEVEL 4 LIBRARY OF SUBSYSTEM-COMPONENTS.

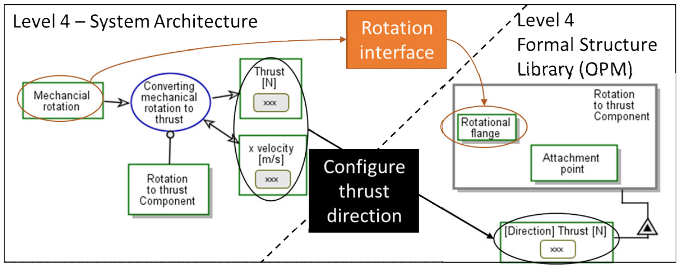

FIGURE 18. LEVEL 3 SYSTEM ARCHITECTURE OBJECT PROCESS PAIRS BEING COMPARED TO THE LEVEL 4 LIBRARY OF SUBSYSTEM-COMPONENTS.

Composition of the Formal Structure from the Level Below can be completed, where relational links are defined to be causal (single headed arrow) or acausal (double headed fish hook). With the Subsystem-Components being connected together based on the flow defined in the System Architecture (FIGURE 14) this is depicted in FIGURE 15.

2.5. Converting OPM Formal Structure to Modelica Formal Structure

As described in the Research Purpose the aim of this research is to create Modelica models which can be used to assess alternative designs. To enable this an architecture of replaceable models is created in Modelica (known in this research a Modelica Formal Structure). This is created directly from the OPM Formal Structure to reduce the resources needed to create a Modelica model.

A procedure of how to conduct this is provided as follows, it makes use of the previous example to provide illustration. Once a new empty model has been created, the interfaces for the Formal Structure must be defined in Modelica as depicted in FIGURE 19. This involves selecting the appropriate connector for the model in Modelica, as such defining in Modelica the interface for the Subsystem. In addition any attributes of interest (x Thrust in FIGURE 19) are set as publically assessable variables. Then, as depicted in FIGURE 20 to create a Modelica model for the architecture defined in the OPM Formal Structure the selection of the appropriate Modelica replaceable partial models from the level below is required.

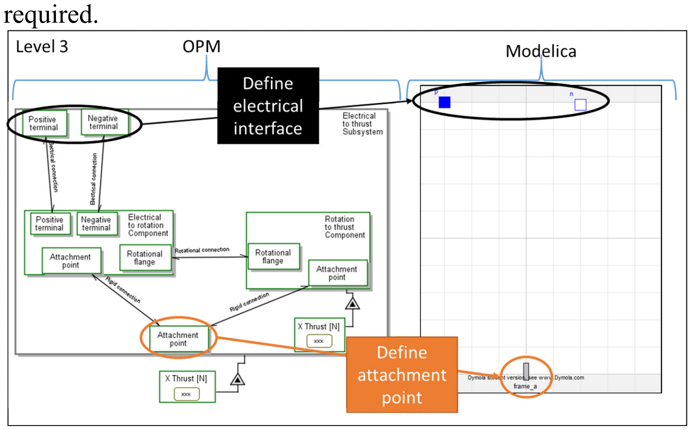

FIGURE 19. LEVEL 3: DEFINING CONNECTORS FOR A SUBSYSTEM IN MODELICA FORMAL STRUCTURE (RIGHT) FROM OPM FORMAL STRUCTURE (LEFT).

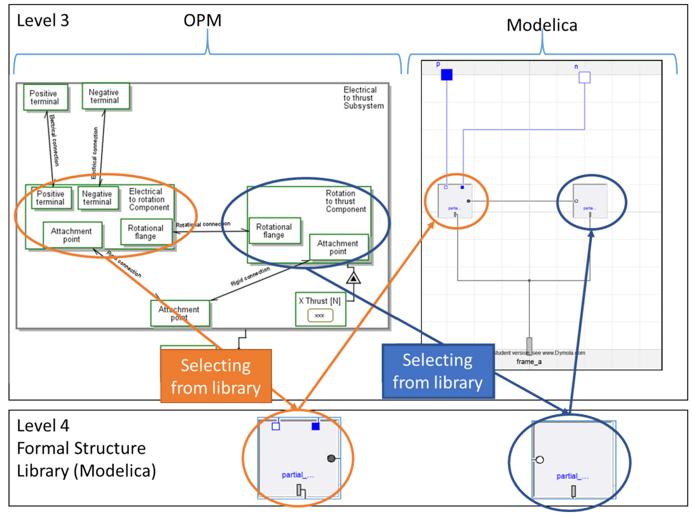

FIGURE 20. LEVEL 3 FORMAL STRUCTURE: SELECTING MODELICA REPLACEABLE PARTIAL COMPONENTS FROM THE SUBSYSTEM-COMPONENTS LIBRARY.

Alternative Composition by Modelica Block Placement enables the rapid creation of alternative designs by placing alternative implementations of the blocks on the Modelica Formal Structure this is represented in FIGURE 8 and FIGURE 13.

3. STEP-BY-STEP METHODOLOGY

This section presents a methodology to enable the reader to synthesize and assess competing System-Level Designs utilizing the tools and core concepts described previously in this paper. The individual steps are explicitly explained to holistically piece together the individual concepts presented previously.

3.1. Overview

The steps to realize the flow are listed explicitly and are the same as those presented in FIGURE 4 (referencing the letters on that diagram), further if the step is additionally represented on FIGURE 7 this is also noted for reference. Detail of the steps including diagrams of examples for Solar-Boat and definitions of terms are in subsequent sections of this paper:

Step A: Identifying and Decomposing Functional Architecture – Primary Value.

Step B: Identifying Subsystems Required for Modelling the System of Interest (FIGURE 7).

Step C: Reviewing and Selecting Assessment Scenarios.

Step D: Configuring Specific Assessment Scenarios:

- **Step D1:** Varying Assessment Scenario inputs.
- **Step D2:** Defining each Assessment Scenario's Value Function, Weight, Simulation Length and Data Extraction Method.

Step E: Synthesizing System of Interest Designs in OPM and Modelica:

- **Step E1:** System of Interest Process Decomposing (FIGURE 7).

Step E2: Assigning Subsystem-Components to Enable the Processes (FIGURE 7).

Step E3: Mapping to a Formal Structure in OPM (FIGURE 7).

Step E4: Mapping to a Formal Structure in Modelica (FIGURE 7).

Step E5: Composing Alternatives in Modelica (FIGURE 7).

Step F: Composing each System of Interest Alternative into each Assessment Scenario for Simulation (FIGURE 7).

Step G: Simulating every Assessment Scenario and System of Interest Alternative Combination (FIGURE 7).

Step H: Consolidating Simulation Results with MODA (FIGURE 7).

Step I: Reviewing Results.

Step J: Modify and Repeat or Move to Detailed Design.

3.2. Explanation by Example

A simple usage example for the Solar-Boat is presented which expands the methodology steps and the models presented previously in this paper as part of the description of the methodology.

Step A: Identifying and Decomposing Functional Architecture – Primary Value diagram (known as System Diagram in [4,5]) is drawn to initially define the most high level value deriving functionality of the system as shown in FIGURE 21. It has a single process "Racing in Solar-Boat Race Event" which is enabled by the System of Interest (Solar-Boat) and the attributes which are to be varied by the process (Solar-Boats dynamic states and the Race ranking). Additionally weather is an input to the process.

However such a process lacks detail to create valid alternatives for assessment. As such the primary value process is decomposed into sub-processes which represent the functionality all valid alternative System of Interest designs are expected to perform. This is represented in FIGURE 22 where the process "Racing in Solar-Boat Race Event" is decomposed into "Floating", "Driving forward", "Turning", "Handling disturbance" and "Ending race". Each of these indicate by an effects link which dynamic states of Solar-Boat are affected by the process (to avoid diagram clutter only those for "Floating" (z position [m]) and "Driving forward" (x velocity [ms^{-1}]) are displayed).

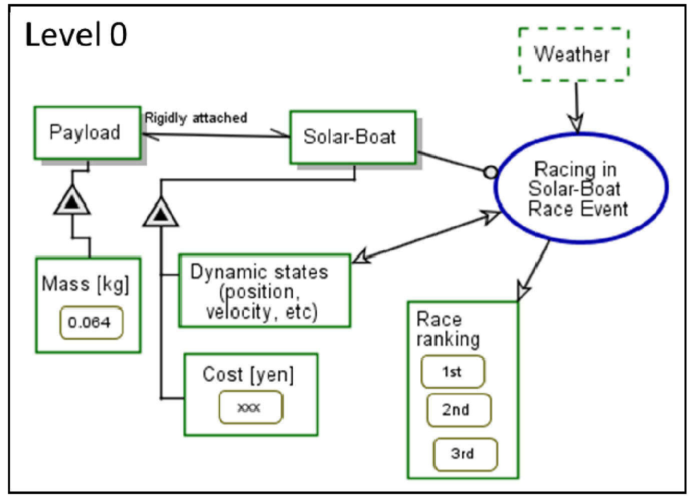

FIGURE 21. LEVEL 0: FUNCTIONAL ARCHITECTURE – PRIMARY VALUE FOR SOLAR-BOAT.

Step B: Identifying Subsystems Required for Modelling the System of Interest is dependent on the purpose of the modeling activity different processes in the decomposed project primary value (FIGURE 22) may or may not be appropriate for modeling (at potentially different levels of detail). For example if only a simple Solar-Boat prototype is to be developed it might be appropriate to only model the "Floating" and "Driving forward" processes. However these processes are not of sufficient detail for modeling so should be further decomposed. An example of such functional decomposition of "Driving forward" process into "Displacing less dense volume", "Converting solar to electrical" and "Converting electrical to thrust" is displayed in FIGURE 12. But to define what Subsystems must be implemented, objects must be assigned to enable the processes and as such develop a System Architecture. An example is shown in FIGURE 23 where "Buoyancy generation", "Solar to electrical" and "Electrical to thrust" Subsystems have been defined. Similar decomposition and subsystem assignment can occur for the other processes. Decomposition of "Floating" results in the "Displacing less dense volume" process and therefore the need for the "Buoyancy generation" Subsystem. FIGURE 24 subsequently consolidates the identified Subsystems and attributes of interest of the System of Interest (Solar-Boat). All alternatives are expected to implement all of these.

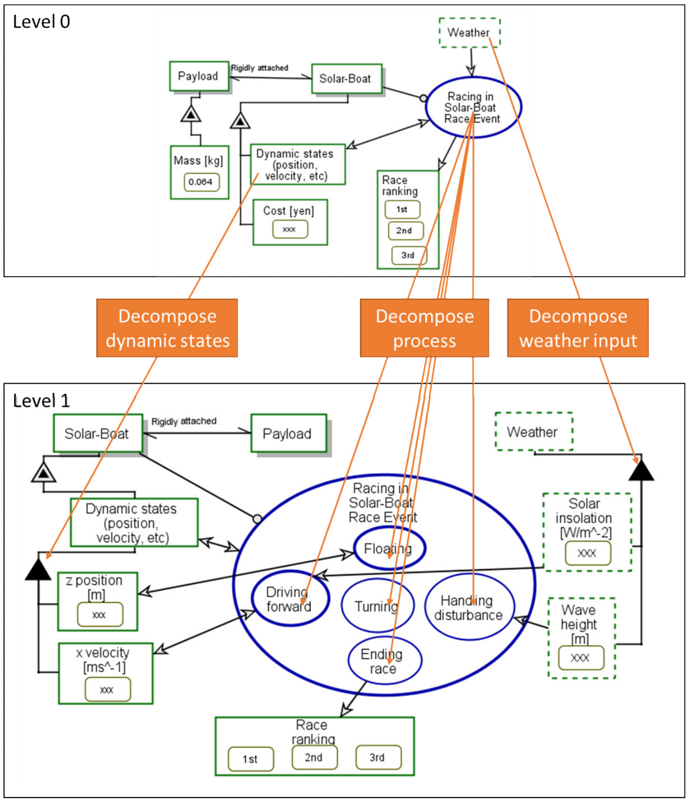

FIGURE 22. LEVEL 0 TO LEVEL 1: DECOMPOSING THE PROJECTS PRIMARY VALUE INTO SUB-PROCESSES.

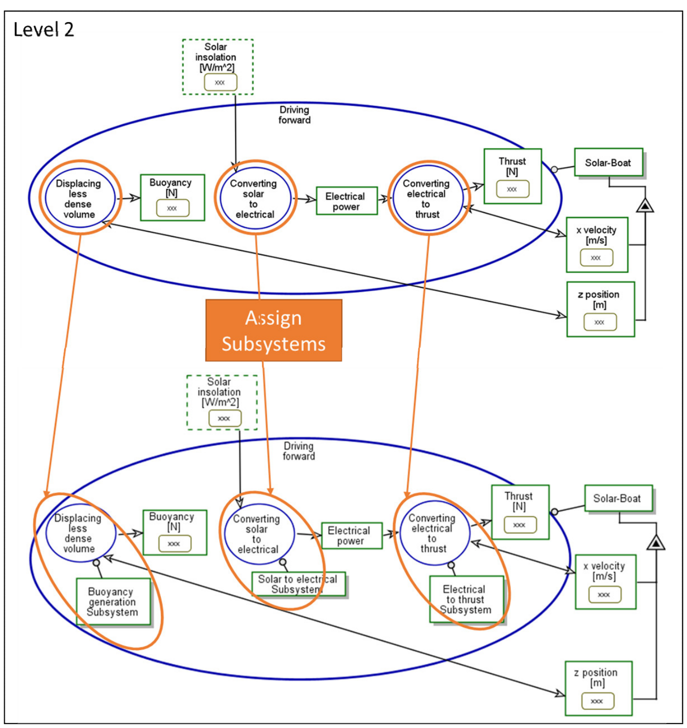

FIGURE 23. LEVEL 2: SYSTEM ARCHITECTURE FOR THE DECOMPOSED "DRIVING FORWARD" PROCESS.

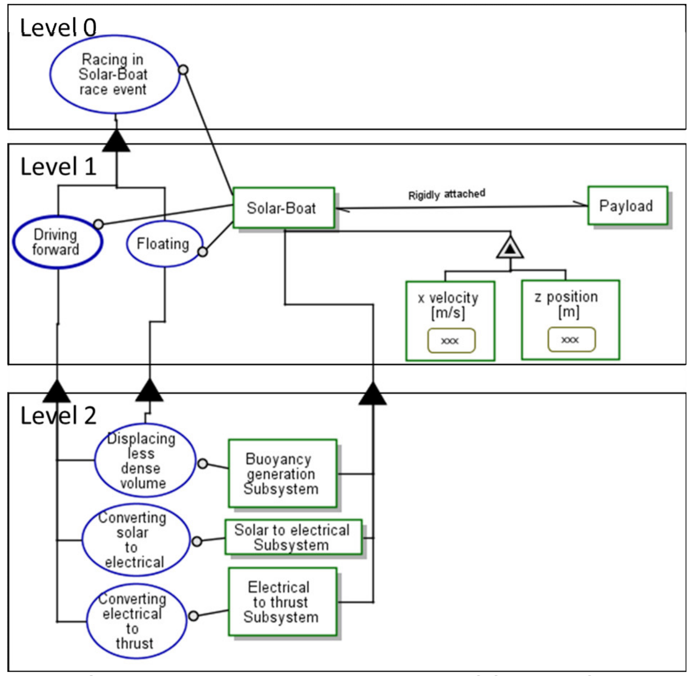

FIGURE 24. LEVEL 1 AND LEVEL 2: SOLAR-BOAT SUBSYSTEMS AND ATTRIBUTES OF INTEREST.

Step C: Reviewing and Selecting Assessment Scenarios is used to develop a method to assess the System of Interest in completing the required functions. Selection amongst the processes which were decomposed in Step B should be made to develop Assessment Scenarios which are to be used to assess the System of Interest. In this case it is assumed "Floating" and "Driving forward" are selected. FIGURE 25 displays this selection, including the attributes of the System of Interest varied by the Assessment Scenario processes and their inputs.

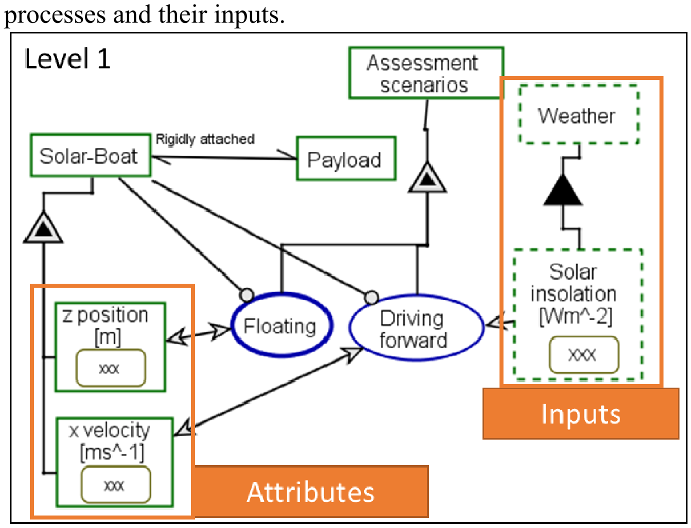

FIGURE 25. LEVEL 1: SELECTED ASSESSMENT SCENARIOS.

Step D: Configuring Specific Assessment Scenarios must be completed such that simulation models can be created, which is explained in the following steps.

Step D1: Varying Assessment Scenario inputs involves defining explicitly the values in the inputs to the process (in the running example this would be the specific solar insolation conditions) for the purpose of later creating a specific model to simulate. For the "Floating" Assessment Scenario no variation is required. For the "Driving Forward" Assessment Scenario the input solar insolation is varied. As show in the first column of TABLE 2.

Step D2: Defining each Assessment Scenario's Value Function, Weight, Simulation Length and Data Extraction Method is done for the later MODA computation combining the results of multiple Assessment Scenarios for each Alternative. With the value function defining a mapping between the attribute of interest to value space (e.g. FIGURE 11 of [14]). Simulation Length indicates how long to simulate the Assessment Scenario. Data Extraction Method describes how to extract one value from the time series e.g. mean, maximum or minimum. An assigned weight to the Assessment Scenario is used to combine the results of multiple Assessment Scenarios for one design Alternative. For the running example the last four columns of TABLE 2 are given where for simplicity all Assessment Scenarios are assumed to have an equal weight (i.e. 0.25). Minimum acceptable performance and stretch goals can be set based on past experience and required performance.

TABLE 2. SPECIFIC ASSESSMENT SCENARIOS.

Assessment Scenario name	Measure of interest	Min acceptable perform	Stretch goal	Sim time (s)	Data Extract
Floating	z position (m)	-0.1	-0.4	70	Mean
Best ever insolation (870 Wm^2) straight line driving	x velocity (m/s)	2	4	3	Max
Average insolation (550 Wm^2)	x velocity (m/s)	1.5	3	3	Max
Worst ever insolation (260 Wm^2)	x velocity (m/s)	0.5	2.5	3	Max

Step E: Synthesizing System of Interest Designs in OPM and Modelica is enabled by making use of the hierarchy and a defined set of model types in OPM and Modelica.

Step E1: System of Interest Process Decomposing requires the decomposition of functionality from the System Architecture of the System of Interest which was created in Step B. The purpose of Step E1 is to decompose the functionality of Subsystems such that alternative implementations of the Subsystems can be identified. An example of this is shown in FIGURE 26 where the "Converting electrical to thrust" process is decomposed into "Converting electrical energy to rotation" and "Converting mechanical rotation to thrust". The other System of Interest processes "Displacing less dense volume" and "Converting solar to electrical" are decomposed into a single processes each such that the hierarchy's consistency is maintained.

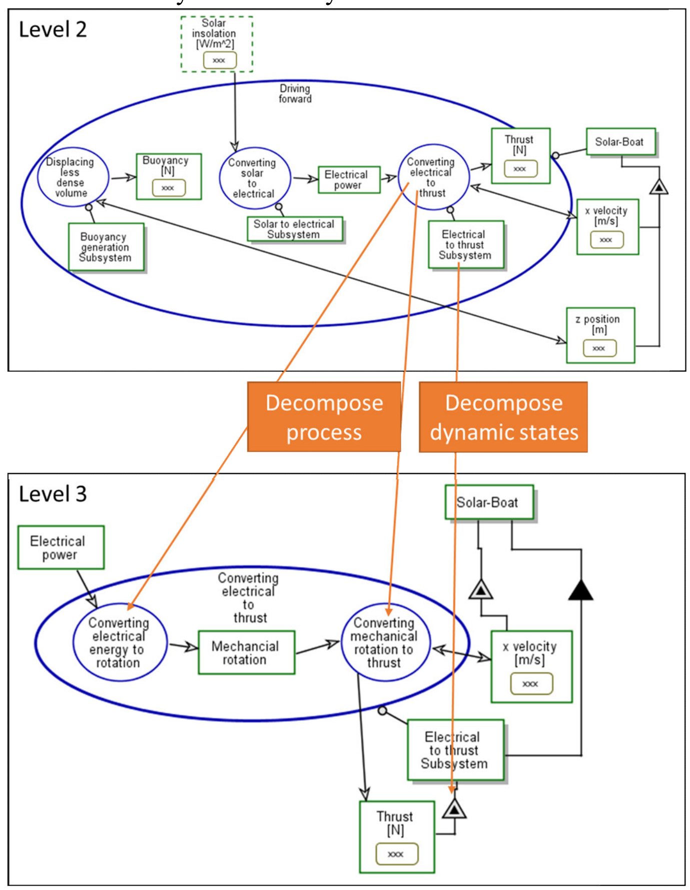

FIGURE 26. LEVEL 2 TO LEVEL3: DECOMPOSING THE SYSTEM ARCHITECTURE "CONVERTING ELECTRICAL TO THRUST" TO A FUNCTIONAL ARCHITECTURE.

Step E2: Assigning Subsystem-Components to Enable the Processes is completed to develop a System Architecture (as in forms which are capable of implementing the processes). An example of this is shown in FIGURE 14. A naming convention similar to that utilized for the naming of objects in Step B should be used (based on the input and output of the process). This step is based on the assumption that each process is enabled by an individual object forming a process and object pair in the System Architecture diagram. For the "Electrical to thrust" Subsystem these are "Electrical to rotation" and "Rotation to thrust", while "Buoyancy generation" and "Solar to electrical" Subsystems have components "Buoyancy generation" and "Solar to electrical" respectively due to single processes decomposition.

Step E3: Mapping to a Formal Structure in OPM is required given the System Architecture does not define specifically the connections between the objects such that a numerical model in Modelica can be developed. To achieve this as per the detailed description at the start of this paper, starting at the lowest level of hierarchy each OPM process and object pair of the System Architecture is compared to a library keyed on the process and object pairs inputs and outputs. This component from the library is then selected and incorporated into the higher level model. As such Level 2 is depicted being created in FIGURE 27 from the Formal Structure of Level 3. It should be noted the temperature connection is only introduced at this point as the modeler did not consider it when completing the Functional Architecture decomposition, but given the Level 4 model contains the temperature connection it is incorporated.

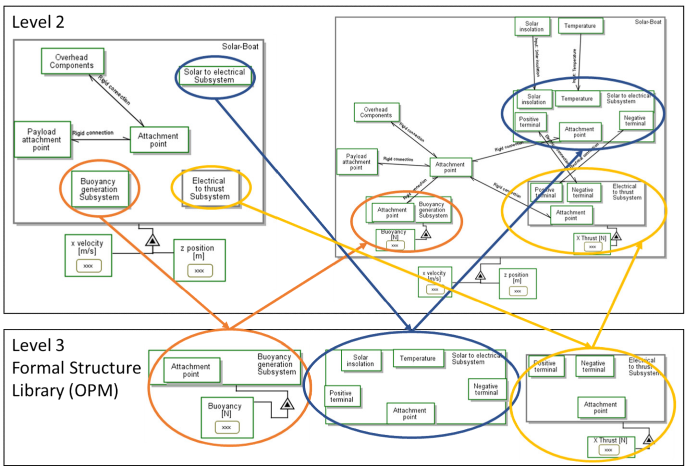

FIGURE 27. LEVEL 2: COMPOSING A FORMAL STRUCTURE (OPM) OF THE SOLAT-BOAT OUT OF LEVEL 3 SUBSYSTEMS.

Step E4: Mapping to a Formal Structure in Modelica was described in depth previously in this paper. As per that description the process of defining Formal Structure of one level of hierarchy from the interfaces defined by a lower level enables Formal Structure for all levels to be created. Based on this Level 2 is depicted being created in FIGURE 28 with a previously defined Modelica replaceable partial component.

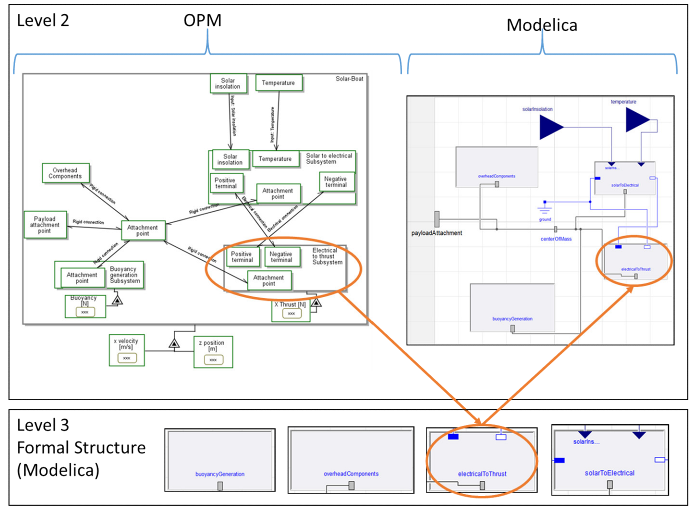

FIGURE 28. LEVEL 2: COMPOSING A FORMAL STRUCTURE (MODELICA) OF THE SOLAT-BOAT OUT OF LEVEL 3 SUBSYSTEMS.

Step E5: Composing Alternatives in Modelica is where various alternative Solar-Boats are composed from various alternative Subsystems created by the variation of Subsystem-Components in the Modelica Formal Structure. For this study the "Buoyancy generation" and "Solar to electrical" subsystems are fixed (no hull or solar panel array variation respectively). The "Electrical to thrust" subsystem is varied however by varying the components used as "Electrical to rotation" and "Rotation to thrust", i.e. a different motor and propeller are used. Six different "Electrical to thrust" subsystems are proposed utilizing two different motors (Low Mass-LM and High Mass-HM) and three different two blade propellers (160mm, 200mm and 220mm diameters). Resulting in six different "Electrical to thrust" subsystems which with the single choices of "Buoyancy generation" and "Solar to electrical" subsystems results by way of composition on the Level 2 Modelica Formal Structure six Solar-Boat alternatives (which are named after the motor and propellers they use).

Step F: Composing each System of Interest Alternative into each Assessment Scenario for Simulation is depicted in FIGURE 29 and FIGURE 13 to obtain a models which are ready for simulation each Alternative System of Interest design must be composed into each Assessment Scenario Formal Structure in Modelica. For the running case study of six Solar-Boat alternatives being composed into the four specific Assessment Scenarios results in twenty four models ready for simulation.

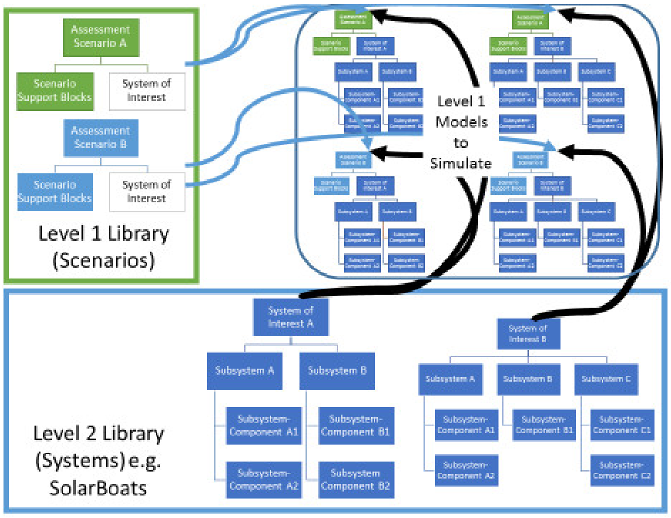

FIGURE 29. DEPICTION OF COMBINING ASSESSMENT SCENARIOS WITH SYSTEMS TO SIMULATE.

Step G: Simulating every Assessment Scenario and System of Interest Alternative Combination results in time series data which explicitly includes the variable of interest specified for the Assessment Scenario (i.e. twenty four time series).

Step H: Consolidating Simulation Results with MODA is to provide a fast initial comparison of the performance of the various Alternative System of Interest designs. Multi Objective Decision Analysis (MODA) is employed using the value function, weighting and data extraction rules defined in Step D to compute the total weighted value for each alternative. Consolidating the time series results from the twenty four simulation runs by way of MODA produces FIGURE 30, where the total weighted value for each alternative (x-axis) is the total height of the column which is broken down into contributions by each Assessment Scenario.

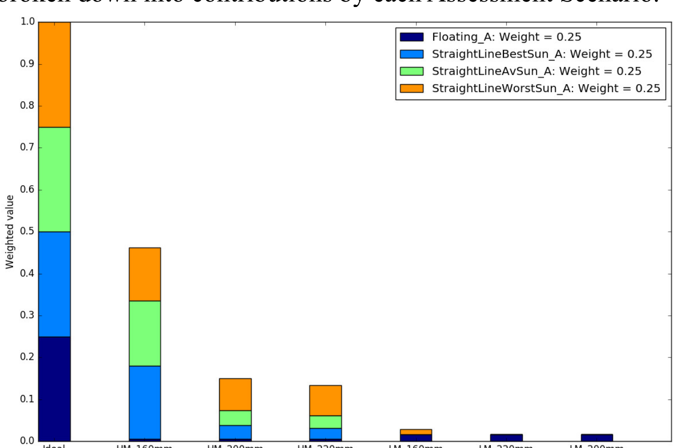

FIGURE 30. MODA PROCESSED RESULTS FROM SIMULATION VARYING MOTORS AND PROPELLERS.

Step I: Reviewing Results initially involves a review of FIGURE 30 the results of Step H. However such results lack the detail into what caused performance to be better or worse for a particular alternative. As such the full results from simulation of Step G are available for further review and assessment. In such a review of the Simulation Results the other alternatives are found to have lower speeds due to lower thrust. Further investigation into Subsystem-Components reveals the angular velocity of the low mass motor is far from nominal speed of the motor when compared to the high mass motor. However the low mass designs do perform better in the floating scenario.

Step J: Modify and Repeat or Move to Detailed Design depends on the engineers preference, design study purpose and results of Step I, the project might immediately move into detailed design, or one, some or all of the steps of the methodology described might be completed again with the aim of understanding existing alternative designs better (i.e. Assessment Scenario variation) or development of additional alternative designs (e.g. component variation, architecture variation).

3.3. Another Iteration

Given ideally a Solar-Boat alternative could be developed with the low mass motor (good floating performance) which operates closer to its nominal speed more designs are to be explored. As such variation of the Functional Architecture a new Formal Structure is created onto which new Solar-Boat alternatives are created and assessed using the same infrastructure as the previous design iteration. As explained as follows:

Steps A, B, C and D handle Subsystem identification and Assessment Scenario specification. These need no modification.

Step E requires modification to vary the alternative designs considered.

Step E1 involves a review of the Functional Architecture. Given the target is to change the motor spin speed to produce more thrust, FIGURE 26 Functional Architecture of “Electrical to thrust” Subsystem is reviewed and new processes “Changing rotation speed” process inserted between the existing processes of “Converting electrical energy to rotation” and “Converting mechanical rotation to thrust”.

Step E2 results in a new System Architecture created from the new Functional Architecture. Given a new process has been introduced a new object to enable it must also be introduced (“Change rotation speed Component” in FIGURE 31).

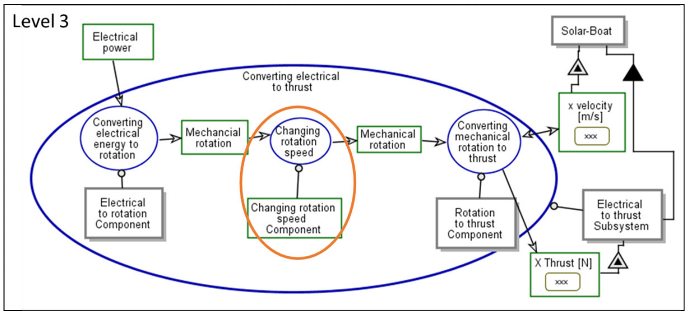

FIGURE 31. LEVEL 3: ALTERNATIVE SYSTEM ARCHITECTURE "CONVERTING ELECTRICAL TO THRUST".

Steps E3 and E4 involve the Formal Structure in OPM and Modelica being updated given the System Architecture has been updated. Which by following the processes described previously and depicted in FIGURE 15 and FIGURE 16 Formal Structure can be created with a position for the "Change rotation speed Component" introduced. It should be noted that the interface for the "Electrical to thrust" subsystem will not vary and as such Level 2 and Level 1 require no architecture changes. Subsequently the updated Formal Structure in OPM can be used to update the Formal Structure in Modelica which is shown on the left side of FIGURE 14 in [14]. Resulting in a new "Electrical to thrust" subsystem Formal Structure.

Step E5 (composition of alternatives) can now be performed utilizing the new "Change rotation speed Component" in the new Formal Structure. As in, new alternative "Electrical to thrust" Subsystems can be created by the composition of Subsystem-Components. Keeping the existing two motors and three propellers a 3:1 gearbox is introduced for the heavy motor and a 13:1 gearbox for the low mass motor. Resulting in six alternative "Electrical to thrust" Subsystems (one example utilizing a particular motor, gearbox and propeller is displayed on the right side of FIGURE 14 in [14]). Which when composed into the Level 2 Formal Structure while keeping the existing hull and solar panel array results in six alternative Solar-Boats which are named corresponding to the motor, gearbox and propeller they utilize.

Steps F and G involves composing the six new alternative designs into the four specific Assessment Scenarios Formal Structure (Step F) resulting in twenty four separate models which are then simulated (Step G).

Steps H and I are completed resulting in FIGURE 32 where two alternatives incorporating the low mass motors (LM_13_200mm and LM_13_220mm) now deliver greater than that of any high mass motor configurations as a result of better performance in the "Driving forward" and "Floating" Assessment Scenarios.

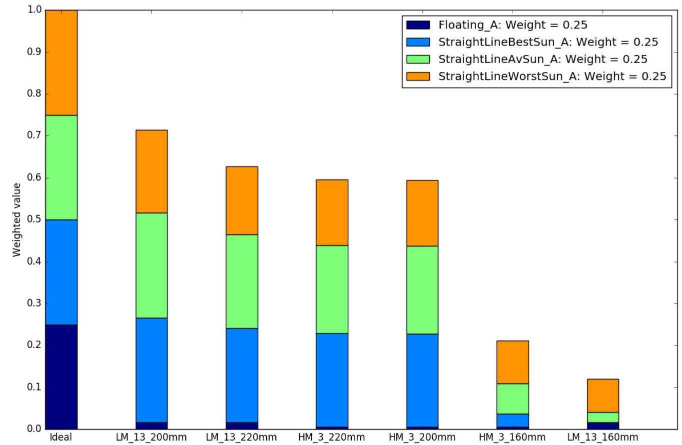

FIGURE 32. MODA PROCESSED RESULTS FROM SIMULATION VARYING MOTORS, GEARBOXES AND PROPELLERS.

Step J given the better performance it now be appropriate to move into detailed design of a prototype based on this System-Level Design.

4. DISCUSSION

4.1. Benefits

The proposed tools and methodologies aim to offer the following benefits:

- Logical synthesis of hierarchical Formal Structure for new product development. The conceptual modeling of OPM has been harnessed to develop a numerical model.
- Fast synthesis of alternative system and subsystem designs by using the common architectures.
- Fast consistent assessment of each alternative designs value by automated simulation and comparison.

4.2. Novelty

The following points of novelty are noted from previous research:

- Using numerical modeling and OPM for design. Previous OPM and numerical simulation [9,10] with MATLAB/Simulink was for the purpose of gaining greater understanding of existing descriptive models rather than the synthesis of engineering systems based on components.
- Attempt to formally combine OPM and Modelica.
- Defining a product structure logically from its functionality to perform a trade study. In [13,16] a predefined product structure was used.

4.3. Further Research

Currently the behavior modelled in Modelica created under the current scheme assumes the processes all occur simultaneously at all times, however real systems demonstrate processes which are not occurring at all times (e.g. time triggering of a process or if/then rules) which the OPM language supports. As such the methodology should be expanded to incorporate such behaviors. In addition the current approach of mapping Functional Architecture to Formal Structure is made under the assumption of very simple modularity (one process is enabled by one object), with no provision for situations where two subsystems enable a process or vice-versa without merging such subsystems and processes into simple process and object pairs.

Mapping from System Architecture to the Formal Structure requires a library of components keyed with System Architecture process and enabling object pairs. Existing libraries are not described in such a way and so the arrangement of such libraries needs exploration.

Further, the currently demonstrated system is of very low complexity, demonstration on a larger more complex project and demonstrating value would provide more validity to the approach.

5. CONCLUSIONS

A methodology and set of tools integrating OPM (for qualitative description) and Modelica (for numerical simulation) to synthesize alternative System-Level Designs and then assess those designs was presented with the aim of decreasing the time to develop new systems. This involved:

- Identifying the required functionality of the system by decomposing its primary functionality in OPM to develop a hierarchy.
- Identifying common Assessment Scenarios to assess Alternative System of Interest designs.
- Synthesizing a Formal Structure on which alternative System of Interest's can be created by defining a series of mappings between various model types in OPM (Functional Architecture, System Architecture and Formal Structure).
- Mapping the OPM Formal Structure to a Modelica Formal Structure such that a series of partial Modelica models are developed.
- Rapidly composing Alternative System of Interest designs by moving up the hierarchy populating the Modelica Formal Structure with Modelica models from a library ready for simulation.
- Simulating each Alternative System of Interest design for each Assessment Scenario. Utilizing the common Formal Structure to enable this.
- Consolidating each Alternative System of Interest's predicted performance by way of Multi Objective Decision Analysis (MODA).

A demonstration utilizing the methodology was presented using a solar powered autonomous boat as an example.

REFERENCES

[1] INCOSE, 2014, A world in motion: Systems Engineering Vision 2025, INCOSE.

[2] Sutherland, J., Kamiyama, H., Aoyama, K., and Oizumi, K., 2015, "Systems Engineering and the V-Model: Lessons from an Autonomous Solar Powered Hydrofoil," 12th International Marine Design Conference (IMDC), Tokyo Japan.

[3] Ulrich, K., and Eppinger, S., 2011, Product Design and Development, McGraw-Hill/Irwin, New York.

[4] Dori, D., 2002, Object-process methodology: a holistic systems paradigm, Springer, Berlin ; New York.

[5] ISO, 2015, ISO/PAS 19450:2015 Automation systems and integration -- Object-Process Methodology.

[6] Grobshtein, Y., Perelman, V., Safra, E., and Dori, D., 2007, "Systems modeling languages: OPM versus SysML," Systems Engineering and Modeling, 2007. ICSEM'07. International Conference on, IEEE, pp. 102–109.

[7] Wang, R., Agarwal, S., and Dagli, C. H., 2015, "OPM & color petri nets based executable system of systems architecting: A building block in FILA-SoS," Systems Conference (SysCon), 2015 9th Annual IEEE International, pp. 554–561.

[8] Yaroker, Y., Perelman, V., and Dori, D., 2013, "An OPM conceptual model-based executable simulation environment: Implementation and evaluation," Syst. Eng., **16**(4), pp. 381–390.

[9] Bolshchikov, S., Renick, A., Mazor, S., Somekh, J., and Dori, D., 2011, "OPM Model-Driven Animated Simulation with Computational Interface to Matlab," 2011 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 193–198.

[10] Dori, D., Renick, A., and Wengrowicz, N., 2015, "When quantitative meets qualitative: enhancing OPM conceptual systems modeling with MATLAB computational capabilities," Res. Eng. Des., pp. 1–24.

[11] INCOSE, 2015, Wiley: INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition - INCOSE.

[12] SEBoK, 2015, "Guide to the Systems Engineering Body of Knowledge (SEBoK)."

[13] Cilli, M. V., and Parnell, G. S., 2014, "Systems engineering tradeoff study process framework," 24th INCOSE Int'l Symposium, Las Vegas, NV.

[14] Sutherland, J., Oizumi, K., Aoyama, K., Takahashi, N., and Eguchi, T., 2016, "System-Level Design Trade Studies by Multi Objective Decision Analysis (MODA) utilizing Modelica," 1st Japanese Modelica Conference, Tokyo Japan.

[15] Crawley, E., Cameron, B., and Selva, D., 2015, System Architecture: Strategy and Product Development for Complex Systems, Prentice Hall, Boston.

[16] Edwards, S., Cilli, M. V., Peterson, T., Zabat, M., Lawton, C., and Shelton, L., 2015, "Whole Systems Trade Analysis," 25th INCOSE Int'l Symposium, Seattle, Seattle, WA, USA.