System-Level Design Trade Studies by Multi Objective Decision Analysis (MODA) utilizing Modelica

Joshua Sutherland, Kazuya Oizumi, Kazuhiro Aoyama[1] Naoki Takahashi[2] Takao Eguchi[3]

[1]Department of System Innovation, The University of Tokyo, Japan,
{joshua,oizumi,aoyama}@m.sys.t.u-tokyo.ac.jp
[2]Dassault Systèmes K.K., Tokyo, Japan naoki.takahashi@3ds.com
[3]Shinko Research Co. Ltd, Tokyo, Japan eguchi.takao@kobelco.com

Abstract

This paper describes an approach and tool to conduct System-Level Design Trades Studies utilizing Modelica by way of Multi Objective Decision Analysis (MODA). Requirements for this being identified from the problems experienced on student Solar-Boat project.

The proposed approach and tool utilizes Modelica to predict performance of different competing alternative designs and MODA as a way to consistently compare those alternatives subject to a range of Assessment Scenarios.

To enable alternative designs to be created with low effort the replaceable feature of Modelica components is used such that the alternatives can share common architectures subject to a defined hierarchy which includes the Assessment Scenario itself.

A tool was created to automate the placement of alternative designs into the Assessment Scenarios, run the simulations and consolidate the results via MODA. Examples utilizing the approach and tool to predict performance of competing Solar-Boat designs and compare them is provided.

Keywords: Trade Studies, Assessing Alternative Designs, System-Level Design

1 Introduction

System-Level Design is defined in (Ulrich *et al*, 2011) to "include the definition of the product architecture and the decomposition of the product into subsystems and components". Expanding to describe what an engineer must achieve with the System-Level Design, it is expected that there is sufficient level of detail to enable the system being designed to be assessed from the perspective of predicting its performance and the cost sufficiently accurately while simultaneously informing what is acceptable to be designed at the detailed design stage, as such bounding the number of alternatives at the detailed design stage to a reasonable level.

Further the development of a System-Level Design should involve the comparison of alternative competing designs from which one or many might be selected for further detailed design.

As described by (Parnell *et al*, 2014) trade studies (or tradeoff studies) play a central role in decision management and can be applied throughout a systems lifecycle. With the term "tradeoff" implying that there may be the need to forgo one objective to obtain a desired level in another. As such trade studies are suitable for usage in System-Level Design which includes the selection of a design from a set of competing alternatives.

1.1 Solar-Boat Project Description

Every summer on Japan's Lake Biwa multiple university teams participate in a competition to race fully automatous solar powered boats over a 20km course which they have designed, manufactured and tested over the previous year. The University of Tokyo, Department of Systems Innovation regularly participates in this event, where all boats are subjected to the following rules: Maximum 2m^2 solar panels, 25Wh lead based batteries and ability to carry a 64g payload. Figure 1 shows an example from 2014 of the hydro foiling craft constructed by University of Tokyo students.

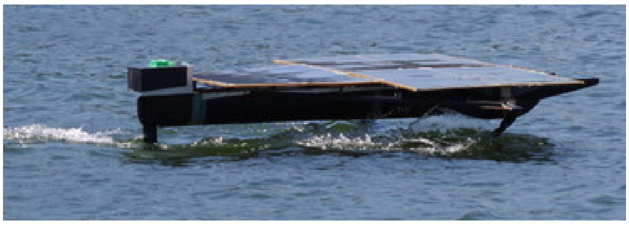

Figure 1. 2014 University of Tokyo Solar-Boat.

1.2 Problems with Previous Solar-Boat Projects

In (Sutherland *et al.* 2015) a detailed analysis of the activities conducted on the Tokyo 2014 Solar-Boat project was conducted, the resulting summary of problems mapped to Lifecycle Stages (LS) is listed in Table 1. Reviewing the listed problems, the lack of design exploration and performance prediction at LS3: System-Level Design focused on a design target identified in LS2: Concept Development resulted in further problems downstream where alternatives generation and simulated performance prediction were not the norm. As such much trial and error based on physically realized components was performed wasting resources.

Table 1. Solar-Boat Lifecycle Stages, 2014 problems and some proposed solutions.

Lifecycle Stage (LS)	*2014 problems*	*Proposed solutions*
LS1: Clarify	Slow to acquire initial knowledge.	Provide knowledge in models.
LS2: Concept Development	Unclear of the design target.	Complete trade-off analysis of multiple designs using models to simulate performance.
LS3: System-Level Design	Little exploration of alternatives or their predicted outcomes.	
LS4: Detail Design	Little prediction of performance.	
LS5: Production, Test and Refinement	Based on trial and error.	
LS6: Race	Lost race due to faults which could have been predicted with modeling.	
LS7: Knowledge Transfer		

Given that design exploration and performance prediction were conducted inadequately on the 2014 project it is important to survey work products generated and used related to early lifecycle stage numerical simulation. The work products which did exist were numerical models in Excel and MATLAB/Simulink. These models while somewhat useful for predicting performance suffered from the following problems:

1. Lack of modularity, preventing: Reuse of existing model components in new situations; generating alternative designs out of different module combinations; different students developing models independently and subsequently virtually integrate.
2. Lack of adequate libraries, resulting in: inaccurate approximations to complex components (e.g. a solar panel array is a "power generator" of a particular efficiency, rather than a component which interacts with a circuit current and voltage).
3. Lack of holistic model resulting in subsystems being designed in isolation (e.g. Powertrain designed separately to the main system structure).
4. Lack of infrastructure to assess and compare alternative designs consistently with each other making it unclear as to what design has been selected for what reason.
5. Lack of access to variables which the models were not "designed" to provide. Much modification is required to Excel and MATLAB/Simulink to expose a new variable of interest.

1.3 Proposing Solutions

Based on these problems identified at the early lifecycle stages some high level solutions were proposed in (Sutherland, 2016) to help alleviate these problems (also shown in Table 1) by:

1. Providing knowledge in models.
2. Completing trade-off analysis of multiple designs using models to simulate performance.

Given the different types of knowledge required to be stored and mechanisms to explore alternative designs it is proposed by (Sutherland, 2016) to utilize a conceptual modeling language (OPM, Object Process Methodology (ISO, 2015)) for LS2: Concept Development and numerical modeling language with subsequent simulation (Modelica) for LS3: System-Level Design. In this paper the proposed usage of an automation framework for expediting the completion of trade studies utilizing Modelica for LS3: System-Level Design is explored.

2 Developing Requirements for the Tools and Approach while Reviewing Existing Literature

Table 2 details a set of requirements for a trade study tool and approach which aims to address the issues with the previous Solar-Boat projects approach. A brief comparison to existing tools and methodologies is provided in the following sections which the focuses on an adequate numerical modeling and simulation language (Section 2.1) and systematic approach (Section 2.2).

2.1 Modelica

The use of Modelica subject to a logical approach can address many of the requirements identified in Table 2. Describing this explicitly:

1. The replaceable keyword enables subsystems and components to be replaced subject to a defined interface.
2. Large libraries of standard components exist and new ones can be developed based on equations quickly.
3. Components integrate across multiple domains.
4. Simulations provide access to all the variables of the equations which describe the components behavior.

Table 2. Solar-Boat previous numerical modeling problems and requirements for the proposed system.

2014 numerical modelling problems	*Requirement for proposed system*
1) Lack of modularity	Can replace components and subsystems with any other which is compliant to a defined interface.
2) Lack of adequate libraries	Access to a range of library components. Can develop new library components quickly.
3) Lack of holistic model	Integrate multiple engineering domains concurrently.
4) Lack of infrastructure to assess and compare	Can assess and compare all alternative designs consistently and automatically.
5) Lack of access to variables which the model was not "designed" to provide	Can review the details of individual components performance.

2.2 Trade Study Approaches

The International Council of Systems Engineering (INCOSE) provides by way of description in (INCOSE, 2015) and (SEBoK, 2015) a decision management process which is intended for trade studies. Two implementations of this process are provided in (Cilli *et al*, 2014) and (Edwards *et al.*, 2015). In these implementations a common architecture of subsystems is defined for the system of interest, variation of the subsystems within this architecture enables alternative System-Level Designs to be generated. Each of these alternatives are then assessed by a common set of performance metrics which have been mapped to the functional objectives of the system of interest. In the case of (Cilli *et al*, 2014) this involved mapping the amount of value derived for a particular functional objective from a prediction of its performance by way of a value function. Multiple objectives are then combined by way of weighting to enable Multi Objective Decision Analysis (MODA). In (Cilli *et al*, 2014) performance prediction is provided by subject matter experts, while in (Edwards *et al.*, 2015) simulation is used, but the simulation model does not use acausal interactions between the individual components which make up the model. Instead subject matter experts define the interaction between components based on the equations and data they wish to utilize. This process is somewhat opaque.

As such, while the approaches used by (Cilli *et al*, 2014) and (Edwards *et al.*, 2015) to implement the INCOSE decision management process can form a basis of an approach, it is proposed for this paper and the Solar-Boat project to utilize Modelica as the numerical modeling tool such that the benefits described in Section 2.1 can be realized when completing a trade study.

3 Proposed Tools and Approach

A high level flow diagram of the proposed tools and approach, developed and demonstrated is shown in Figure 2. It is described as having three important processing elements (in green on Figure 2) listed as: Model Builder, Simulation Runner and Results Processor.

The required initial inputs of the tools and approach (in orange on Figure 2) take the form of Comparison.xml detailing what Assessment Scenarios and alternative System-Level Designs to consider and a library of Modelica models which are the Assessment Scenarios and alternative System-Level Designs referenced by the Comparison.xml. With the Assessment Scenario describing how to assess a design alternative subject to a set of stated conditions.

Ultimately the aim of running through the approach is to generate insight (in black on Figure 2) into the performance and cost characteristics of alternative designs, which can occur through the reviewing consolidated Multi Objective Decision Analysis (MODA) results or reviewing detailed raw results of the .mat file (in blue on Figure 2) generated during the simulation of the model associated with each design alternative for each Assessment Scenario (blue on Figure 2).

More detailed descriptions are provided in subsequent sections for the items in Figure 2: Inputs to the approach described in Section 3.1 while processing elements and their subsequent outputs are described in Section 3.2.

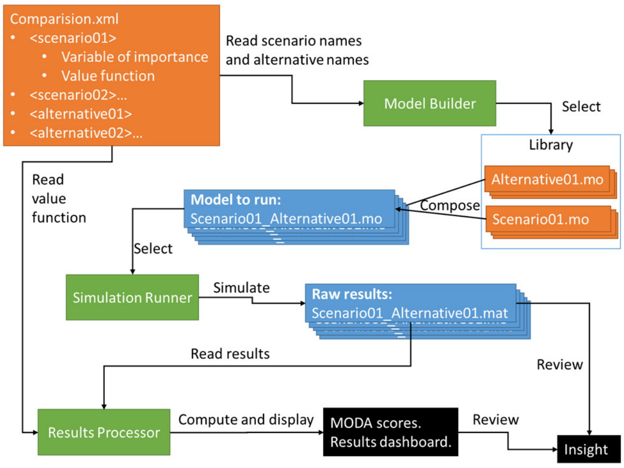

Figure 2. Proposed trade study approach utilizing Modelica. Green: System processing elements. Orange: Inputs. Blue: Intermediate results. Black: Final results.

3.1 Inputs

3.1.1 Comparison.xml

Comparison.xml is an input to the approach, it is an XML file listing and describing what Assessment Scenarios to complete (including time to simulate and how to processes its results, described in more detail in Section 3.2.3) followed by a listing of alternative System-Level Designs to assess. A code snippet is provided below of an example file (truncated and modified for simplicity).

```
<system_name="solarBoat"/>
<scenario name="Floating">
      <variable="z_top_of_hull"/>
      <variable_units name="m"/>
      <value_func_direct name="neg"/>
      <min_accep_perform val="-0.1"/>
      <stretch_goal val="-0.4"/>
      <weight val="0.5"/>
      <sim_length val="70"/>
      <extract_data_type name="mean"/>
 </scenario>
 <scenario name="StraightLineAvSun">
      <variable="x_velocity"/>
      <variable_units name="m/s"/>
      <value_func_direct name="pos"/>
      <min_accep_perform val="1.5"/>
      <stretch_goal val="3"/>
      <weight val="0.5"/>
      <sim_length val="3"/>
      <extract_data_type name="max"/>
</scenario>
<scenario name="Cost">
    <variable="cost_money"/>
    <variable_units name="yen"/>
    <value_func_direct name="neg"/>
    <min_accep_perform val="300000"/>
    <stretch_goal val="0"/>
    <sim_length val="1"/>
    <extract_data_type name="max"/>
</scenario>
<design name="Ideal"/>
<design name="Boat_Alternative_01"/>
<design name="Boat_Alternative_02"/>
```

3.1.2 Library – Structure and Conventions

To manage complexity the library and the models utilized are divided into distinct hierarchy levels which are outlined in Table 3. Figure 3 pictorially depicts how Level 4 models ultimately combine to form a Level 1 model which can be simulated. Each level of this hierarchy is discussed in the subsequent sections.

In addition assumptions made about the Solar-Boat are explicitly listed to enable a consistent library to be developed by way of setting rules for how these assumptions are implemented in the library. To generalize, these assumptions stem from the bottom up approach used to develop the models, where for example Subsystems are defined by their interface and internal Subsystem-Components. Assumptions associated cost are described in Table 4, assumptions associated with mass are described in Table 5, and those regarding fluid interaction are described in Table 6.

Table 3. Hierarchy levels utilized in the models and libraries.

Hierarchy Level	*Name*	*Example*
Level 1	Assessment Scenario	Straight line good weather
Level 2	System of Interest	Solar-Boat
Level 3	Subsystems	Electrical to Thrust
Level 4	Subsystem-Components	DC Motor

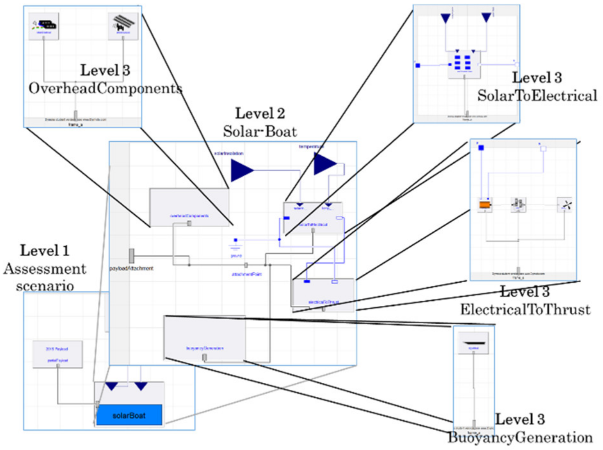

Figure 3. Representation of the different hierarchy levels combining.

Table 4. Assumptions associated with cost and how the assumptions are implemented in the library.

Assumption / Design process decision	*Implementation in Library*
All objects of the Solar-Boat have cost.	All models at Level 2-4 extend “PartialProcurementAttributes” with the single variable cost_money_computed. As such they must expose/compute their cost.
Cost properties of the System of Interest occur from the sum its Subsystem-Components cost properties.	Every Subsystem-Component defines a cost parameter. The cost of the subsystem is the sum of the cost of its components. The same logic follows up the levels.

Table 5. Assumptions associated with mass and degrees of freedom and how the assumptions are implemented in the library.

Assumption / Design process decision	*Implementation in Library*
All objects of the Solar-Boat have mass.	All models at Level 2-4 extend "PartialMassAttributes" with the single variable mass_computed. As such they must expose/compute their mass. Which might be the sum of lower level component masses.
Mass properties of the System of Interest occur from the sum its Subsystem-Components mass properties.	Every Subsystem-Component attaches a mass component from the Mechanics.MultiBody library to its Frame_a connector.
System of Interest is a single rigid body in a 3D world.	All Subsystems and components expose a Modelica.Mechanics. MultiBody.Interfaces. Frame_a connector.
The number of degrees of freedom in motion required at different times if the lifecycle varies.	The use of Modelica.Mechanics. MultiBody.Joints. Prismatic to prevent motion on degrees of freedom which are not going to be considered.

Table 6. Assumptions associated with fluids and their implementation in the library.

Assumption / Design process decision	*Implementation in Library*
All objects immersed in a fluid generate a drag_force and buoyancy_force.	Any models Level 2-4 expected to be immersed in a fluid extend "PartialInAFluidAttributes" with the variables drag_force and displaced_volume exposed. As such these must be computed.

3.1.3 Library Level 1 – Assessment Scenario

The Assessment Scenario is a Modelica model which aims to provide the necessary infrastructure around a replaceable partial model of the System of Interest (Solar-Boat) the necessary inputs to represent the scenario for simulation. Figure 4 shows an example of one such of these for driving in a straight line subject to average sun conditions. With Figure 4 (left) showing prior to population with a valid alternative and Figure 4 (right) showing this after having been populated with a valid alternative Solar-Boat design.

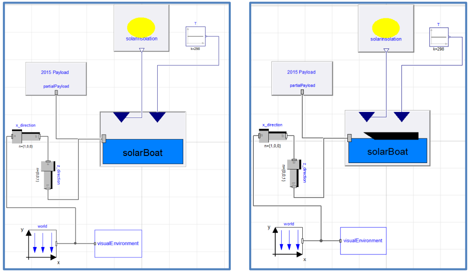

Figure 4. Assessment Scenario: Straight line average sun. Left: Before being populated with a valid Solar-Boat design alternative. Right: After being populated with a design alternative such that it can be simulated.

3.1.4 Library Level 2 – System of Interest

This level describes the system which is being attempted to be designed and assessed (i.e. Solar-Boat). As such all valid alternative designs should be compliant with the interface used for the System of Interest in the Assessment Scenarios. In addition the variables of interest defined in the Comparison.xml (e.g. x_velocity) must be declared such that they can be extracted by the Results Processor. Figure 5 shows the partial model interface, while Figure 6 (left) shows an example architecture created by the population with partial replaceable Subsystems interfaces. In this case four Subsystems are utilized: Solar to Electrical, Electrical to Thrust, Buoyancy generation and Overhead mass components. This architecture is then populated with Subsystem models to generate a Solar-Boat alternative as shown in Figure 6 (right).

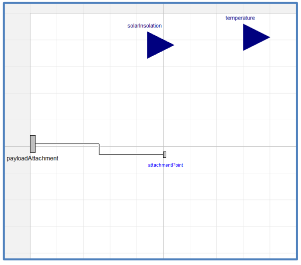

Figure 5. Partial model defining the interface of the System of Interest (Solar-Boat).

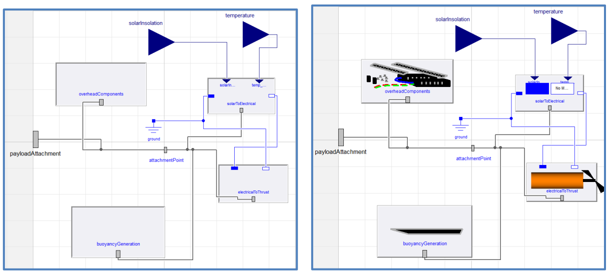

Figure 6 Left: Extending the partial model of the System of Interest (Solar-Boat) and subsequently defining an architecture by placing partial Subsystems on it. Right: Populating this architecture with Subsystems.

3.1.5 Library Level 3 – Subsystems

Subsystems have an interface to define their interaction at the System of Interest level. An example is shown in Figure 7 for an electrical to thrust Subsystem. A Subsystem architecture (Figure 8 left) can then be defined for the interface (Figure 7) by the population with partial replaceable Subsystem-Component interfaces. Subsequent population with Subsystem-Components results in an alternative Subsystem being defined (Figure 8 right).

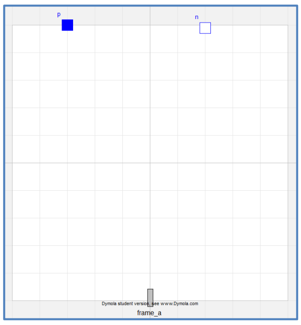

Figure 7. Partial model defining the interface of a Subsystem (electrical to thrust).

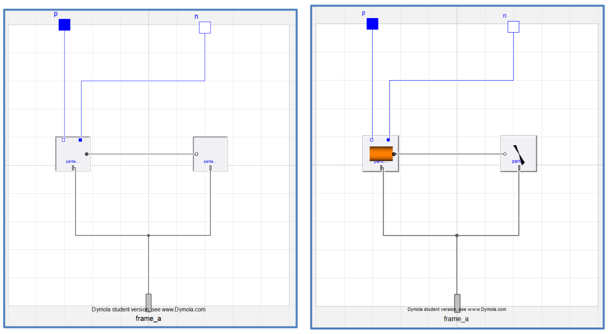
p
n
Dymola student version, see www.Dymola.com
frame_a
p
n
Dymola student version, see www.Dymola.com
frame_a

Figure 8. Left: Extending the Electrical to thrust partial model and subsequently defining an architecture by placing partial Subsystem-Components on it. Right: Populating this architecture with Subsystem-Components.

3.1.6 Library Level 4 – Subsystem-Components

The lowest level of the defined hierarchy are Subsystem-Components. Similar to the other levels an interface is used to define their interaction at the higher levels, as shown in Figure 9. However architecture implementation takes a different form, generally being made of additional models (custom and standard library) which have their parameters provided by way of redeclaring a partial record. The aim of this approach is to create a large library of components based on the specification sheets of commercial products which can be transferred to a record in the Modelica language. As per the assumptions discussed in Section 3.1.2 each Subsystem-Component must declare a mass and cost which can then be used to compute the mass and cost of the Subsystem it is included in. Further as shown at the bottom of Figure 10 each Subsystem-Component includes a mass component from Mechanics.MultiBody library.

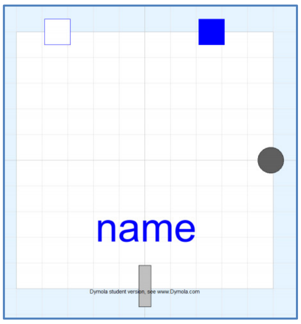

Figure 9. Interface of Subsystem-Component (Electrical to rotation).

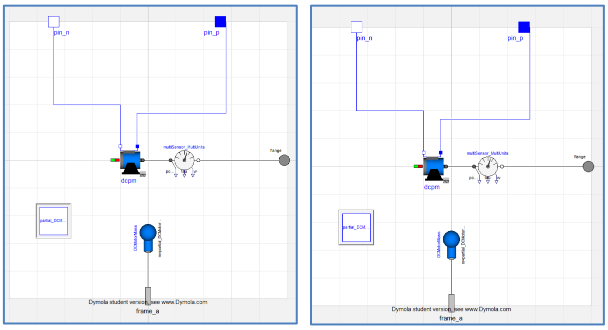

Figure 10. Left: Implementing the Electrical to rotation component with a partial record. Right: Populating the partial record to create a Subsystem-Component.

3.2 Processing Elements

In this section the processing elements of the approach proposed (shown in Figure 2) which process inputs to generate output are discussed. This was implemented with Python code as Dymola and OpenModelica were unable to automate the variation of Modelica blocks or the consolidation of multiple results.

3.2.1 Model Builder

The Model Builder processing element generates a Modelica model for each combination of Assessment Scenario and System of Interest (Solar-Boat) alternative described in the Comparison.xml file. The Model Builder requires that the Assessment Scenarios and System of Interest (Solar-Boat) alternatives named in the Comparison.xml are available from the library. This is achieved programmatically by duplicating existing model for the Assessment Scenario and manipulating the .mo text file to change the Solar-Boat alternative to the one for assessment.

3.2.2 Simulation Runner

Simulation Runner subsequently simulates all the models created by Model Builder for the simulation length specified in the Comparison.xml file. This is achieved programmatically by utilizing Dymola's python interface. The subsequent results (in the .mat file) can then be further reviewed by the engineer if they wish.

3.2.3 Results Processor

The results processor extracts for each model simulated (Assessment Scenario and design alternative pair) the time series of the raw simulation results the variable of interest for each Assessment Scenario to measure the System of Interest's performance (e.g. max x_velocity). This extraction is enabled by Dymat python package (Dymat, 2015). For simplicity weighted sum MODA scheme described in (Cilli *et al*, 2014) was used which is explained as follows.

The extracted variable (e.g. max x_velocity) is used to compute unweighted value by the utilization of the value function (see Figure 11 for an example) which is defined in Comparison.xml and maps performance on a particular scenario to unweighted value (minimum acceptable performance corresponding to zero value and stretch goal corresponding to value of 1). The unweighted value is then multiplied by the weight assigned in Comparison.xml to create weighted value for that scenario. Summing the for all the scenario runs for an individual alternative design results in the total weighted value for that particular alternative design. With the "ideal system" having a value of 1 as its performance is assumed to always be at the stretch goal and the weights sum to 1.

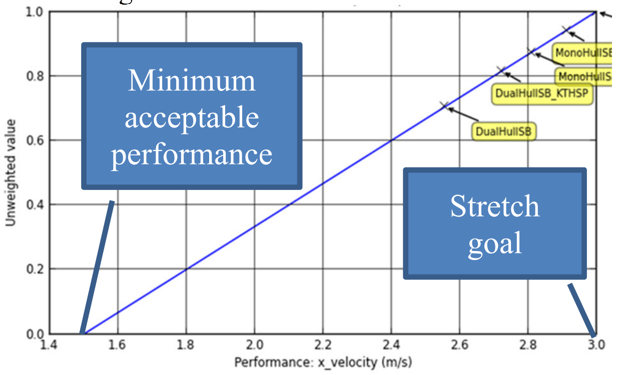

Figure 11. Example linear value function.

These results can then be consolidated on to a single dashboard (example shown in Figure 12). On all charts the y-axis displays the total weighted value for the alternative. For the top chart and middle chart the x-axis indicates the alternatives being considered. With the middle chart further displaying a breakdown of the weighted value contributions from each Assessment Scenario. For the bottom scatter chart, each point on the bottom chart indicates a design alternative of the System of Interest and the x-axis indicating cost in yen for that particular design alternative.

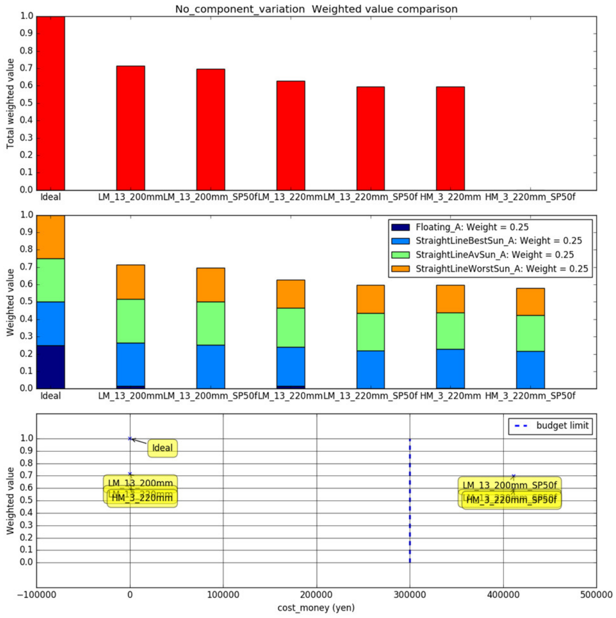

Figure 12. Example output of the Results Processor. Top: Total weighted value (y-axis) computed for all the Assessment Scenarios (x-axis) for all the Alternative System of Interests. Middle: Breakdown of the weighted value contributions from each assessment scenario. Bottom: Weighted value (y-axis) compared to cost in yen (x-axis) for each alternative design. (Results are from Section 4.3 comparison of the introduction of new solar panels).

4 Demonstration

This section provides examples of the utilization of the approach to demonstrate how it can rapidly enable the fast comparison of alternative Solar-Boats. The assessment scenarios used are described in Table 7, which are used to define a Comparison.xml and library of Level 1 models. For simplicity all weights were set equally to 0.25 in all the demonstrations, the Solar-Boat architecture of Figure 6 (left) is utilized.

4.1 Component Variation

An initial set of Solar-Boat alternative designs are outlined in Table 8 and created as models by populating the Solar-Boat architecture of Figure 6 (left). These designs are identical other than the variation in the Subsystem-Components used for a direct drive electrical to thrust Subsystem. The Subsystem-Component variation involves motor variation (high mass and low mass variants) and propeller variation. By following the flow diagram of Figure 2 for the Assessment Scenarios of Table 7 and alternative designs of Table 8, it is possible to generate the results as shown in Figure 13 automatically. Reviewing these results it is possible to see three alternatives fail to meet the minimum acceptable performance on at least one scenario (red ring on Figure 13). The complex interaction between boat mass, water line, thrust, drag and velocity has simplified into a single chart.

Table 7. Assessment Scenarios used in the demonstration.

Measure of interest	*Scenario conditions*	*Min accep perform*	*Stretch goal*	*Data type*
Top of hull z position (m)	Floating	-0.1	-0.4	Mean
x velocity (m/s)	Best ever insolation (870 Wm^2)	2	4	Max
x velocity (m/s)	Average insolation (550 Wm^2)	1.5	3	Max
x velocity (m/s)	Worst ever insolation (260 Wm^2)	0.5	2.5	Max

Table 8. Alternative Solar-Boat designs created by electrical to thrust Subsystem variation (H = High mass motor, L = Low mass motor).

Alternative name	*Buoyancy Generation*	*Solar To Elec*	*Elec To Thrust*
HM_160mm	Single hull	FT-136SE	H motor: No gearbox: 160mm prop
HM_200mm	Single hull	FT-136SE	H motor: No gearbox: 200mm prop
HM_220mm	Single hull	FT-136SE	H motor: No gearbox: 220mm prop
LM_160mm	Single hull	FT-136SE	L motor: No gearbox: 160mm prop
LM_200mm	Single hull	FT-136SE	L motor: No gearbox: 200mm prop
LM_220mm	Single hull	FT-136SE	L motor: No gearbox: 220mm prop

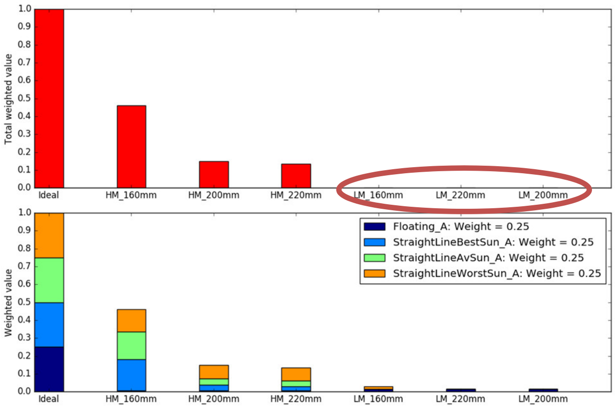

Figure 13. Multi Objective Decision Analysis results for Solar-Boat alternatives of Table 8.

4.2 Subsystem – Architecture Variation

Given the approach makes use of a common architecture for the Solar-Boat it is possible to rapidly compare alternative Solar-Boat designs utilizing different Subsystem architectures. As such Figure 14 displays an alternative electrical to thrust Subsystem architecture (incorporating a gearbox) to the one used previously (Figure 8). As such it is possible to create alternative Solar-Boats utilizing this architecture. Creating several alternatives by varying the Subsystem-Components in this model and simulating results in Figure 15 (where the designs have significant performance increase over the results of Section 4.1).

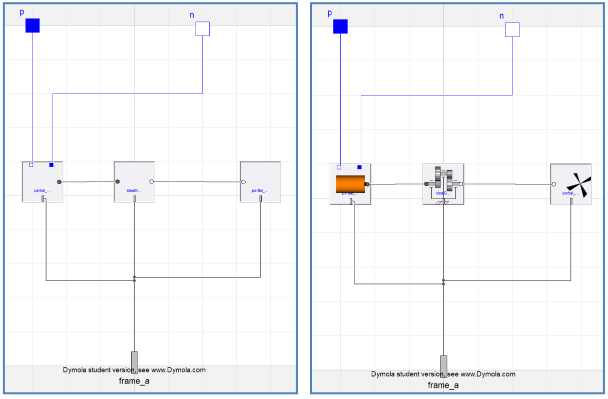
p
n
partial_
idealG...
partial_
Dymola student version, see www.Dymola.com
frame_a
p
n
Dymola student version, see www.Dymola.com
frame_a

Figure 14. Left: Extending the electrical to thrust partial model but defining a different Subsystem architecture to that in Figure 8 by incorporating a gearbox. Right: Populating this architecture with Subsystem-Components.

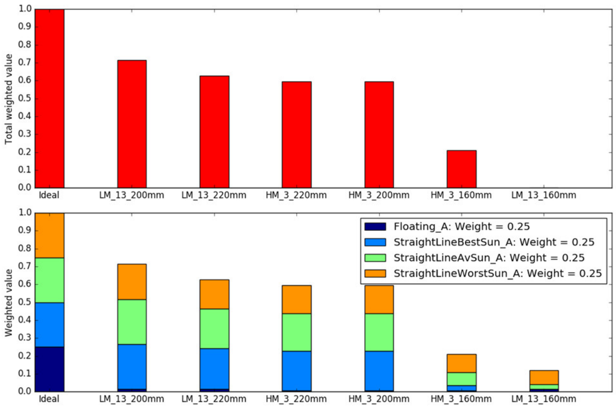

Figure 15. Multi Objective Decision Analysis results for Solar-Boat alternatives incorporating a gearbox.

4.3 Cost Benefit of a New Component

Further component exploration of interest could involve the performance evaluation associated with incorporating higher efficiency, high mass and high cost solar panels (solar to electrical Subsystem). Creating alternatives based on these and simulating results in Figure 12. The bottom chart clearly displays the large cost of the new solar panels (exceeding the project budget). While the weighted total value of alternatives incorporating the panels is not significantly different to those utilizing existing panels. Indicating they are not a wise purchase.

5 Discussion

The proposed approach described in this paper has a number of benefits when compared to other approaches. By clearly describing each Assessment Scenario and processing the results formally by way of Multi Objective Decision Analysis (MODA) each alternative design is compared consistently and decision making is simplified. Further, the utilization of defined interfaces and common architectures based on them enables alternative designs compliant with the Assessment Scenarios to be created quickly. By using Modelica as the modeling language the engineer benefits from accessibility to the rich simulation results of many variables and can compose System-Level Designs using extensive existing libraries.

However the approach and tool has further work to be done to it to make it more useful including: automation of the generation of alternative designs, support for parameter variation and implementation on a more complex design.

6 Conclusions

This paper has described an approach and tool for performing System-Level Design trade studies using Modelica. In the form of a Model Builder, Simulation Runner and Results Processor which take a suitable library and XML description file as input.

The aim of the approach was to consistently assess multiple design alternatives and summarize their results for fast comparison.

This was achieved by defining a common interface for the System of Interest (Solar-Boat) and placing it in an Assessment Scenario as a replaceable partial model into which programmatically, different SolarBoat alternatives were placed by the Model Builder. Each of these was then simulated and the results processed by Multi Objective Decision Analysis (MODA).

The rapid automated assessment of the alternatives and processing of results by MODA enables engineers to quickly understand the benefits of different designs, but by retaining the rich results associated with Modelica simulation further (manual) analysis can be performed to gain greater insight into how individual components are performing.

This was demonstrated for some simple examples of several Solar-Boat alternative System-Level Designs being subject to the same four different assessment scenarios.

References

Cilli, M. V., & Parnell, G. S. (2014). Systems engineering tradeoff study process framework. In 24th INCOSE Int'l Symposium, Las Vegas, NV.

DyMat 0.7 (2015), A package for reading and processing the result files of Dymola and OpenModelica. https://pypi.python.org/pypi/DyMat

Edwards, S., Cilli, M. V., Peterson, T., Zabat, M., Lawton, C., & Shelton, L. (2015). Whole Systems Trade Analysis. In 25th INCOSE Int'l Symposium, Seattle. Seattle, WA, USA.

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

ISO. (2015). Automation systems and integration -- Object-Process Methodology (No. ISO/PAS 19450:2015).

Parnell, G. S., Cilli, M. V., & Buede, D. (2014). Tradeoff Study Cascading Mistakes of Omission and Commission. INCOSE International Symposium, Las Vagas.

SEBoK. (2015). Guide to the Systems Engineering Body of Knowledge (SEBoK).

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

Sutherland, J. (2016, March 4). Knowledge Management and System-Level Design Tools utilizing OPM and Modelica for a Student Solar-Boat Project (Master's Thesis). University of Tokyo, Tokyo Japan.

Ulrich, K., & Eppinger, S. (2011). Product Design and Development (5 edition). New York: McGraw-Hill/Irwin.