Knowledge Management and System-Level Design Tools utilizing OPM and Modelica for a Student Solar-Boat Project

Joshua Sutherland

2016/02/03

Overview

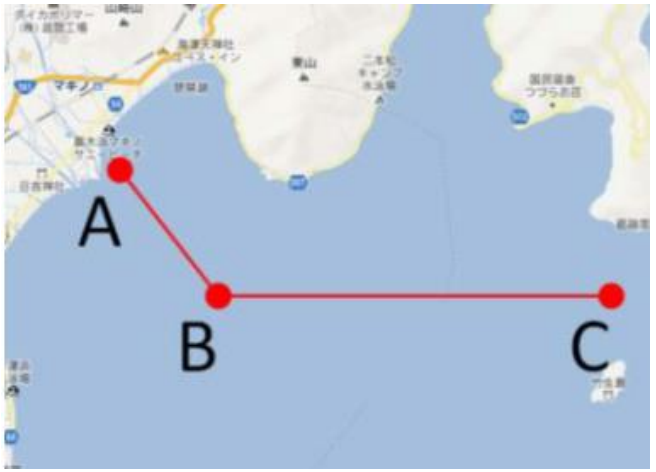
1. Problems and proposed solutions identified from the 2014 Solar-Boat project
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. Examples and demonstrations
4. Discussion
5. Conclusions
6. PhD plans

Overview

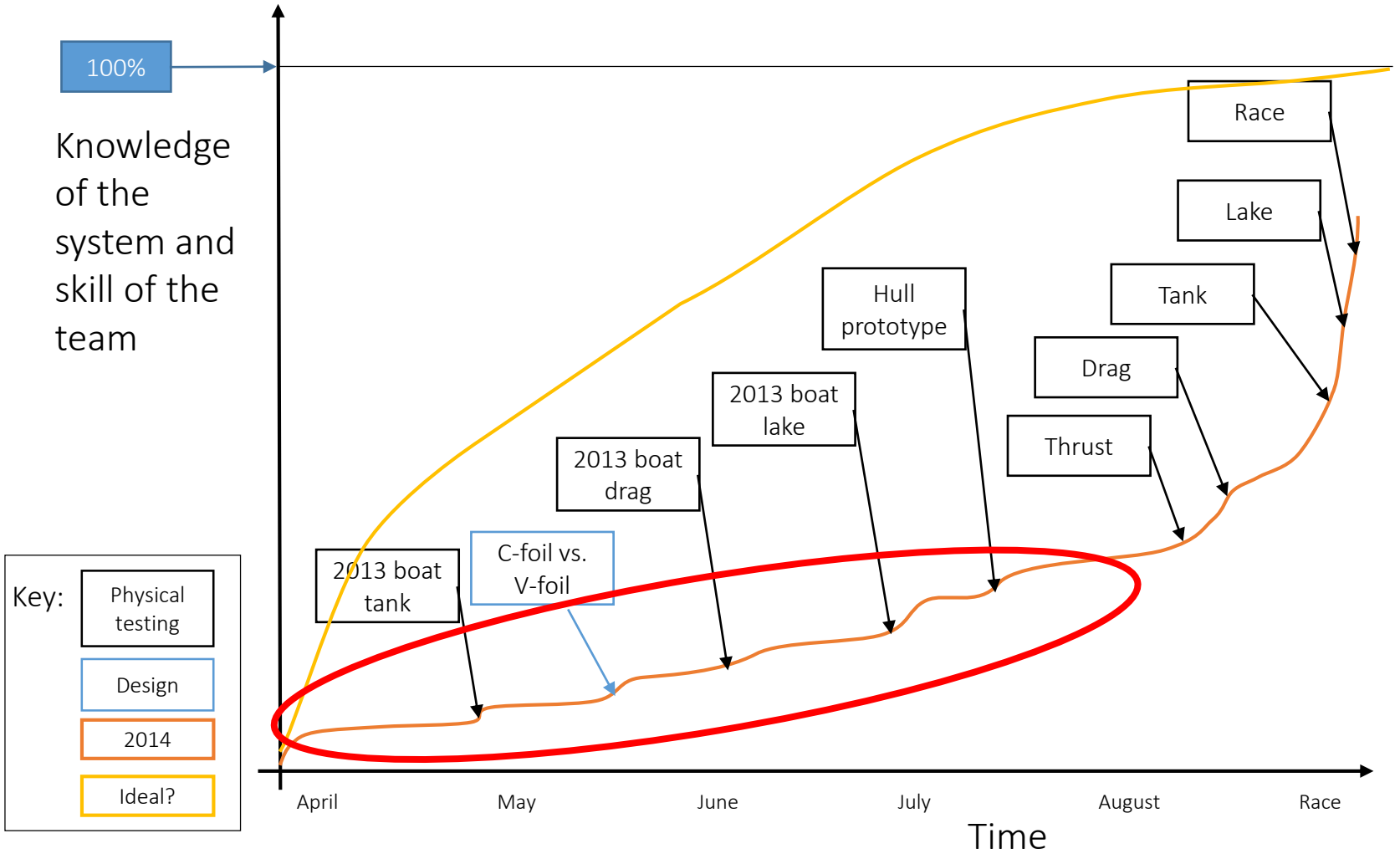
1. **Problems and proposed solutions identified from the 2014 Solar-Boat project**
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. Examples and demonstrations
4. Discussion
5. Conclusions
6. PhD plans

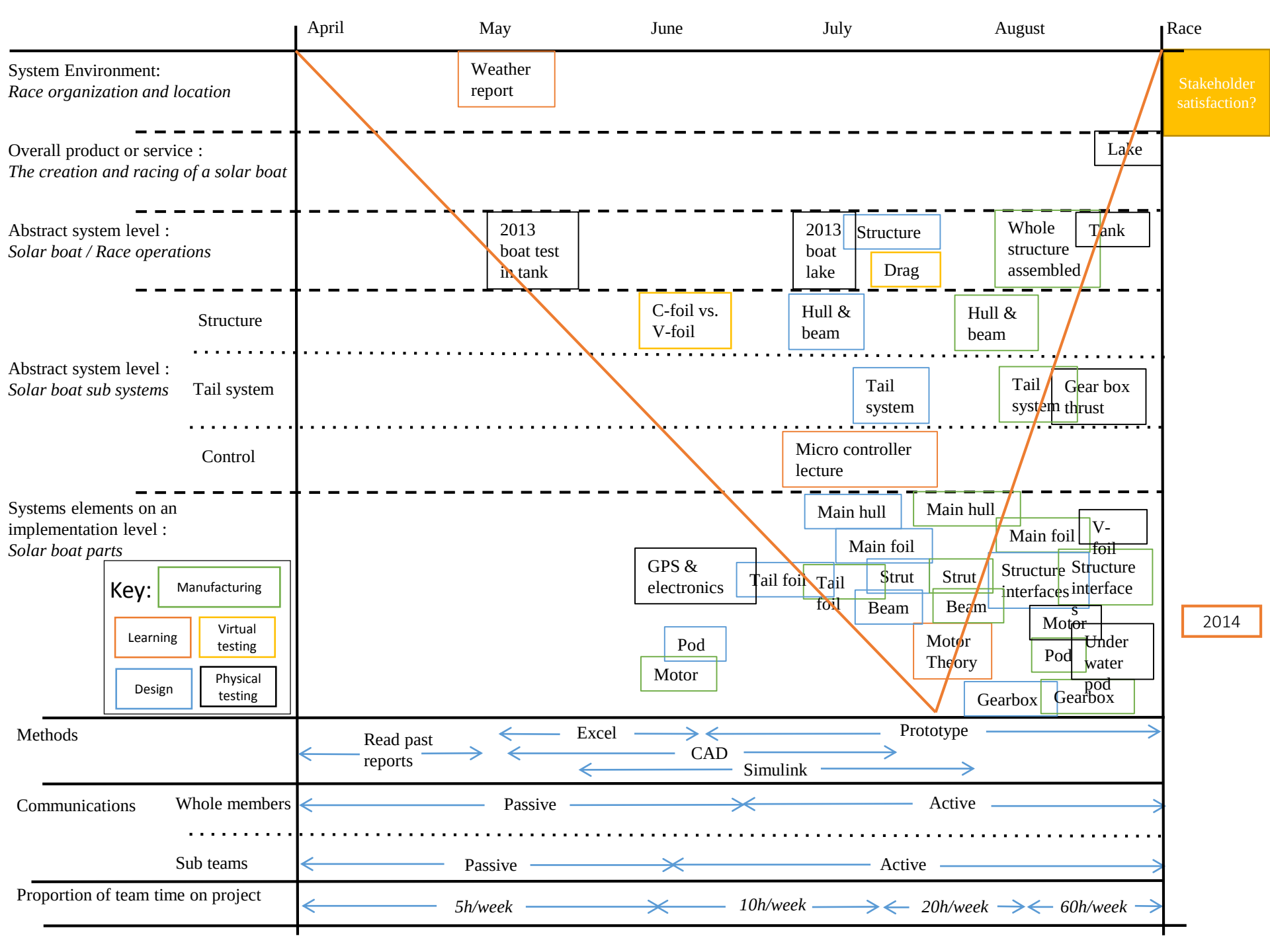
Knowledge Management and System-Level Design Tools utilizing OPM and Modelica for a Student Solar-Boat Project

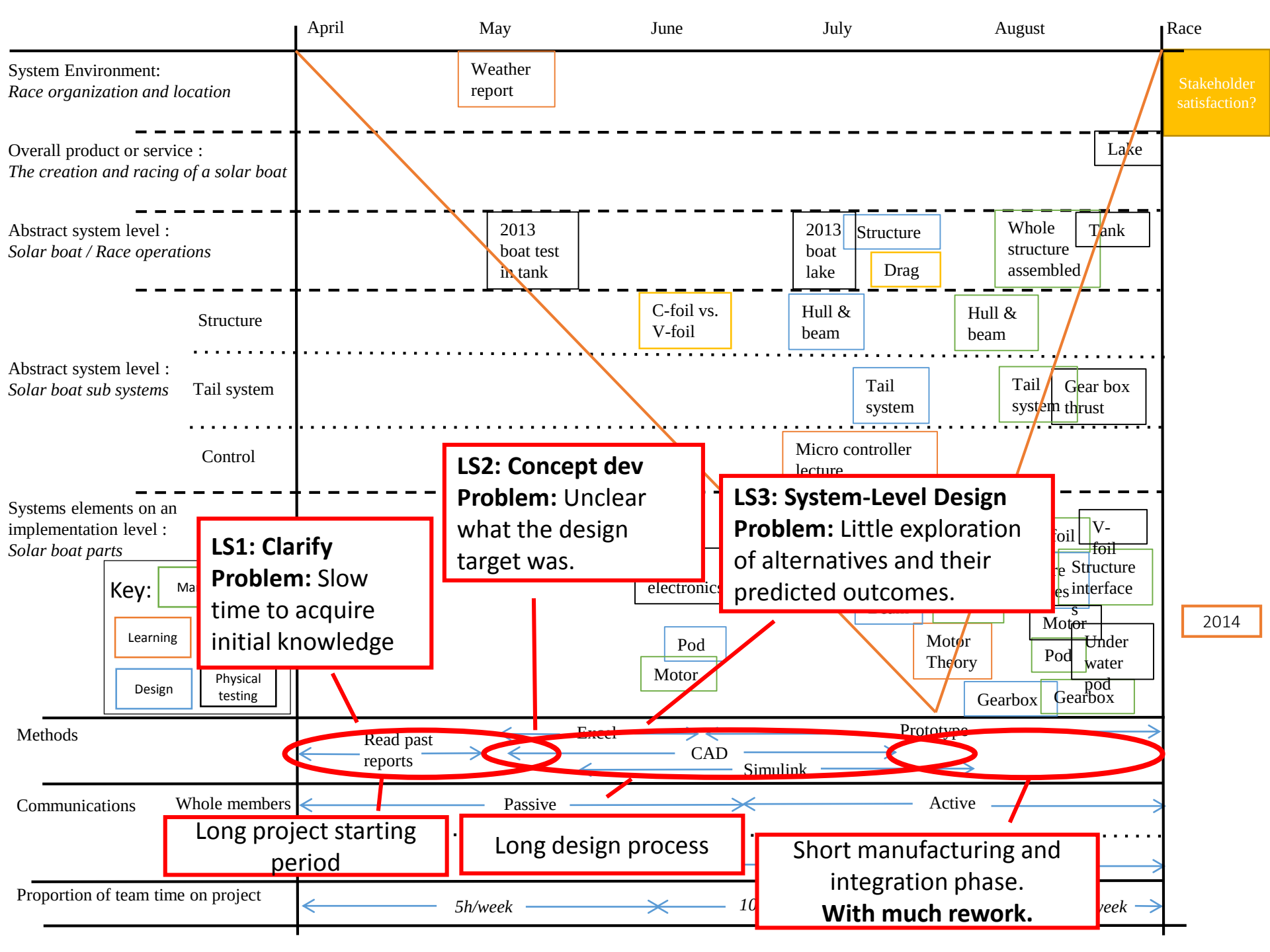
- Lake Biwa (Japan) competition requirements:
  - Max 2m^2 of solar panels
  - Max 25Wh of lead based batteries
  - Complete the 20km course autonomously
  - Very small required cargo (64g 68mmx46mmx18mm)



Knowledge / Skill Growth curve: SolarBoat 2014




April
May
June
July
August
Race
System Environment:
Race organization and location
Weather report
Stakeholder satisfaction?
Overall product or service :
The creation and racing of a solar boat
Lake
Abstract system level :
Solar boat / Race operations
2013 boat test in tank
2013 boat lake
Structure
Drag
Whole structure assembled
Tank
Structure
C-foil vs. V-foil
Hull & beam
Hull & beam
Abstract system level :
Solar boat sub systems
Tail system
Tail system
Tail system
Gear box thrust
Control
Micro controller lecture
Systems elements on an implementation level :
Solar boat parts
Main hull
Main hull
Main foil
V-foil
Main foil
GPS & electronics
Tail foil
Tail foil
Strut
Strut
Structure interfaces
Structure interfaces
Beam
Beam
Motor
Pod
Motor
Motor Theory
Pod
Under water pod
Gearbox
Gearbox
Key:
Manufacturing
Learning
Virtual testing
Design
Physical testing
2014
Methods
Read past reports
Excel
CAD
Simulink
Prototype
Communications
Whole members
Passive
Active
Sub teams
Passive
Active
Proportion of team time on project
5h/week
10h/week
20h/week
60h/week


April
May
June
July
August
Race
System Environment:
Race organization and location
Weather report
Stakeholder satisfaction?
Overall product or service :
The creation and racing of a solar boat
Lake
Abstract system level :
Solar boat / Race operations
2013 boat test in tank
2013 boat lake
Structure
Drag
Whole structure assembled
Tank
Structure
C-foil vs. V-foil
Hull & beam
Hull & beam
Abstract system level :
Solar boat sub systems
Tail system
Tail system
Tail system
Gear box thrust
Control
Micro controller lecture
Systems elements on an implementation level :
Solar boat parts
LS1: Clarify
Problem: Slow time to acquire initial knowledge
LS2: Concept dev
Problem: Unclear what the design target was.
LS3: System-Level Design
Problem: Little exploration of alternatives and their predicted outcomes.
Key:
Learning
Design
Physical testing
electronics
V-foil
Structure
interface
Motor
Pod
Motor
Motor Theory
Pod
Under water pod
Gearbox
Gearbox
2014
Methods
Read past reports
Excel
CAD
Simulink
Prototype
Communications
Whole members
Passive
Active
Long project starting period
Long design process
Short manufacturing and integration phase.
With much rework.
Proportion of team time on project
5h/week

Idealized SolarBoat lifecycle stages

Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design	LS4: Detail Design	LS5: Production, Test and Refinement	LS6: Race	LS7: Knowledge transfer
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design	Define 3D specifications of components.	Incrementally build, test and refine the boat.	Race the boat and repair as needed.	Package up knowledge for the next years team.

Focus on the early lifecycle stages

Idealized SolarBoat lifecycle stages

Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design
Identified problems:	Slow time to acquire initial knowledge	Unclear what the design target was	Little exploration of alternatives and their predicted outcomes

Some **alternative** SolarBoat physical architectures



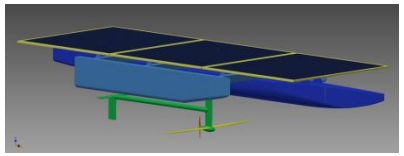










Idealized SolarBoat lifecycle stages

Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design
Identified problems:	Slow time to acquire initial knowledge	Unclear what the design target was	Little exploration of alternatives and their predicted outcomes
Proposed solutions:	Provide knowledge in models	Complete trade-off analysis of multiple designs using models to simulate performance	

Idealized SolarBoat lifecycle stages

Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design
Identified problems:	Slow time to acquire initial knowledge	Unclear what the design target was	Little exploration of alternatives and their predicted outcomes
Proposed solutions:	Provide knowledge in models	Complete trade-off analysis of multiple designs using models to simulate performance	

What knowledge for management

	Content:
Race:	Race rules
	Environmental inputs
SolarBoat project:	Project intension
	Resources
	Design processes
	Manufacturing processes
	Testing processes
SolarBoat:	Design (including alternative designs)
	Predicted performance
	Tested performance

All of which change with time

Idealized SolarBoat lifecycle stages

Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design
Identified problems:	Slow time to acquire initial knowledge	Unclear what the design target was	Little exploration of alternatives and their predicted outcomes
Proposed solutions:	Provide knowledge in models	Complete trade-off analysis of multiple designs using models to simulate performance	
Problems / Difficulties with implementing solutions:	• What languages? • Integrate multiple languages? • Keeping models update	• Framework to assess all alternative designs • Comparing a reasonable number of alternatives • Numerical optimization vs. exploratory approaches	

Idealized SolarBoat lifecycle stages

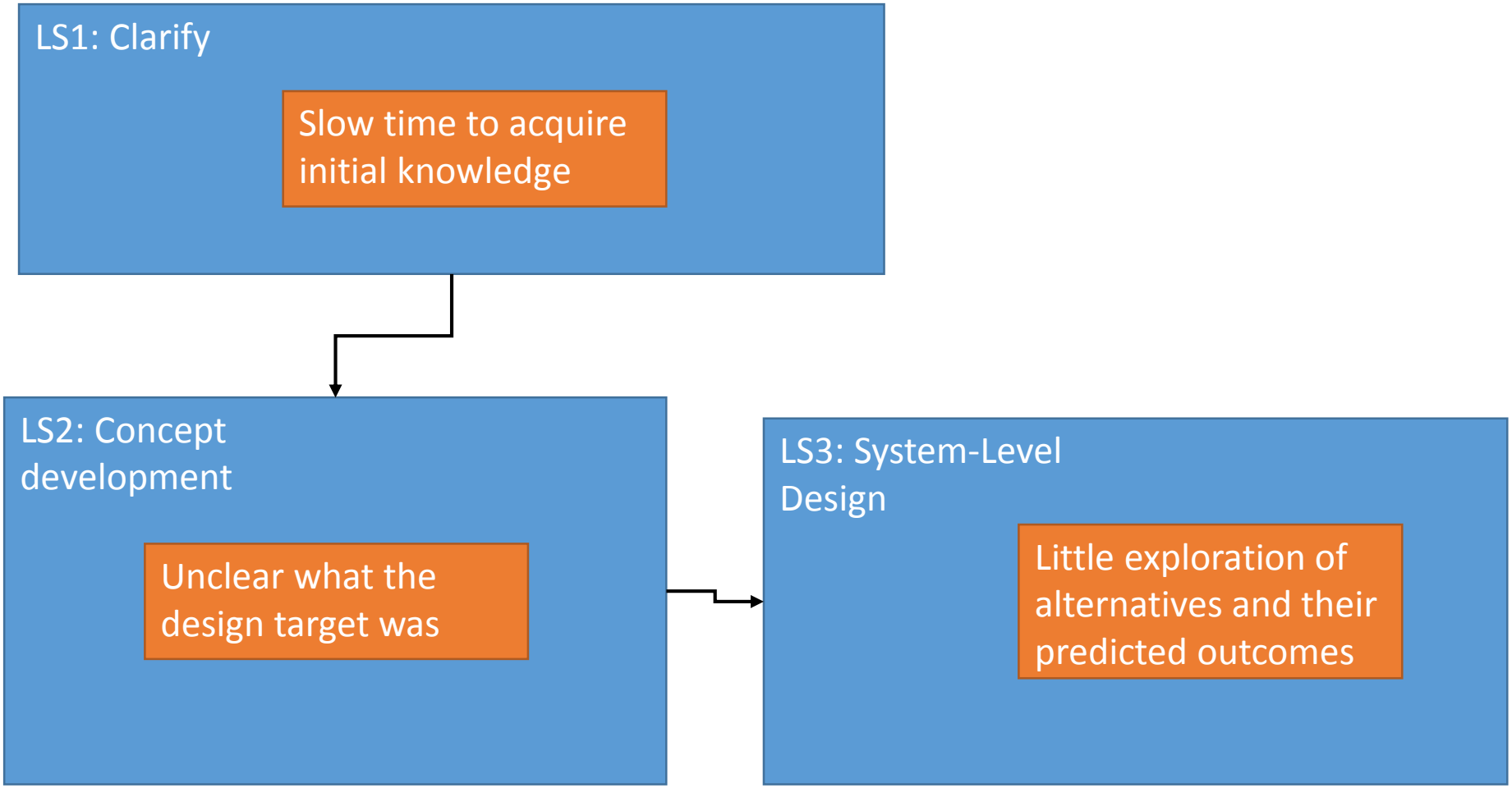
Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design
Identified problems:	Slow time to acquire initial knowledge	Unclear what the design target was	Little exploration of alternatives and their predicted outcomes
Proposed solutions:	Provide knowledge in models	Complete trade-off analysis of multiple designs using models to simulate performance	
Problems / Difficulties with implementing solutions:	• What languages? • Integrate multiple languages? • Keeping models update	• Framework to assess all alternative designs • Comparing a reasonable number of alternatives • Numerical optimization vs. exploratory approaches	
Thesis aim:	To propose tools and methodologies to help students: • Manage project knowledge • Explore concept designs		

Overview

1. Problems and proposed solutions identified from the 2014 Solar-Boat project
2. **Proposed tools and methodologies for Knowledge Management and System-Level Design**
3. Examples and demonstrations
4. Discussion
5. Conclusions
6. PhD plans

Proposed tools and methodologies for Knowledge Management and System-Level Design

2014 Problems



Proposed tools and methodologies for Knowledge Management and System-Level Design

Design process questions

LS1: Clarify

1. What is project **value**?
2. What are the **measures of performance**?
3. What **resources** are available?

LS2: Concept development

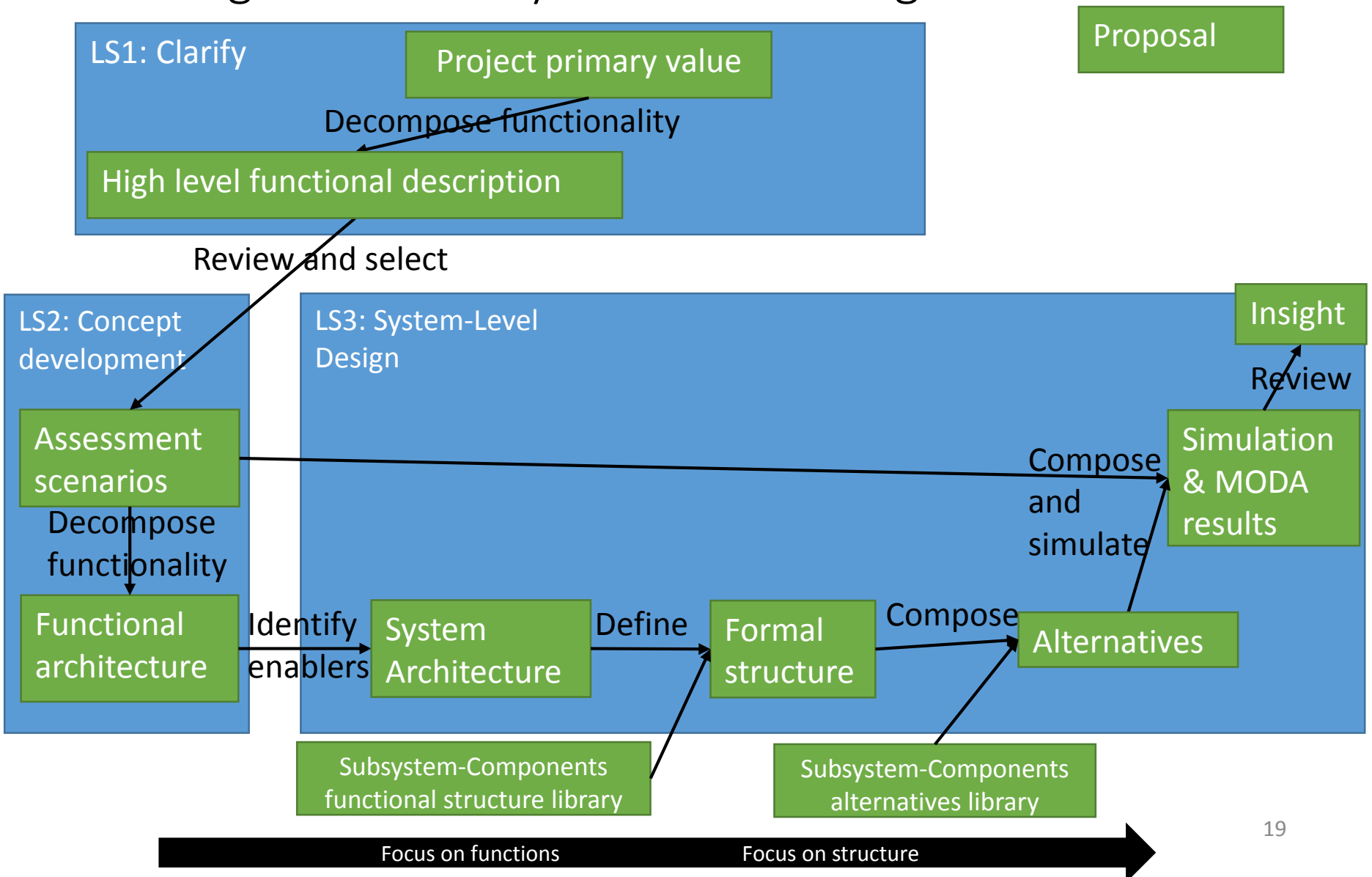
1. What must a **SolarBoat do**?
2. **What subsystems** do they need?
3. How can they all be **assessed and compared**?

LS3: System-Level Design

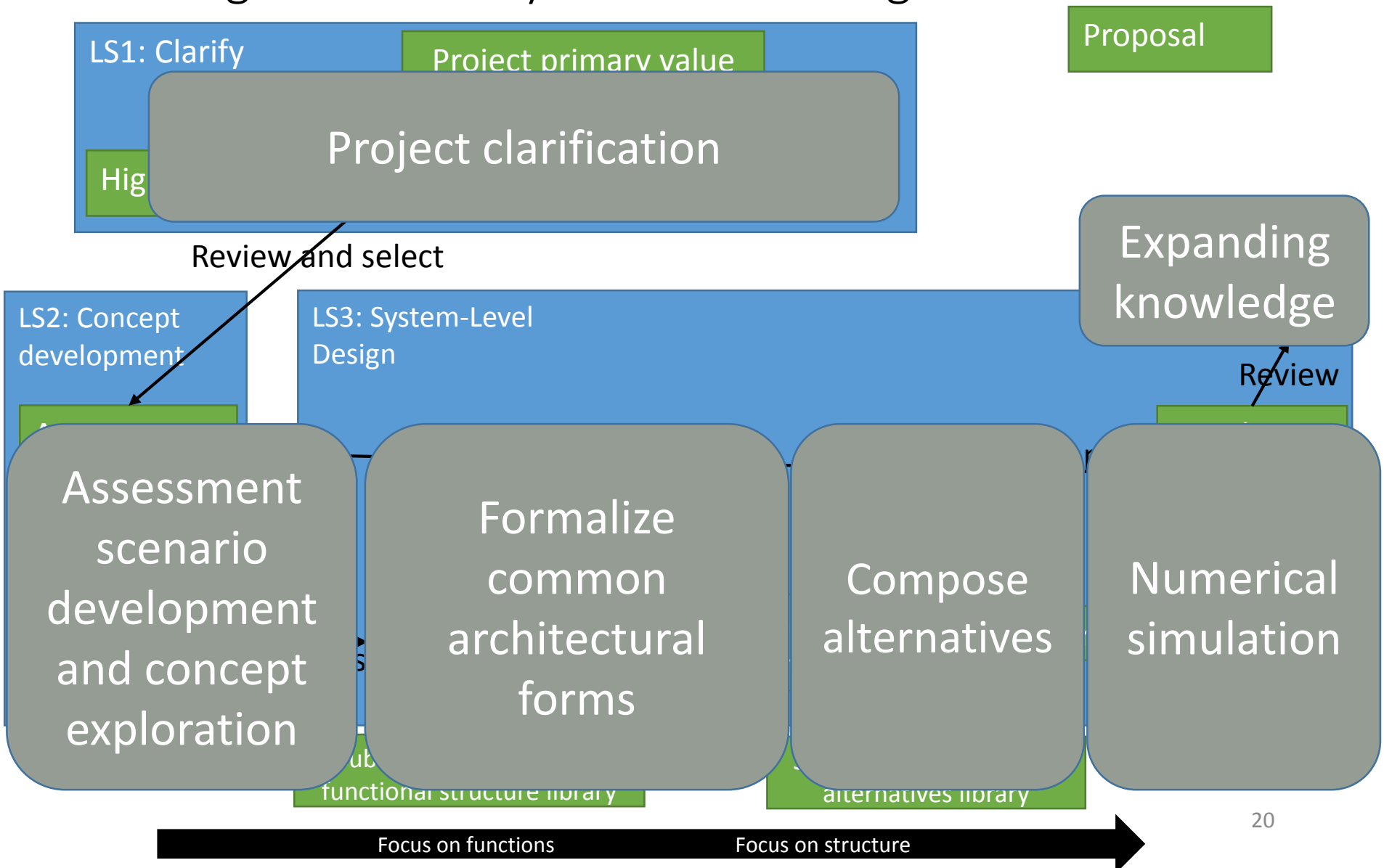
1. What are valid **alternative designs**?
2. What is the **designs predicted performance** how do they **compare**?
3. Are there **better designs**?

Focus on functions → Focus on structure

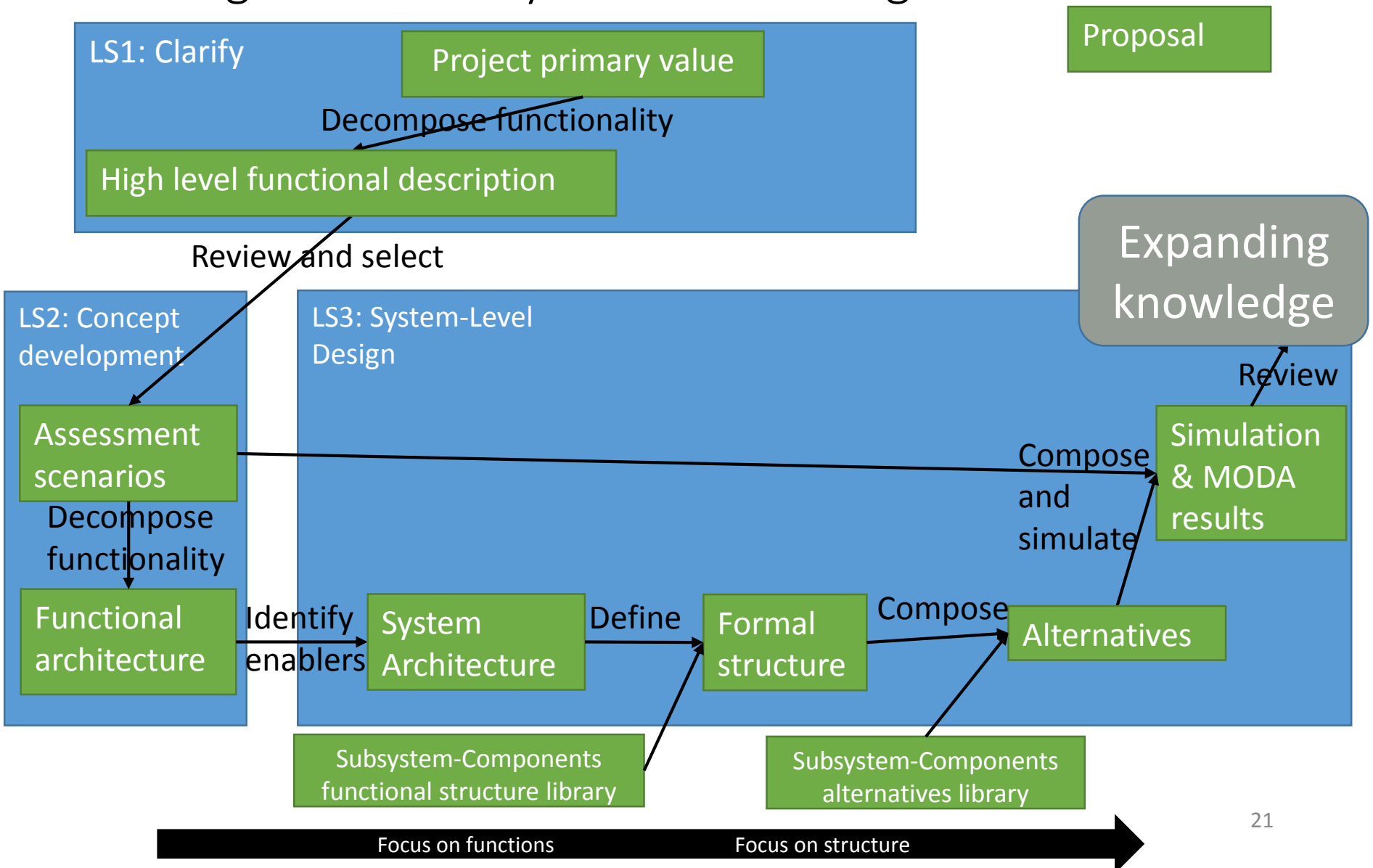
Proposed tools and methodologies for Knowledge Management and System-Level Design



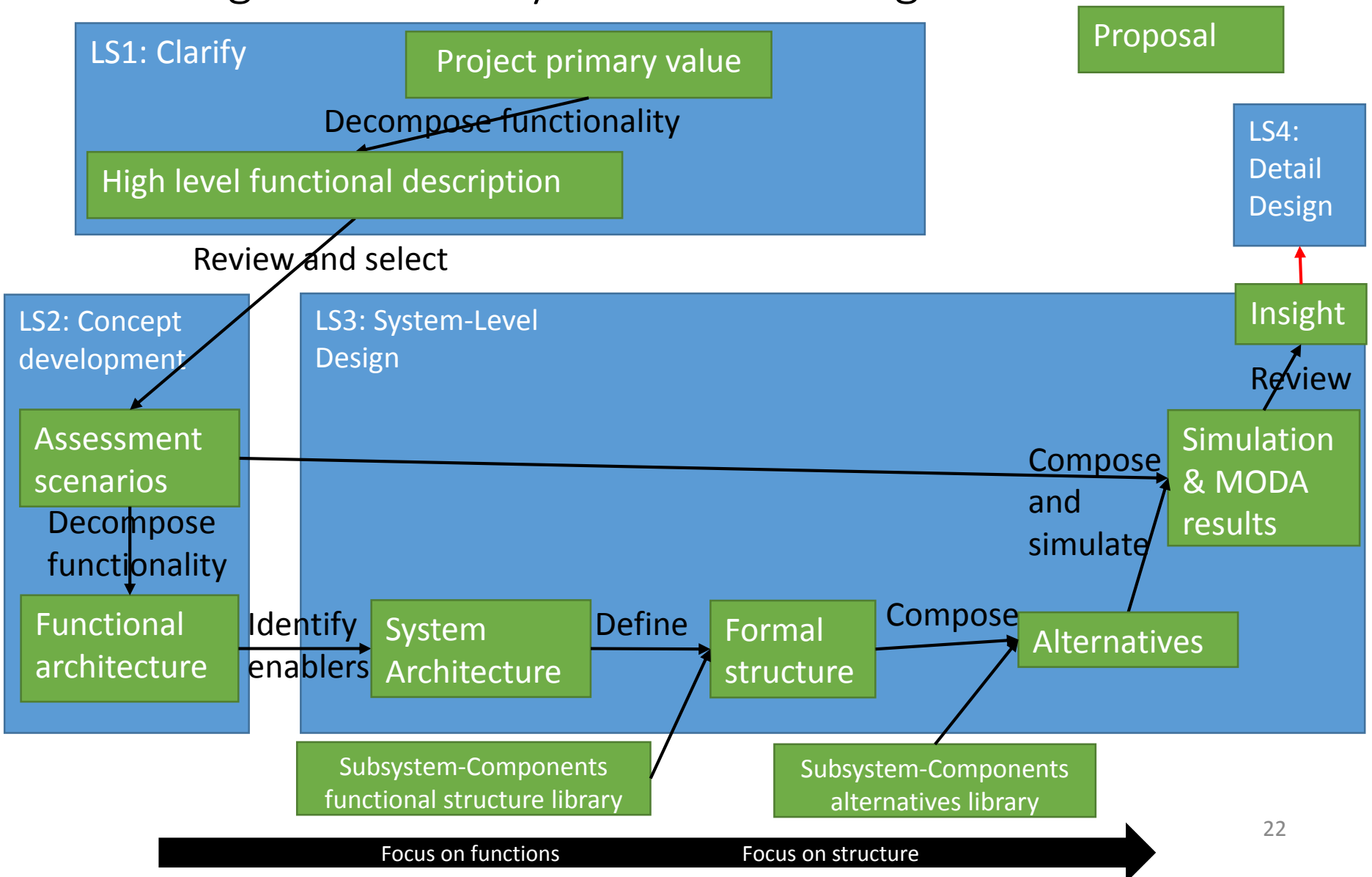
Proposed tools and methodologies for Knowledge Management and System-Level Design



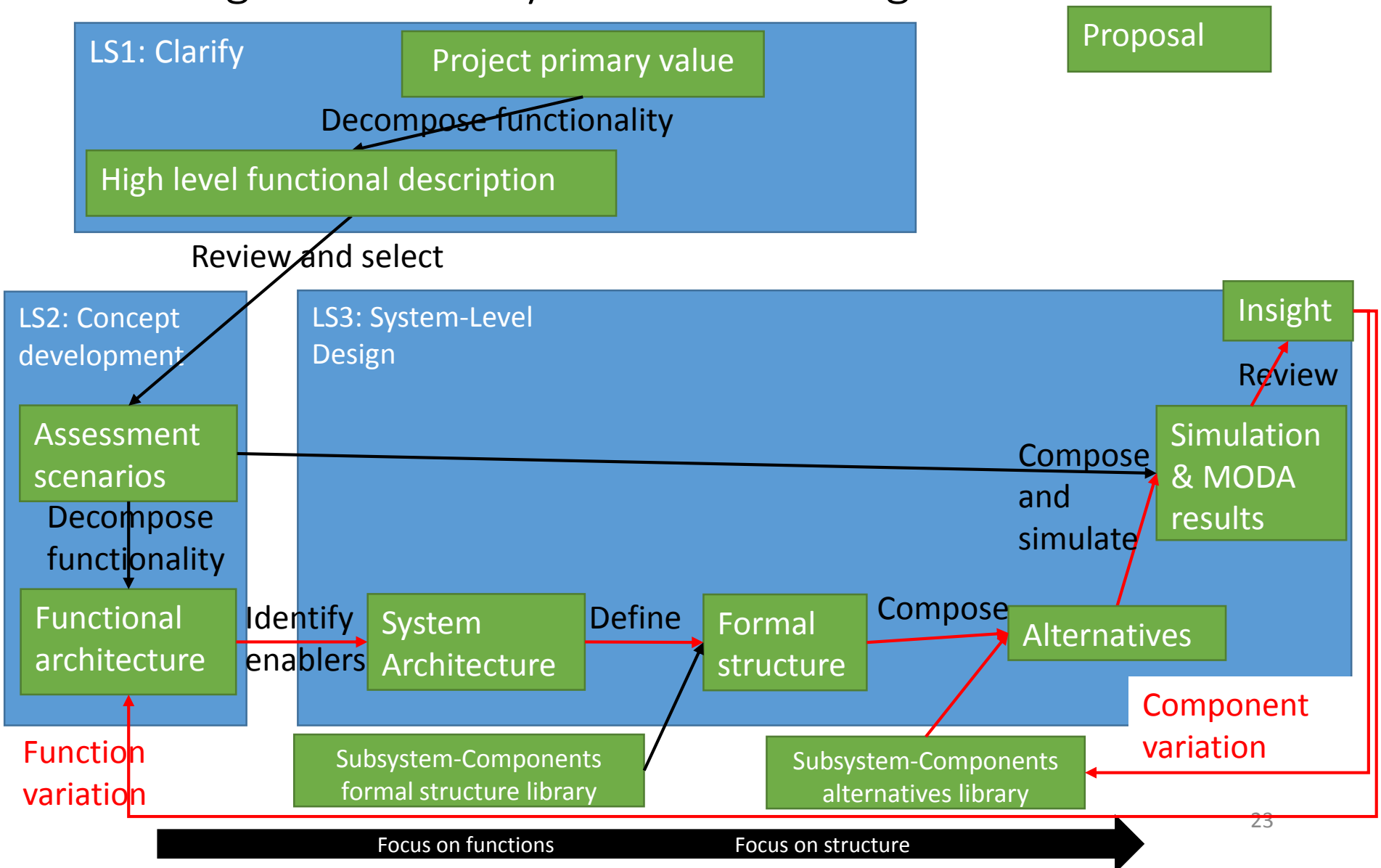
Proposed tools and methodologies for Knowledge Management and System-Level Design



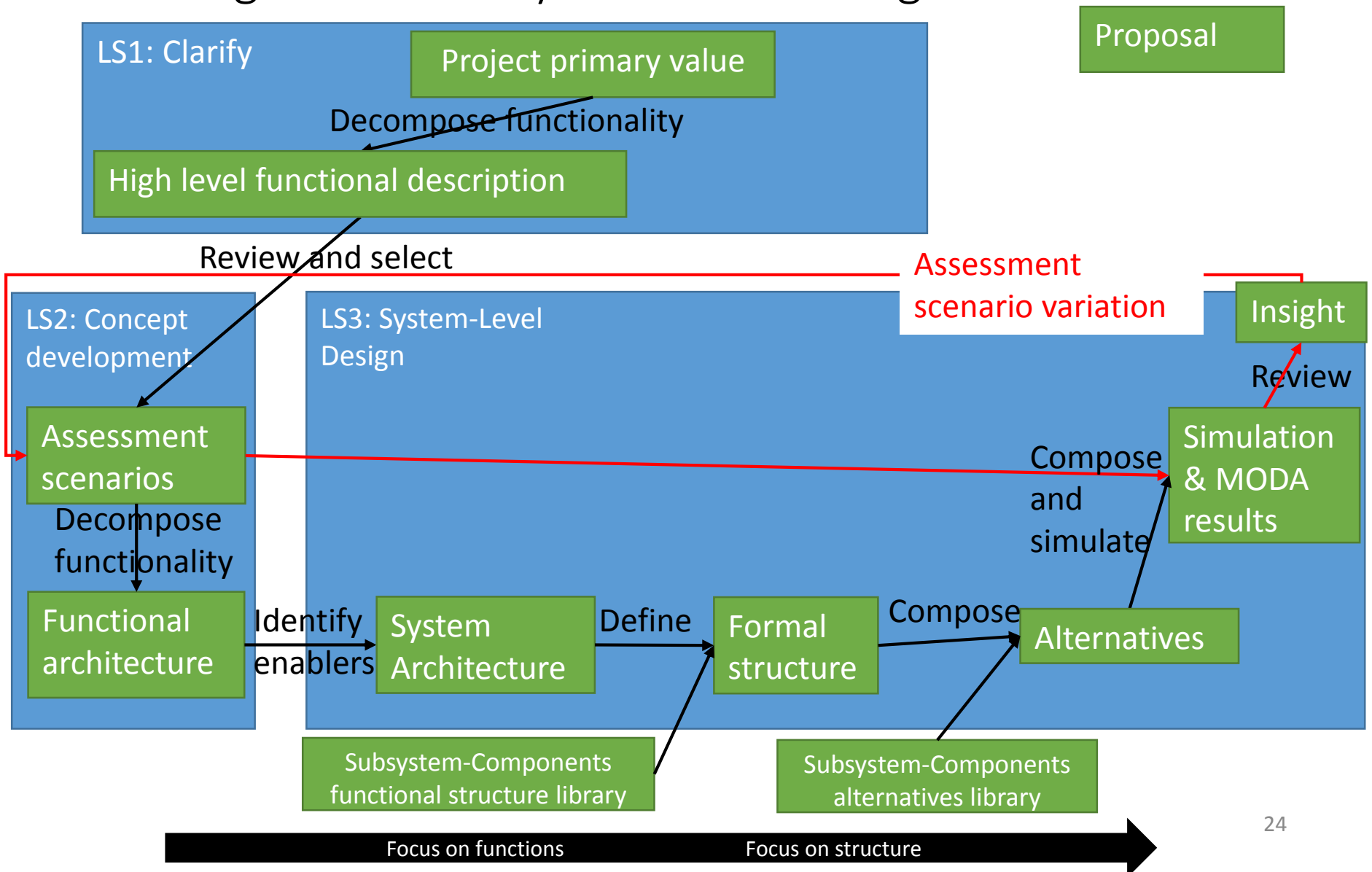
Proposed tools and methodologies for Knowledge Management and System-Level Design



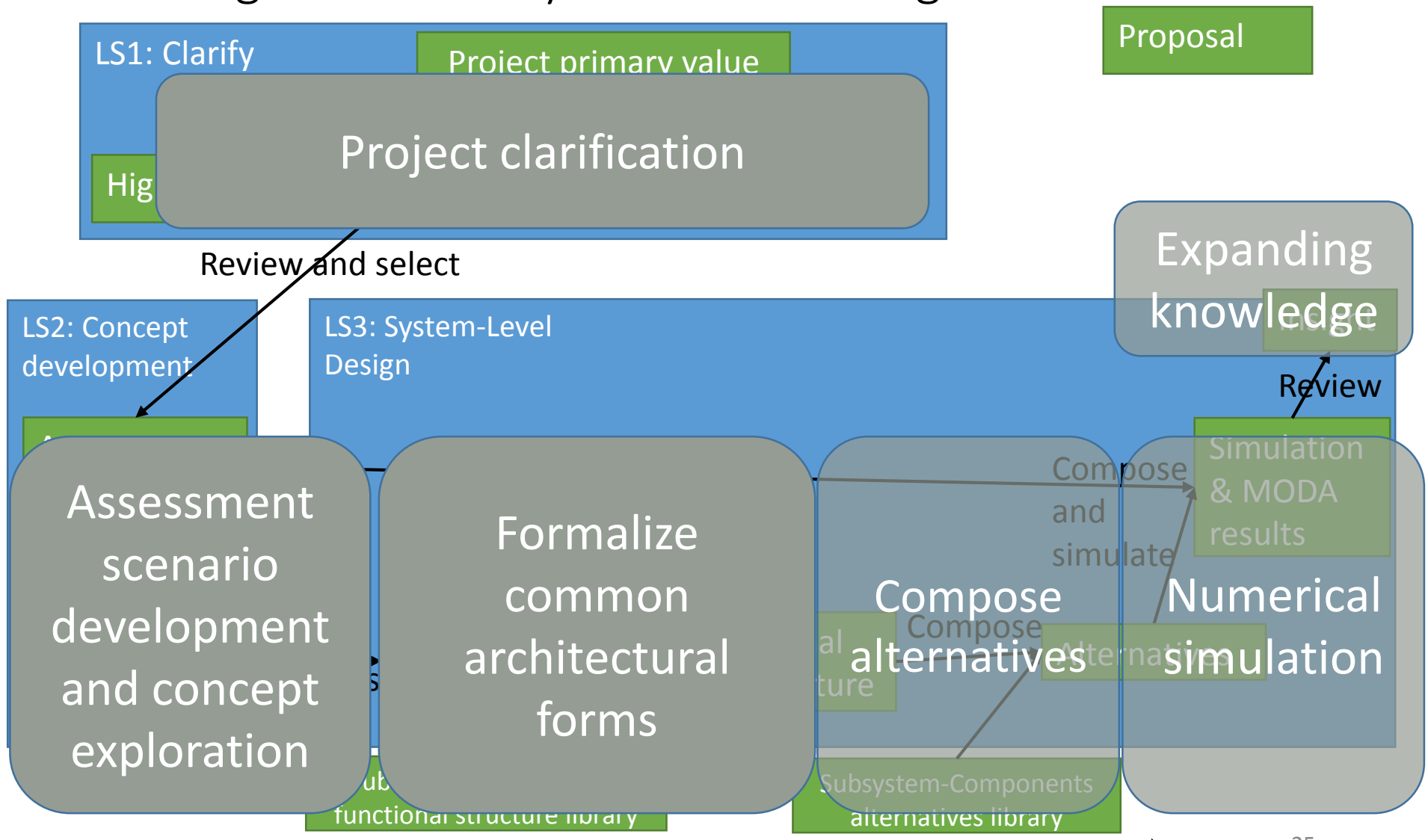
Proposed tools and methodologies for Knowledge Management and System-Level Design



Proposed tools and methodologies for Knowledge Management and System-Level Design

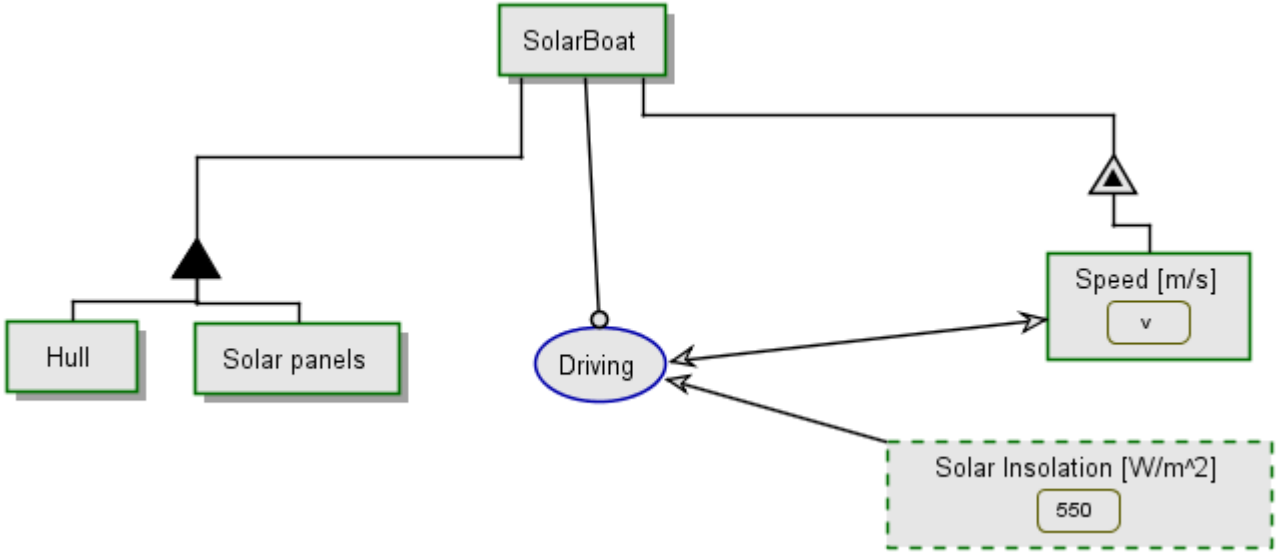


Proposed tools and methodologies for Knowledge Management and System-Level Design



Object Process Methodology (OPM)

- ISO standardized conceptual modeling language
- **Single diagram type** accompanied with text models **behavior** and **structure. Complexity managed** by **hierarchical decomposition**
- Well suited for modeling system for students

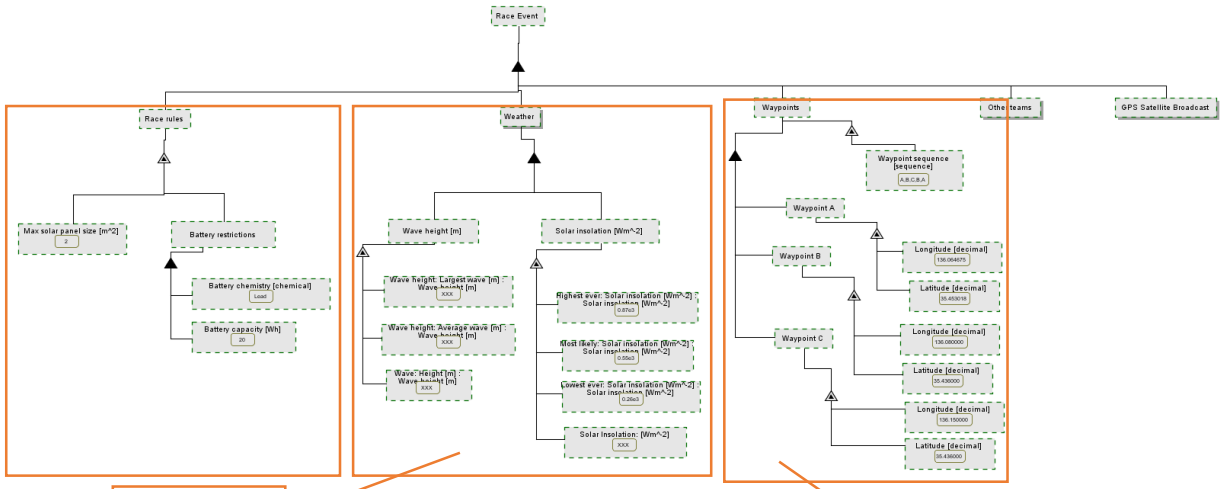


SolarBoat is physical.
SolarBoat exhibits Speed [m/s].
Speed [m/s] is v.
SolarBoat consists of Hull and Solar panels.
Hull is physical.
Solar panels is physical.
Solar Insolation [W/m^2] is environmental.
Solar Insolation [W/m^2] is 550.
Driving requires SolarBoat.
Driving affects Speed [m/s].
Driving consumes Solar Insolation [W/m^2].

Modeling more detail

Race

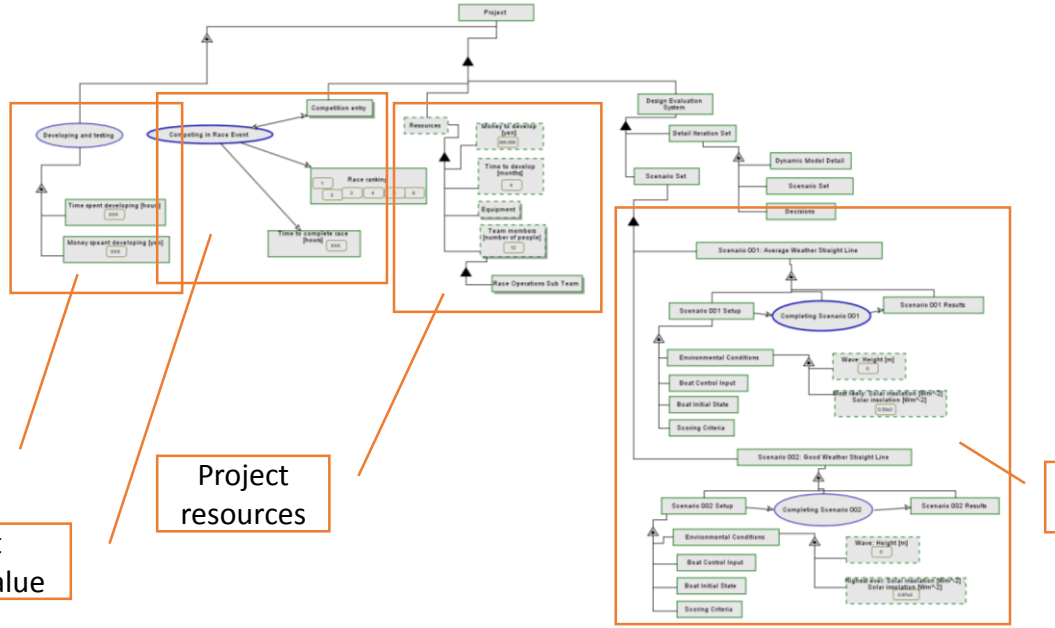
Project



Race rules

Weather ranges

Waypoints



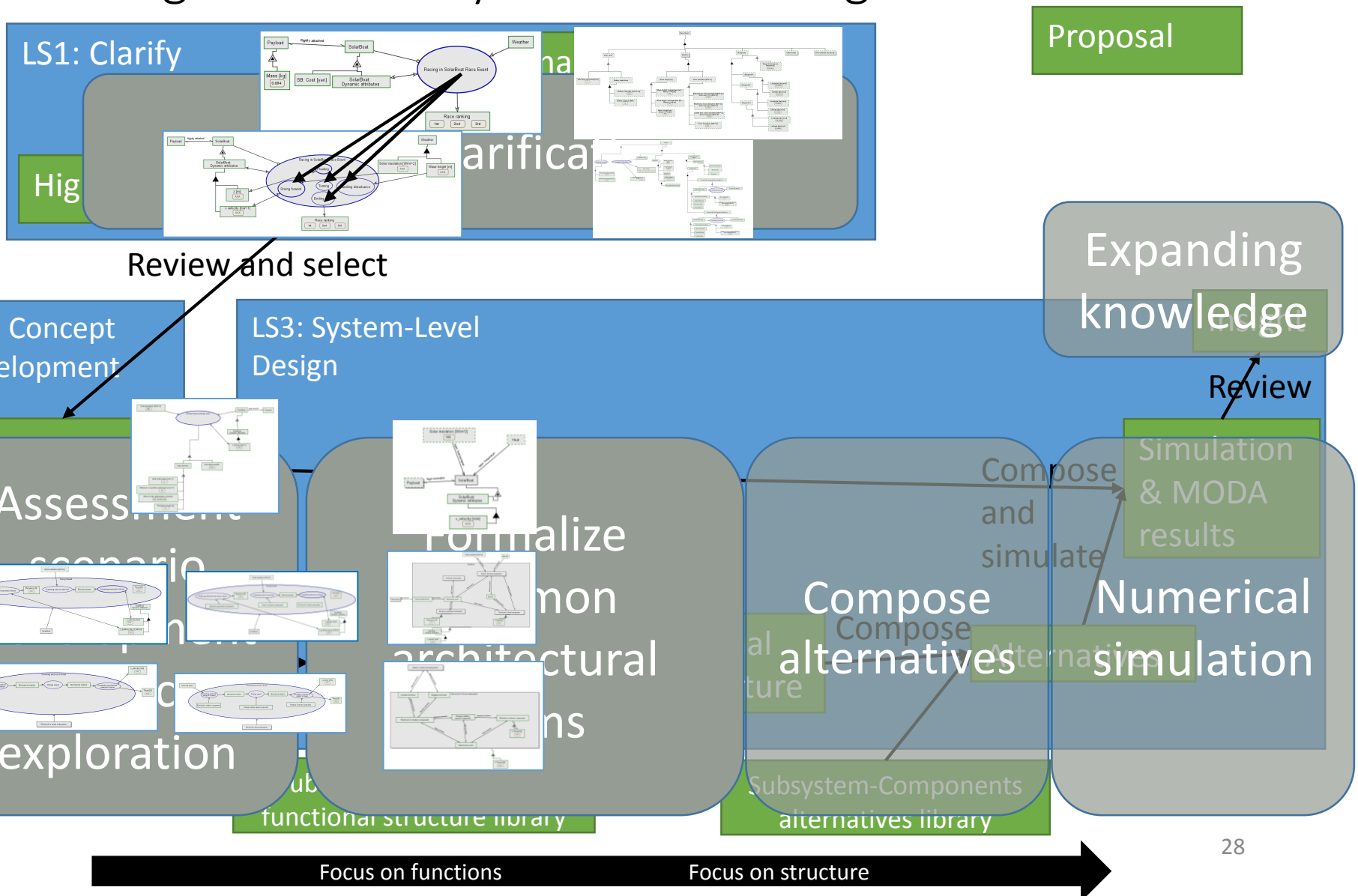
Development and test

Project primary value

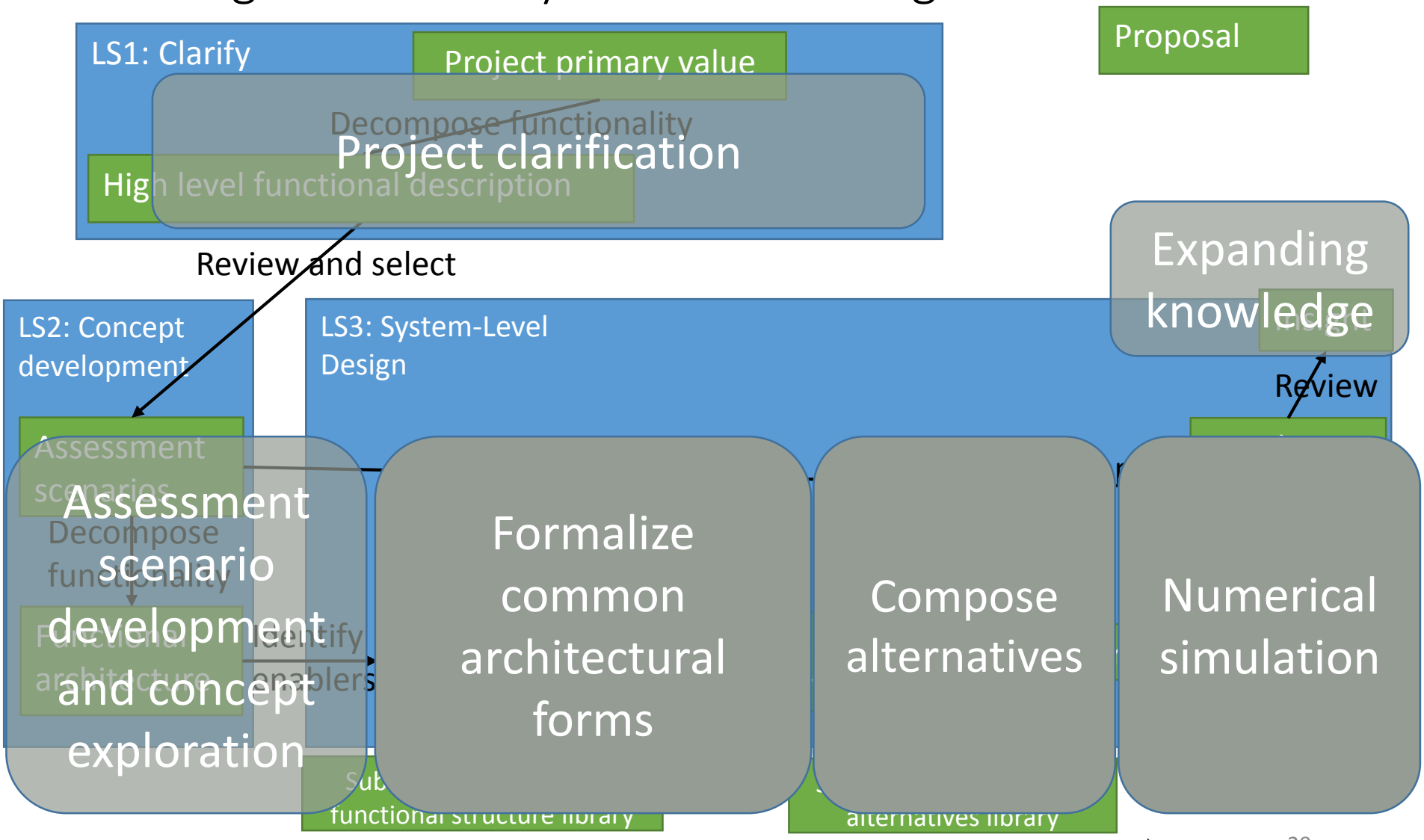
Project resources

Assessment criteria

Proposed tools and methodologies for Knowledge Management and System-Level Design

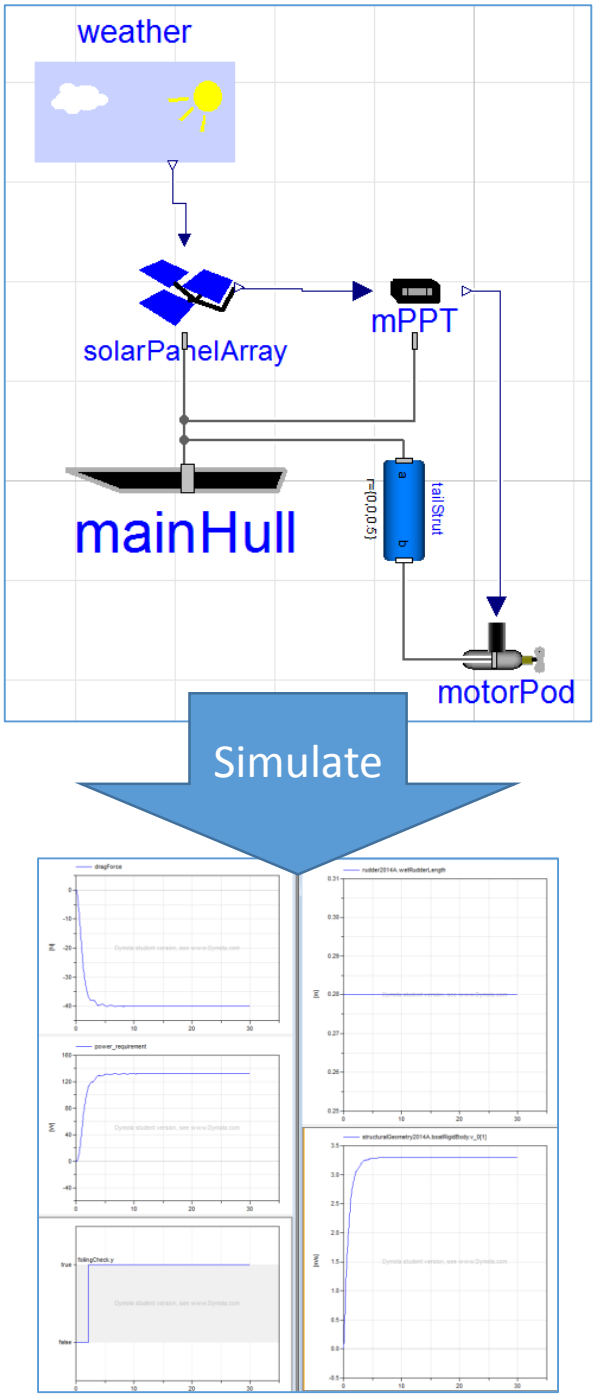


Proposed tools and methodologies for Knowledge Management and System-Level Design

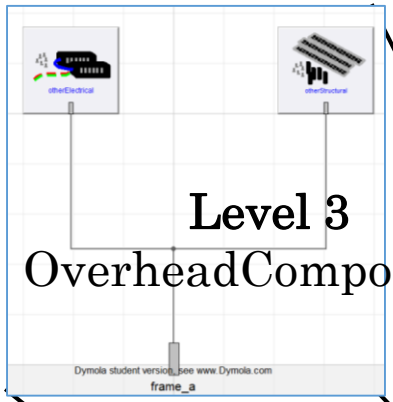


Modelica

- Multi domain numerical simulation modeling system
- Supports acausal component interactions
- Compose using existing parts libraries or make your own
- Supports hierarchy
- Well suited for early stage Solar-Boat modeling

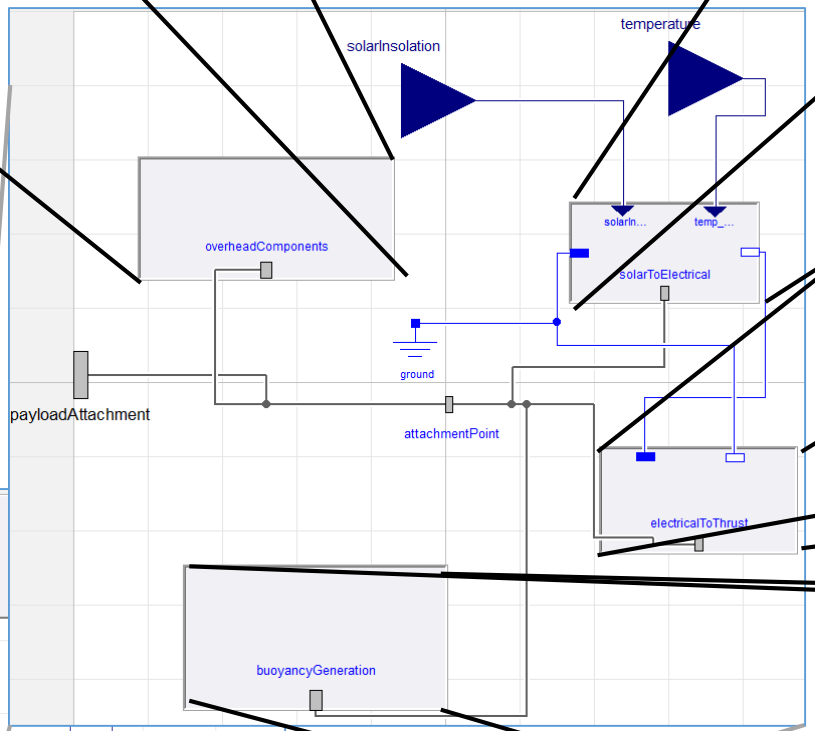


Hierarchy

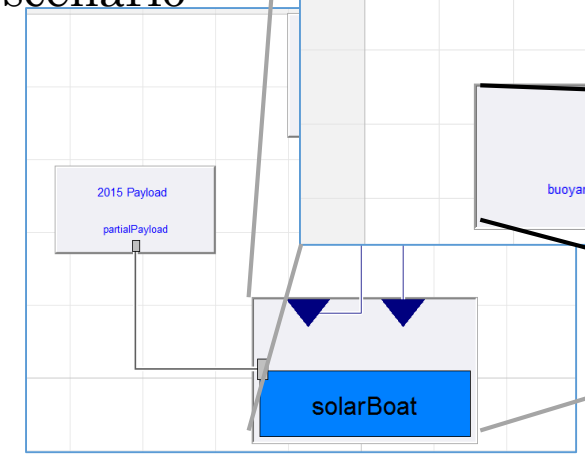


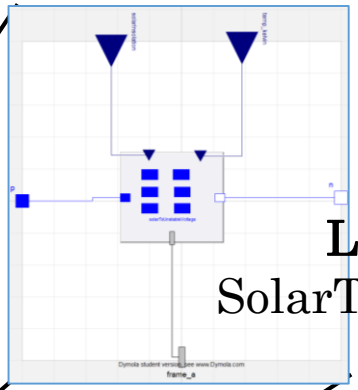
Level 3
OverheadComponents

Level 2
Solar-Boat

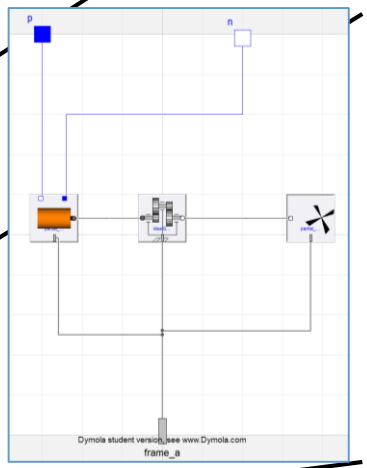


Level 1
Assessment scenario

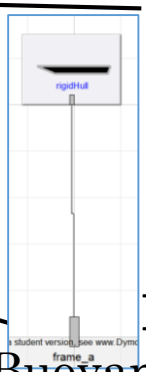




Level 3
SolarToElectrical

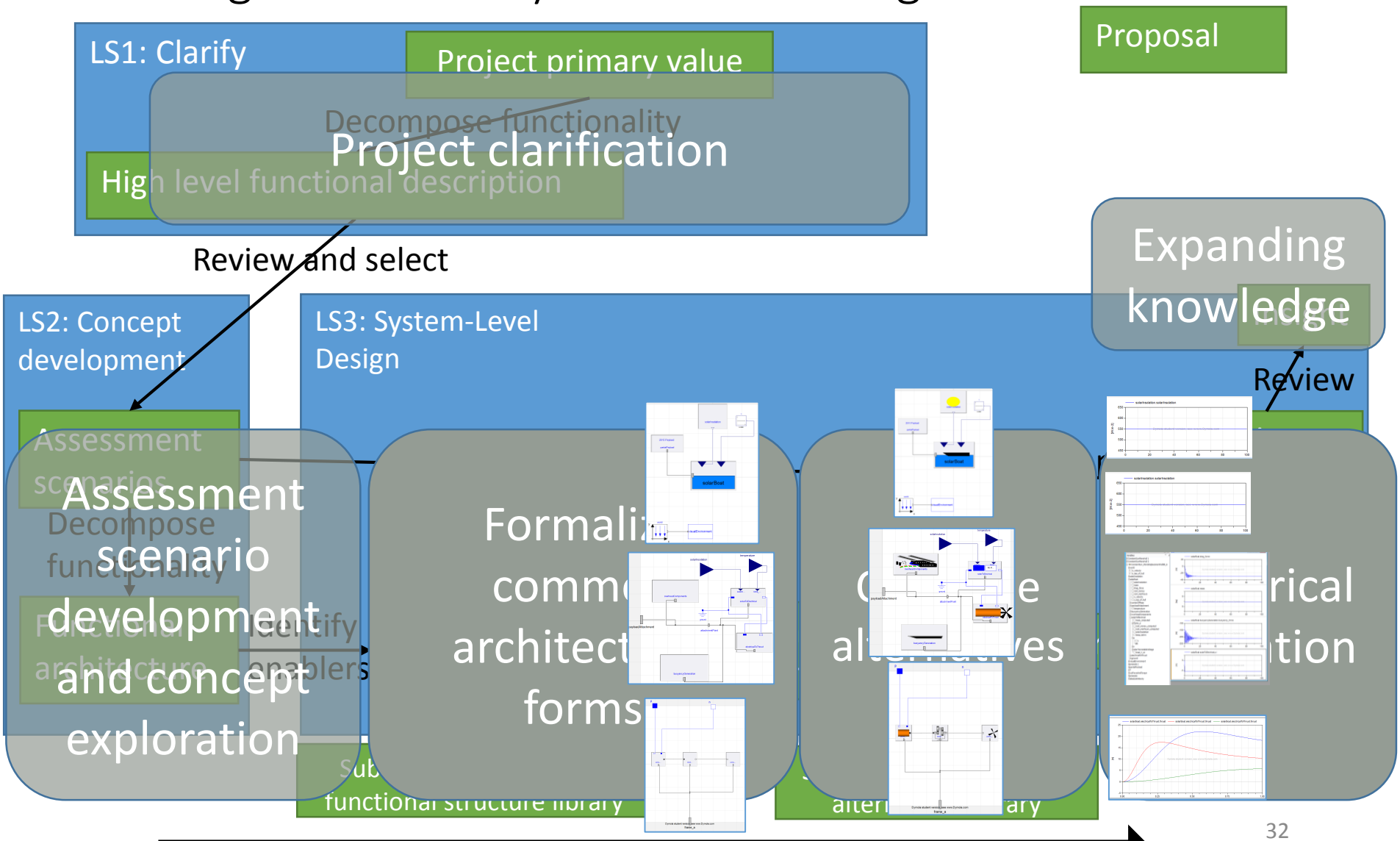


Level 3
ElectricalToThrust

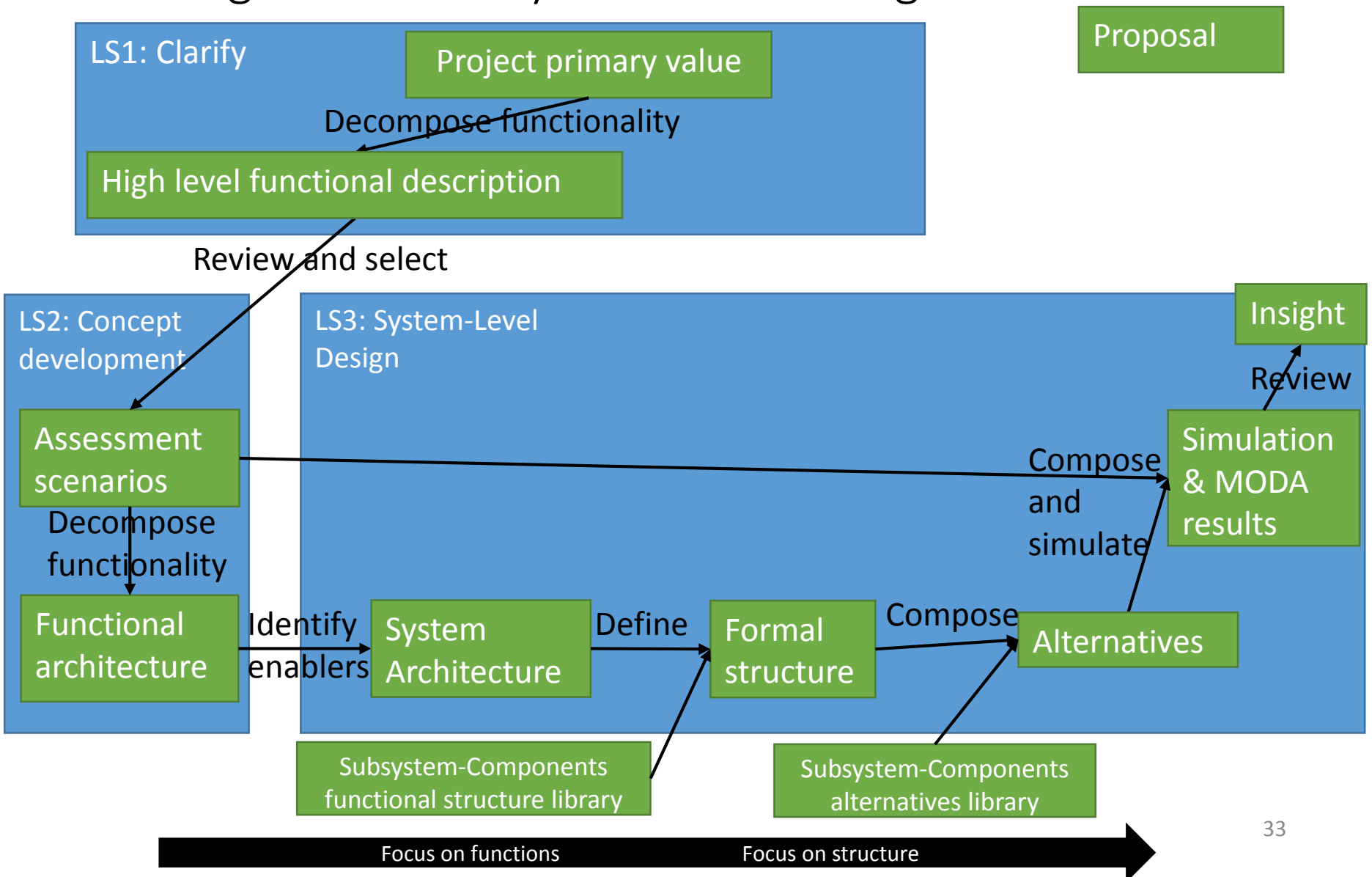


Level 3
BuoyancyGeneration

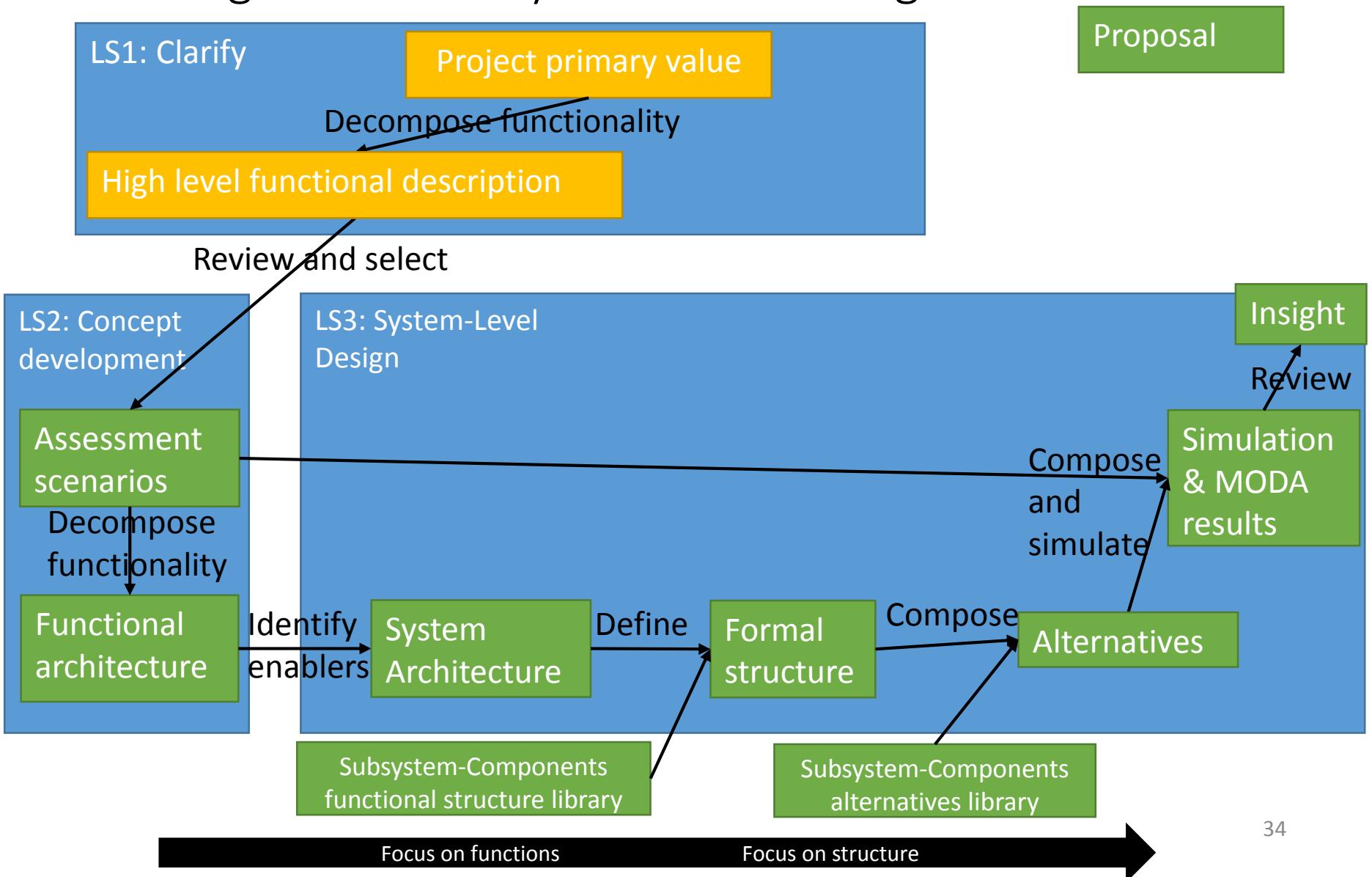
Proposed tools and methodologies for Knowledge Management and System-Level Design



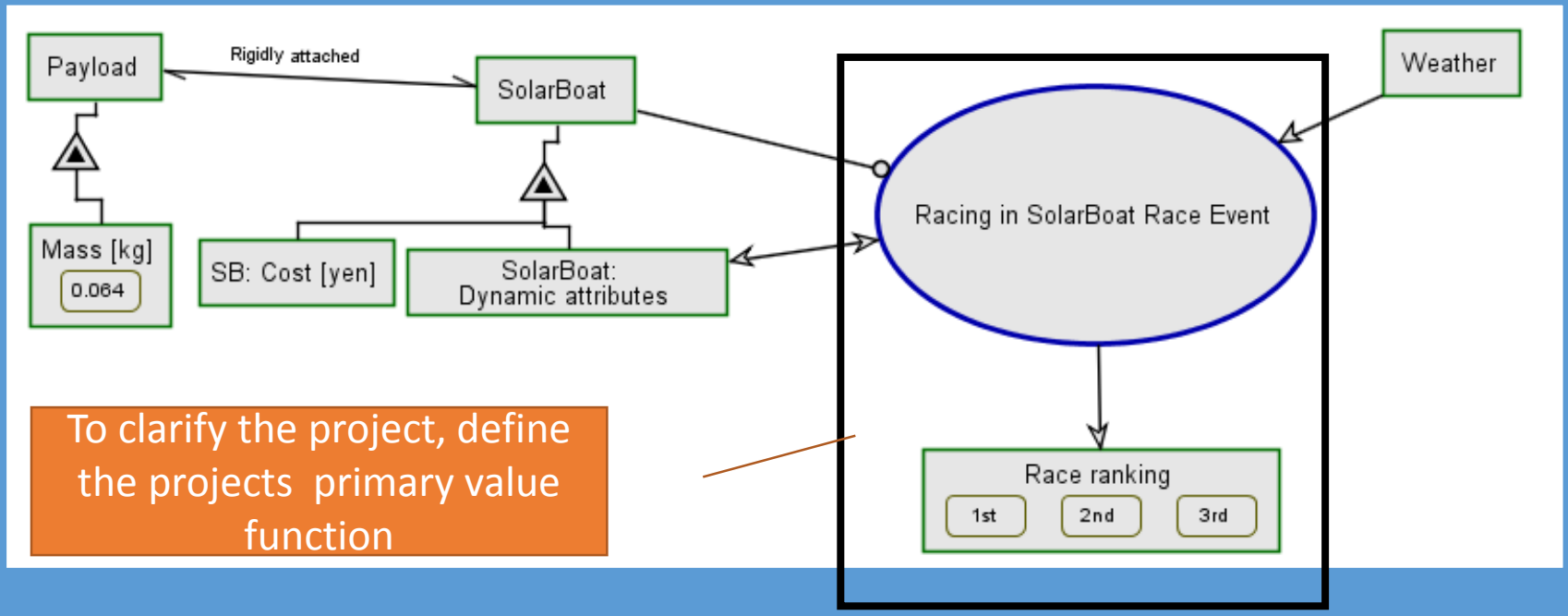
Proposed tools and methodologies for Knowledge Management and System-Level Design



Proposed tools and methodologies for Knowledge Management and System-Level Design

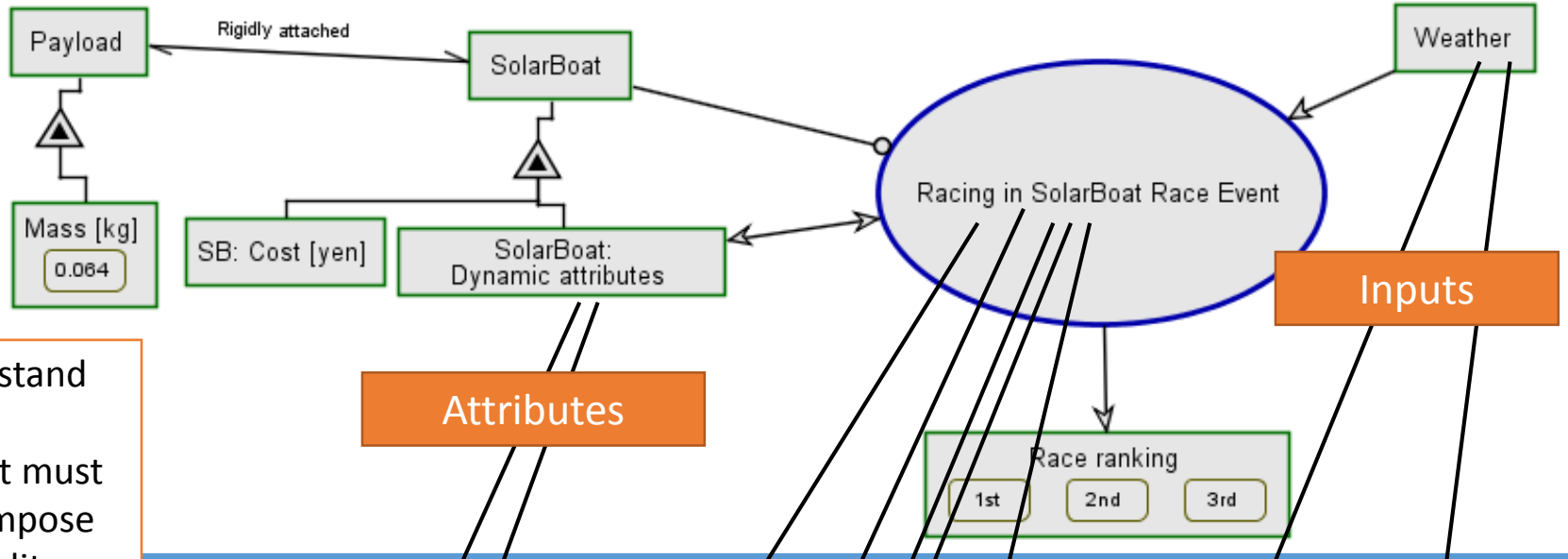


LS1:
Clarify



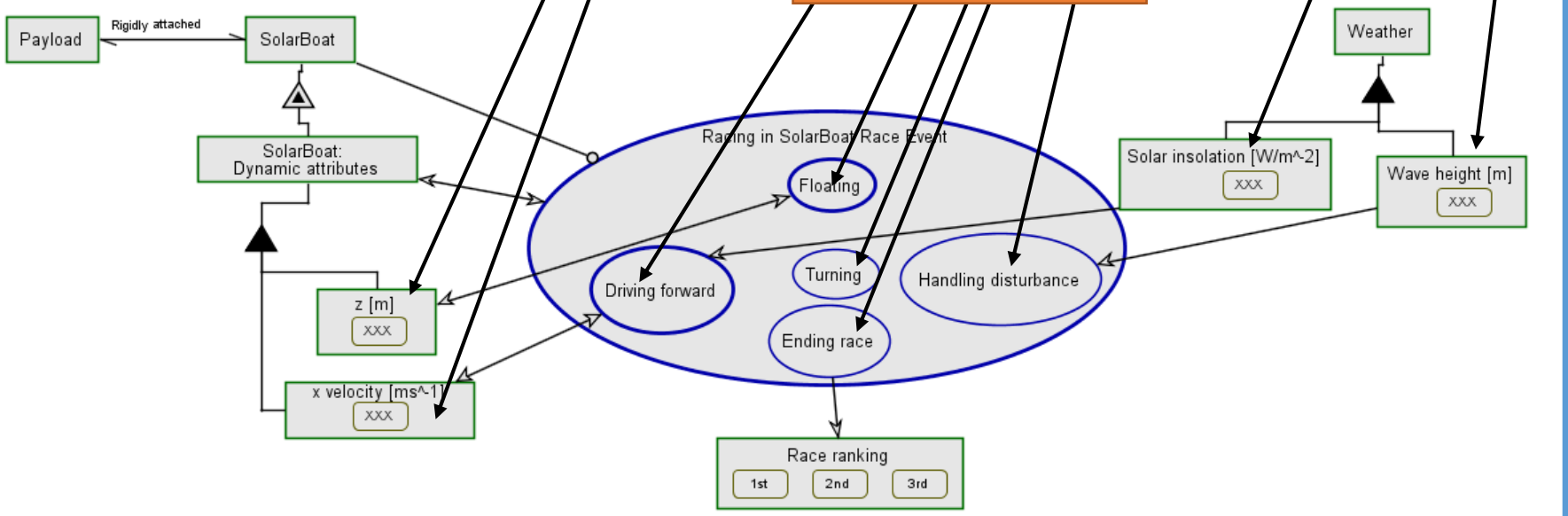
To clarify the project, define the projects primary value function

LS1: Clarify

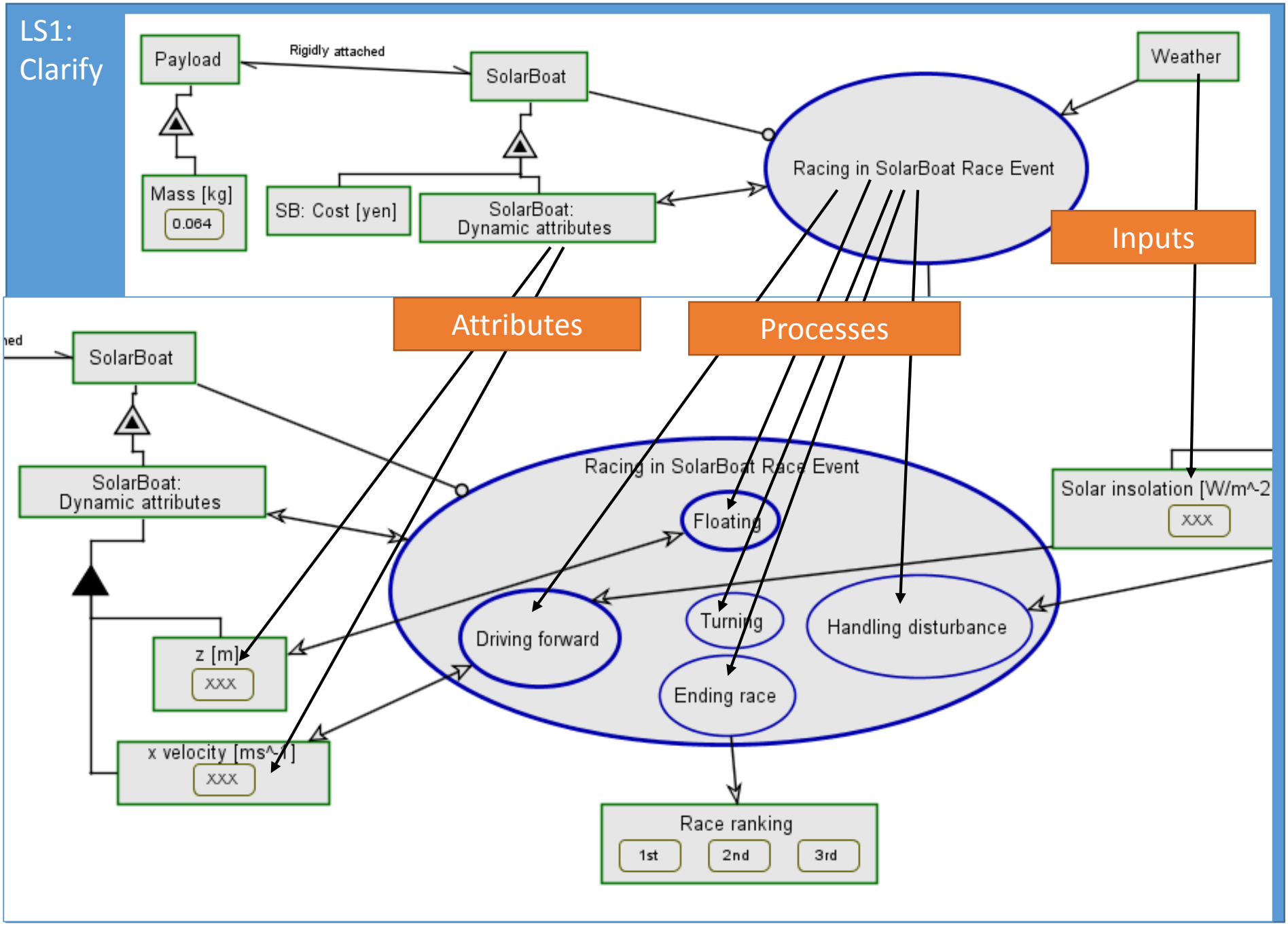


To understand what a SolarBoat must do decompose functionality

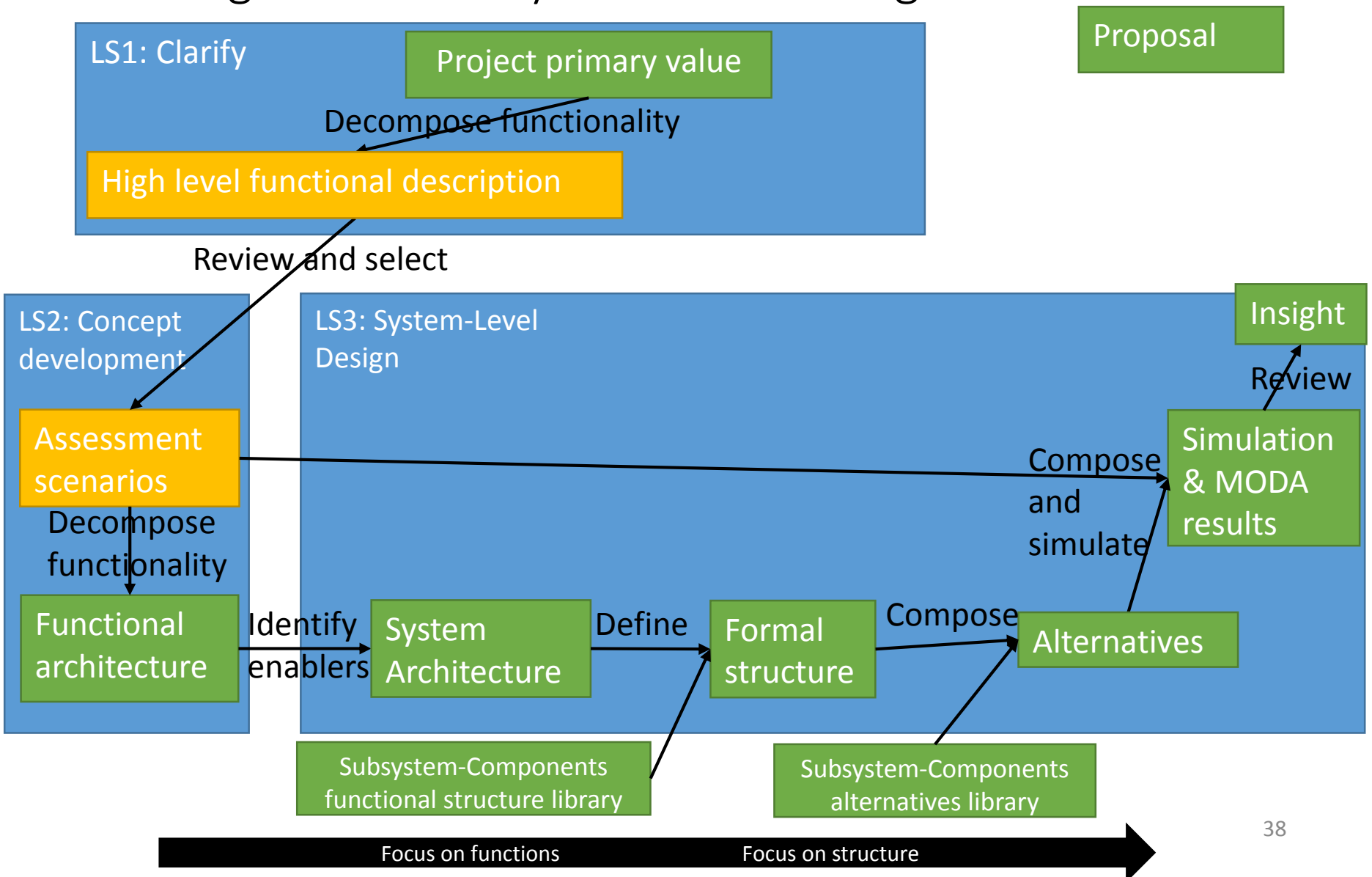
Processes



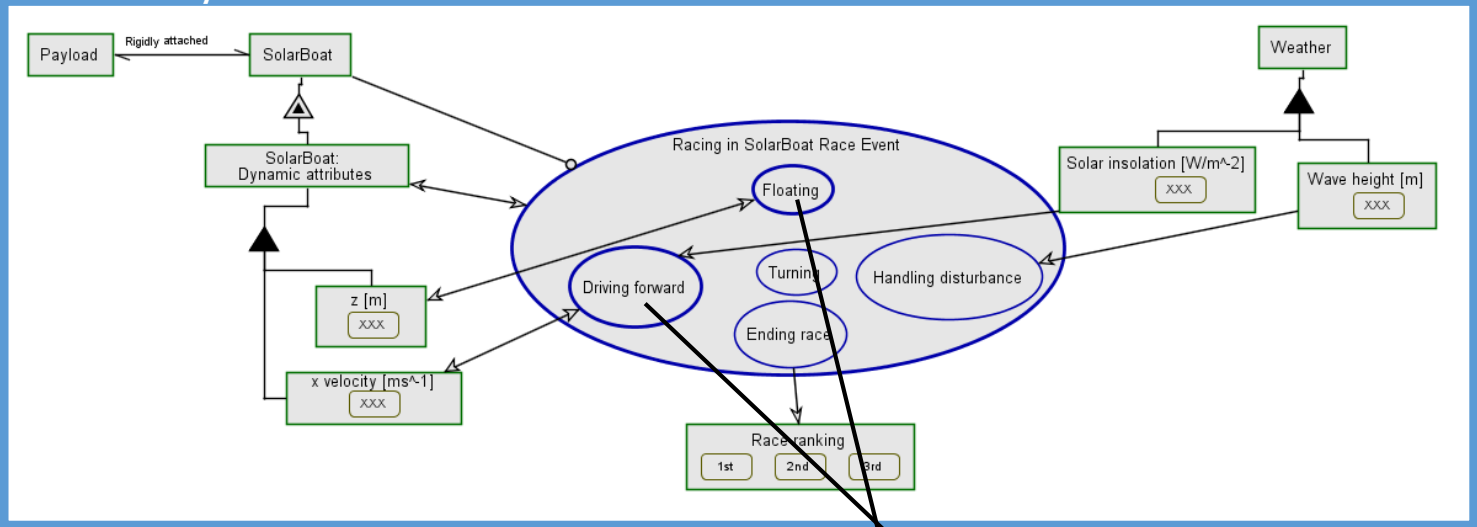
LS1: Clarify


Payload
Rigidly attached
SolarBoat
Weather
Mass [kg]
0.064
SB: Cost [yen]
SolarBoat:
Dynamic attributes
Racing in SolarBoat Race Event
Inputs
Attributes
Processes
SolarBoat
SolarBoat:
Dynamic attributes
Racing in SolarBoat Race Event
Floating
Driving forward
Turning
Handling disturbance
Ending race
Solar insolation [W/m^-2
XXX
z [m]
XXX
x velocity [ms^-1]
XXX
Race ranking
1st
2nd
3rd

Proposed tools and methodologies for Knowledge Management and System-Level Design



LS1: Clarify

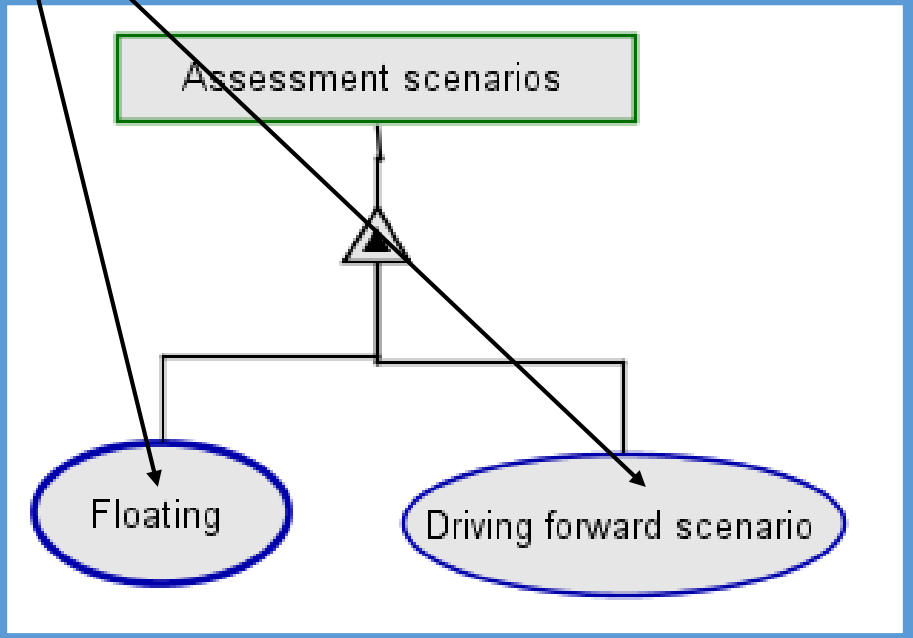


Review and select

LS2: Concept development

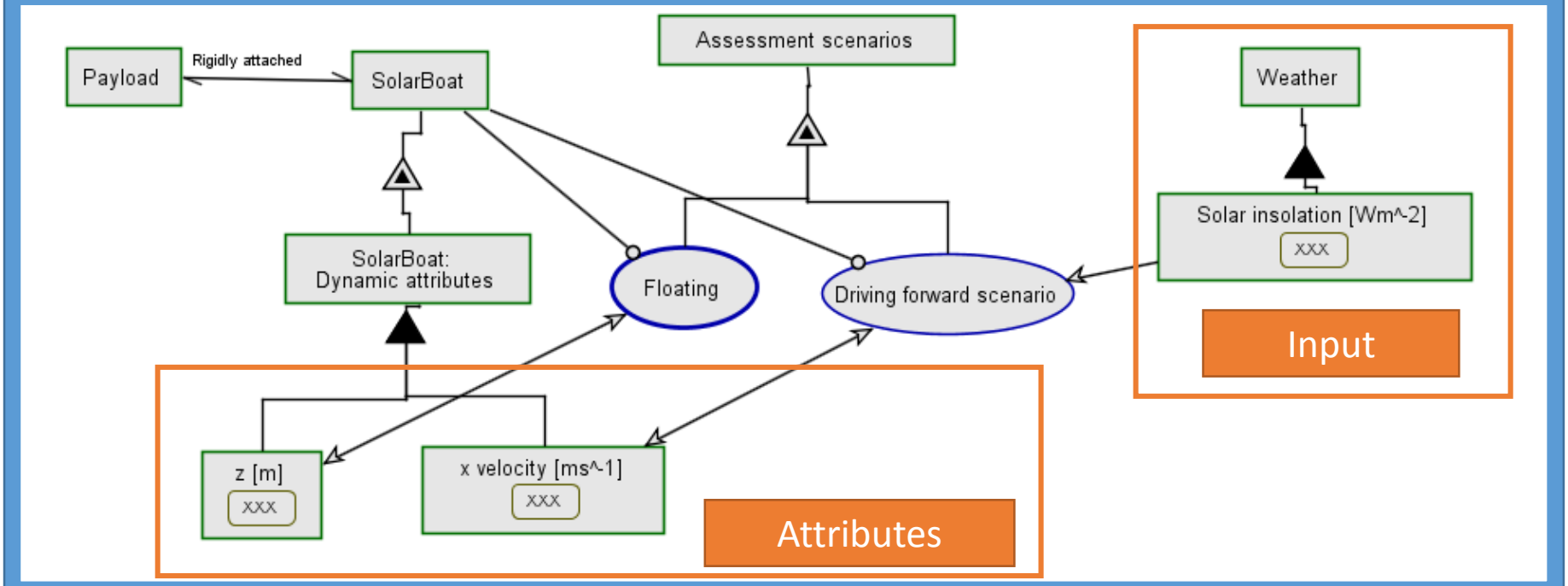
To develop different assessment criteria select some (or all) of the process which the SolarBoat performs.

Basing decomposition on these processes it assumes this is the only functionality of the SolarBoat. i.e. A very simple design.

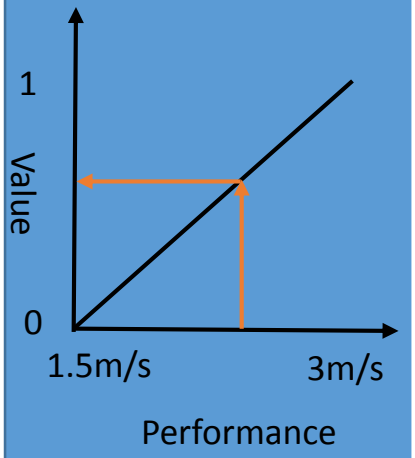


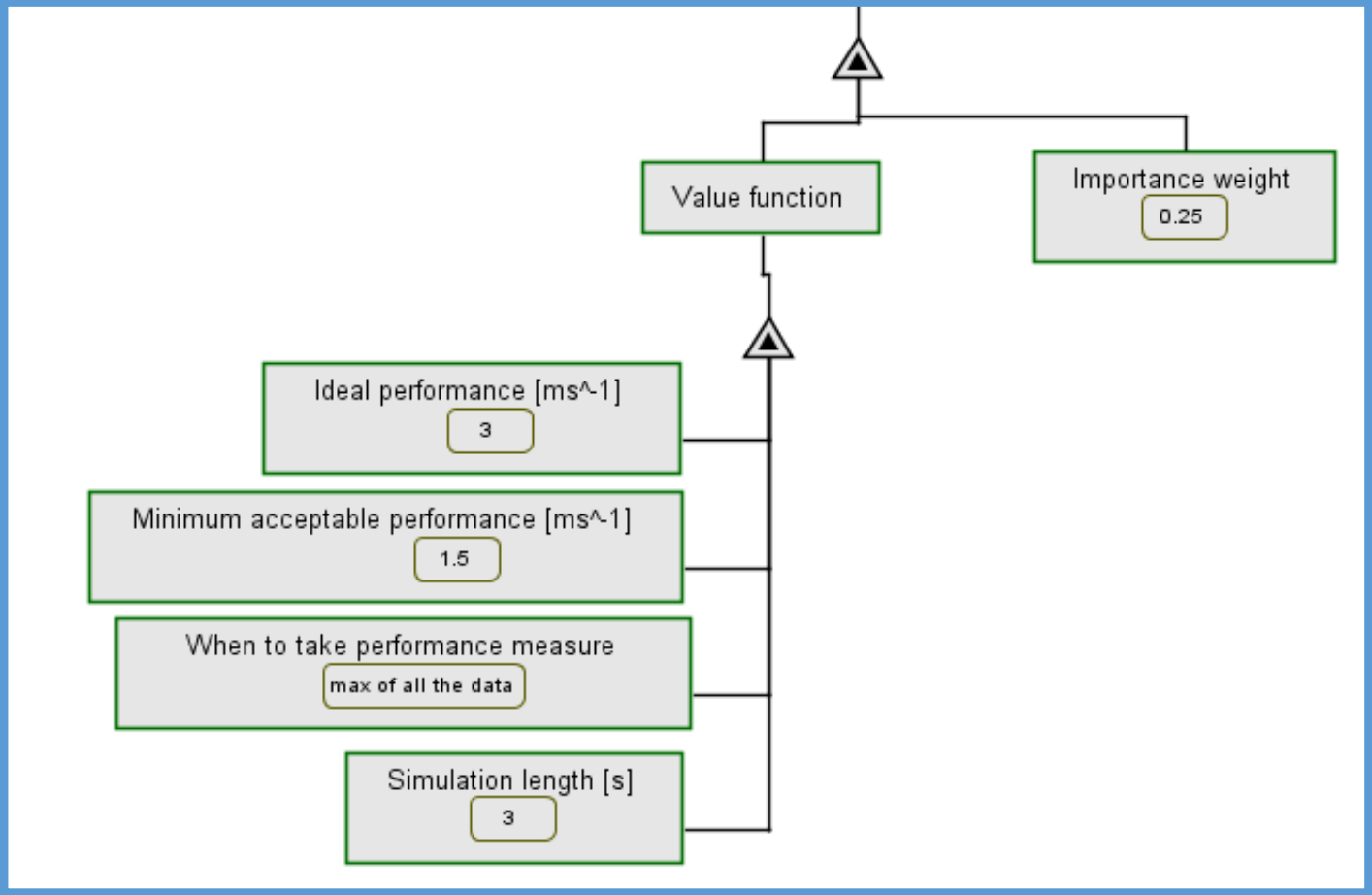
LS2: Concept development

To develop a framework to assess alternative designs.
Expand to display inputs, enablers and attributers of interest.

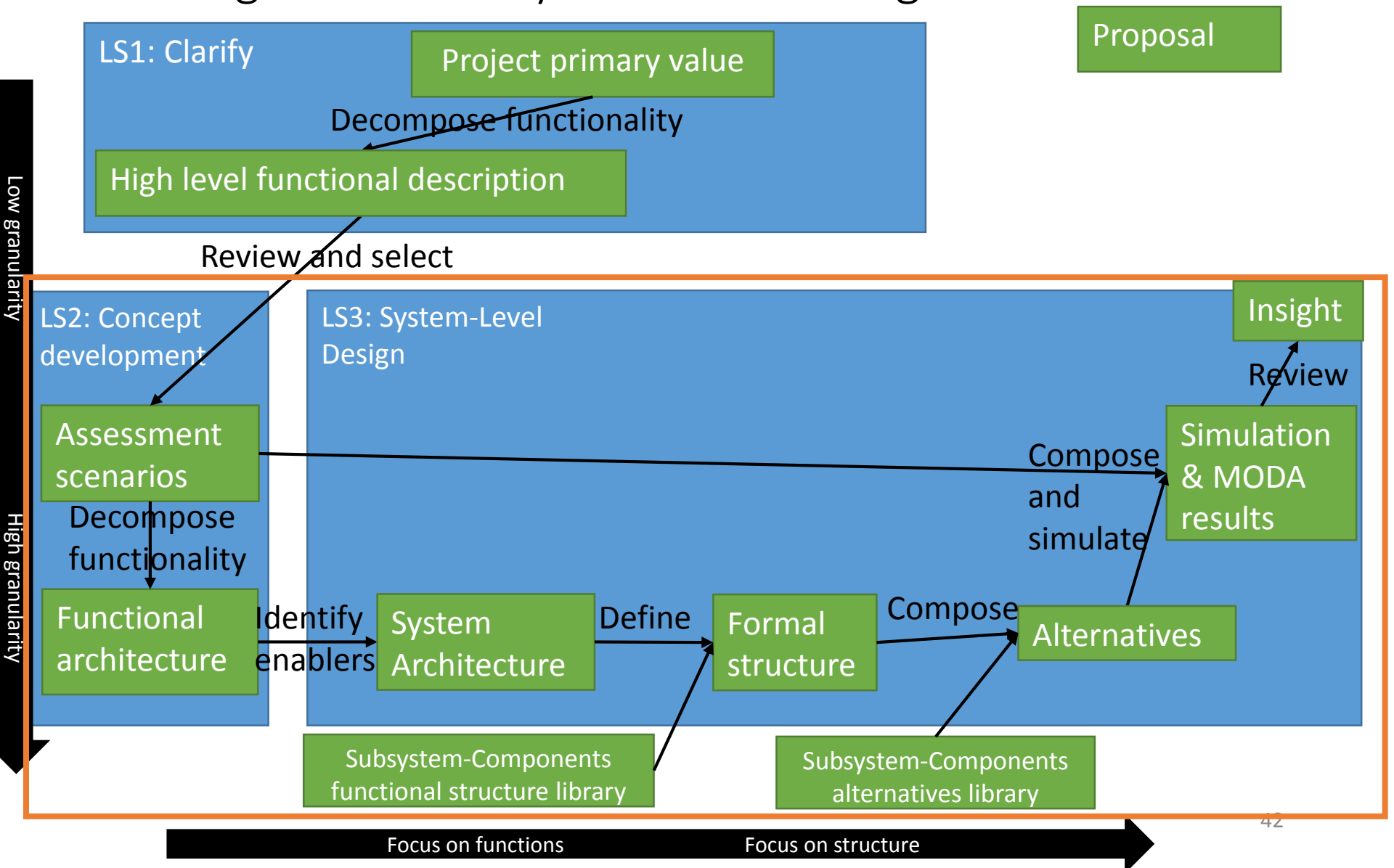


To later combine the results of multiple assessment scenarios by way of **Multi Objective Decision Analysis (MODA)**:
Define ideal performance, minimum acceptable performance and importance weight for the scenario.
Enabling fast comparison of alternatives.





Proposed tools and methodologies for Knowledge Management and System-Level Design



Focus on functions → Focus on structure

	LS2: Concept development	LS3: System-Level Design				
Model type:	**Functional architecture:**	**System architecture:**	**Formal structure (decomposition of structure):**	**Alternative:**	**Simulation results:**	**MODA result:**
Language:	OPM			Modelica		
Level 1 Assessment scenario						
Level 2 System of interest (e.g. Solar-Boat)						
Level 3 Subsystem (e.g. Electrical to thrust)						
Level 4 Subsystem-Components (e.g. Motor)						

Low detail → High detail

Defining hierarchy by OPM decomposition

Low granularity

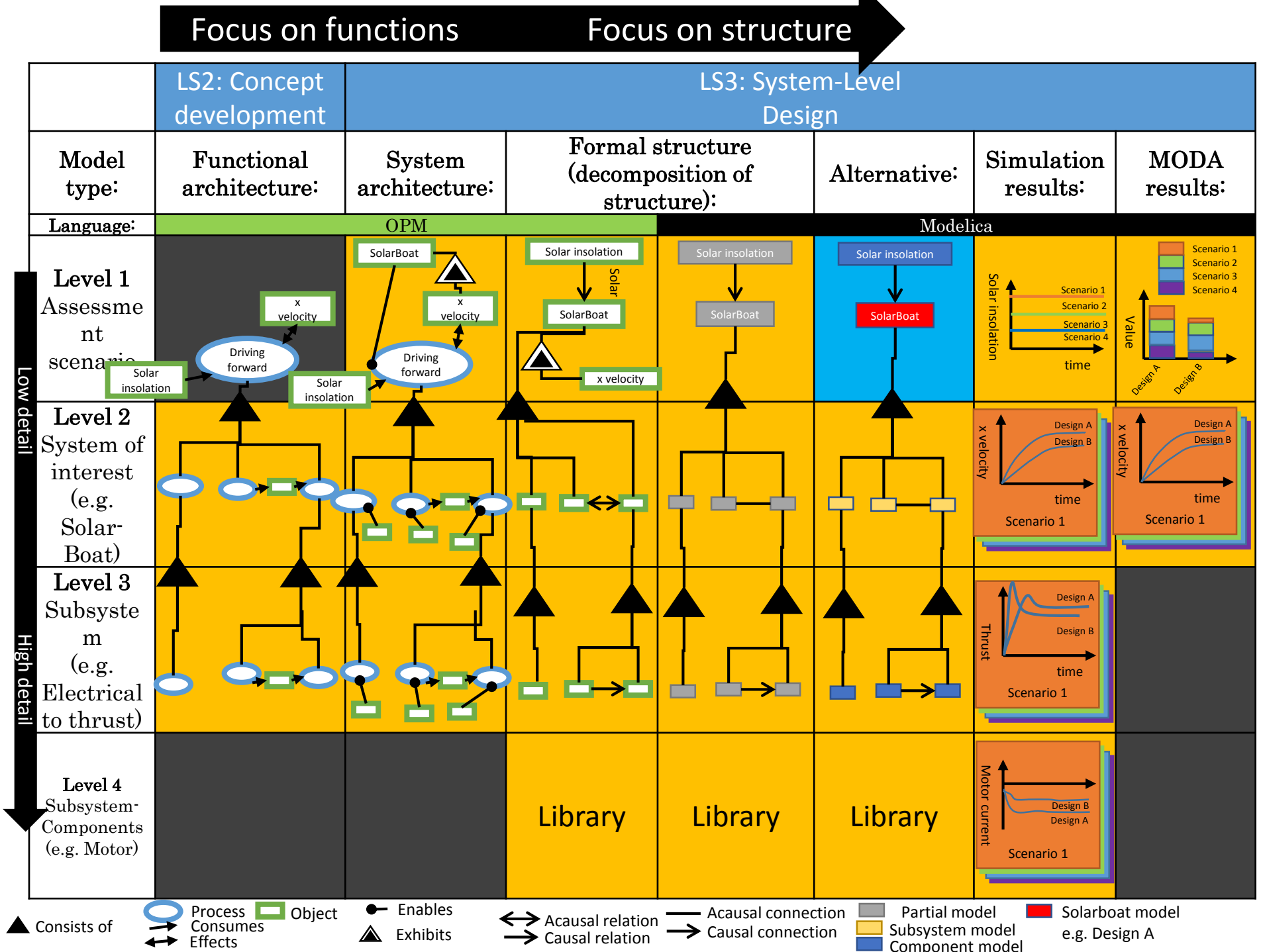
High granularity

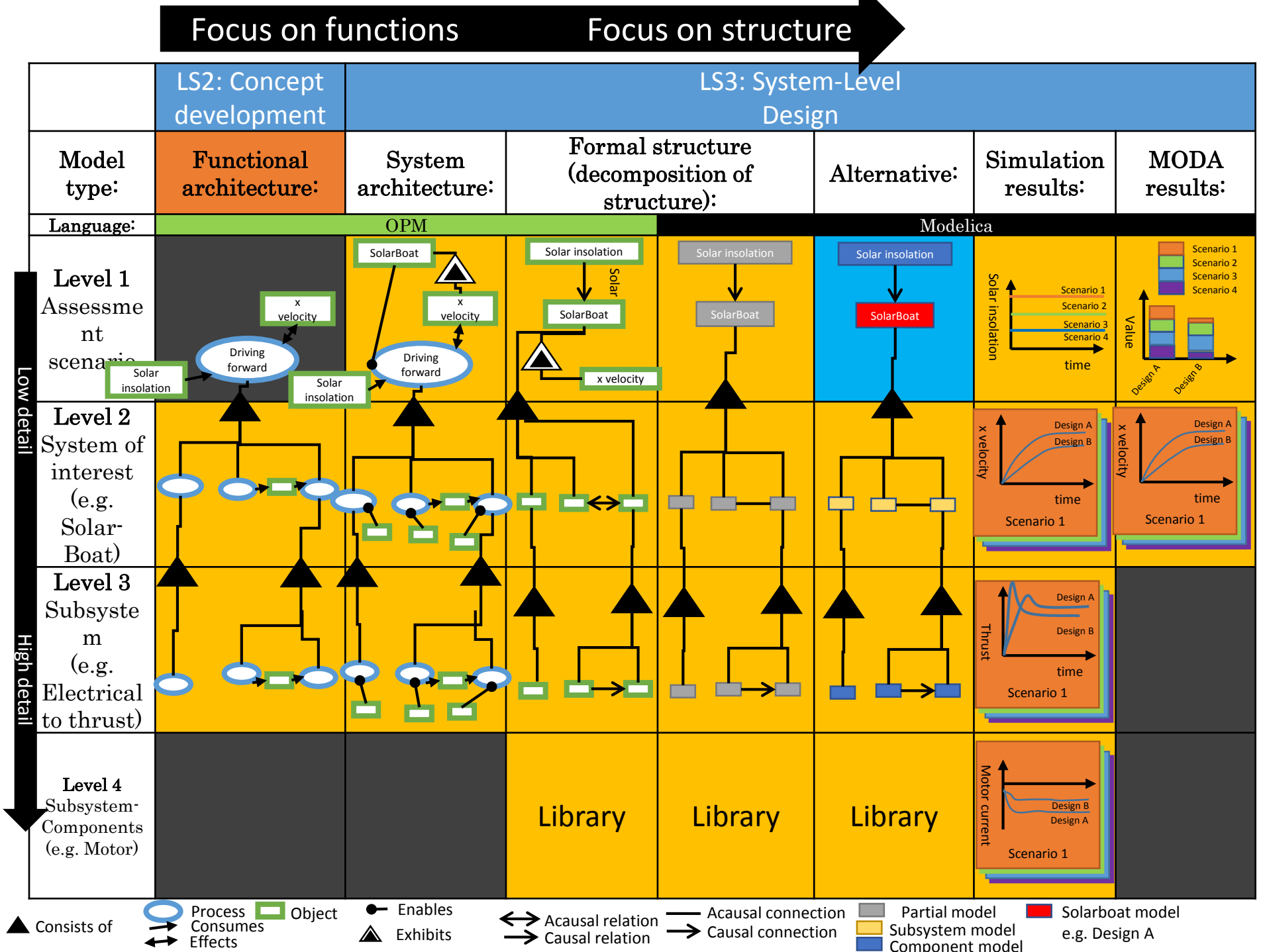
Level	Name	Example processes	Example object
Level 0	OPM “System Diagram” (SD)	Racing in Solar-Boat race event	SolarBoat Race
Level 1	Assessment Scenario, High level functional	Driving in straight line, Floating	Assessment scenario result
Level 2	System of interest	Converting Solar to Electrical, Converting Electrical to Thrust	Solar-Boat
Level 3	Subsystems	Converting Electrical to Rotation, Converting Rotation to thrust	Electrical to Thrust subsystem
Level 4	Subsystem-Components		DC Motor

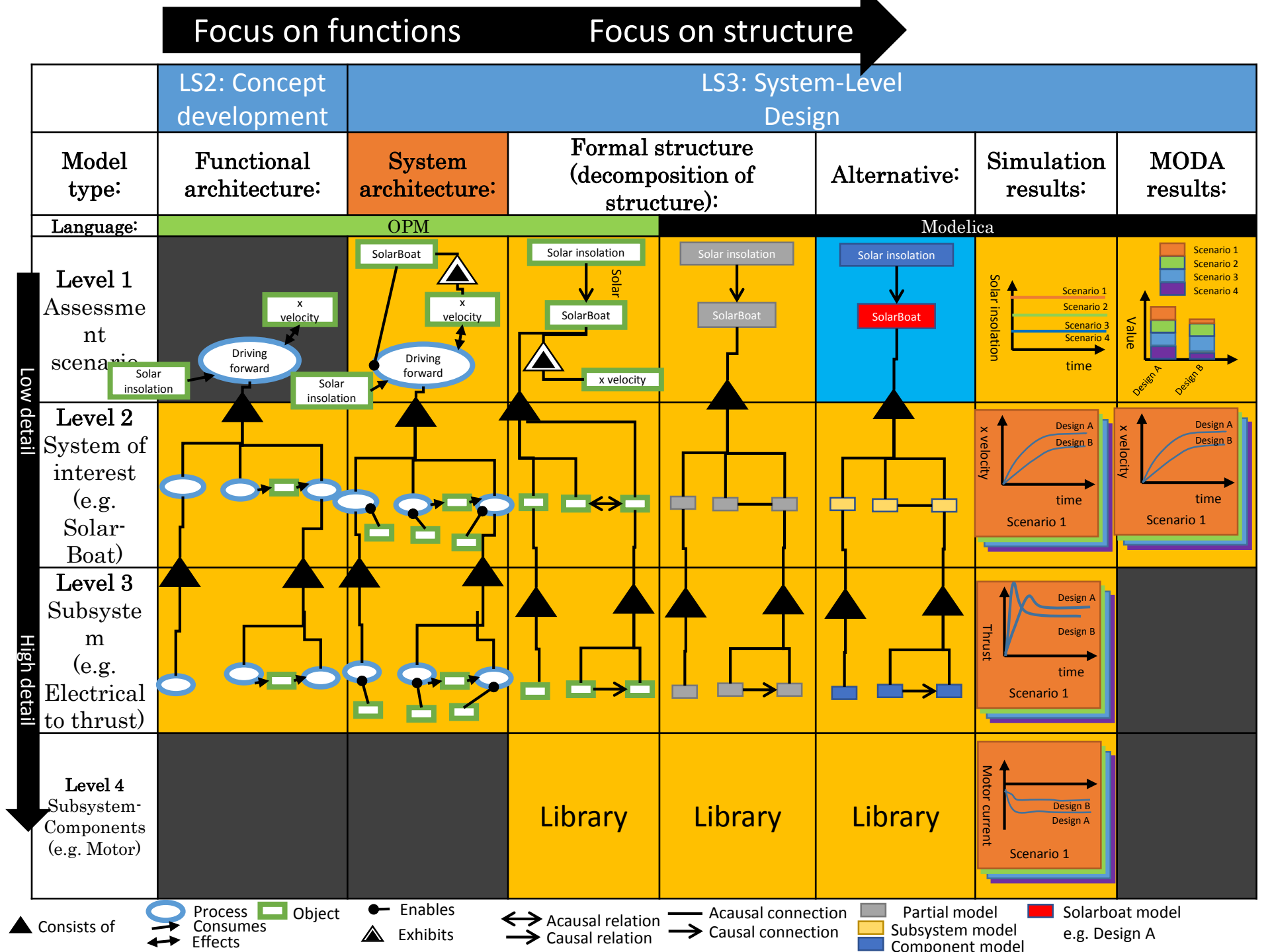
Focus on functions → Focus on structure

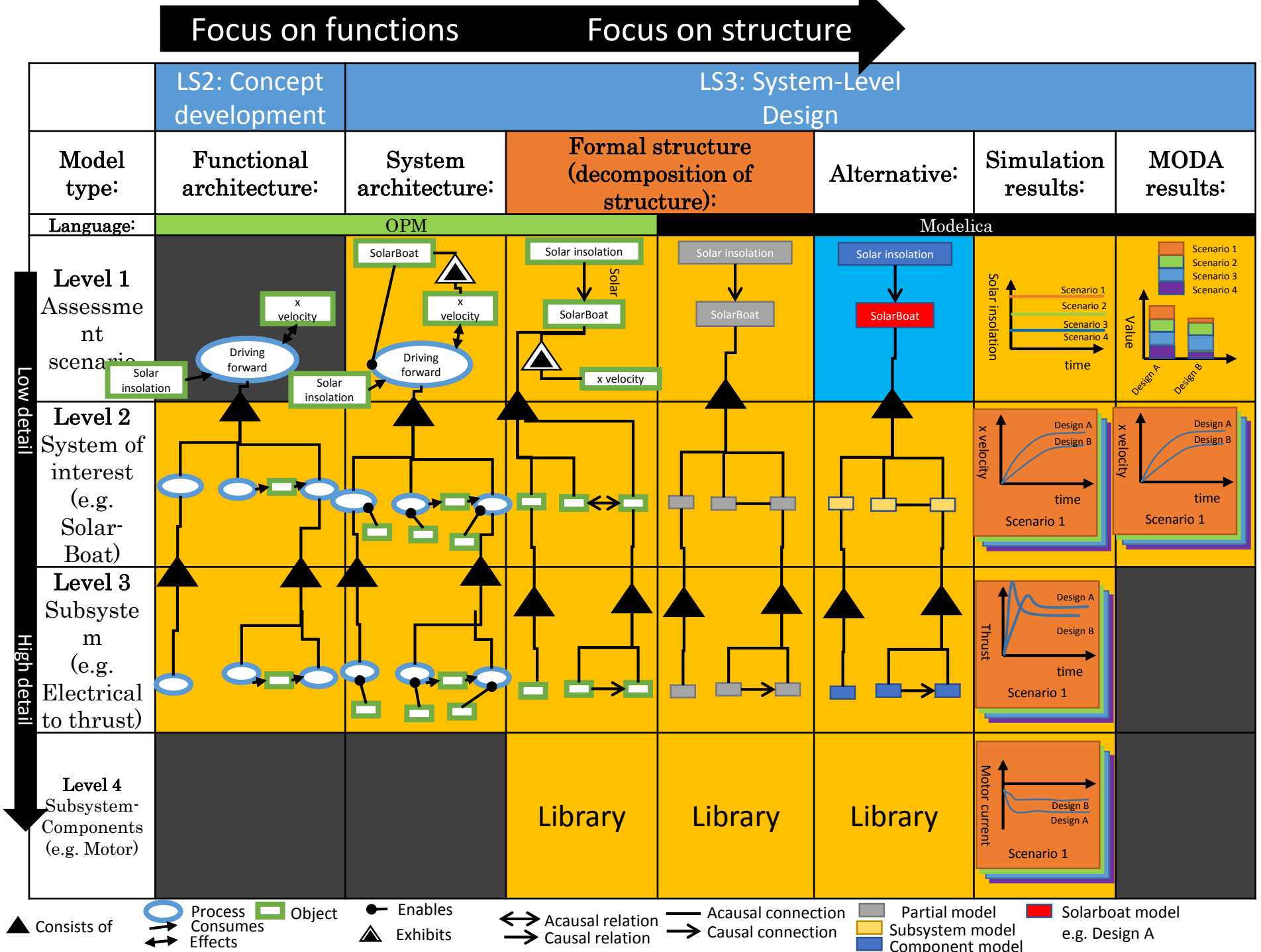
	LS2: Concept development	LS3: System-Level Design				
Model type:	**Functional architecture:**	**System architecture:**	**Formal structure (decomposition of structure):**	**Alternative:**	**Simulation results:**	**MODA result:**
Language:	OPM			Modelica		
Level 1 Assessment scenario						
Level 2 System of interest (e.g. Solar-Boat)						
Level 3 Subsystem (e.g. Electrical to thrust)						
Level 4 Subsystem-Components (e.g. Motor)						

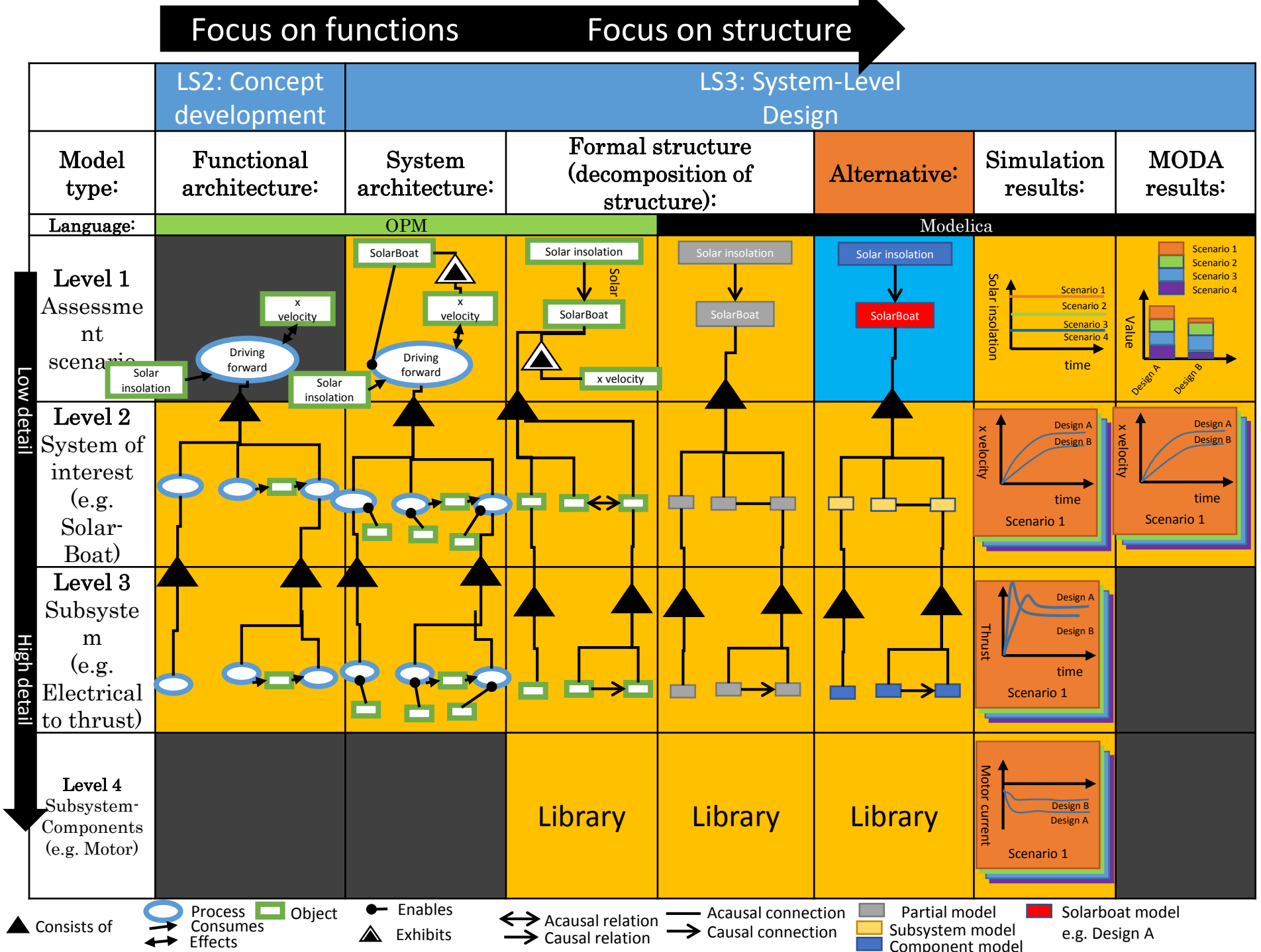
Low detail → High detail

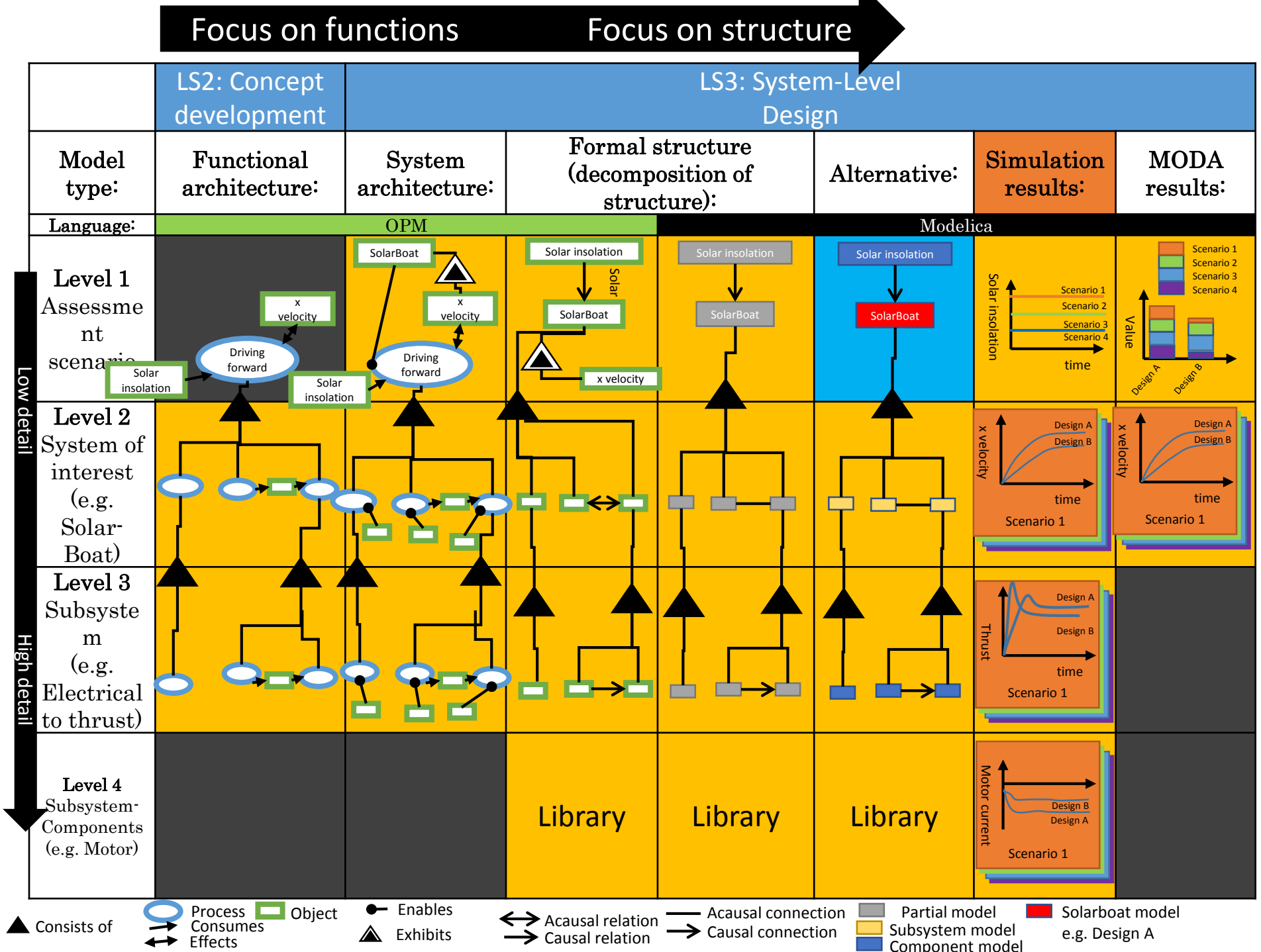

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Solar insolation
Driving forward
x velocity
SolarBoat
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A

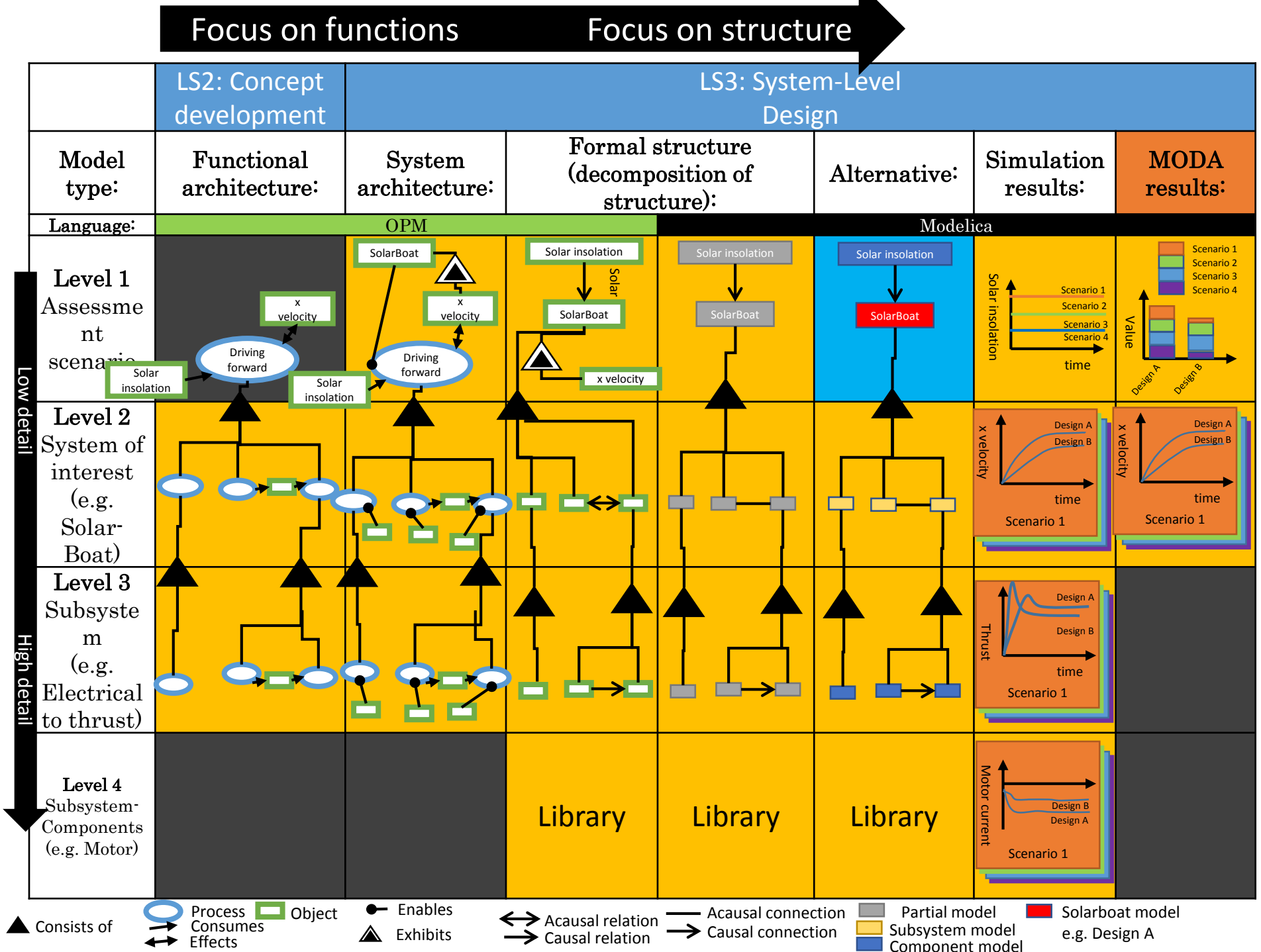

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
SolarBoat
x velocity
Driving forward
Solar insulation
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A


Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Solar insolation
Driving forward
x velocity
SolarBoat
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A


Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Low detail
High detail
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Solar insulation
Driving forward
x velocity
SolarBoat
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A

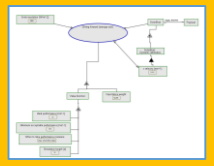
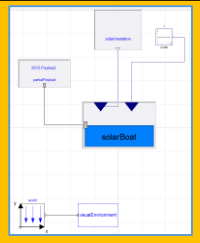
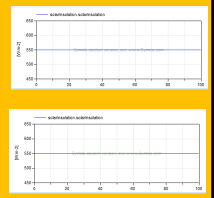
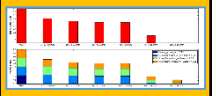
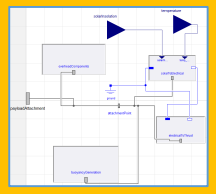
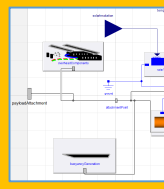
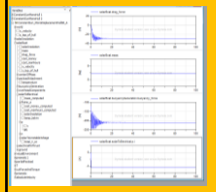
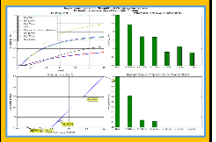
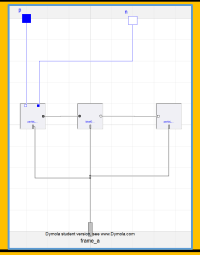
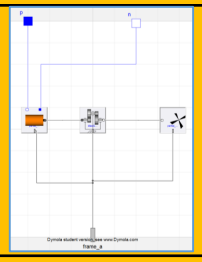
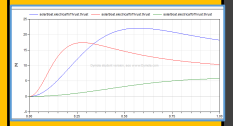
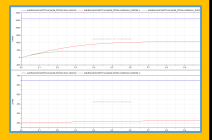

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Solar insolation
Driving forward
x velocity
SolarBoat
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A

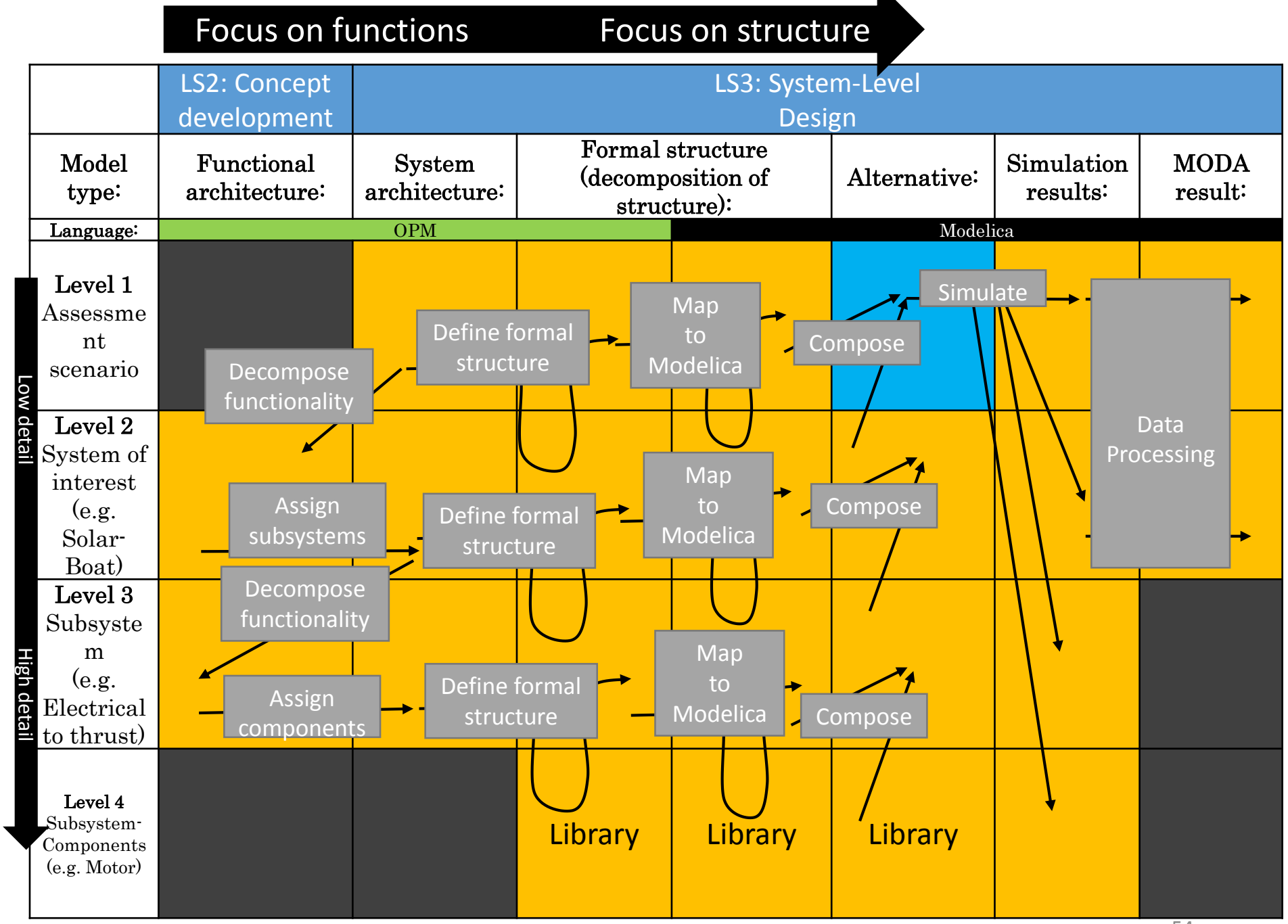

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Low detail
High detail
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Solar insulation
Driving forward
x velocity
SolarBoat
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A

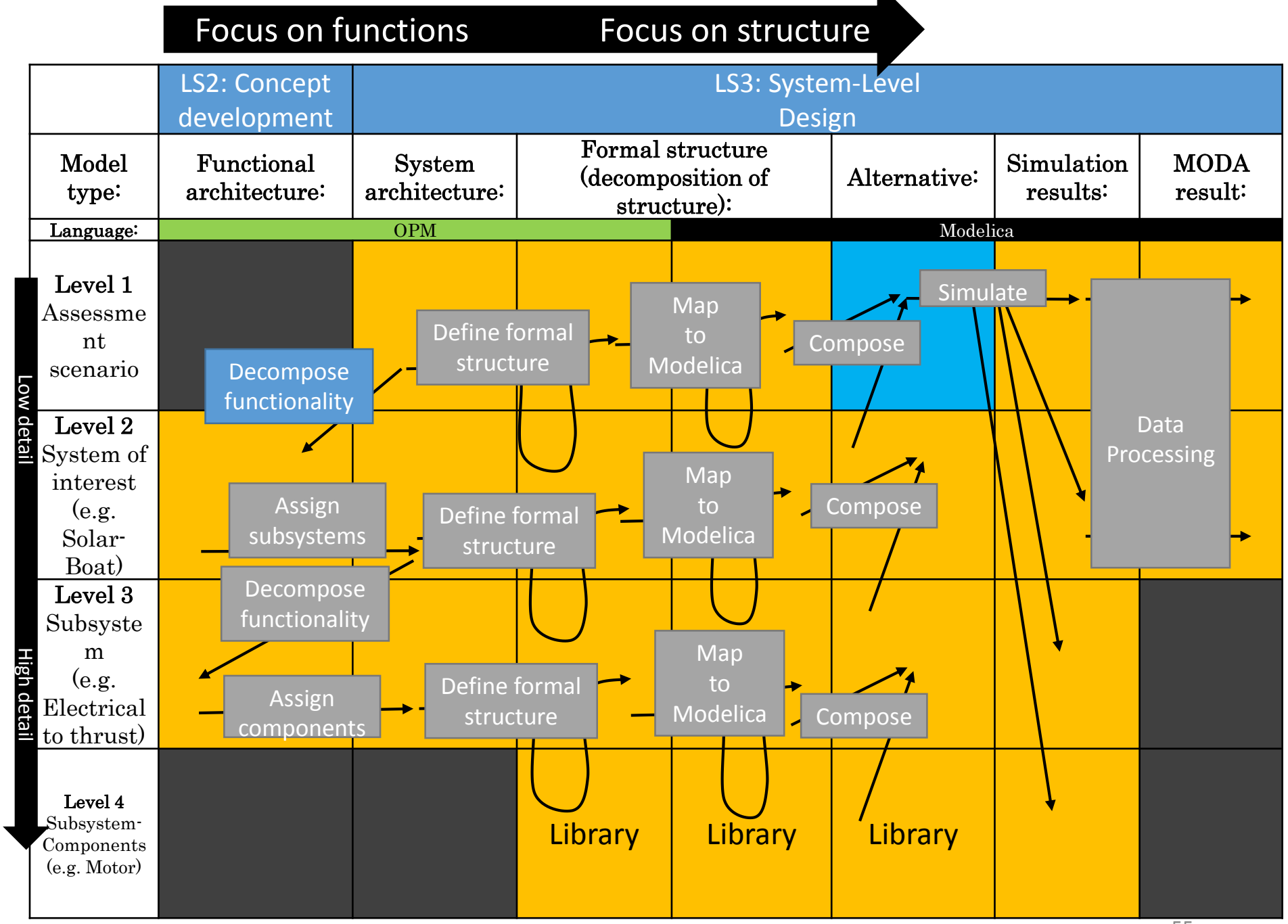

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
x velocity
Driving forward
Solar insulation
SolarBoat
Solar
Library
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
Thrust
Motor current
Consists of
Process
Object
Consumes
Effects
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A

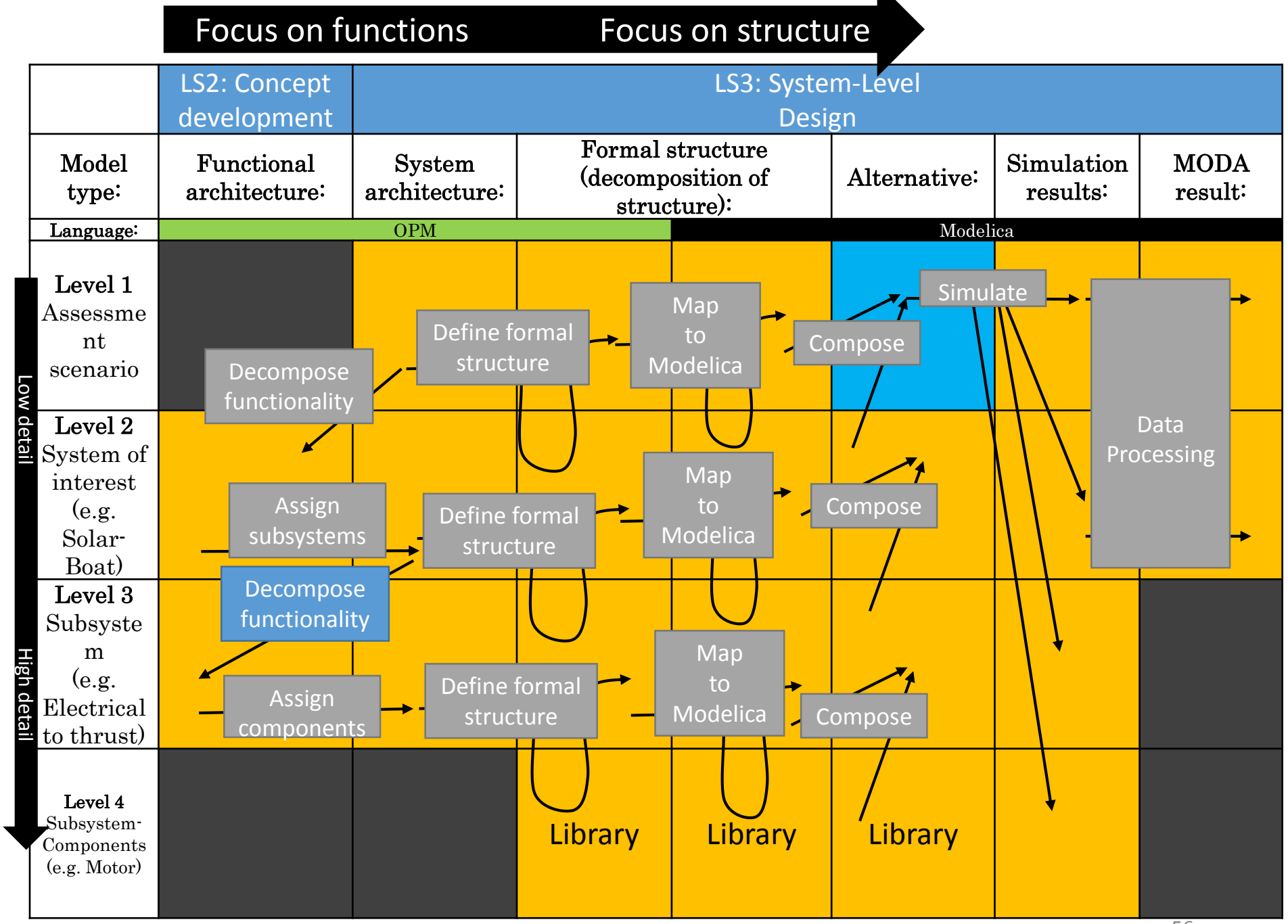
Focus on functions → Focus on structure

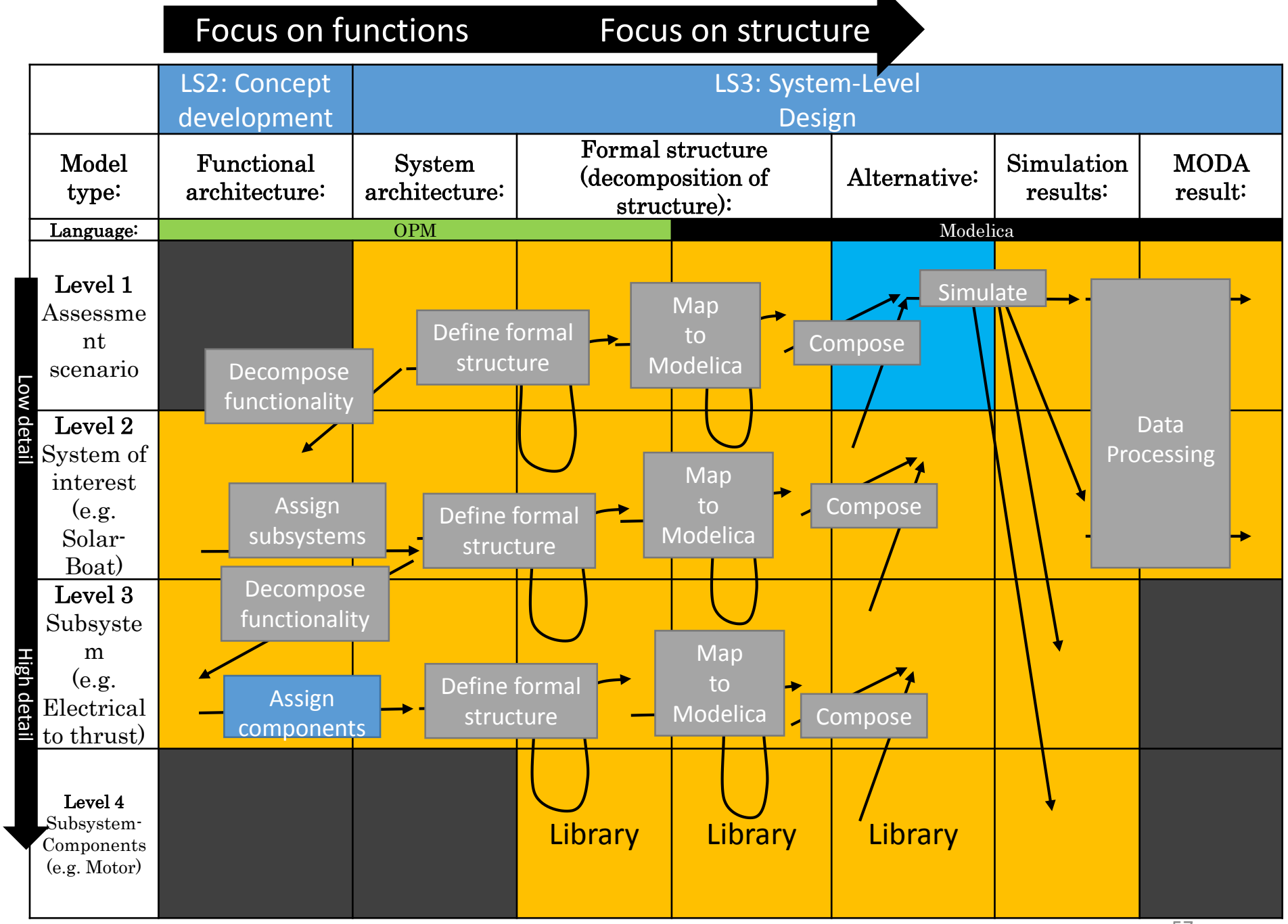
Low detail ↓ High detail

	LS2: Concept development	LS3: System-Level Design				
Model type:	**Functional architecture:**	**System architecture:**	**Formal structure (decomposition of structure):**	**Alternative:**	**Simulation results:**	**MODA result:**
Language:	OPM			Modelica		
Level 1 Assessment scenario						 
Level 2 System of interest (e.g. Solar-Boat)							
Level 3 Subsystem (e.g. Electrical to thrust)							
Level 4 Subsystem-Components (e.g. Motor)			Library	Library	Library		


Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library


Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

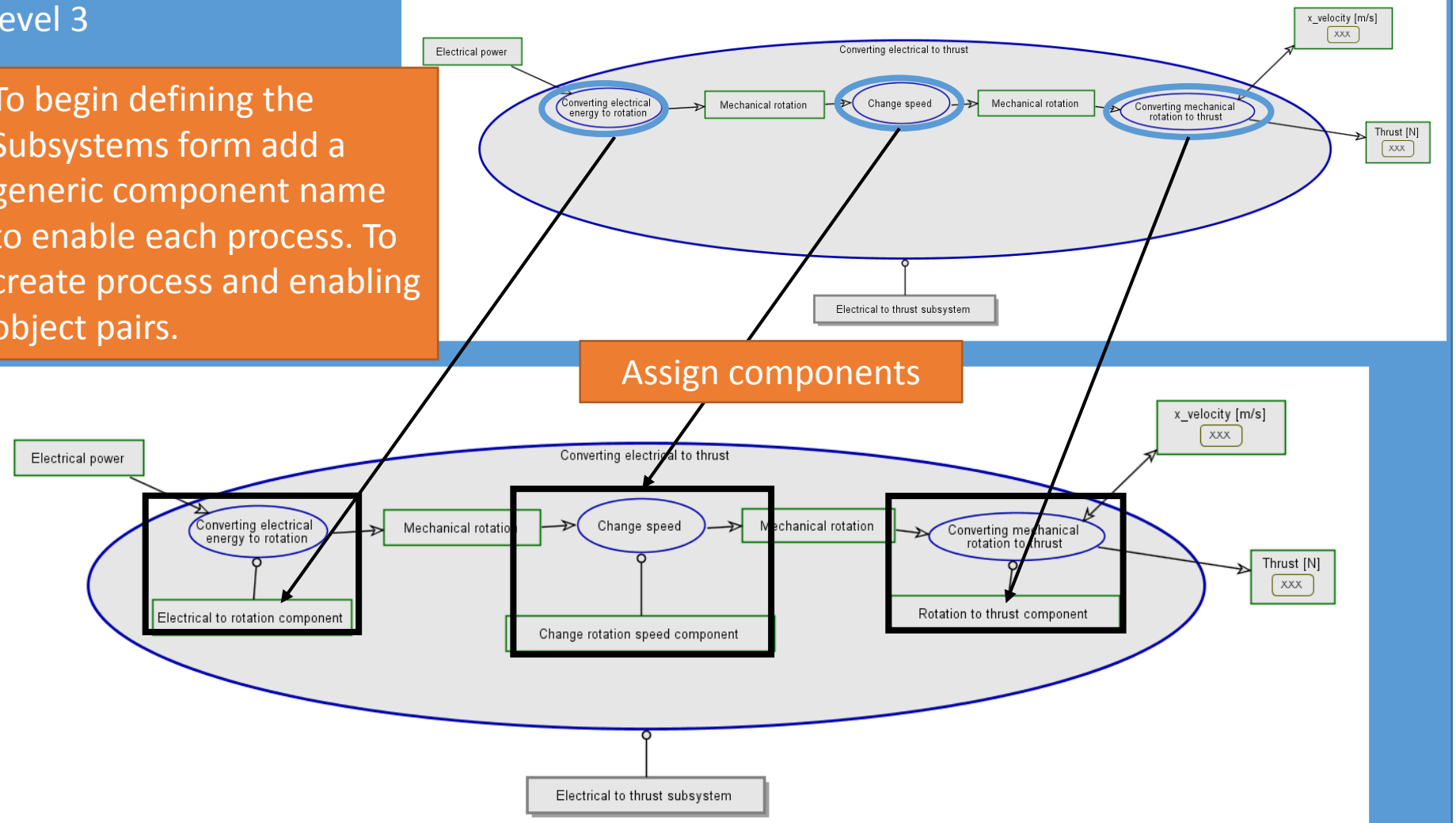

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

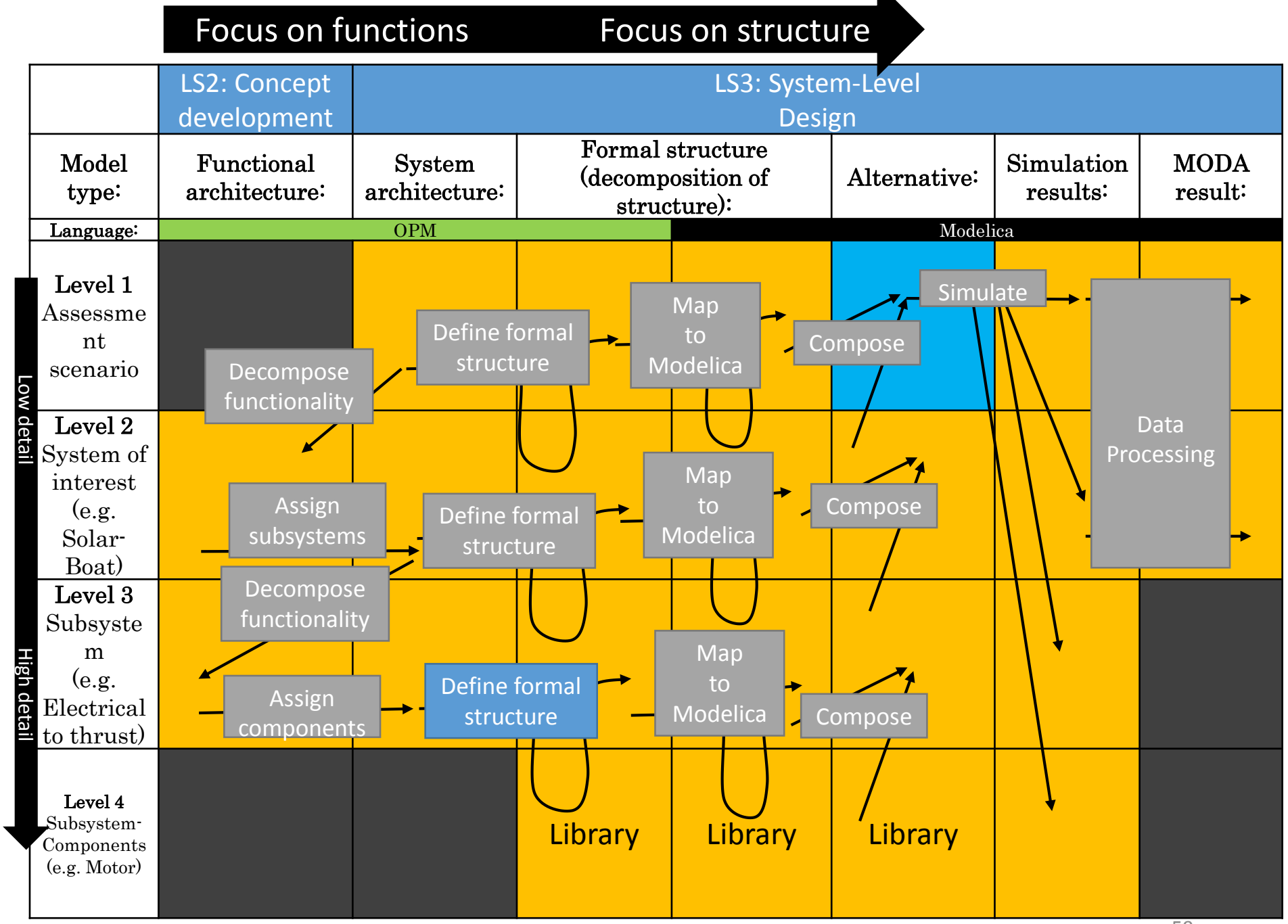

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Assign components

Level 3

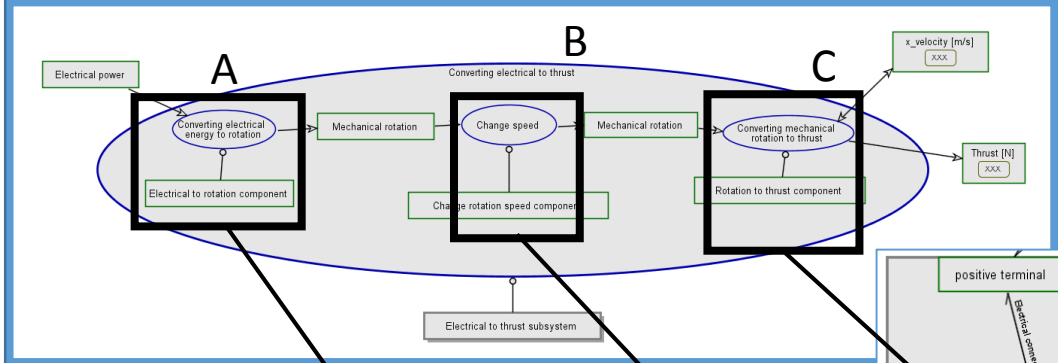
To begin defining the Subsystems form add a generic component name to enable each process. To create process and enabling object pairs.

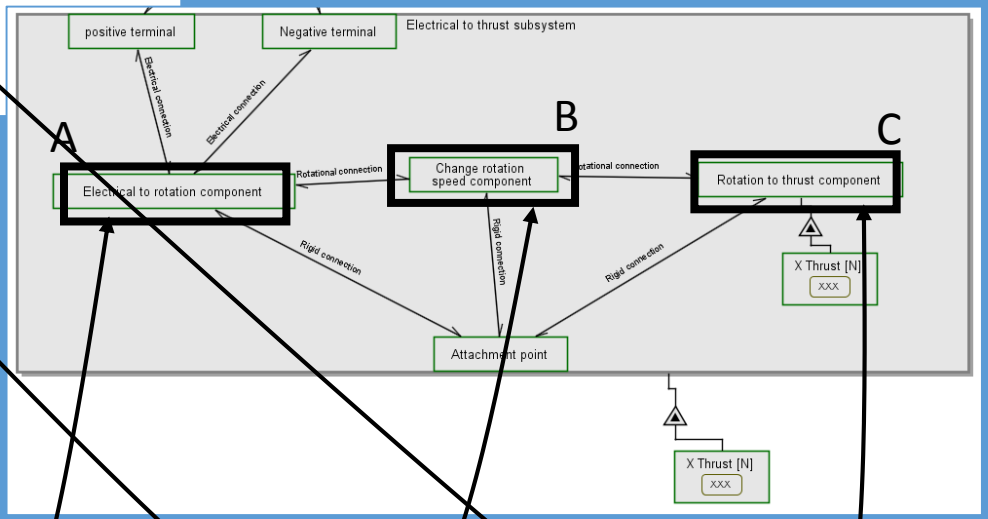



Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Define formal structure

Level 3

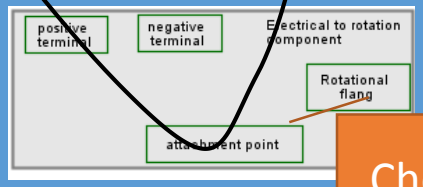


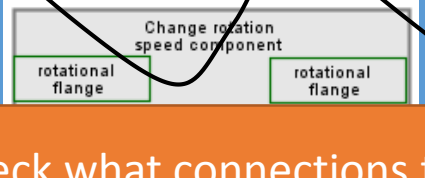


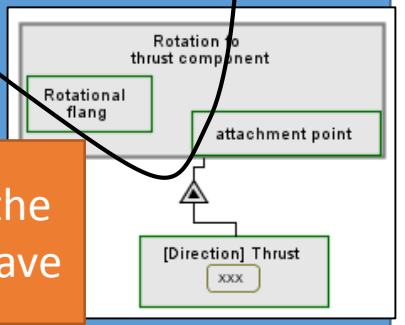
To define formal structure involves:

1) Build “common” formal structure and name it
2) Review library each System architecture object/process pair (A, B & C) to understand the interfaces
3) **Populate**

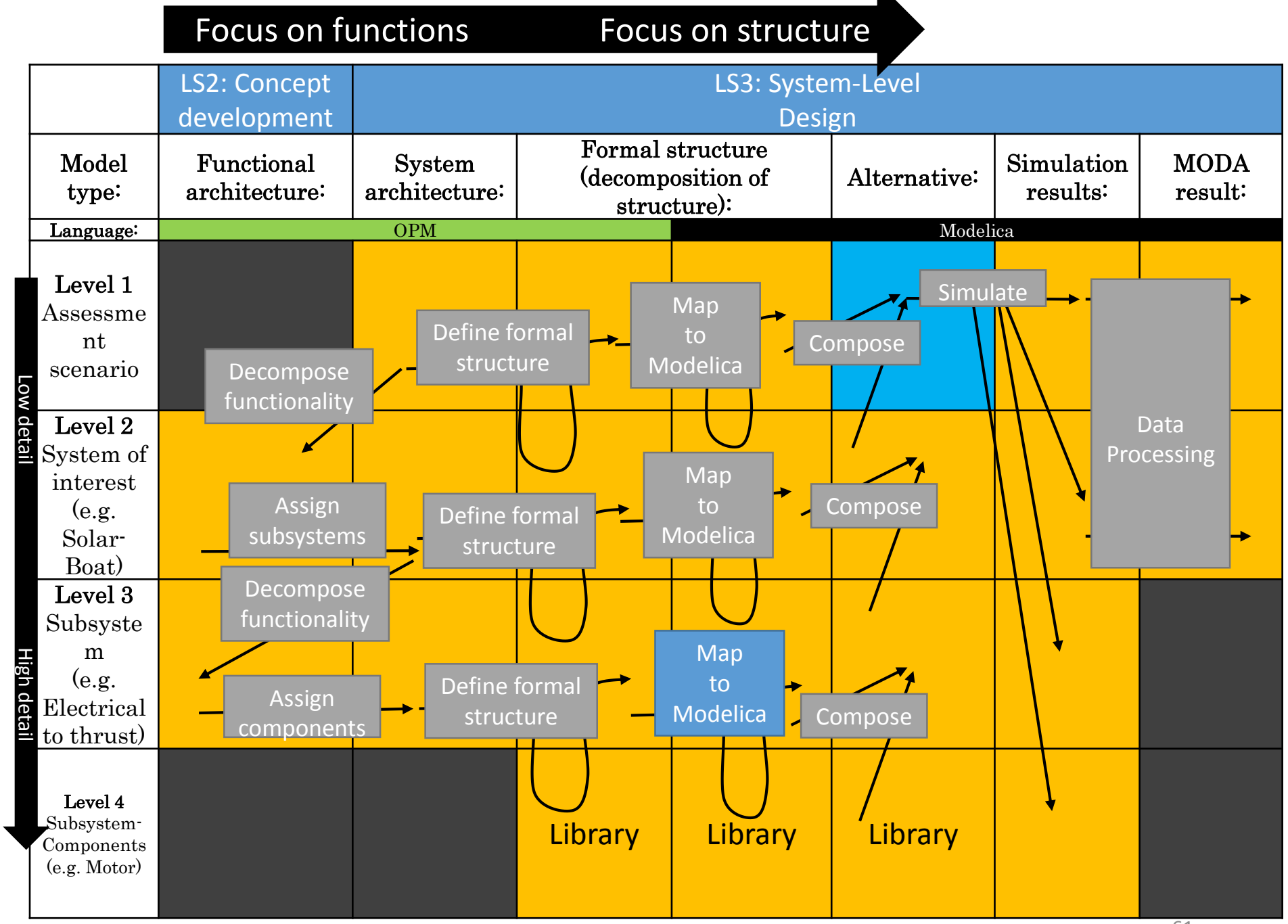
Level 4 (Subsystem-Components library)





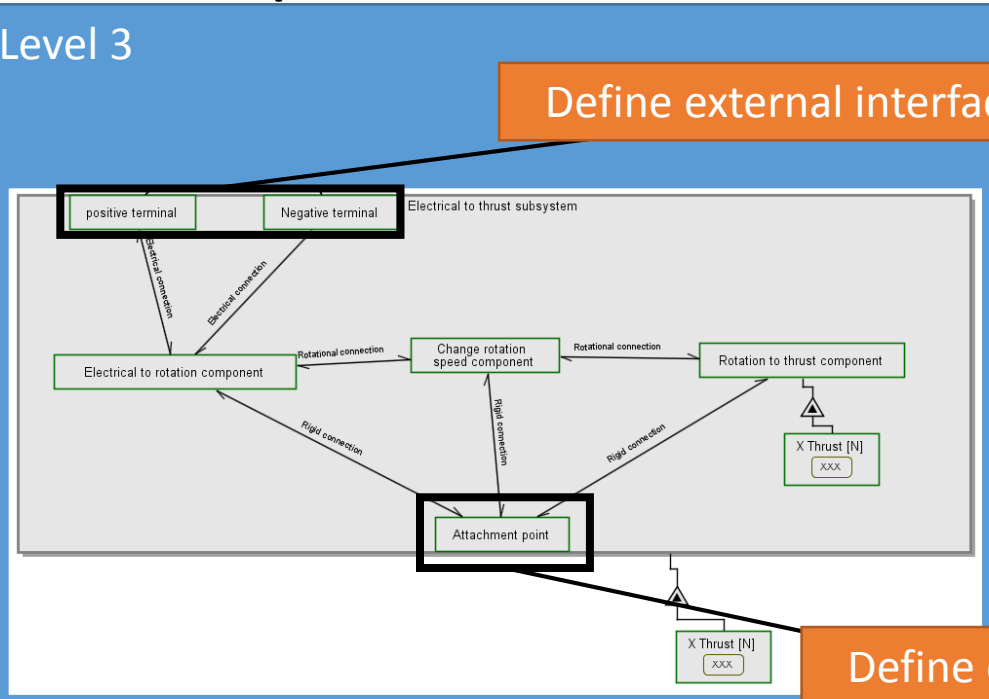


Check what connections the lower level components have

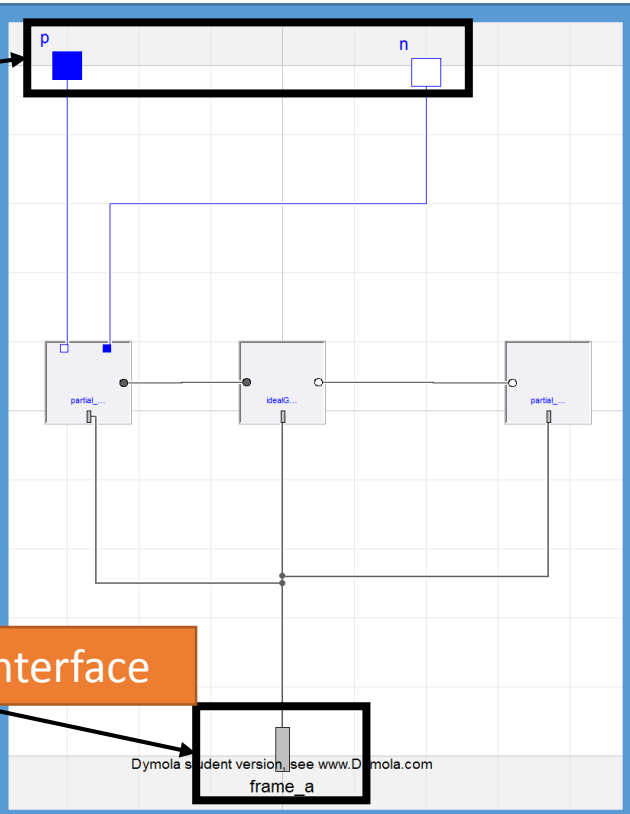

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Map to Modelica

Level 3



Define external interface

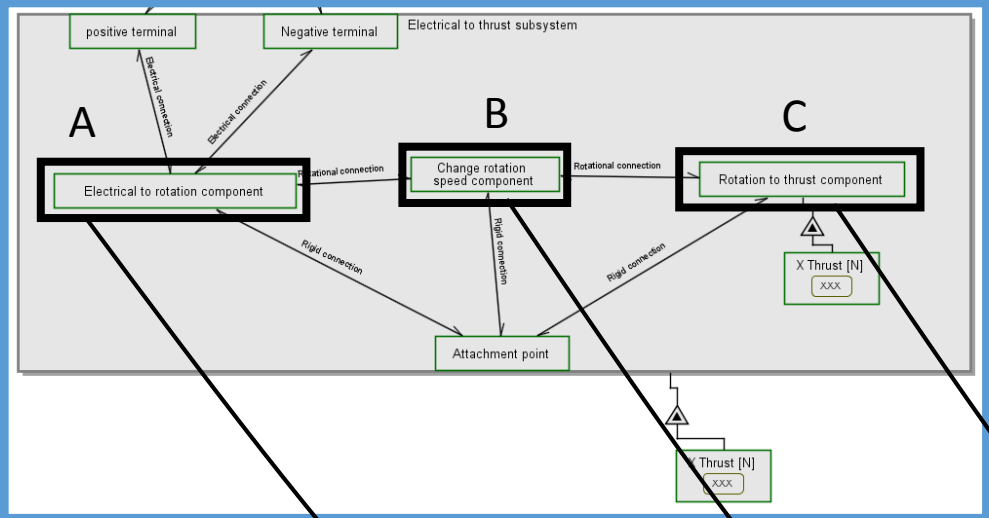


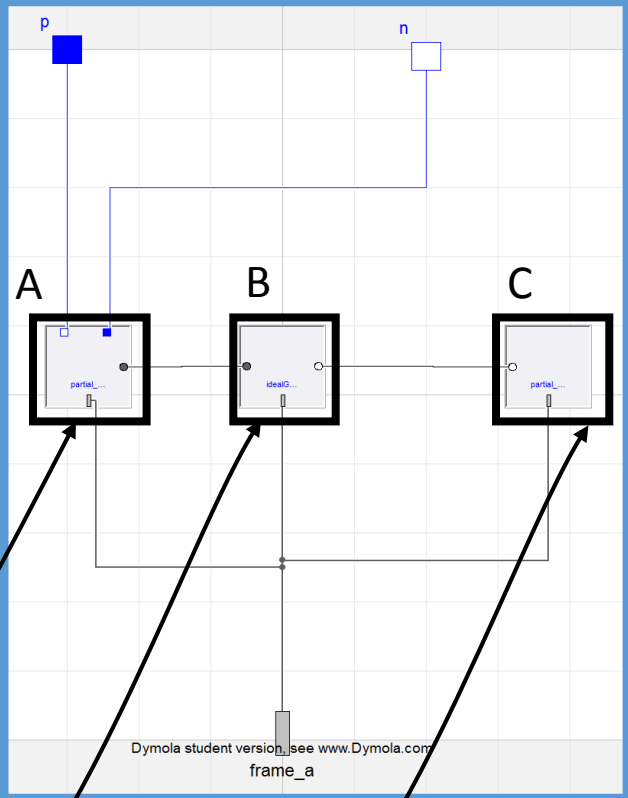
Define external interface

Level 4
(Subsystem-Components library)

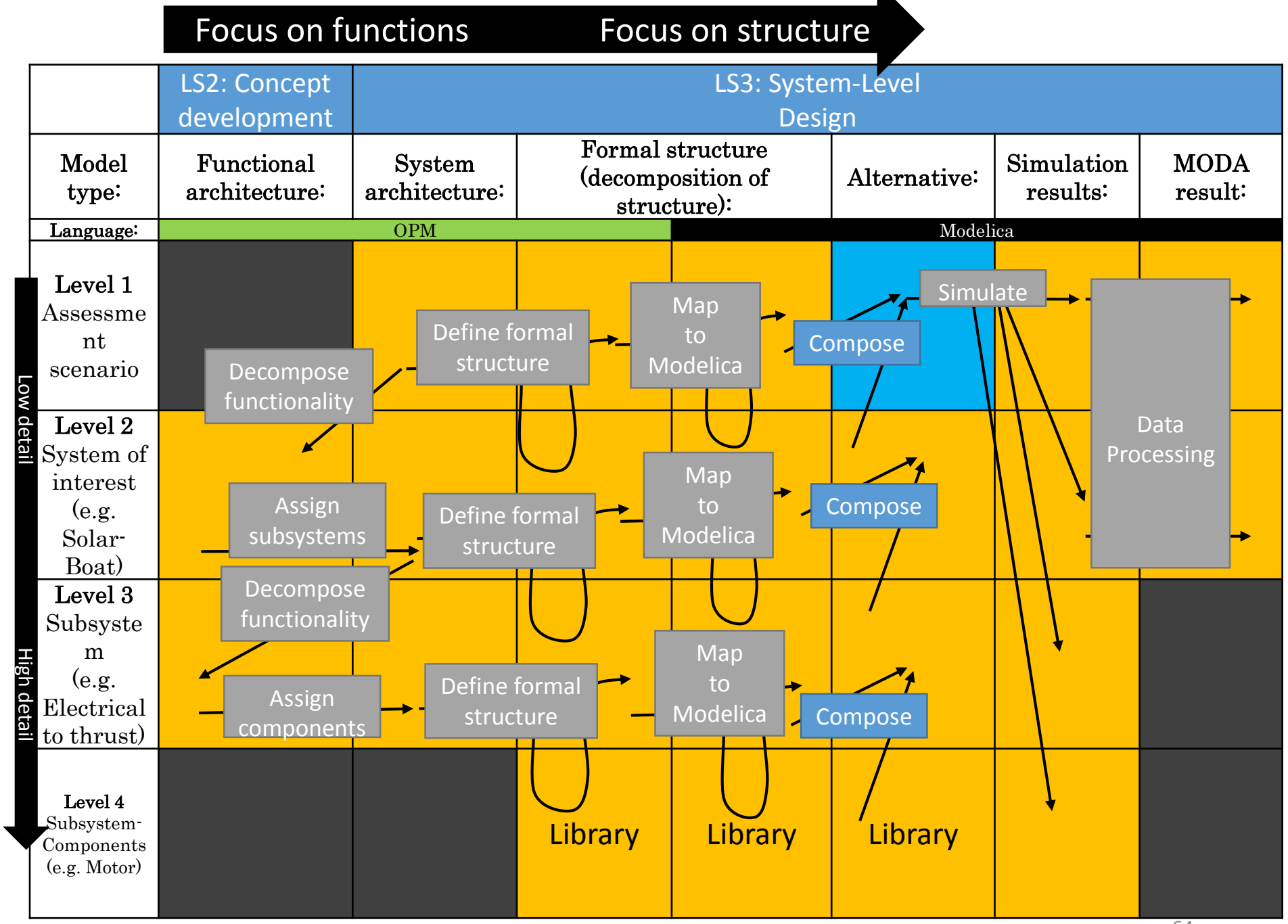
Map to Modelica

Level 3

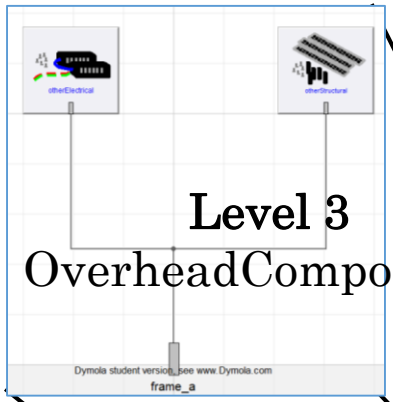




Level 4
(Subsystem-Components library)

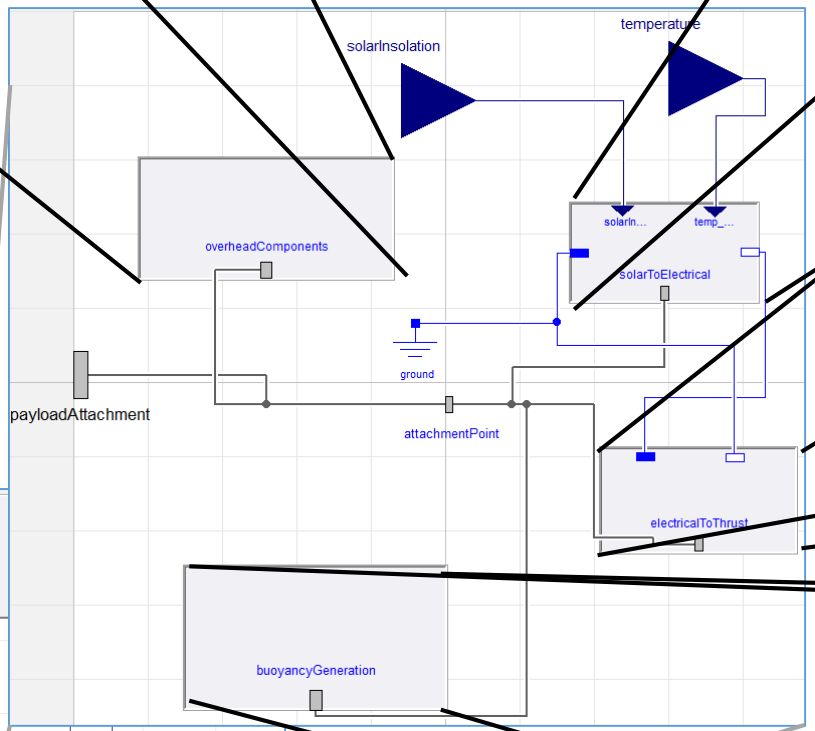

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Compose

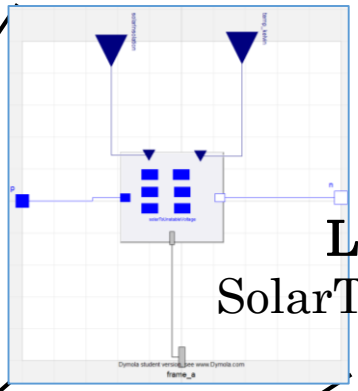


Level 3
OverheadComponents

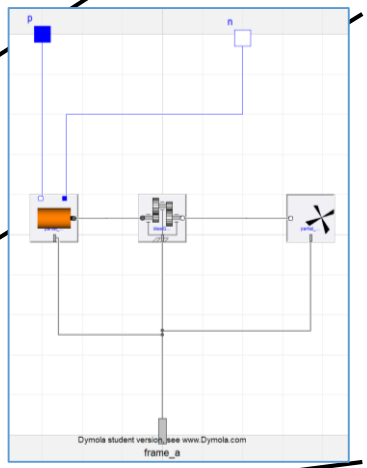
Level 2
Solar-Boat



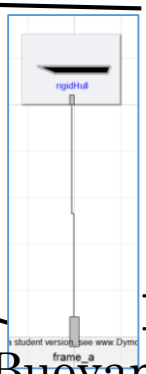
Level 1
Assessment scenario



Level 3
SolarToElectrical

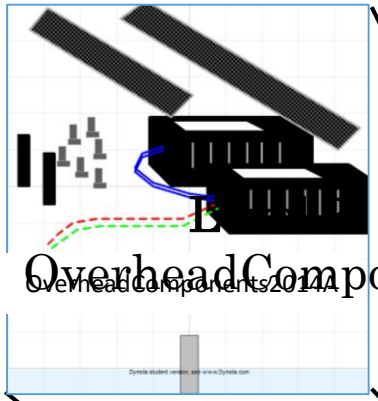


Level 3
ElectricalToThrust

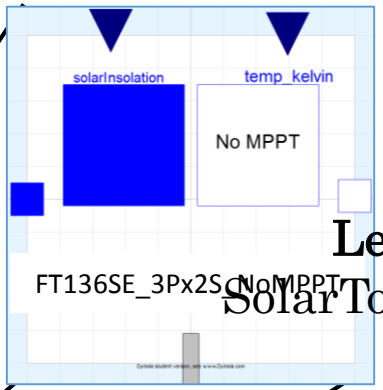


Level 3
BuoyancyGeneration

Compose

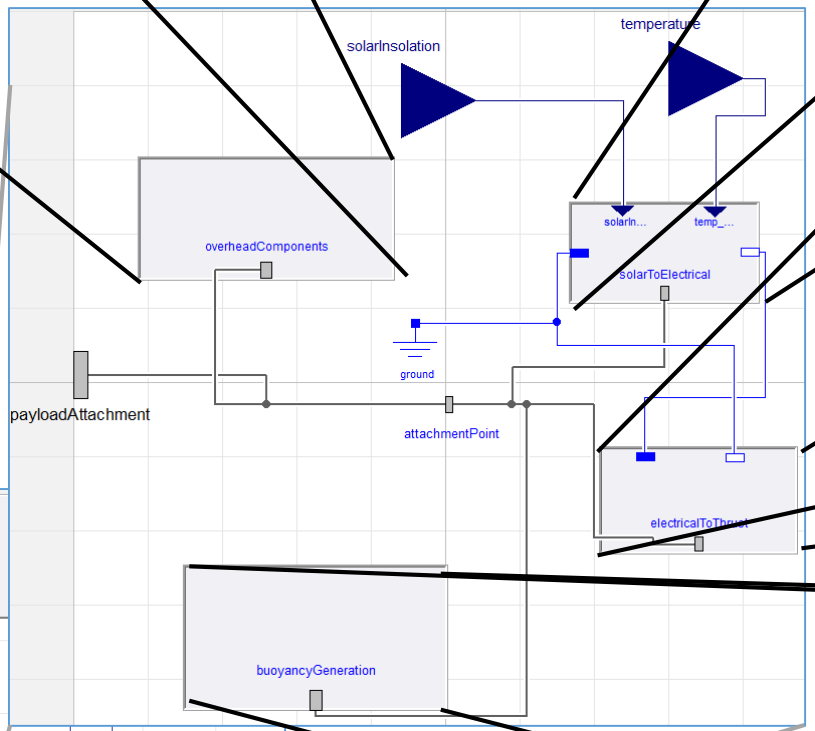


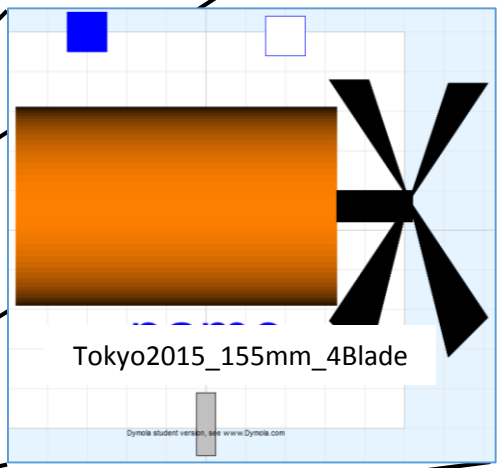
Level 3
OverheadComponents



Level 3
SolarToElectrical

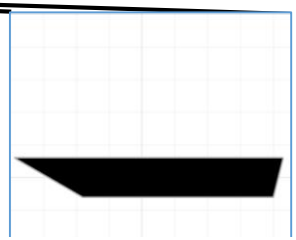
Level 2
Solar-Boat





Level 3
ElectricalToThrust

Level 1
Assessment scenario

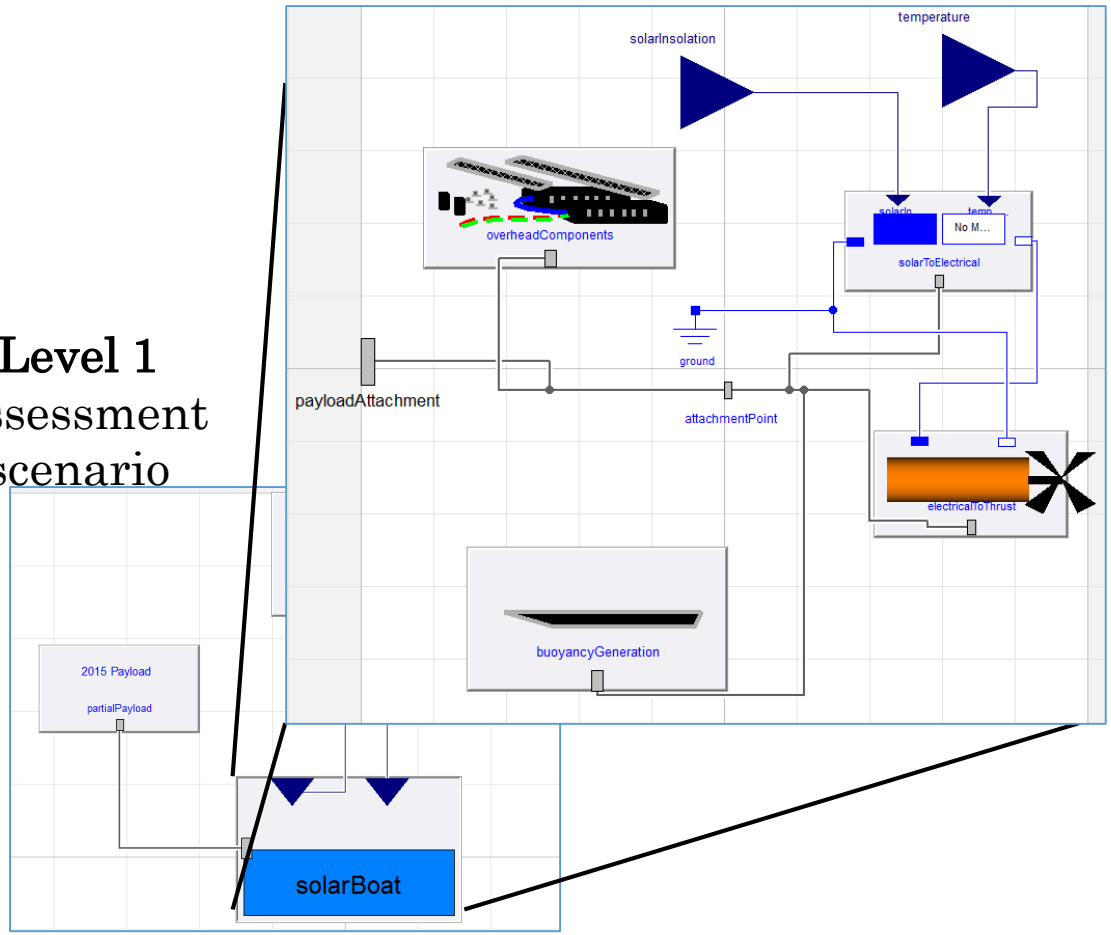


Level 3
BuoyancyGeneration

Compose

Level 2
Solar-Boat

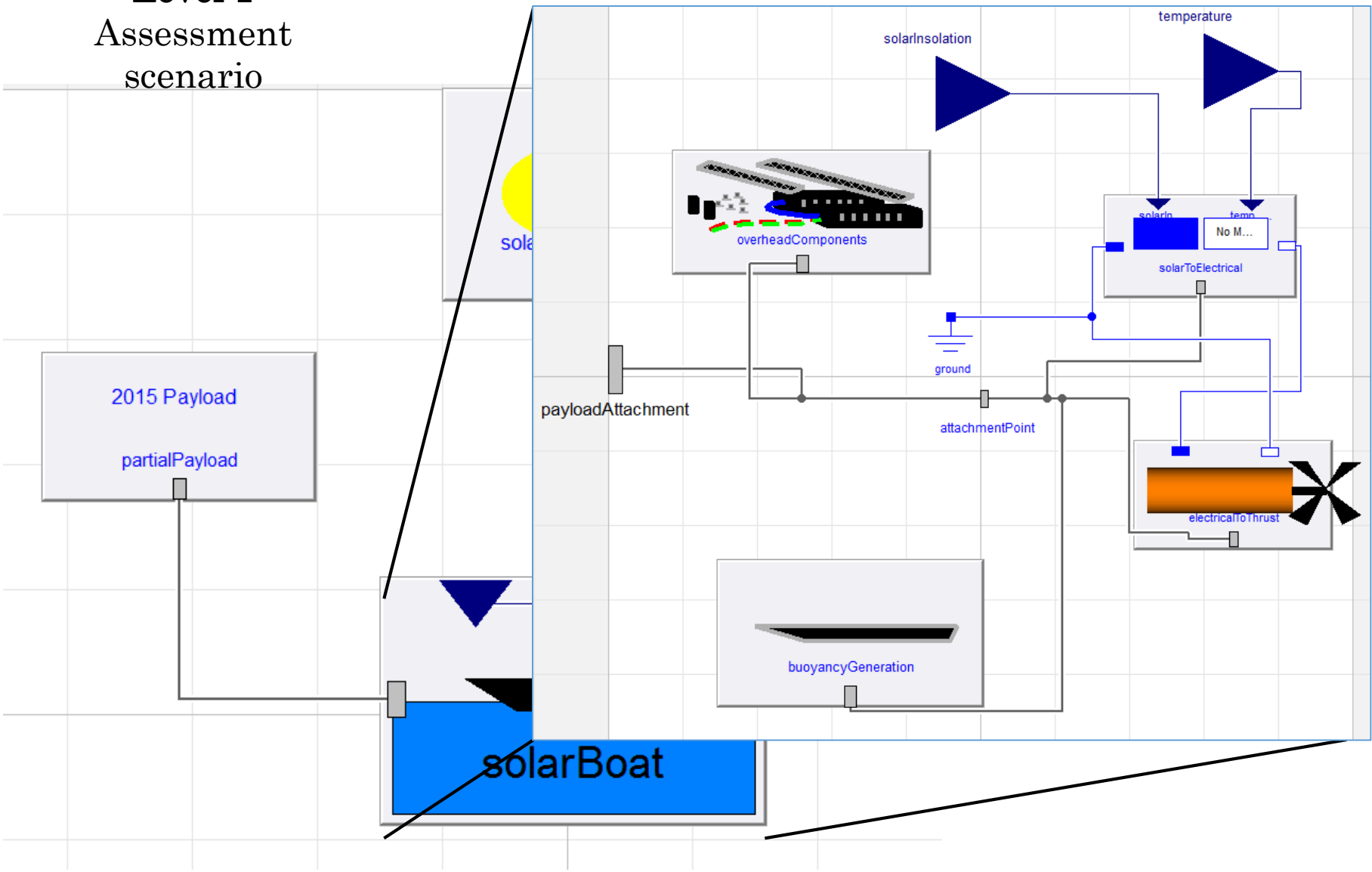
Level 1
Assessment scenario



Compose

Level 1
Assessment scenario

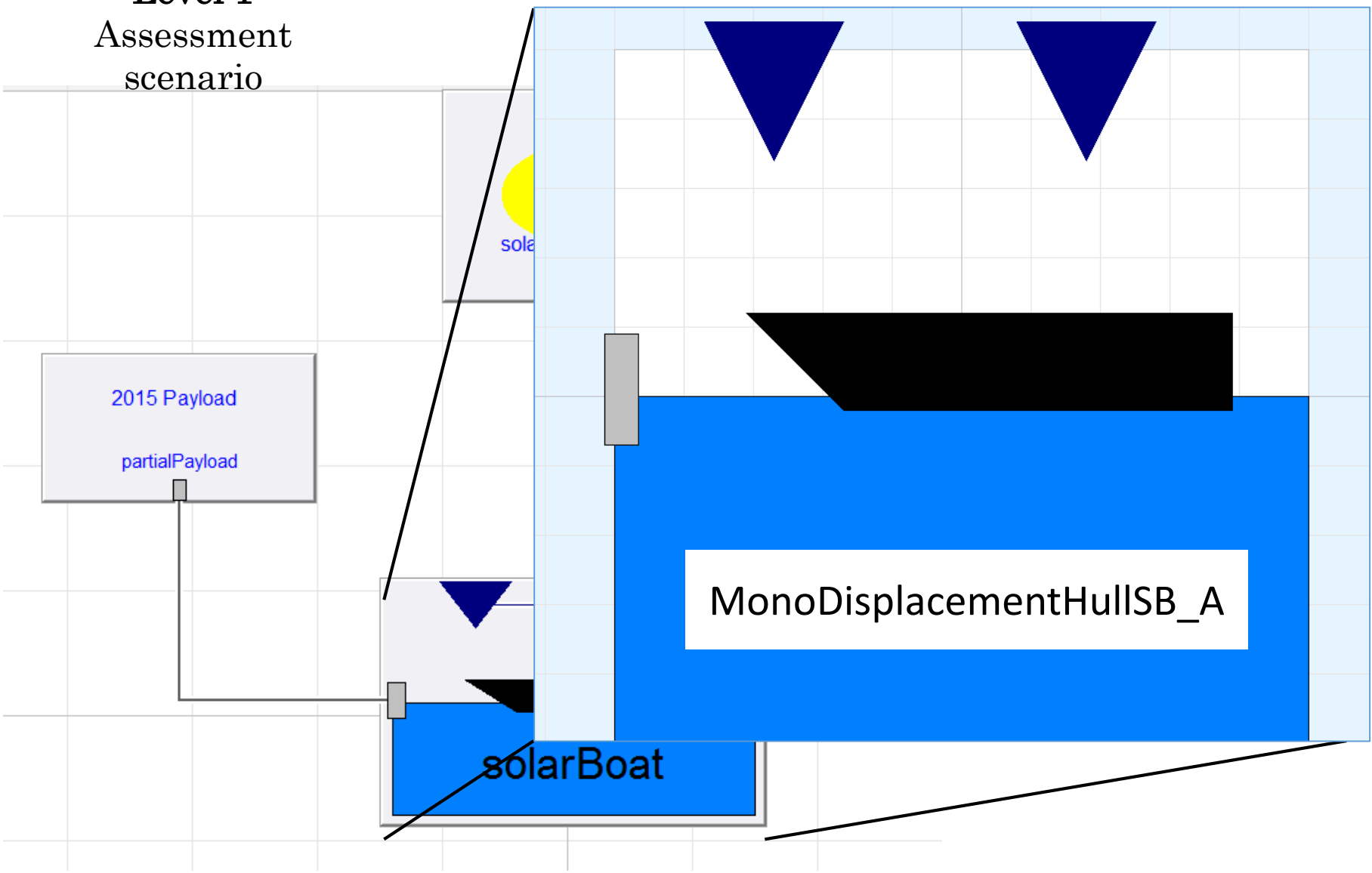
Level 2
Solar-Boat



Compose

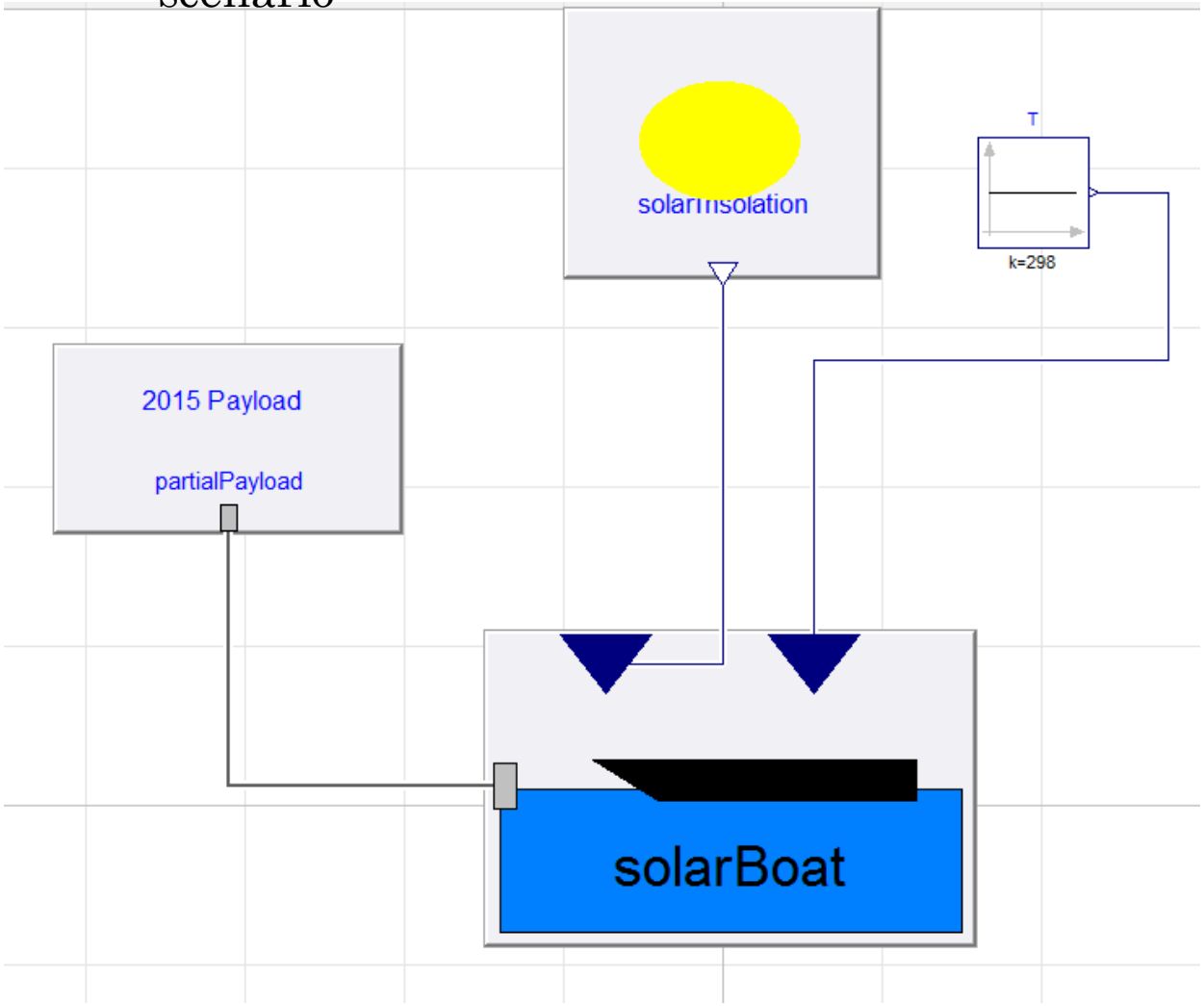
Level 1
Assessment scenario

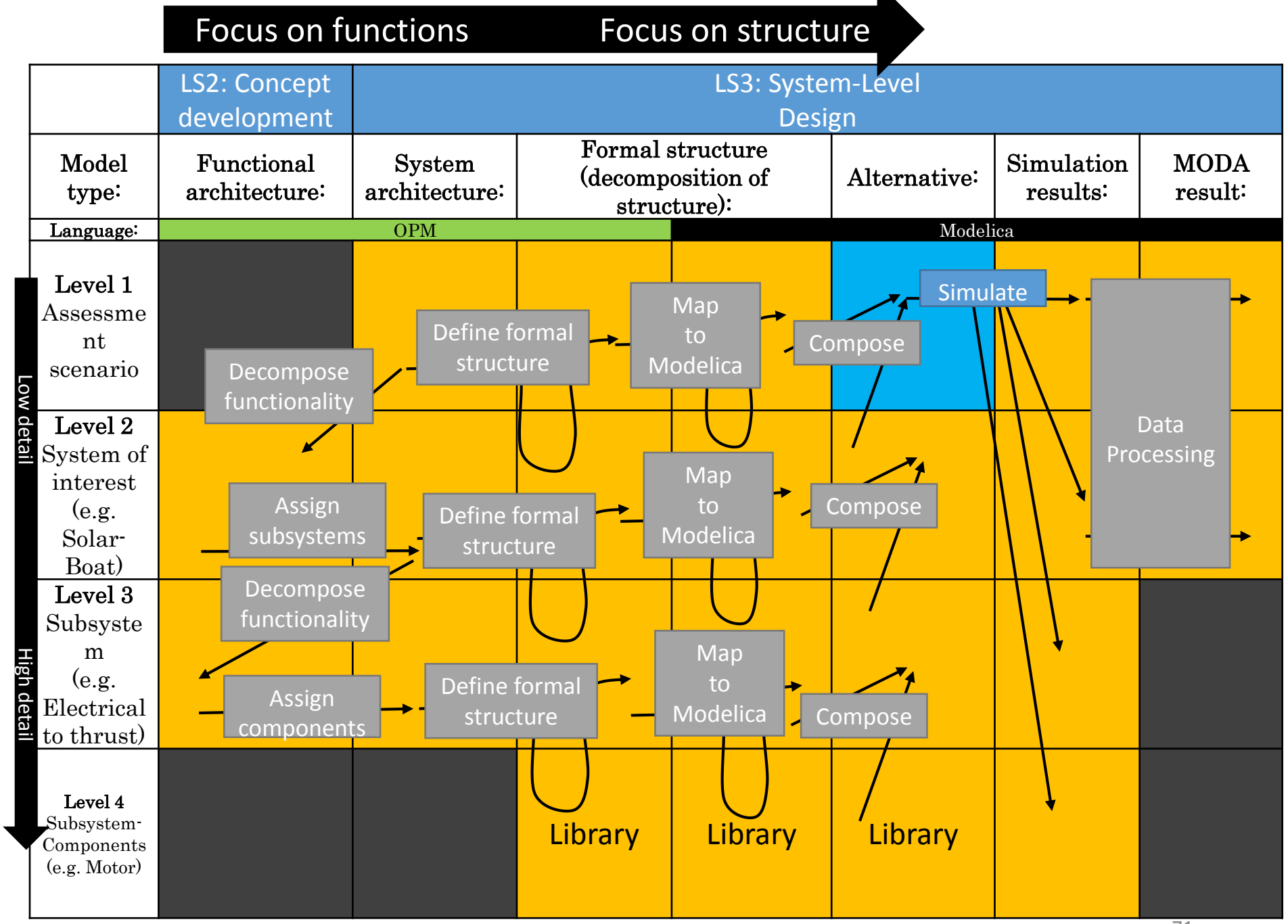
Level 2
Solar-Boat



Compose

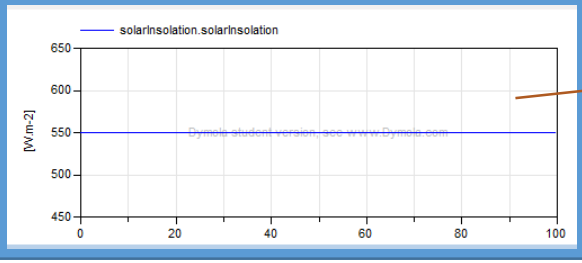
Level 1
Assessment scenario




Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

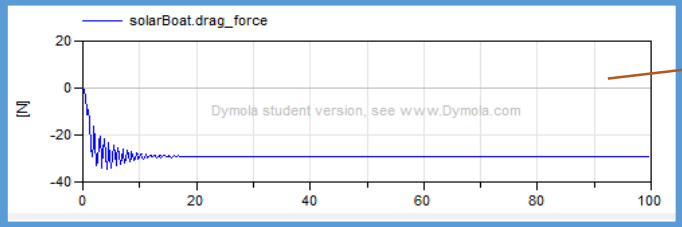
Simulation results – Time series examples (note from different simulation runs)

Level 1



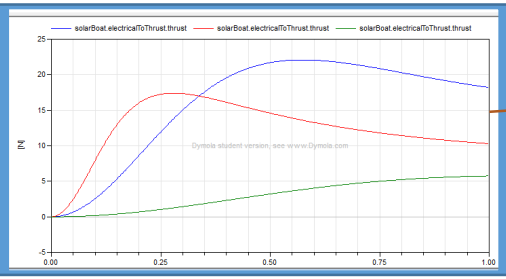
Solar Insolation input [Wm^{-2}]

Level 2



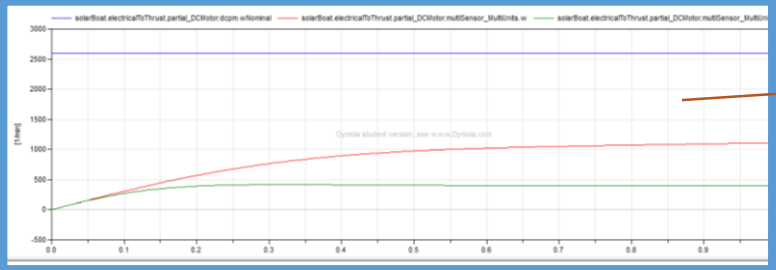
SolarBoat drag force [N]

Level 3

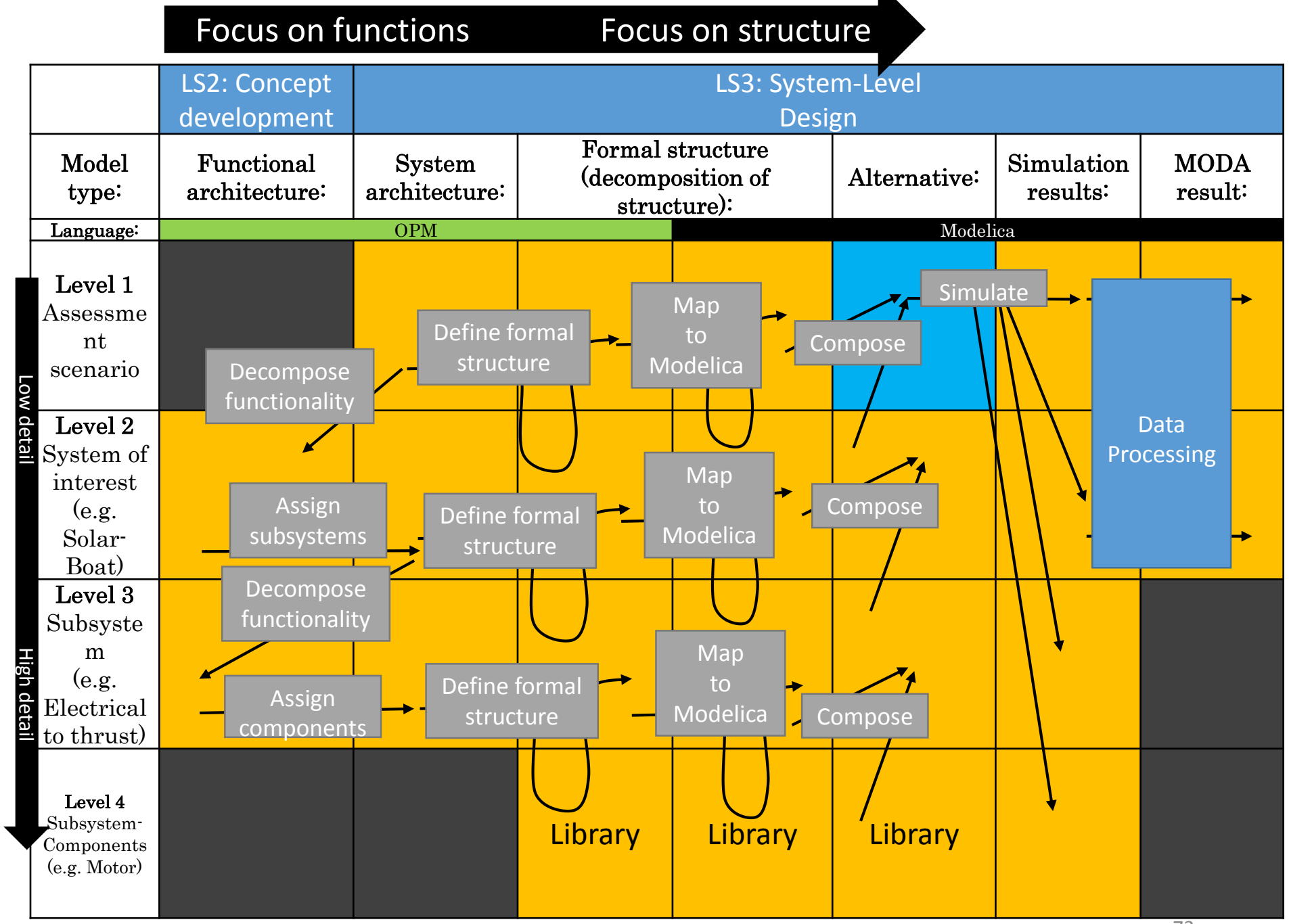


Electrical to Thrust Subsystem, Thrust [N]

Level 4



Motor velocity [RPM]


Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Multi Objective Decision Analysis: Automated implementation

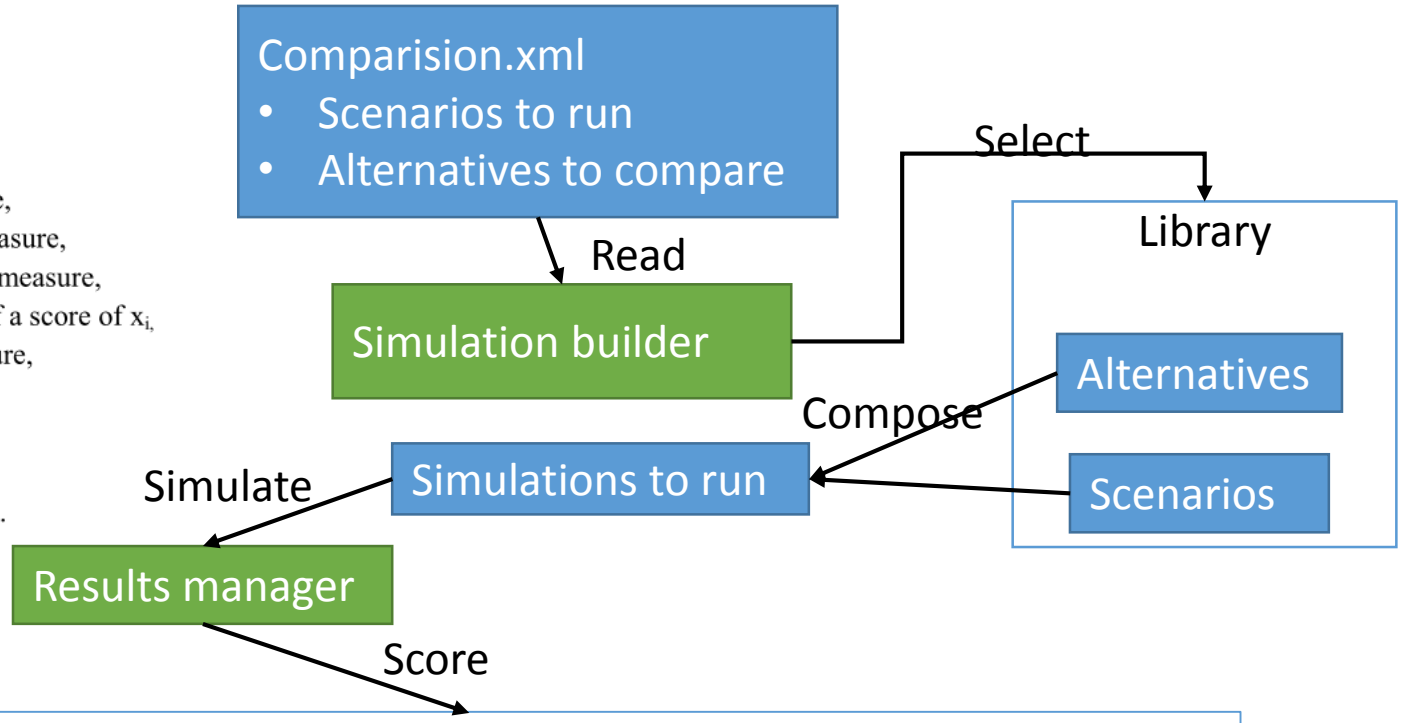
$$v(x) = \sum_{i=1}^{n} w_i v_i(x_i)$$

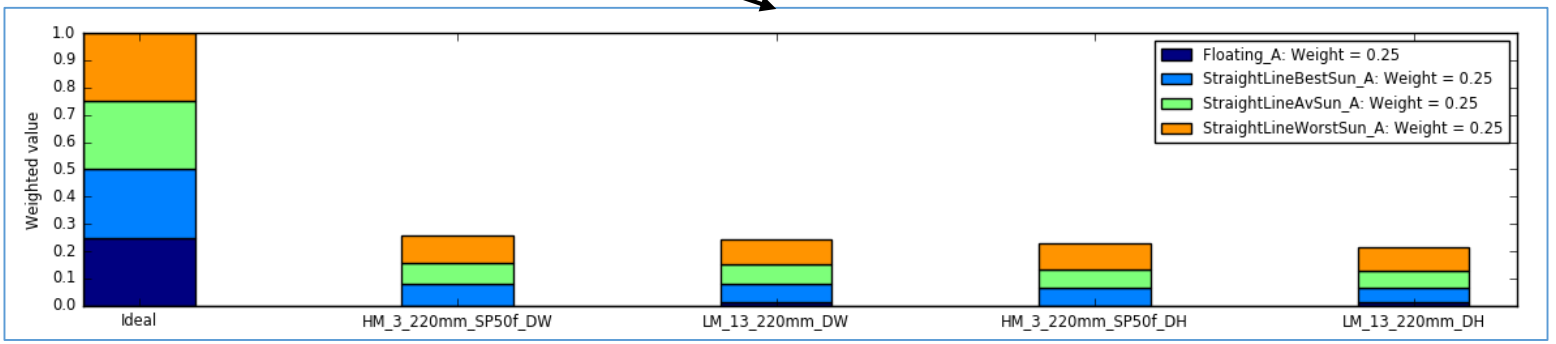
where
v(x) is the alternative's value,
i = 1 to n is the number of the measure,
x_i is the alternative's score on the i^{th} measure,
$v_i(x_i)$ = is the single dimensional value of a score of x_i,
w_i is the weight of the i^{th}measure,

$$\sum_{i=1}^{n} w_i = 1$$

and (all weights sum to one).

From Cilli & Parnell, 2014

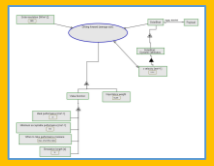
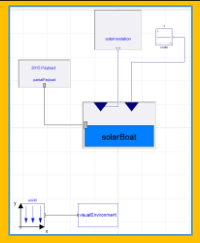
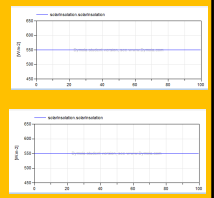
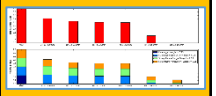
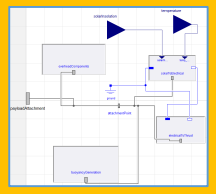
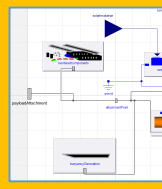
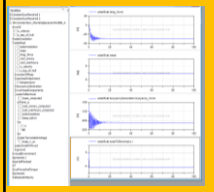
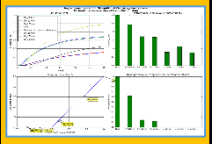
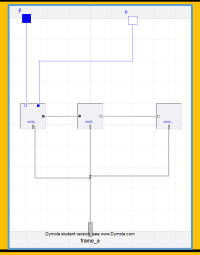
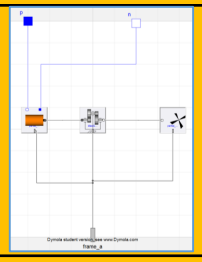
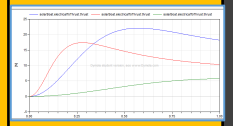
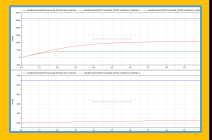




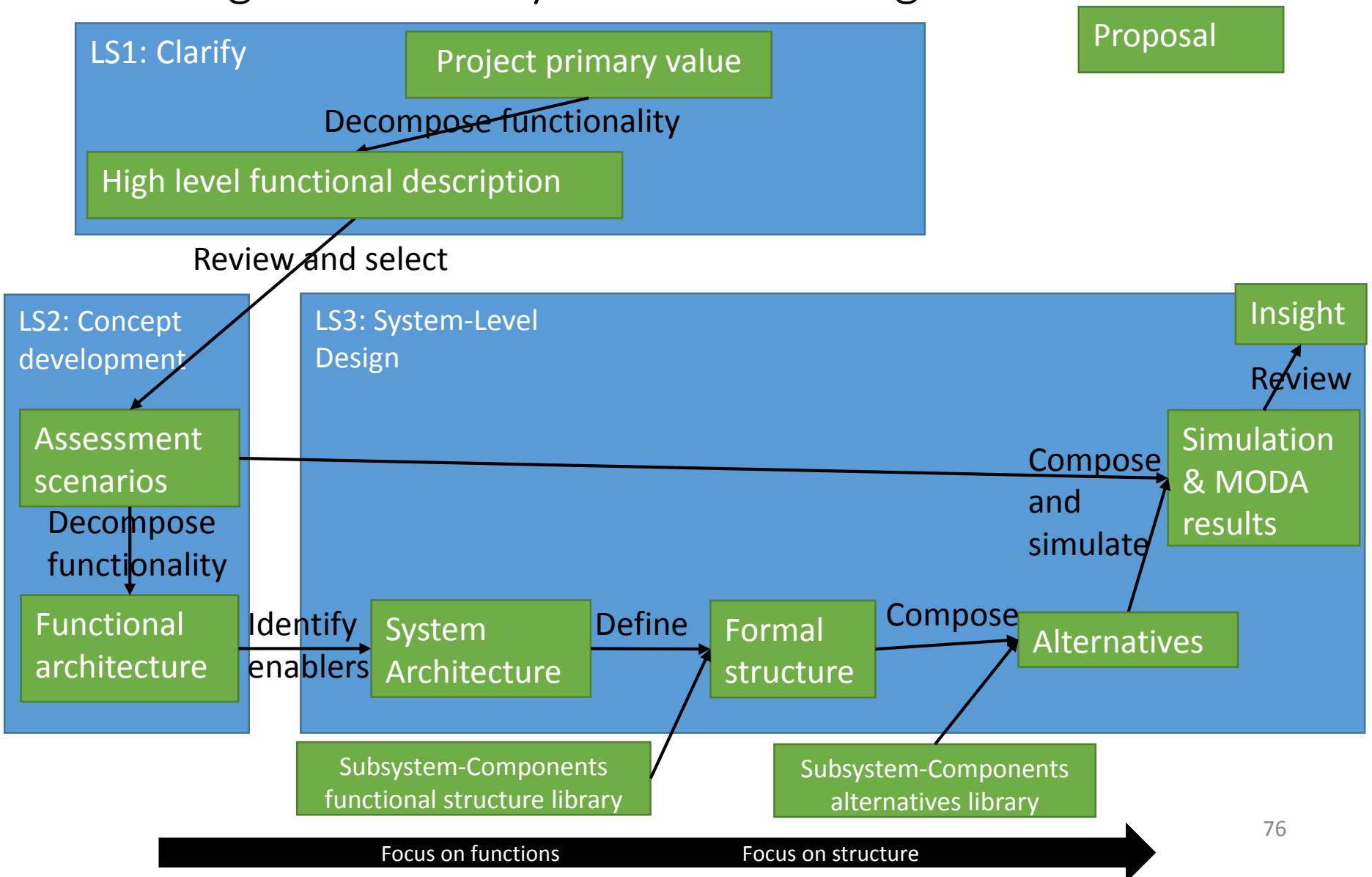
← Alternatives →

Focus on functions → Focus on structure

Low detail ↓ High detail

	LS2: Concept development	LS3: System-Level Design				
Model type:	**Functional architecture:**	**System architecture:**	**Formal structure (decomposition of structure):**	**Alternative:**	**Simulation results:**	**MODA result:**
Language:	OPM			Modelica		
Level 1 Assessment scenario						 
Level 2 System of interest (e.g. Solar-Boat)							
Level 3 Subsystem (e.g. Electrical to thrust)							
Level 4 Subsystem-Components (e.g. Motor)			Library	Library	Library		

Proposed tools and methodologies for Knowledge Management and System-Level Design

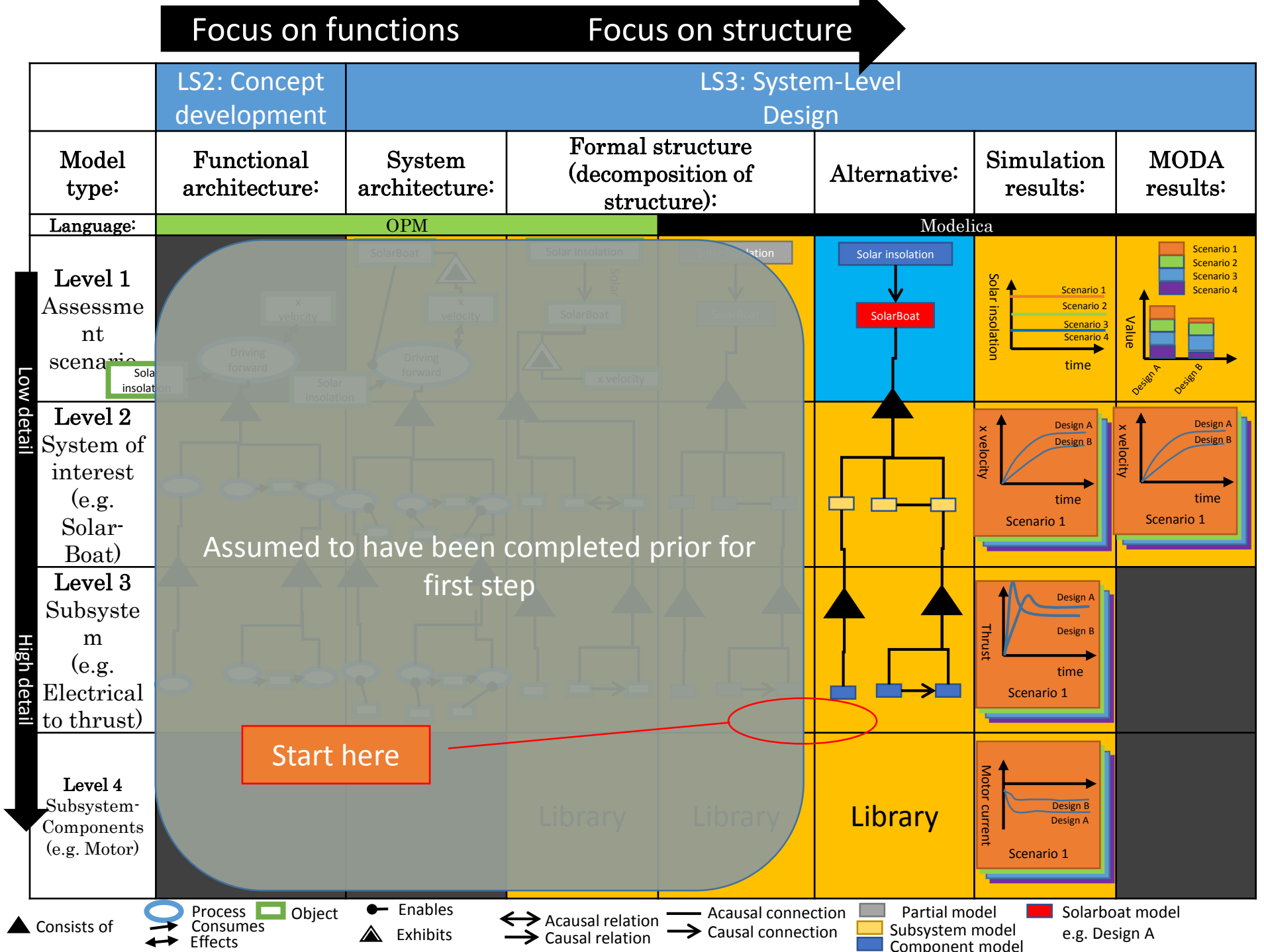


Overview

1. Problems and proposed solutions identified from the 2014 Solar-Boat project
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. **Demonstration**
4. Discussion
5. Conclusions
6. PhD plans

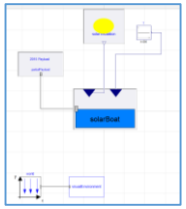
Demonstration

- Student design goal:
  1. Develop initial Solar-Boat design for prototype for standard rules:
     1. Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers)
     2. Design Set 2 (Subsystem Functional Architecture variation: add a motor speed changing device)
     3. Design Set 3 (Subsystem-Component variation with cost impact: compare new expensive solar panels which are more efficient but more heavy)
  2. Experience a rule change. Payload from 0.064kg to 15kg:
     1. Design Set 4 (Rule change: see impact of larger payload)
     2. Design Set 5 (Subsystem Formal Structure variation: Address the payload issue with different buoyancy system & Change assessment approach)

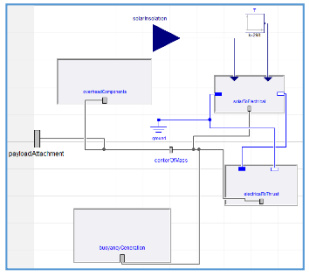

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA results:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Solar insolation
SolarBoat
Scenario 1
Scenario 2
Scenario 3
Scenario 4
time
Value
Design A
Design B
x velocity
Thrust
Motor current
Assumed to have been completed prior for first step
Start here
Library
Consists of
Process
Consumes
Effects
Object
Enables
Exhibits
Acausal relation
Causal relation
Acausal connection
Causal connection
Partial model
Subsystem model
Component model
Solarboat model
e.g. Design A

As such Formal Structures and Assessment Scenarios are ready for population

Level 1
Assessment scenarios

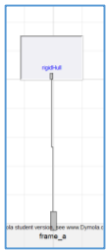


Level 2
Solar-Boat

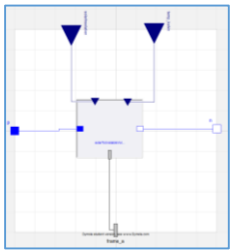


Measure of interest	Scenario conditions	Minimum acceptable performance	Stretch goal	Data extraction type
Top of hull z position (m)	Floating	-0.1	-0.4	Mean
x velocity (m/s)	Best ever insolation (870 Wm^2) straight line driving	2	4	Maximum value
x velocity (m/s)	Average insolation (550 Wm^2)	1.5	3	Maximum value
x velocity (m/s)	Worst ever insolation (260 Wm^2)	0.5	2.5	Maximum value

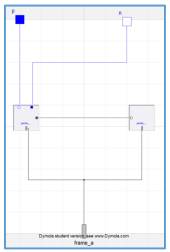
Level 3
BuoyancyGeneration



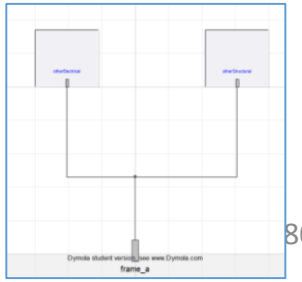
Level 3
SolarToElectrical



Level 3
ElectricalToThrust

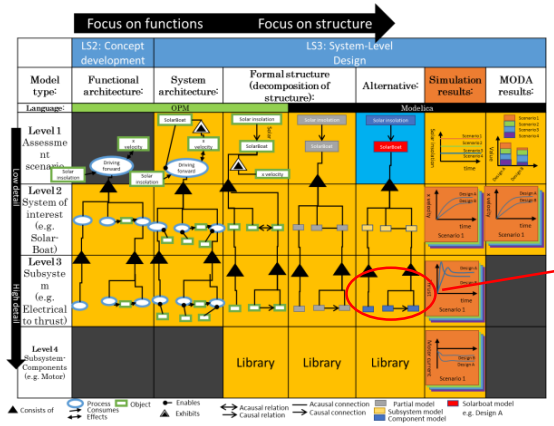


Level 3
OverheadComponents



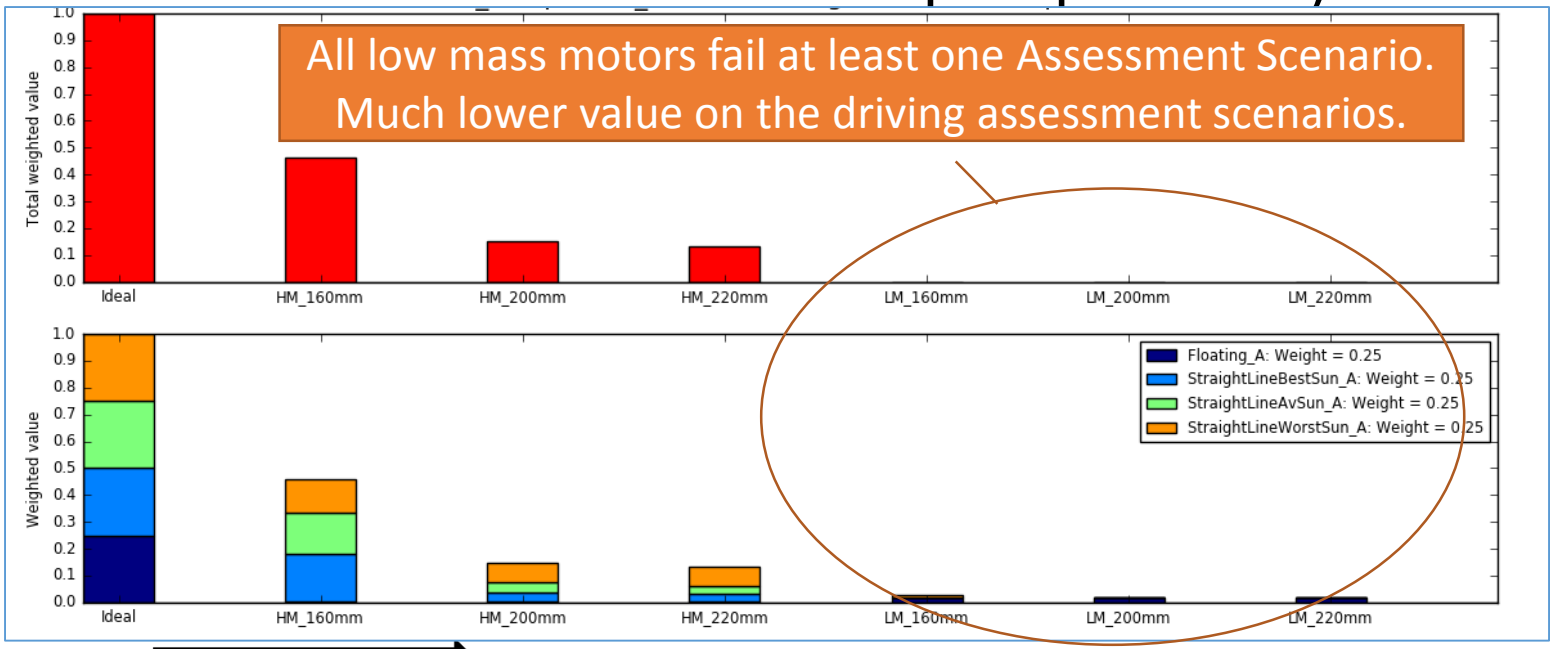
Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers)

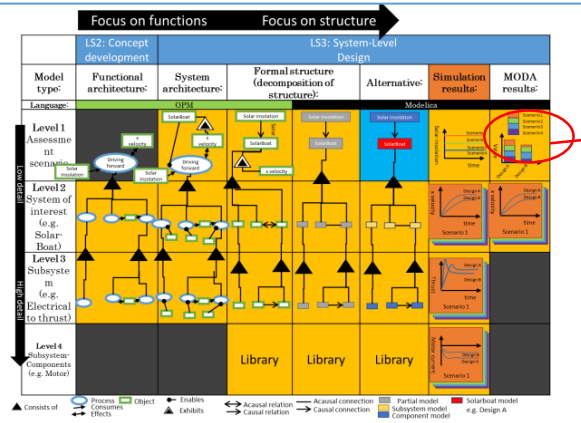
Alternative name	Buoyancy Generation	SolarToElectical	ElectricalToThrust
HM_160mm	Single hull	Old solar panel	H motor: No gearbox: 160mm prop
HM_200mm	Single hull	Old solar panel	H motor: No gearbox: 200mm prop
HM_220mm	Single hull	Old solar panel	H motor: No gearbox: 220mm prop
LM_160mm	Single hull	Old solar panel	L motor: No gearbox: 160mm prop
LM_200mm	Single hull	Old solar panel	L motor: No gearbox: 200mm prop
LM_220mm	Single hull	Old solar panel	L motor: No gearbox: 220mm prop



Variation

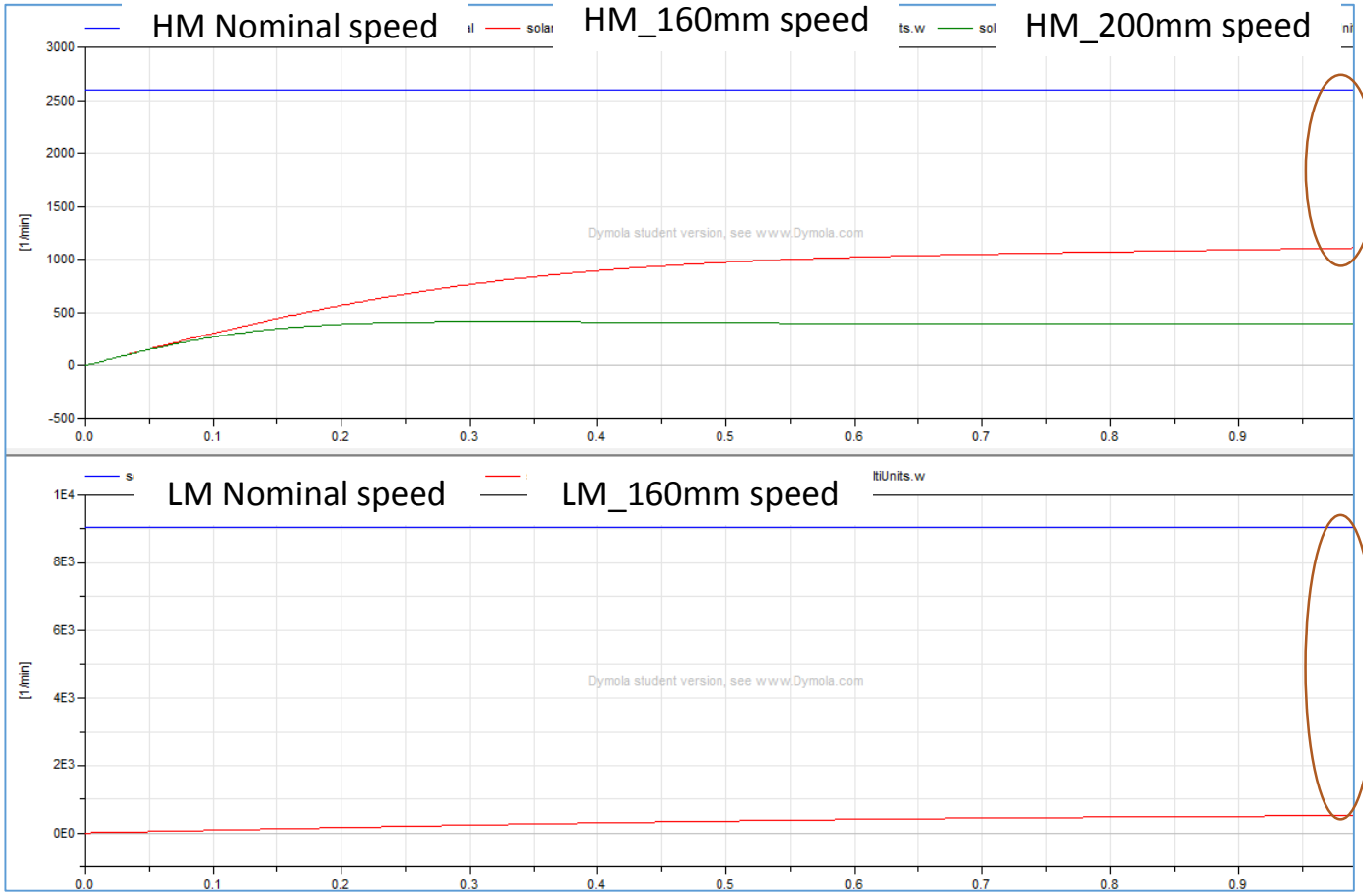
Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers)





Review

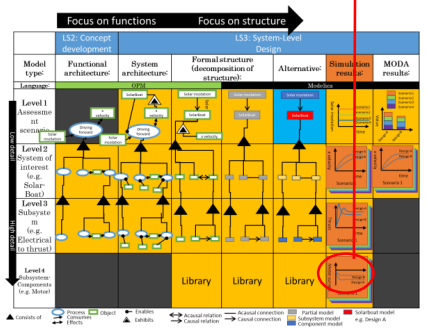
Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers) Compare motor speed



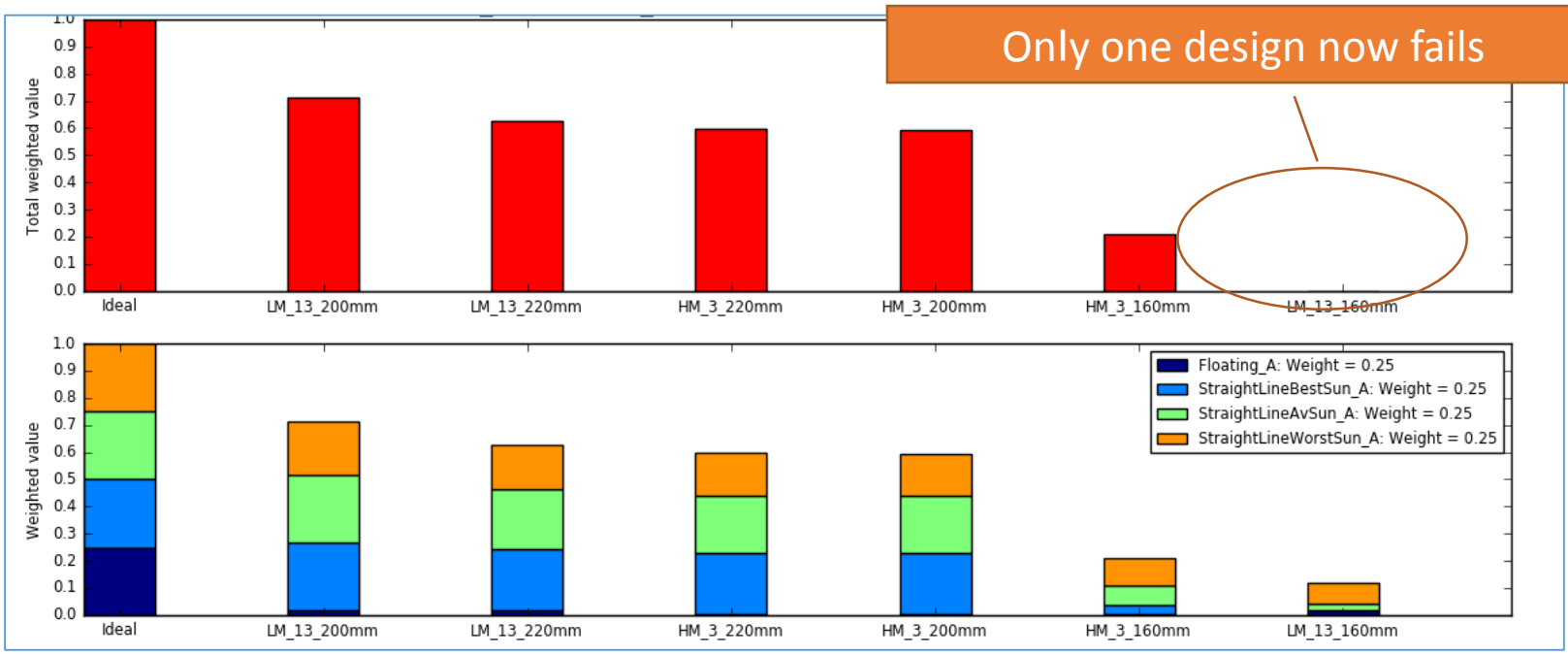
Closer to nominal speed but still quite far from nominal (2.36 times)

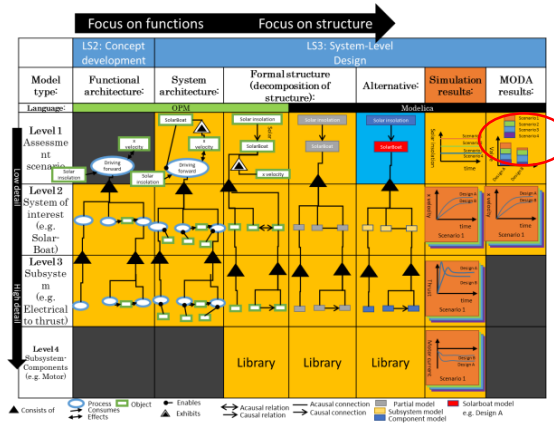
18 times from nominal

Review



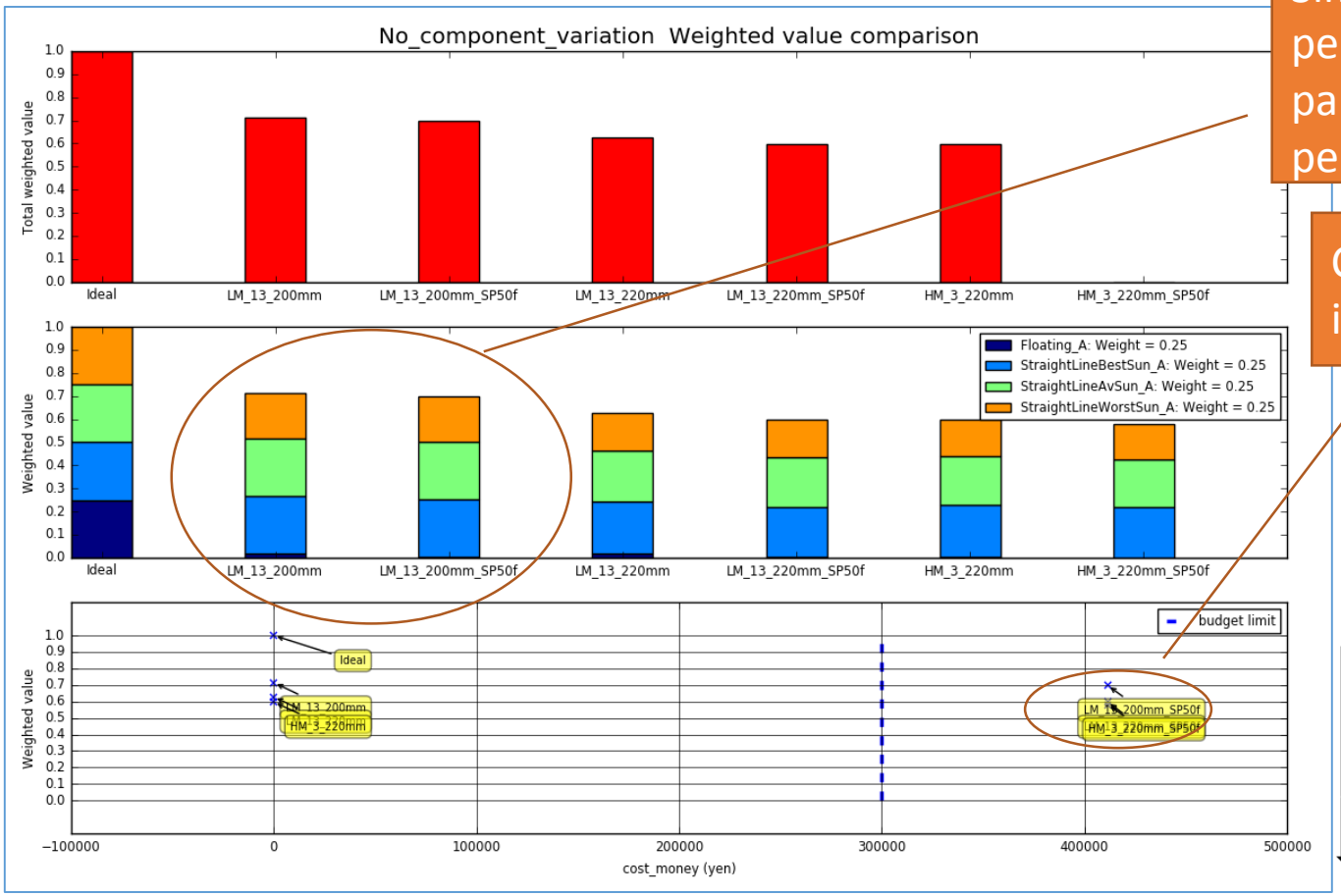
Design Set 2 (Subsystem Functional Architecture variation: add a motor speed changing device) How to get motor to spin closer to nominal speed





Review

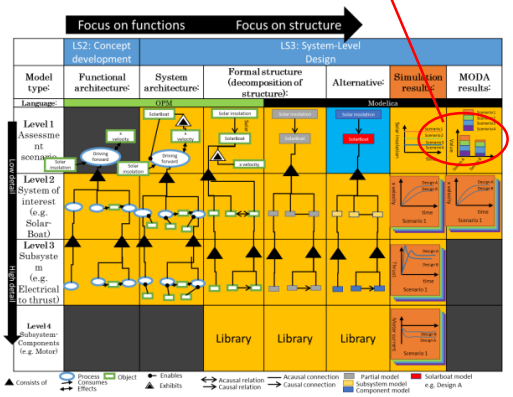
Design Set 3 (Subsystem-Component variation with cost impact: compare new expensive solar panels (7% more efficient but 2x mass over old ones)



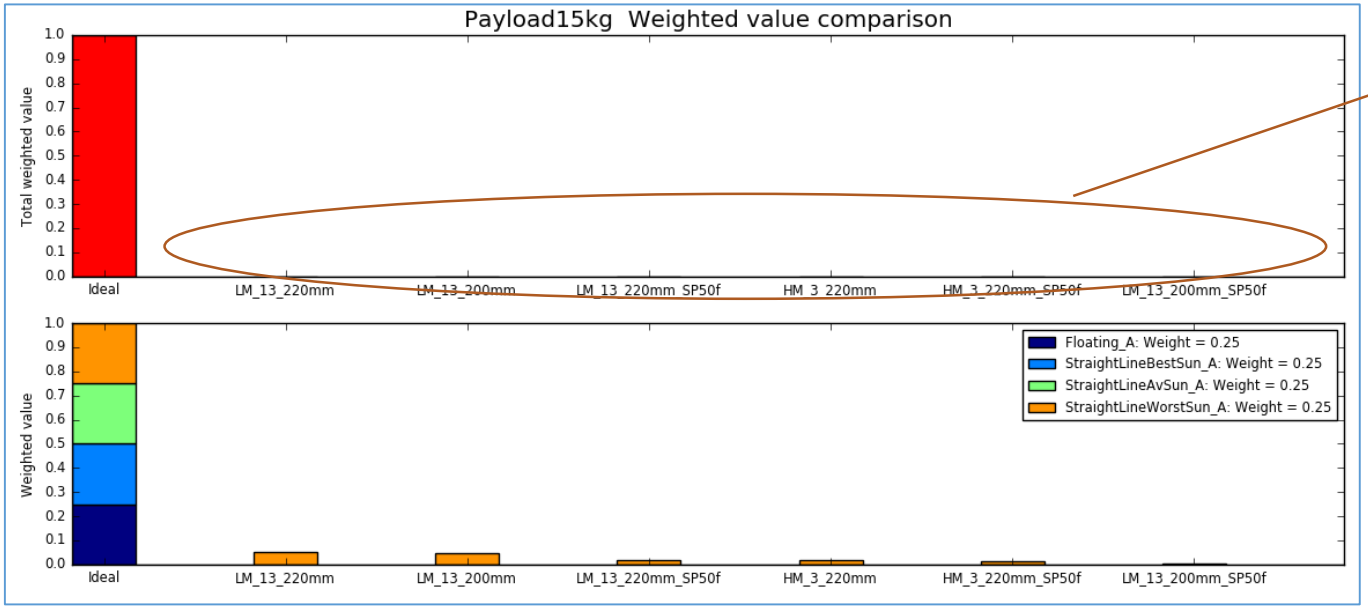
Similar speed performance, but old panel has better floating performance

Cost of the new panels is beyond the budget

Review

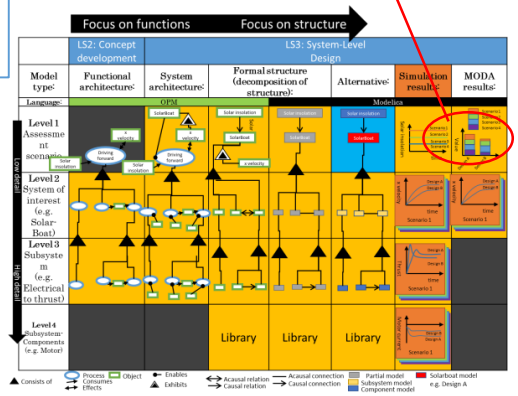


Design Set 4 (Rule change: see impact of larger 15kg payload vs 0.064kg payload)

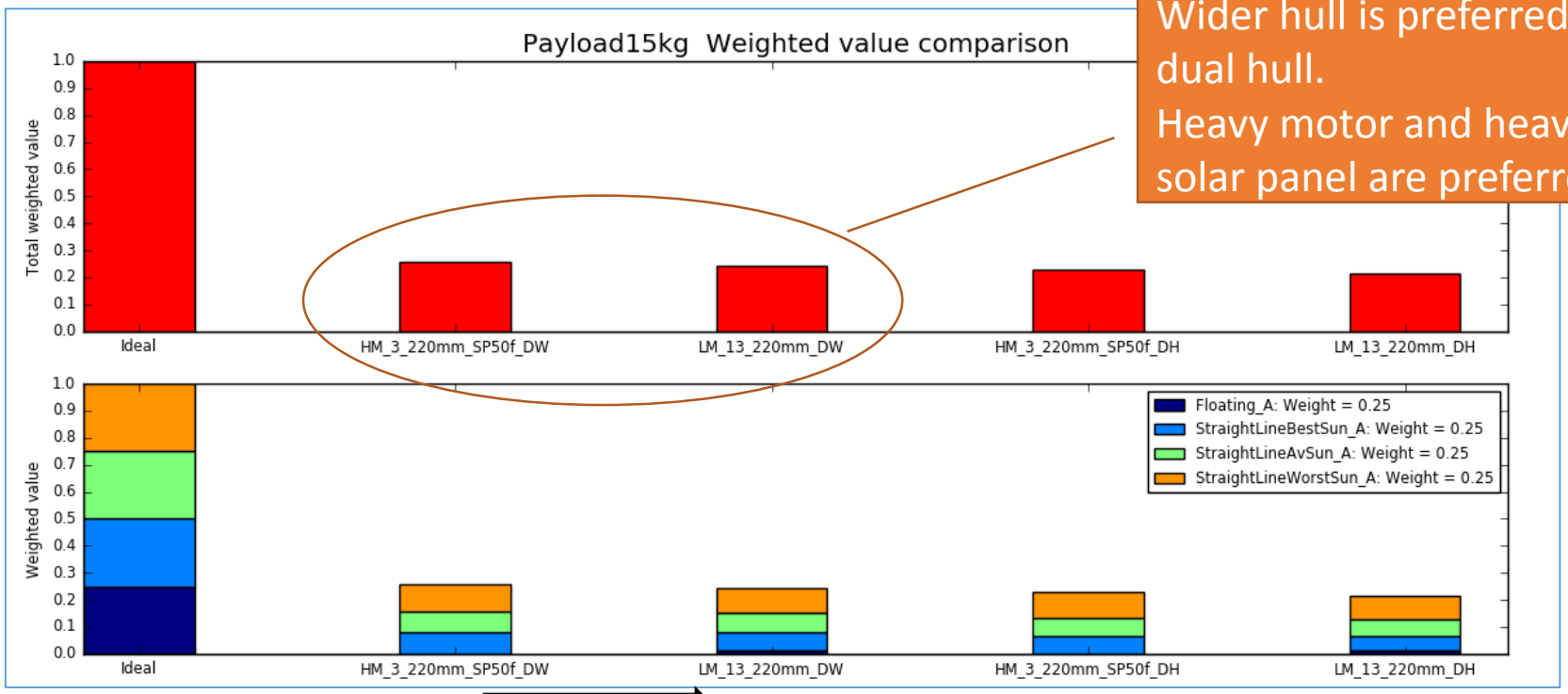


All now fail

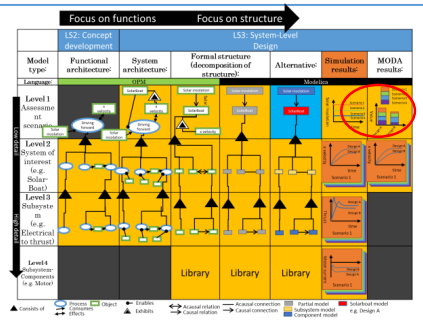
Review



Design Set 5 (Subsystem Formal Structure variation: Address the payload issue with different buoyancy systems. Two hulls or dual hull?



Wider hull is preferred to dual hull.
Heavy motor and heavy solar panel are preferred.



Review

Overview

1. Problems and proposed solutions identified from the 2014 Solar-Boat project
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. Examples and demonstrations
4. **Discussion**
5. Conclusions
6. PhD plans

Discussion – Benefits

- **Logical synthesis** of hierarchical **formal structure** for **new product** development. **Not reliant** on a **pre-provided** one
- **Fast synthesis** of **alternative** system and subsystem **designs**

- **Fast consistent assessment** of **alternative designs** value by **automated simulation** and comparison
- **Deeper** component, subsystem and system **knowledge gain** by **exploring rich simulation** results

- **Systems Engineering foundations** introduced to **students**
- **Tacit** and **document knowledge** is **avoided**. **Integrate** various models and simulation across the **lifecycle**

Discussion – Benefits

Point of novelty:
By functionally decomposing (in OPM) and mapping to a formal structure (in Modelica):
- Object based numerical simulation of OPM is attempted
- Integration of two important languages is attempted

- **Logical synthesis** of hierarchical **formal structure** for **new product** development. **Not reliant** on a **pre-provided** one
- **Fast synthesis** of **alternative** system and subsystem **designs**

- **Fast consistent assessment** of **alternative designs** value by **automated simulation** and comparison
- **Deeper** component, subsystem and system **knowledge gain** by **exploring rich simulation** results

- **Systems Engineering foundations** introduced to **students**
- **Tacit** and **document knowledge** is **avoided**. **Integrate** various models and simulation across the **lifecycle**

Discussion – Shortcomings & Further work

- Methodology logic:
  - Assumes one object enables one process.
    - No explicit provision for situations where two subsystems enable a process or vice-versa. Subsystems are merged.
  - Timing and control logic of behavior:
    - Currently behavior being modelled in Modelica assumes to occur at all times. Real systems exhibit causality when a particular behavior is triggered (e.g. time triggering of a process or if->then).
- Tool implementation:
  - Automate the generation of alternative designs (component placement and parameter variation)
- As such:
  - Needs demonstration on larger more complex projects

Overview

1. Problems and proposed solutions identified from the 2014 Solar-Boat project
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. Examples and demonstrations
4. Discussion
5. **Conclusions and summary**
6. PhD plans

Description and problems of: Early Lifecycle Stages of the Solar-Boat project

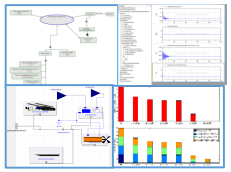
Lifecycle Stage:	LS1: Clarify	LS2: Concept dev	LS3: System-Level Design
Activities:	Review past knowledge	Defining required functions	Comparing and selecting System-Level Design
Identified problems:	Slow time to acquire initial knowledge	Unclear what the design target was	Little exploration of alternatives and their predicted outcomes
Proposed solutions:	Provide knowledge in models	Complete trade-off analysis of multiple designs using models to simulate performance	
Problems / Difficulties with implementing solutions:	• What languages? • Integrate multiple languages? • Keeping models update	• Framework to assess all alternative designs • Comparing a reasonable number of alternatives • Numerical optimization vs. exploratory approaches	
Thesis aim:	To propose tools and methodologies to help students: • Manage project knowledge • Explore concept designs		

Conclusions and summary

- **Methodology** and **tools** integrating **OPM** and **Modelica** to **synthesize alternative** Systems-Level Solar-Boat **Designs** and **assess** those designs was presented with the aim of **increasing** the amount **knowledge** of the system for students including:
  - **Decomposing** the **functionally**
  - **Identifying** common **assessment** scenarios
  - **Synthesizing formal structure**
  - Rapidly composing **alternative designs** by **populating** the **formal structure**
  - **Simulating** each **alternative design** for each **assessment scenario**
  - **Comparing** predicted **performance** of the alternatives

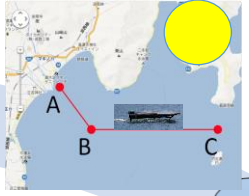
Conclusions and summary

- A **demonstration** of the tools was presented indicating:
  - **Automated comparison** of multiple **alternatives** for **assessment scenarios**
  - **Exploration** of rich **simulation results**
- Using the methodology and tools **knowledge** was **generated**, **stored** and **consumed** at appropriate at appropriate times, aiming to enabling knowledge management success



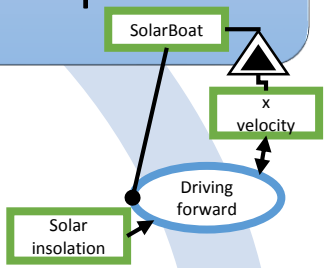
Team change

LS7: Knowledge transfer



LS1: Clarify

LS2: Concept development



Review knowledge

Develop System diagram

Decompose to functional architecture

Refine assessment scenarios

Model, simulate, compare alternative designs

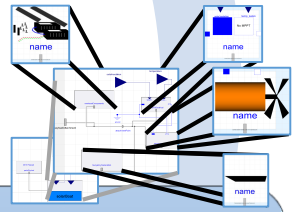
Design, simulate, trade-off subsystems

Manufacture & test with models, simulate modifications

Compare performance to simulations

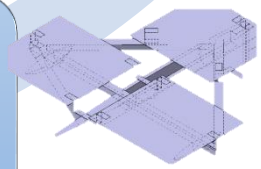
Adjust simulation framework, report, lessons learned

Model-based knowledge

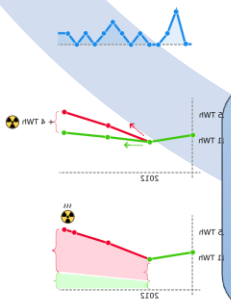


LS3: System-level design

LS4: Detail design



LS5: Production, test, refinement



LS6: Race!

Overview

1. Problems and proposed solutions identified from the 2014 Solar-Boat project
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. Examples and demonstrations
4. Discussion
5. Conclusions
6. **PhD plans**

PhD plan

- Industrial background
- Master’s further work
- Trying to extend to more extensive products
- Problems to solve
- Potential solutions
- Project steps
- Proposed outcome

Industry background

- Systems development in modern times:
  - **No time:**
    - Fast market changes
  - **Too complicated to handle:**
    - Multiple engineering domains integrated to develop high performance high functionality systems
      - Modular and hierarchy clarity reduced to improve performance
  - **Very complex:**
    - Many, many components

- 1D-CAE: Enables composition and assessment of systems quickly and efficiently

Discussion – Shortcomings & Further work

- Methodology logic:
  - Assumes one object enables one process.
    - No explicit provision for situations where two subsystems enable a process or vice-versa. Subsystems are merged.
  - Timing and control logic of behavior:
    - Currently behavior being modelled in Modelica assumes to occur at all times. Real systems exhibit causality when a particular behavior is triggered (e.g. time triggering of a process or if->then).
- Tool implementation:
  - Automate the generation of alternative designs (component placement and parameter variation)

- As such:
  - Needs demonstration on larger more complex projects

More complex

More complicated

Problems with extending the current approach to more large scale products

- Current design method:
  - Does not support non continuous running processes
  - Is too modular. Objects enabling multiple processes and processes being enabled by multiple objects
- Current systems scalability:
  - Not many domains used for modeling
  - Process decomposition is based on the engineers idea not a formal approach

Problems with extending the current approach to more large scale products

- Current design method:
  - Does not support non continuous running processes
  - Is too modular. Objects enabling multiple processes and processes being enabled by multiple objects
- Current systems scalability:
  - Not many domains used for modeling
  - Process decomposition is based on the engineers idea not a formal approach

Goals

1. Improve the design method to enable 1D-CAE method of non continuous, non modular conceptual designs.

2. Develop integrated modeling management system.

3. Develop method for decomposing processes (systems function) and system.

Proposed solutions to meet the goals

1. Improve the design method to enable 1D-CAE method of non continuous, non modular conceptual designs:
   - Define behavior descriptions of components (in OPM) to associate with 1D-CAE models which represent the richer behavior
2. Develop method for decomposing process and system:
   - Identify the systems requirement and decompose into functions (processes and operands) based on priority for modeling
3. Integrated modeling management system:
   - Integrate the OPM conceptual model formally with 1D-CAE model in software

Design target and case study

- “Delight Design” for automotive
  - Integrate into modeling of non traditional domains

- Develop and use the proposed system to synthesize alternative hierarchical automotive designs from functional description and assess them holistically (including from a “Delight Design” viewpoint)

Proposed outcome

- Developed methodology (and associated software implementation) to:
  - Functionally decompose processes associated with a system, including more complex behaviors
  - Enable the building of 1D-CAE models from the functional descriptions
  - Integrate the conceptual and numerical models with software
- Case study presented and applied to an automotive system considering “Delight Design”

Research plan / schedule

- 1st year:
  - Review:
    - Existing functional description mapping to system architecture design methodologies
    - Existing decomposition methodologies
  - Integrate existing OPM conceptual model formally with 1D-CAE model in software
  - Build automotive models incorporating “Delight Design” library and the current design method
    - Write paper based on the updated method and use in automotive “Delight Design”
  - Attempt to address flaws with current design method to enable 1D-CAE method of non continuous non modular conceptual designs

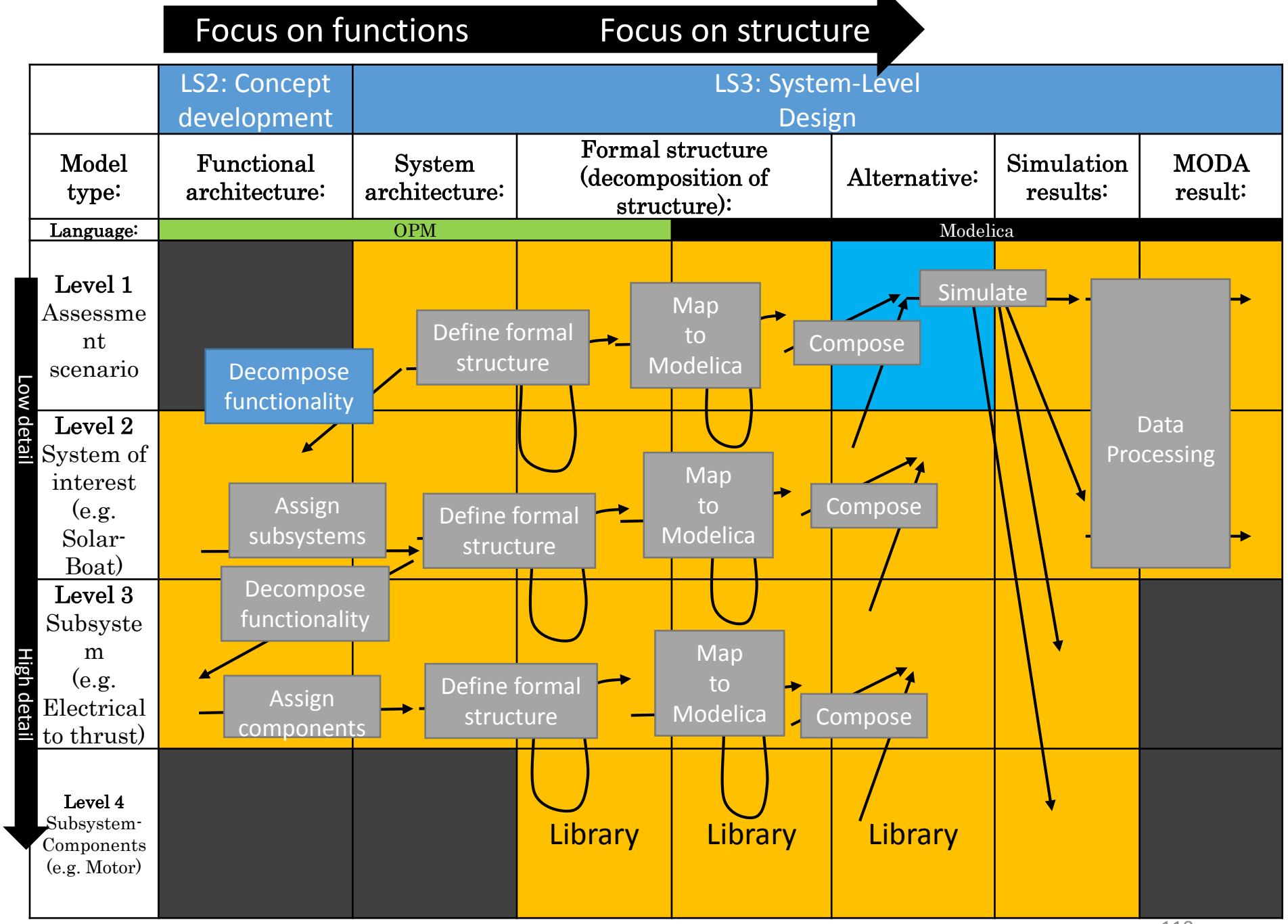
Research plan / schedule

- 2nd year:
  - Develop more formal decomposing process while simultaneously developing functional requirements for an automotive system by decomposition
  - Find flaws in the current implemented system by attempting to develop 1D-CAE models of automotive system and assess alternatives:
    - Write paper on the development and assessment of multiple automotive designs from a “Delight Design” viewpoint
- 3rd year:
  - Iterate based on found flaws
  - Attempt to use system on alternative design target:
    - Write paper

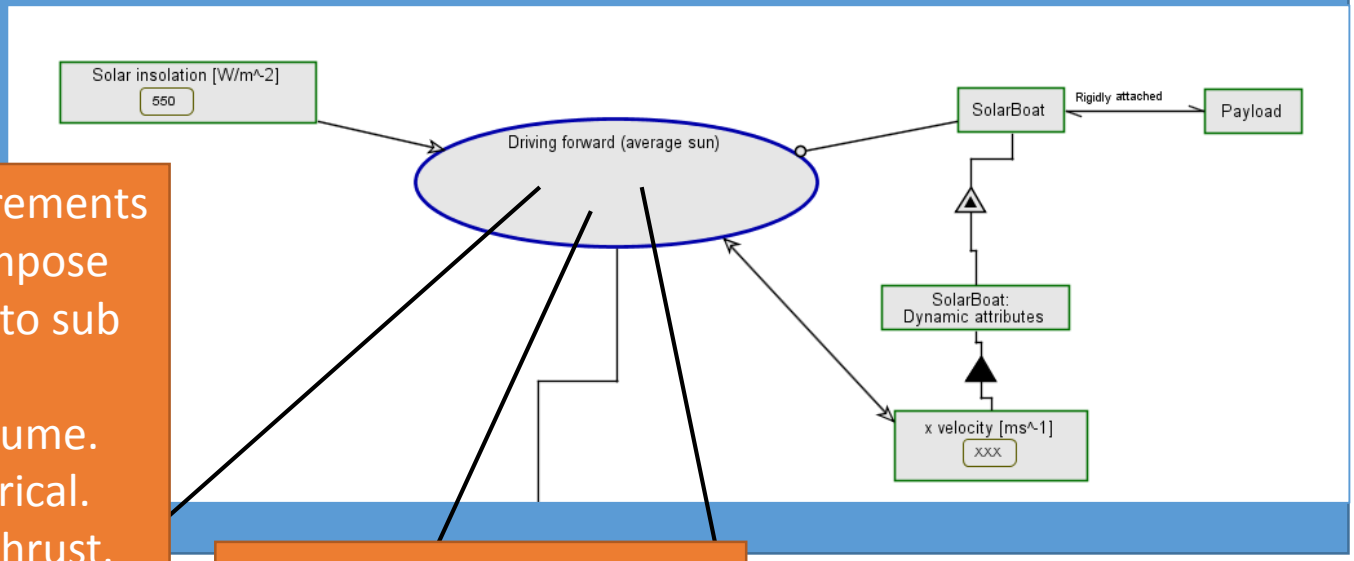
Any questions?

1. Problems and proposed solutions identified from Solar-Boat
2. Proposed tools and methodologies for Knowledge Management and System-Level Design
3. Examples and demonstrations
4. Discussion
5. Conclusions
6. PhD plans

Appendix – Mapping on the chart


Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Level 1

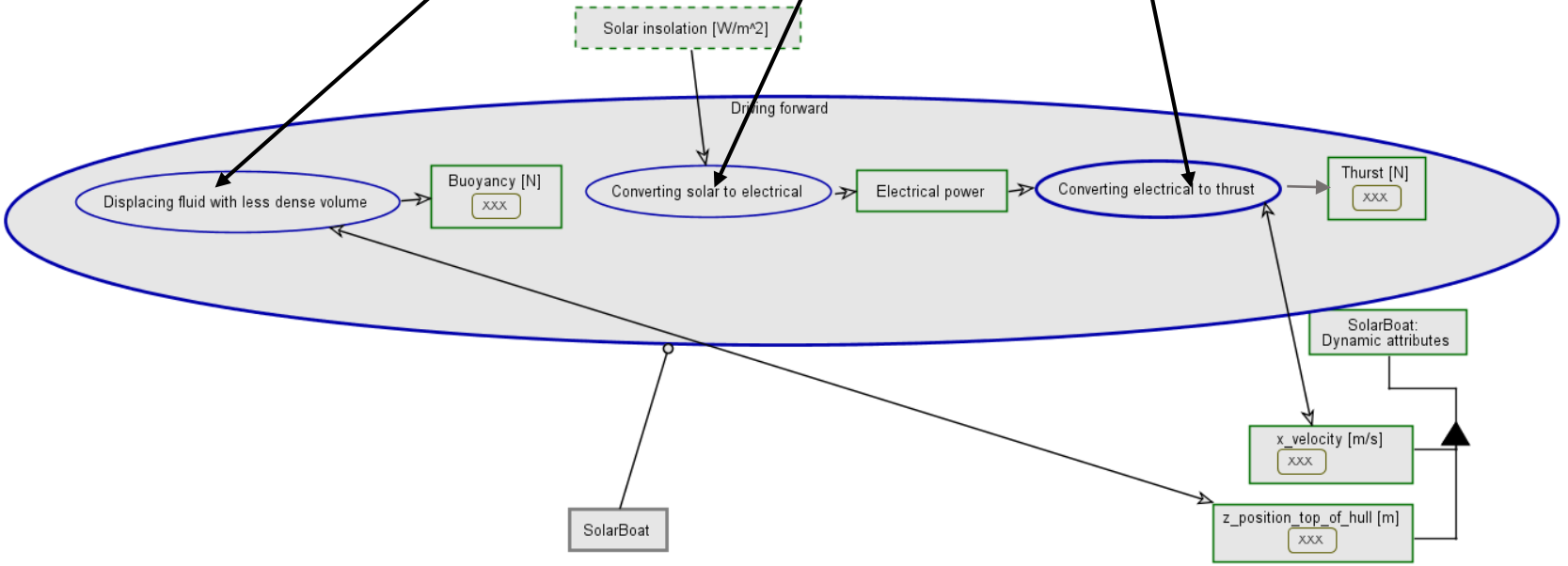


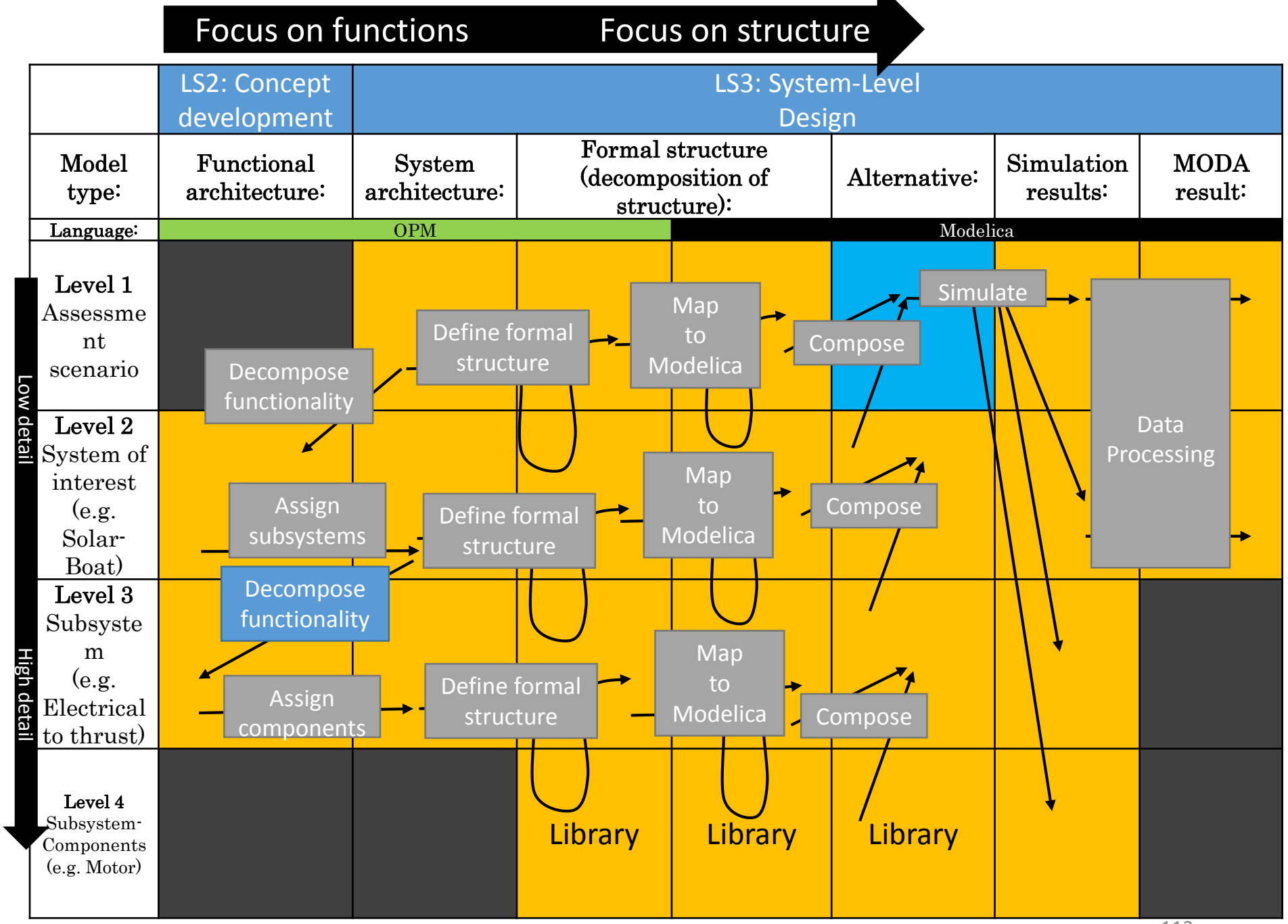
To define functional requirements of Subsystems must decompose “Driving forward” process to sub processes:

* Displacing less dense volume.
* Converting solar to electrical.
* Converting electrical to thrust.

Decompose functionality

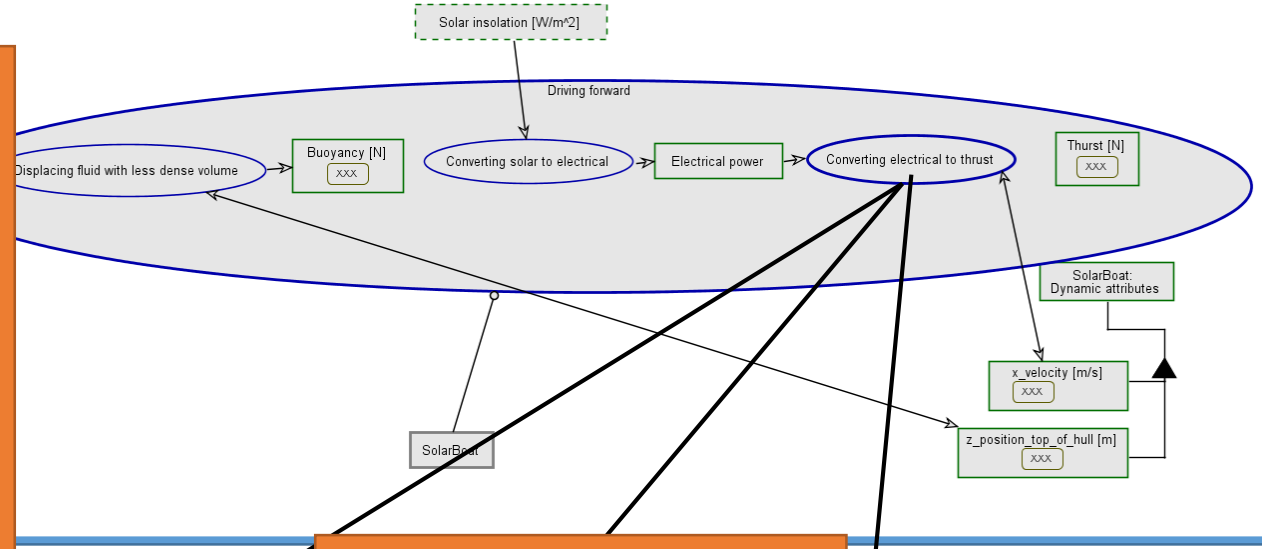
Level 2



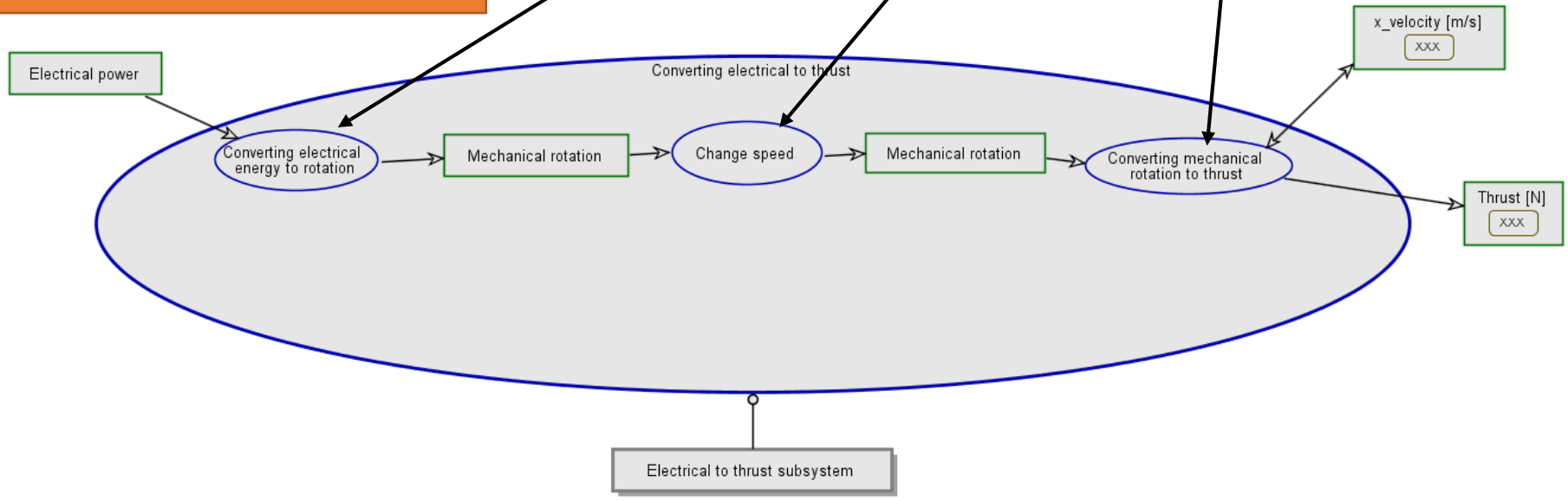

Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

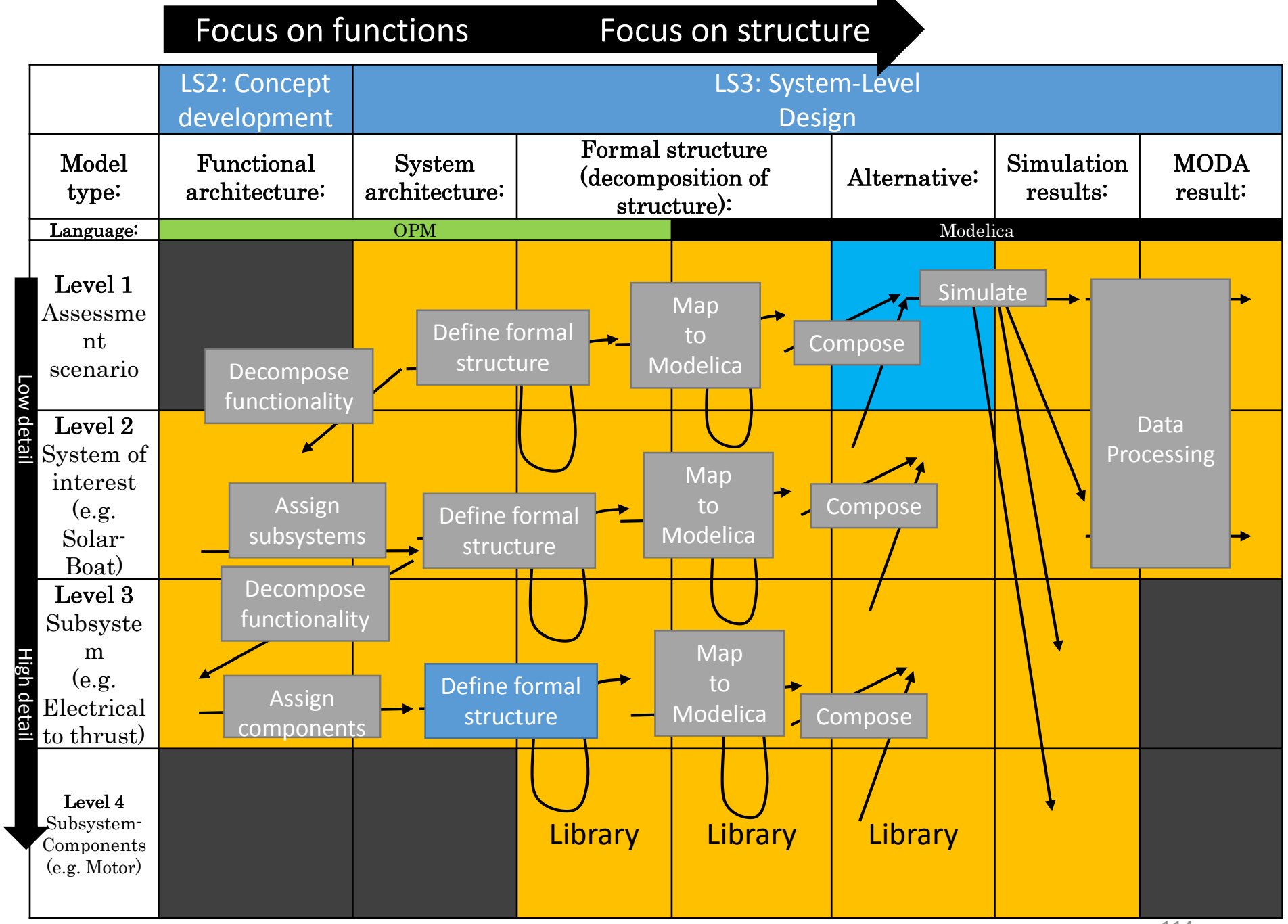
Level 2

To define functional requirements of components must decompose “Converting electrical to thrust” process to sub processes:
* Converting electrical energy to rotation.
* Changing speed.
* Converting mechanical rotation to thrust.



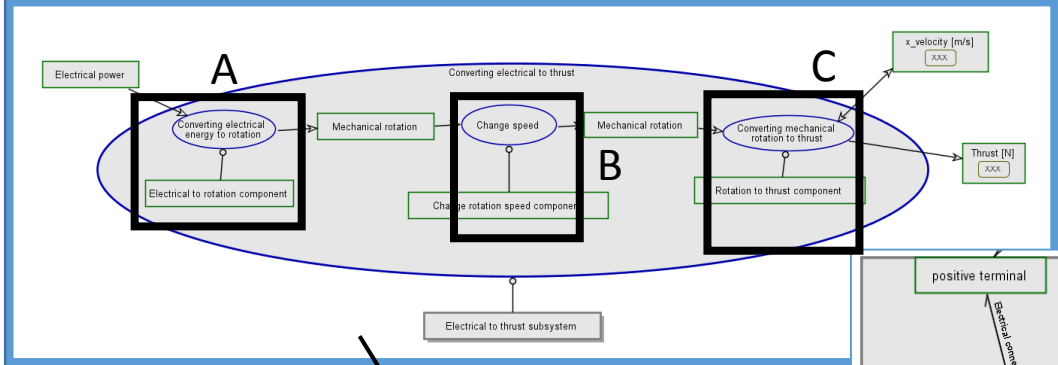
Decompose functionality

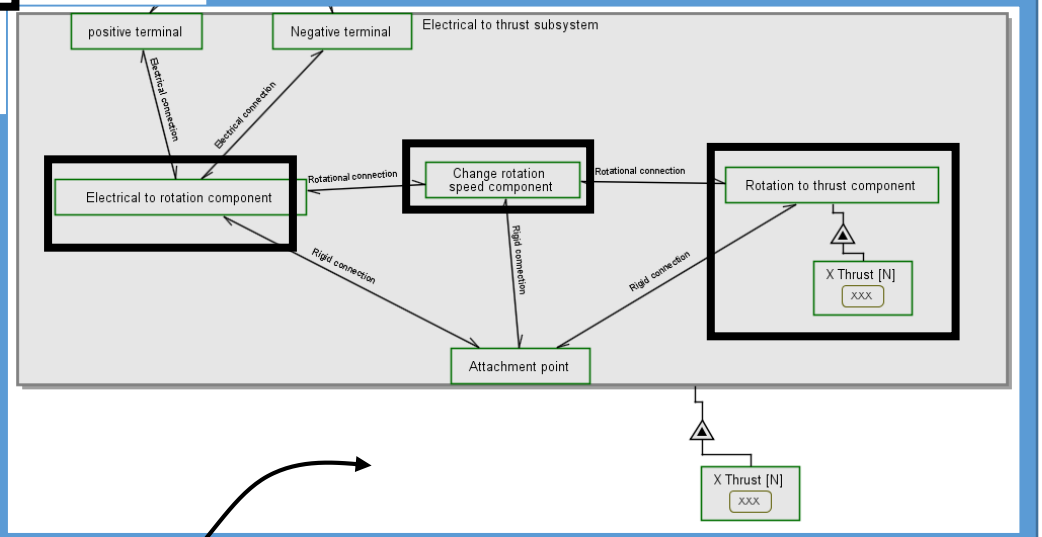



Focus on functions
Focus on structure
LS2: Concept development
LS3: System-Level Design
Model type:
Functional architecture:
System architecture:
Formal structure (decomposition of structure):
Alternative:
Simulation results:
MODA result:
Language:
OPM
Modelica
Level 1 Assessment scenario
Level 2 System of interest (e.g. Solar-Boat)
Level 3 Subsystem (e.g. Electrical to thrust)
Level 4 Subsystem-Components (e.g. Motor)
Low detail
High detail
Decompose functionality
Define formal structure
Map to Modelica
Compose
Simulate
Data Processing
Assign subsystems
Define formal structure
Map to Modelica
Compose
Decompose functionality
Assign components
Define formal structure
Map to Modelica
Compose
Library
Library
Library

Define formal structure

Level 3





To define formal structure involves:

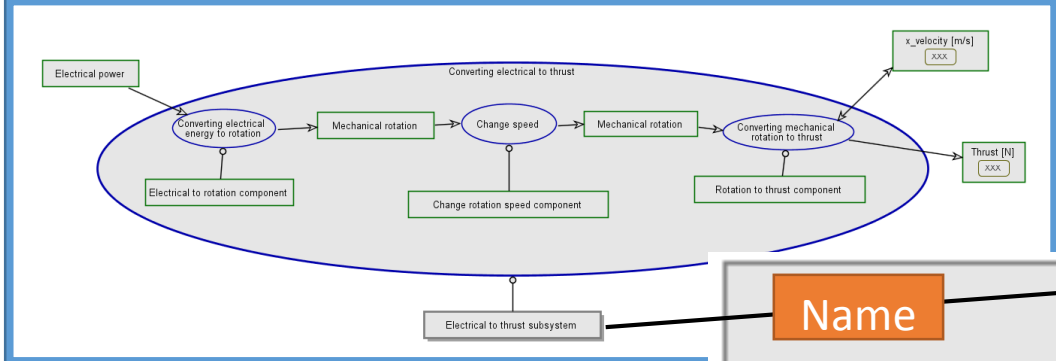
1) Build “common” formal structure and name it
2) Review library each System architecture object/process pair (A, B & C) to understand the interfaces
3) Populate

Level 4 (Subsystem-Components library)

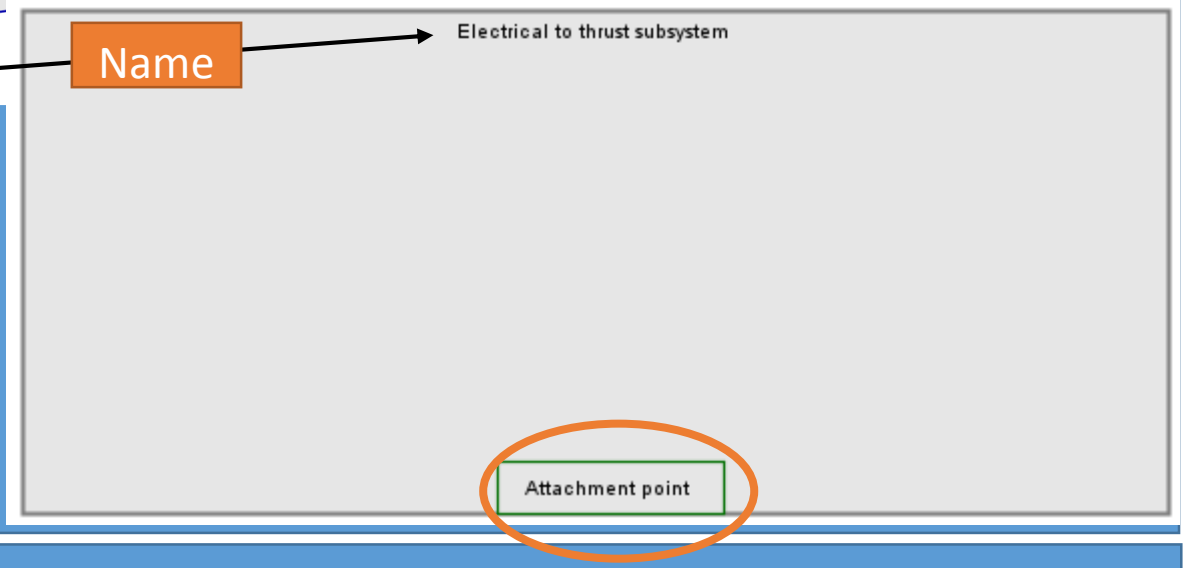
Check what connections the lower level components have

Define formal structure

Level 3



Name → Electrical to thrust subsystem



To define formal structure involves:

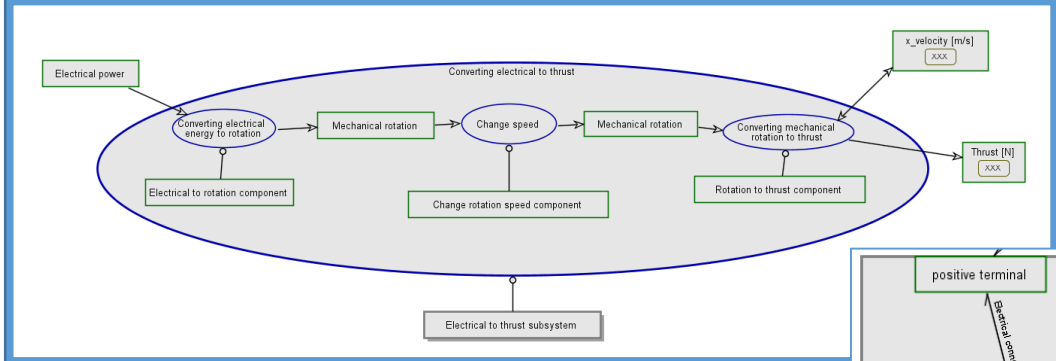
1) **<u>Copy "common" formal structure and name it</u>**
2) Review library each System architecture object/process pair (A, B & C) to understand the interfaces
3) Populate

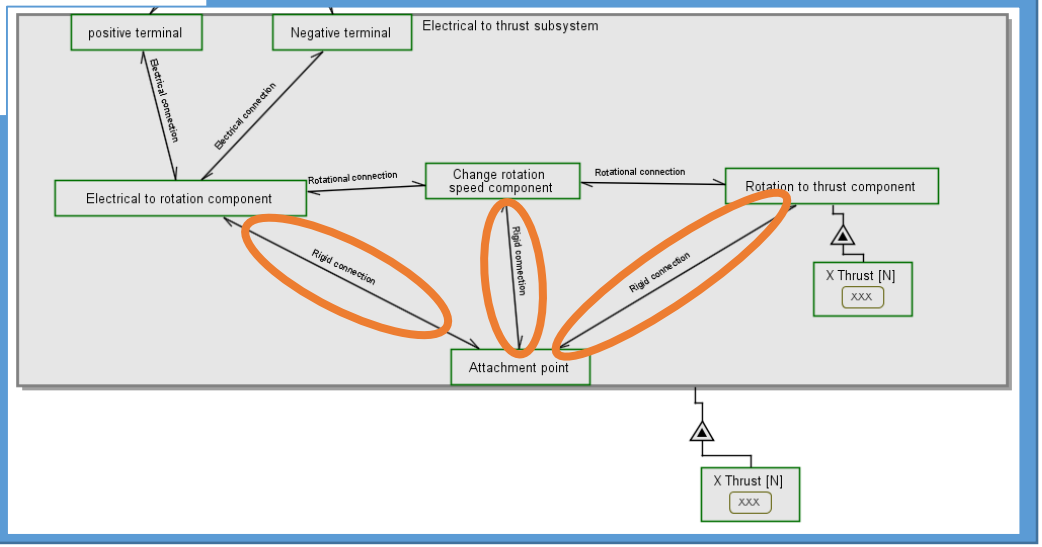
Level 4 (Subsystem-Components library)

Assume SolarBoat is rigid so all components must be attached to something. All components attach to the attachment point.

Define formal structure

Level 3





To define formal structure involves:

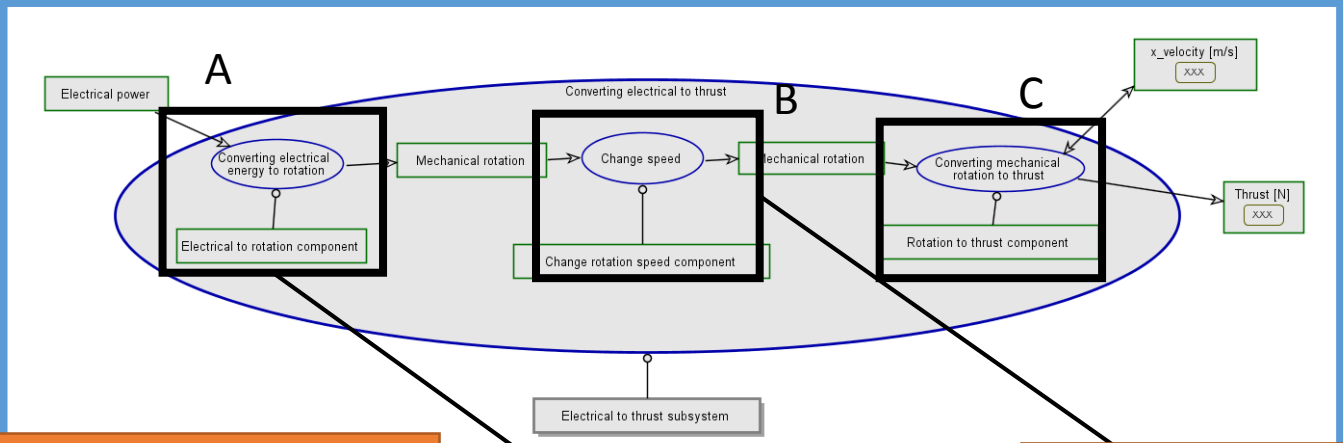
1) **<u>Copy "common" formal structure and name it</u>**
2) Review library each System architecture object/process pair (A, B & C) to understand the interfaces
3) Populate

Level 4 (Subsystem-Components library)

Assume SolarBoat is rigid so all components must be attached to something. All components attach to the attachment point.

Define formal structure

Level 3

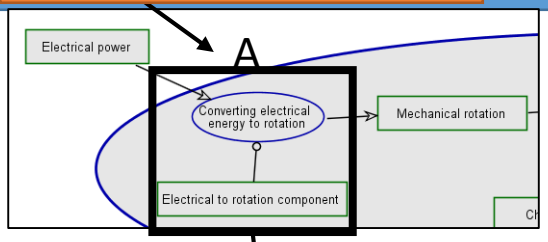


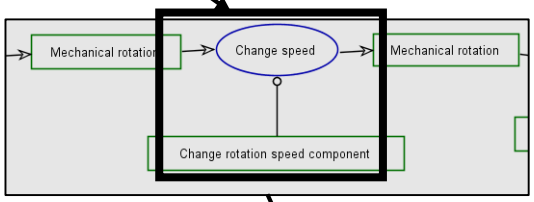
To define formal structure involves:

1) Build "common" formal structure and name it
2) **Review library each System architecture object/process pair (A, B & C) to understand the interfaces**
3) Populate

Process object pair

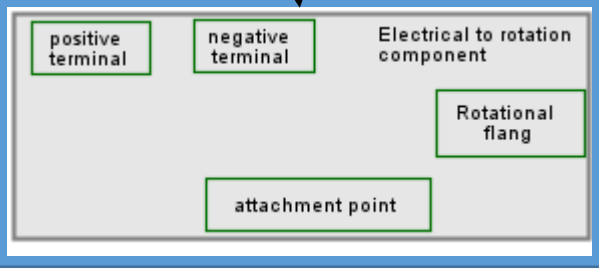
Process object pair

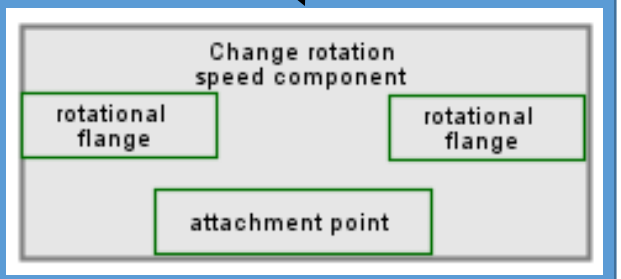




Find in library same behavior and interface description

Find in library same behavior and interface description

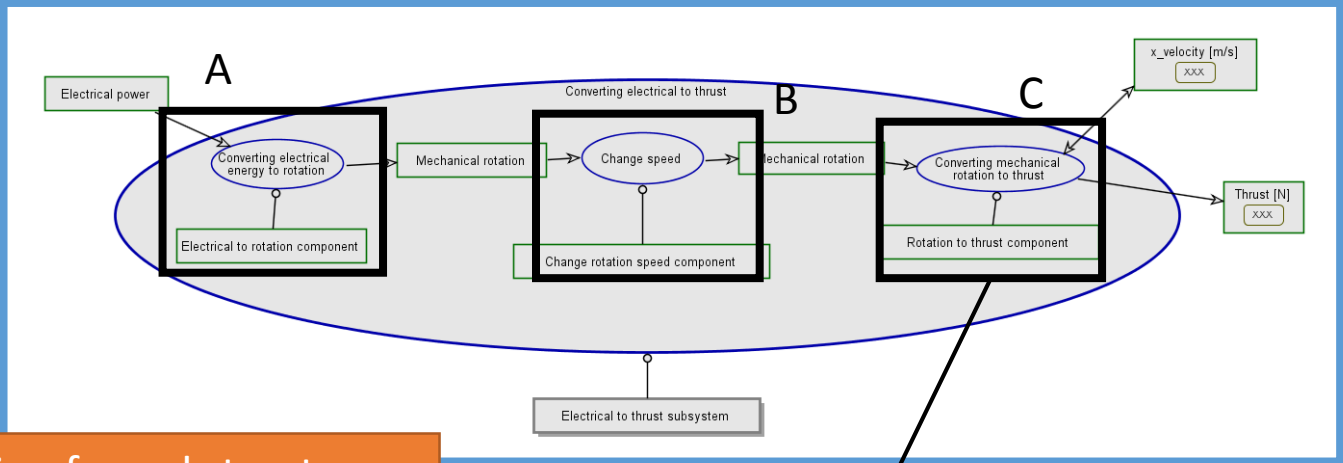




Level 4 (Subsystem-Components library)

Define formal structure

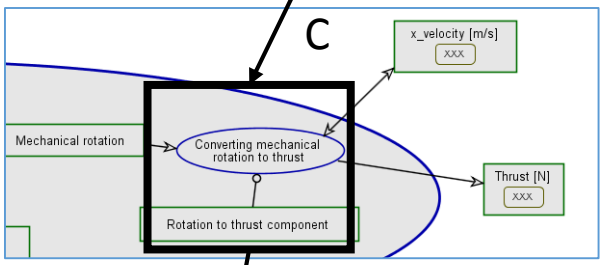
Level 3



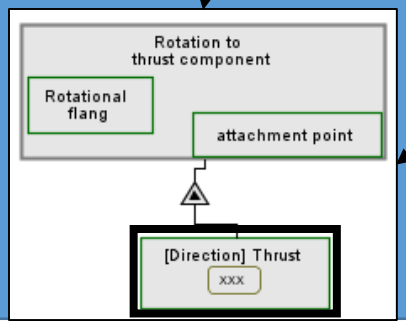
To define formal structure involves:

1) Build "common" formal structure and name it
2) **Review library each System architecture object/process pair (A, B & C) to understand the interfaces**
3) Populate

Process object pair



Find in library same behavior and interface description

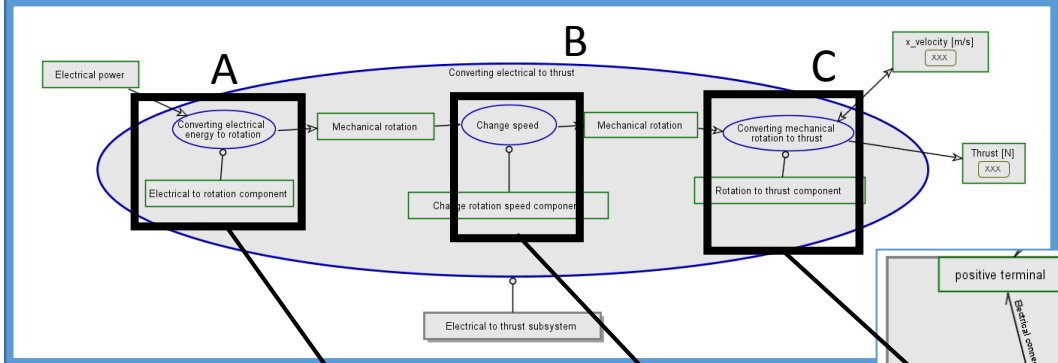


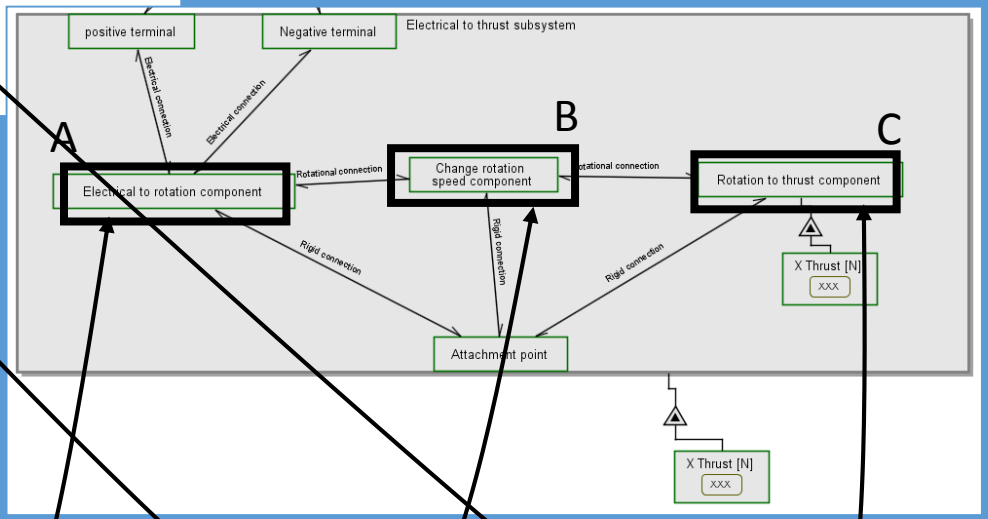
Special case Forces are transmitted through the structure. Mark direction based on variable being effected. i.e. x here

Level 4 (Subsystem-Components library)

Define formal structure

Level 3

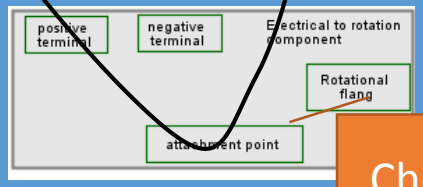


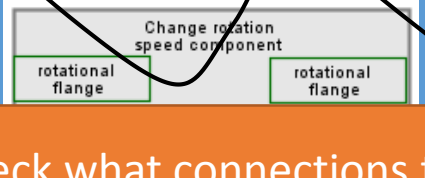


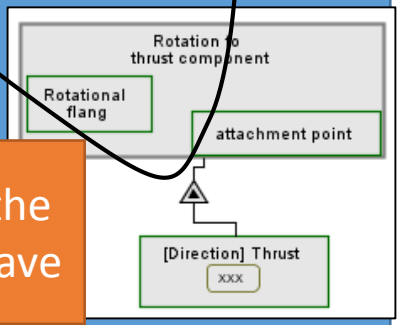
To define formal structure involves:

1) Build “common” formal structure and name it
2) Review library each System architecture object/process pair (A, B & C) to understand the interfaces
3) **Populate**

Level 4 (Subsystem-Components library)



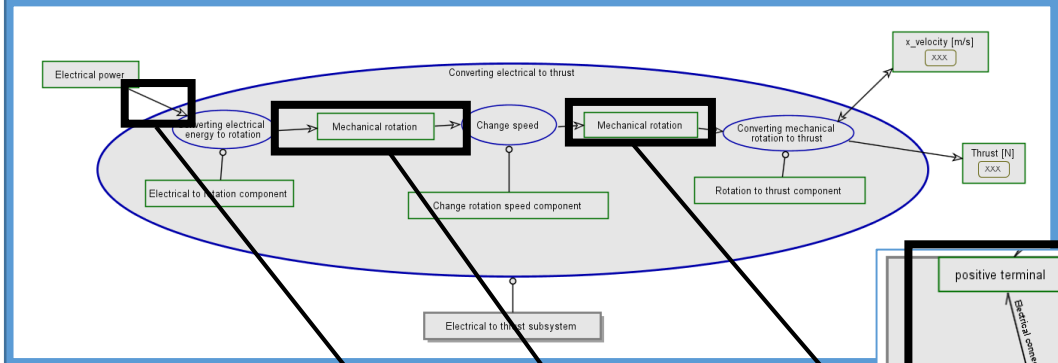


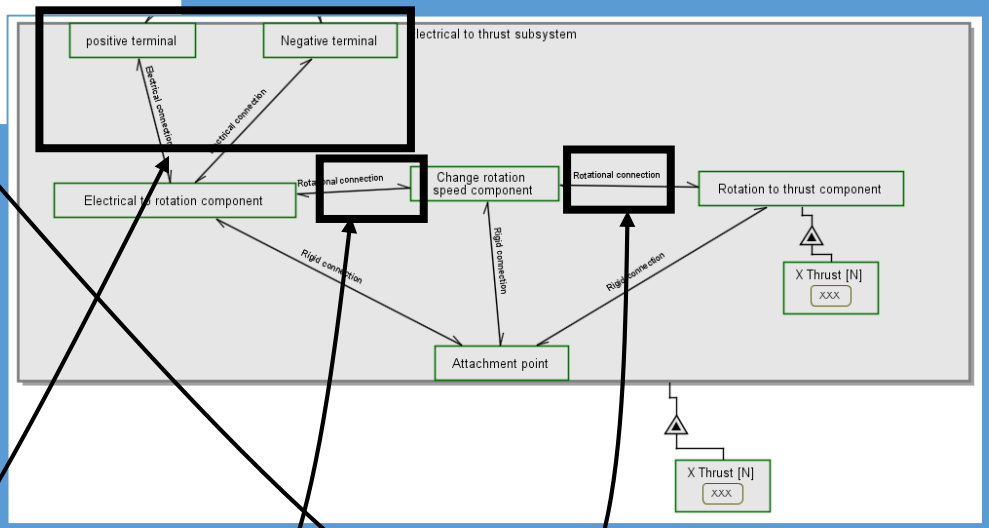


Check what connections the lower level components have

Define formal structure

Level 3

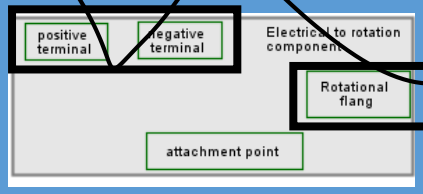


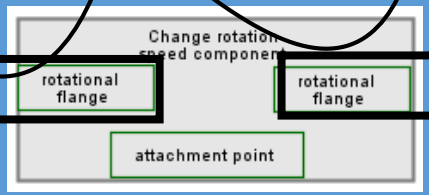


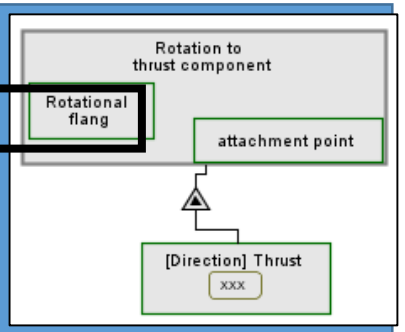
To define formal structure involves:

1) Build “common” formal structure and name it
2) Review library each System architecture object/process pair (A, B & C) to understand the interfaces
3) **Populate**

Level 4 (Subsystem-Components library)

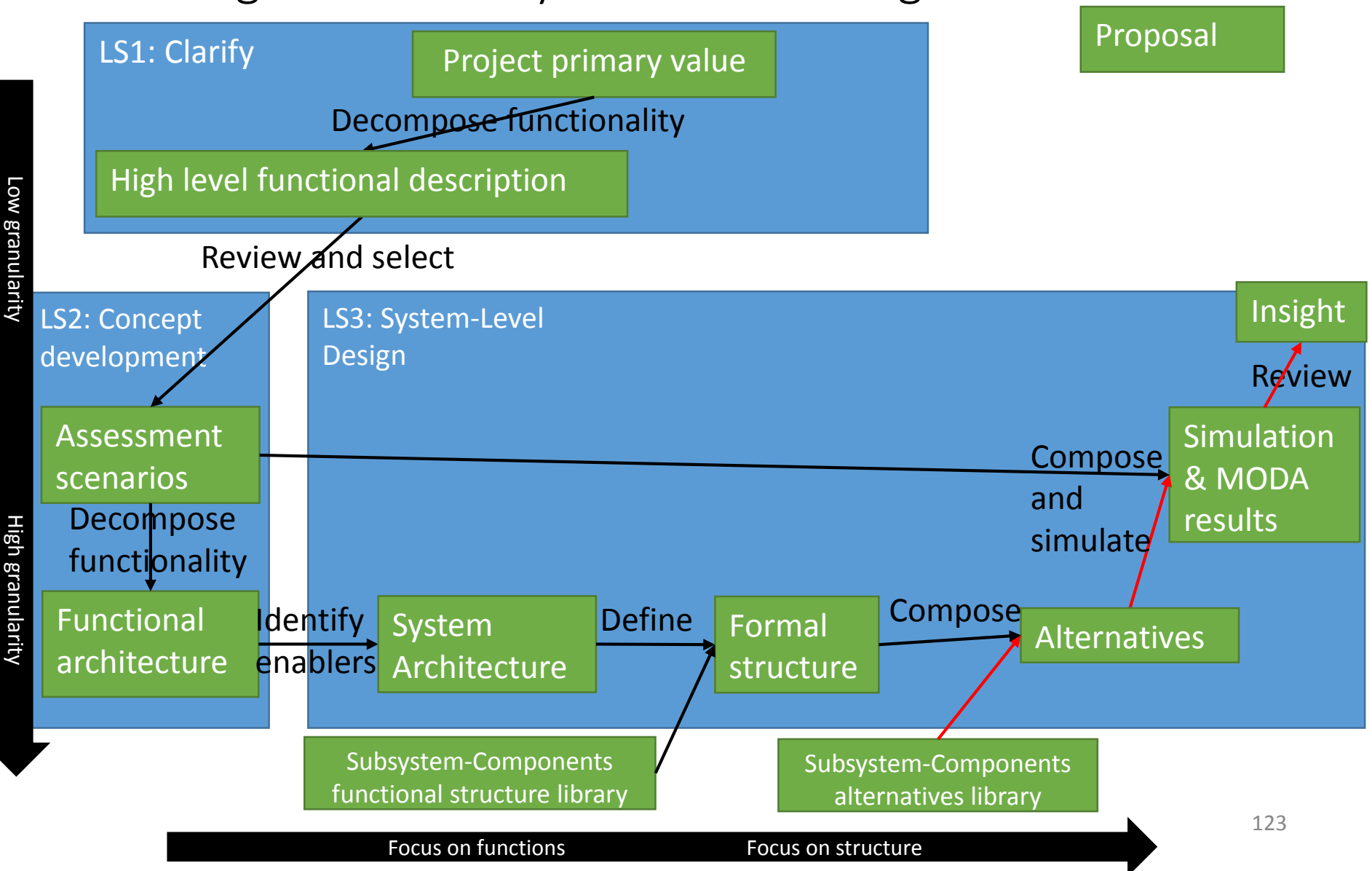






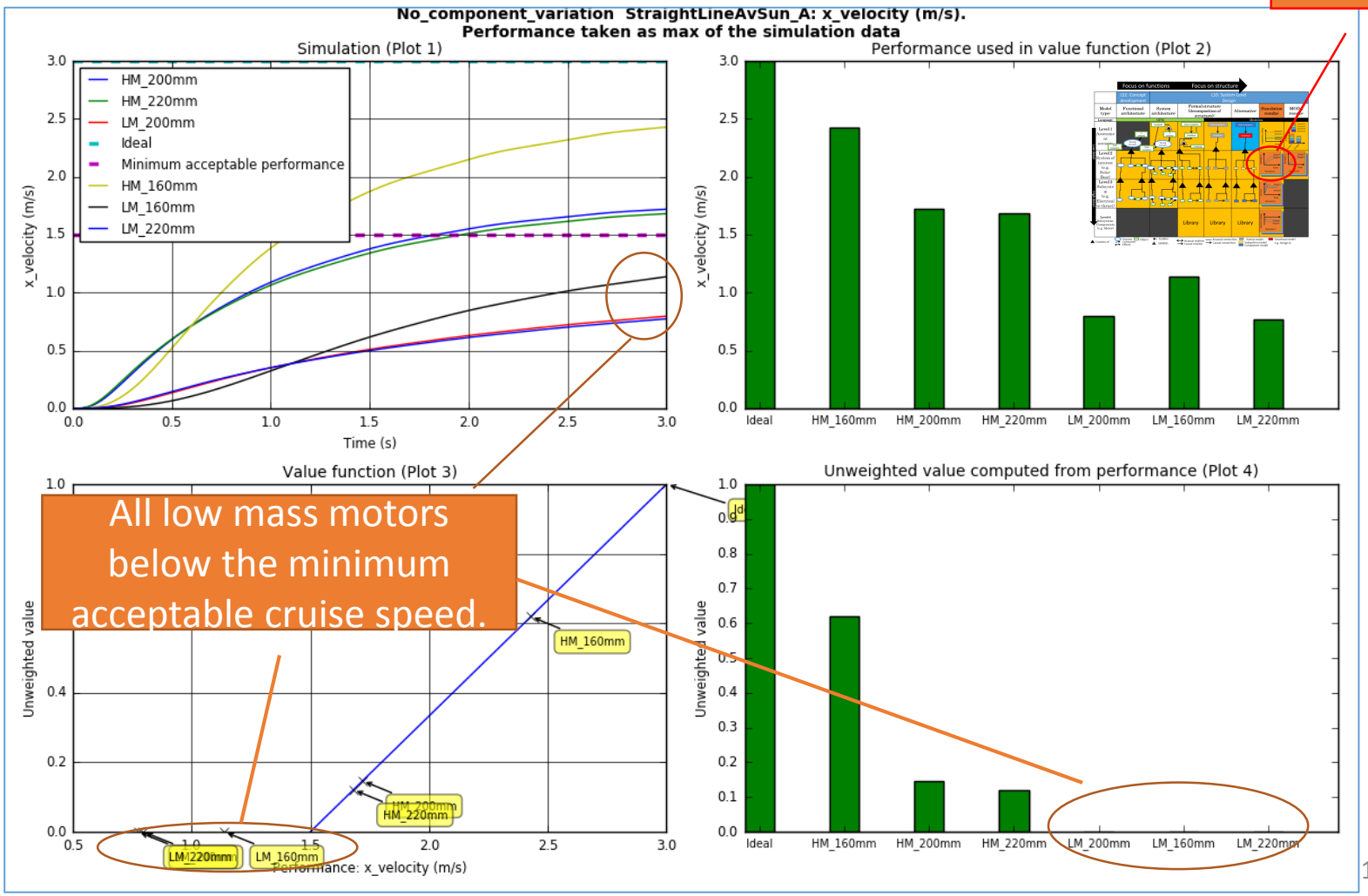
Appendix – Demo

Proposed tools and methodologies for Knowledge Management and System-Level Design

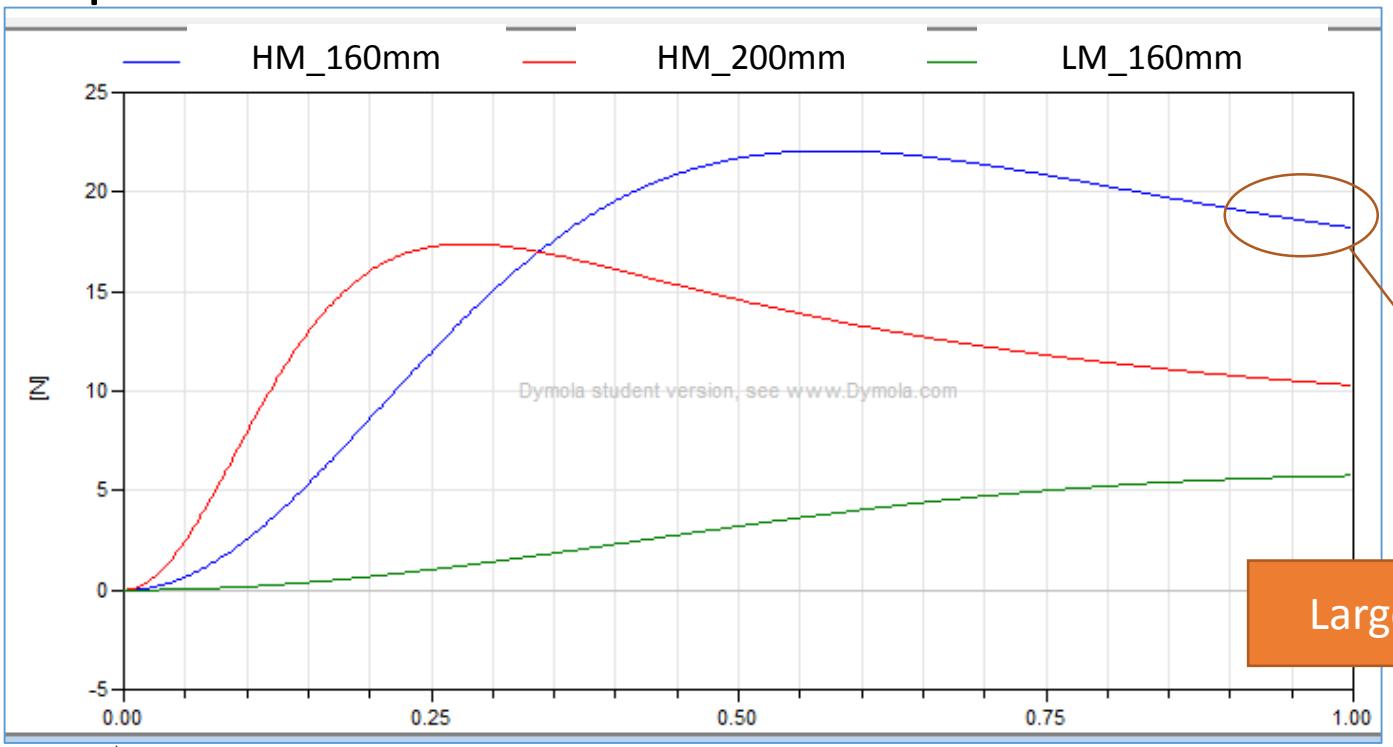


Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers)

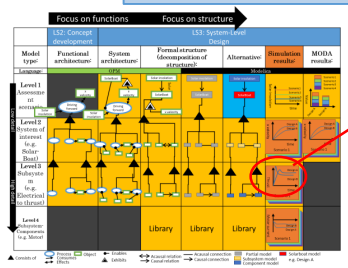
Review



Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers) Compare thrust

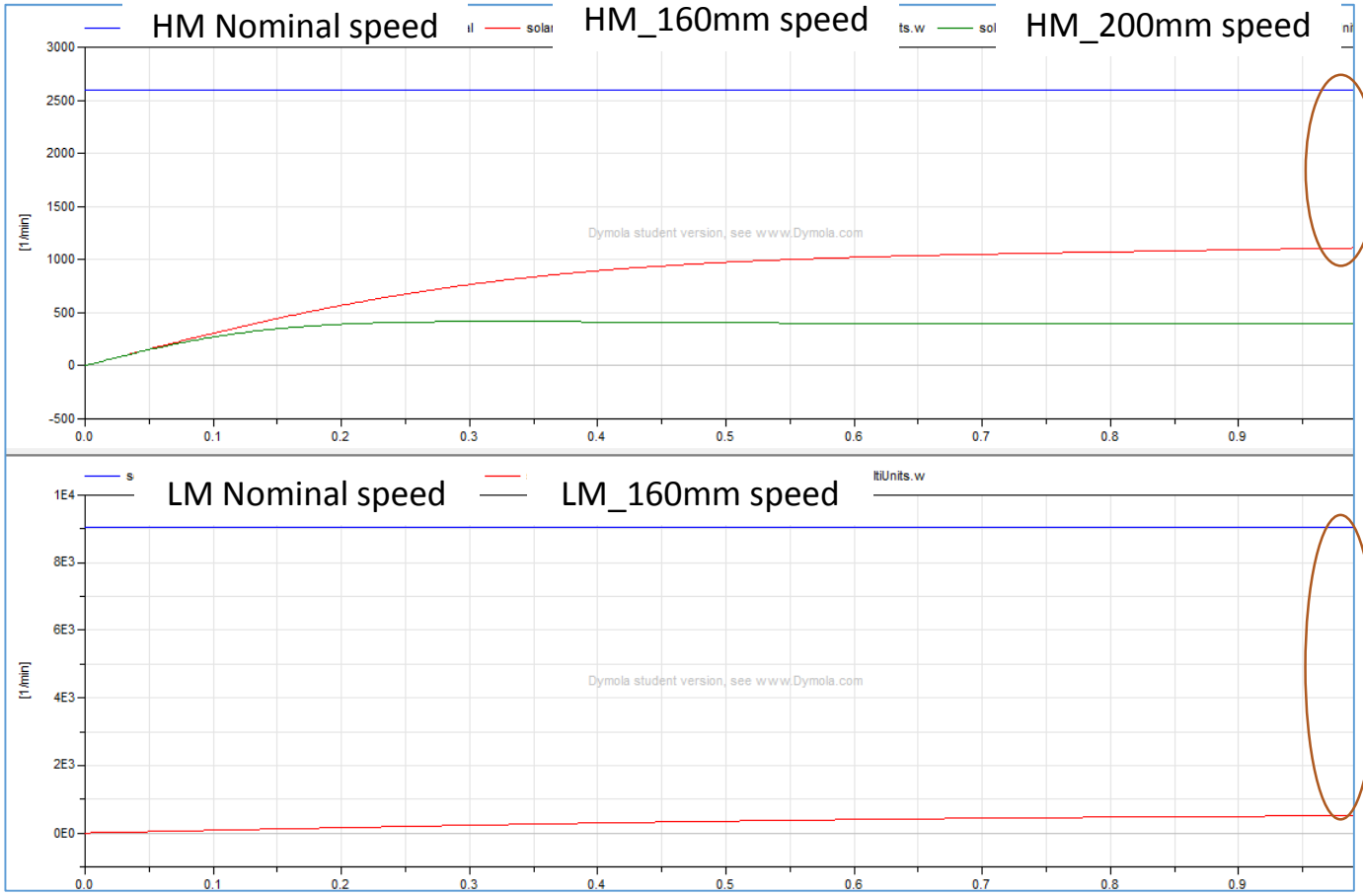


Larger thrust



Review

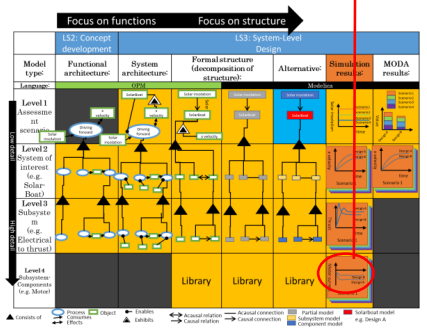
Design Set 1 (Subsystem-Component variation: compare a heavy motor to low mass one with different propellers) Compare motor speed



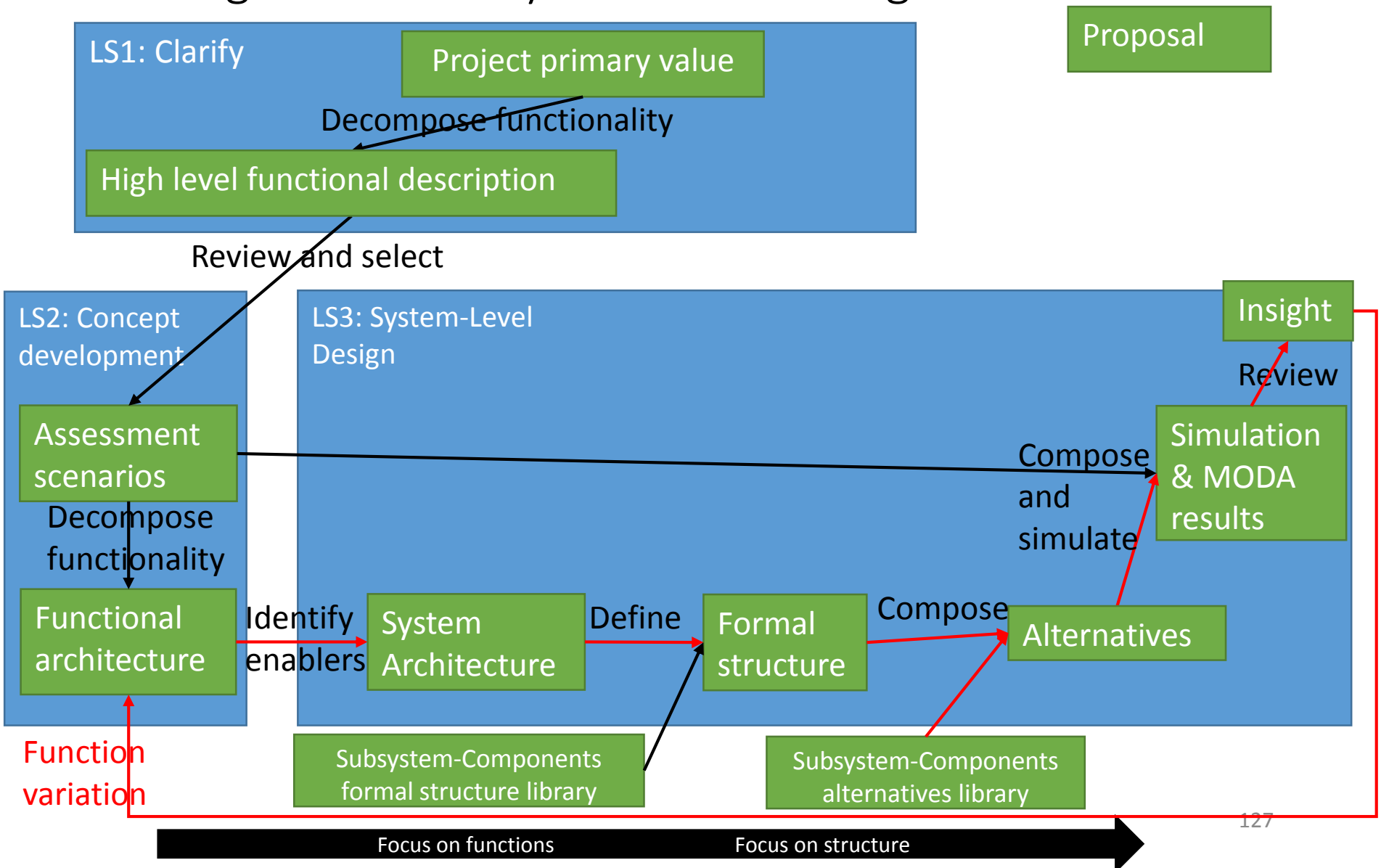
Closer to nominal speed but still quite far from nominal (2.36 times)

18 times from nominal

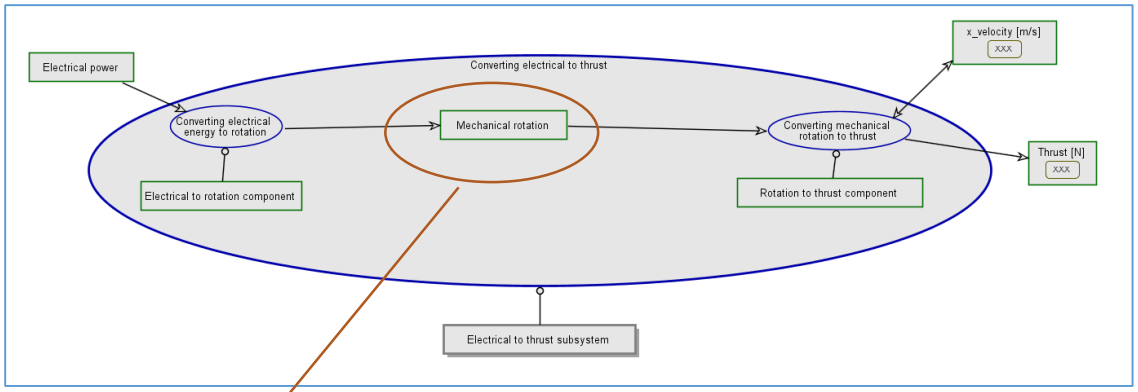
Review



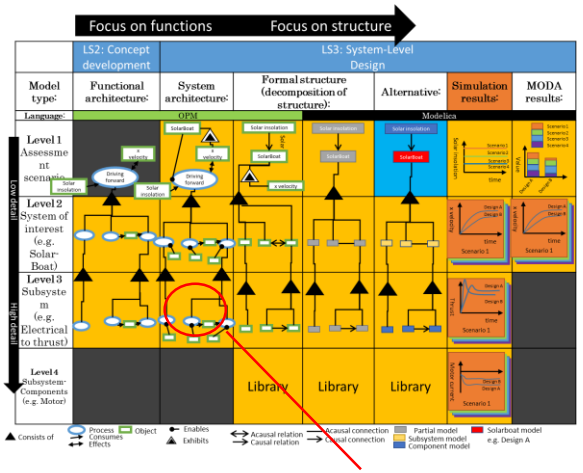
Proposed tools and methodologies for Knowledge Management and System-Level Design



Design Set 2 (Subsystem Functional Architecture variation: add a motor speed changing device)
How to get motor to spin closer to nominal speed

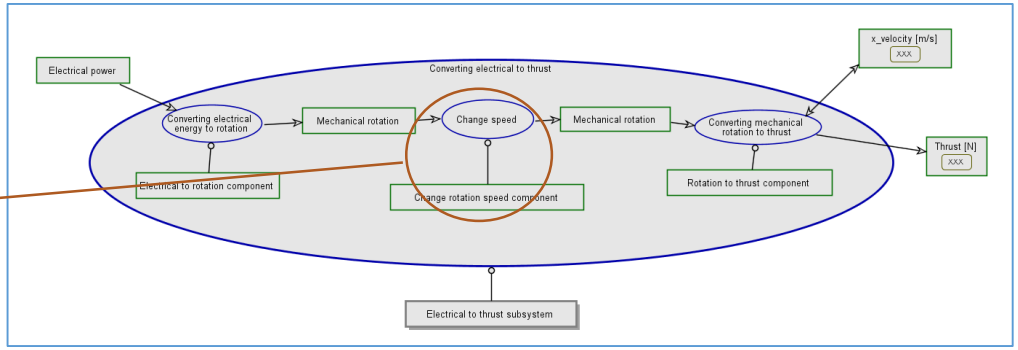


Change here



Review and update

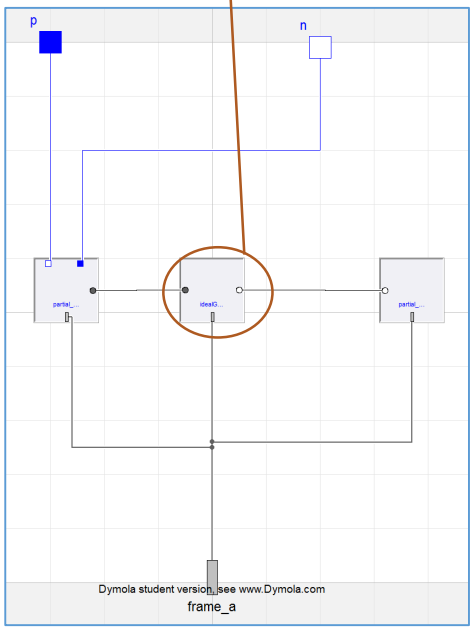
Idea… Add rotation speed changing component

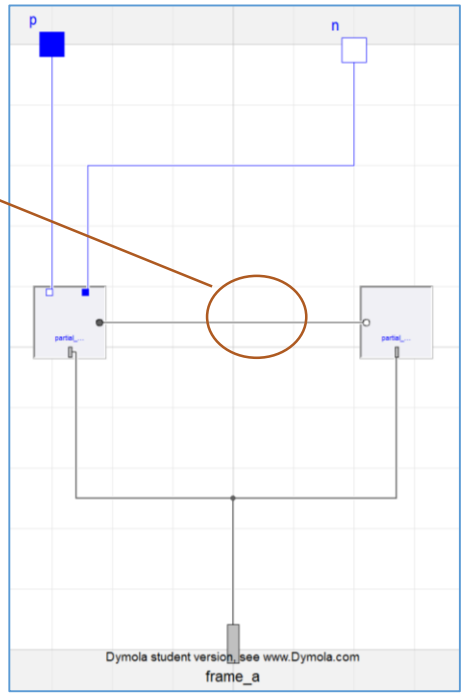


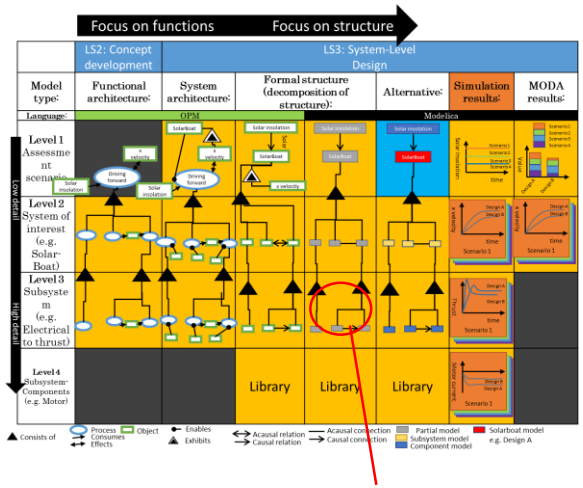
Design Set 2 (Subsystem Functional Architecture variation: add a motor speed changing device)
How to get motor to spin closer to nominal speed

Change here

Idea… Add rotation changing component





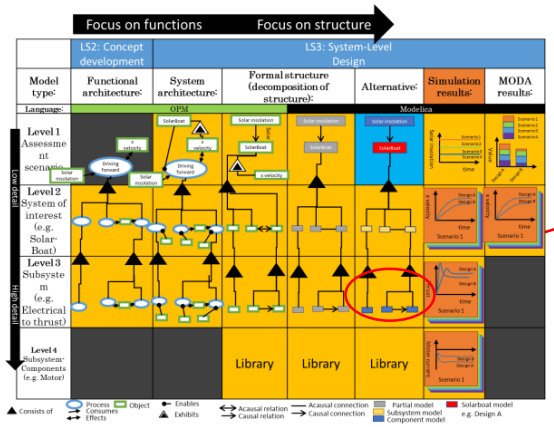


Change

OPM Formal Structure not shown to save time

Design Set 2 (Subsystem Functional Architecture variation: add a motor speed changing device)
How to get motor to spin closer to nominal speed

Alternative name	Buoyancy Genearation	SolarToElectical	ElectricalToThrust
HM_3_160mm	Single hull	Old solar panel	H motor: 3->1 Gearbox : 160mm prop
HM_3_200mm	Single hull	Old solar panel	H motor: 3->1 Gearbox : 200mm prop
HM_3_220mm	Single hull	Old solar panel	H motor: 3->1 Gearbox : 220mm prop
LM_13_160mm	Single hull	Old solar panel	L motor: 13->1: 160mm prop
LM_13_200mm	Single hull	Old solar panel	L motor: 13->1: 200mm prop
LM_13_220mm	Single hull	Old solar panel	L motor: 13->1: 220mm prop



Create new alternatives from (gearboxes from Subsystem Components Alternatives)