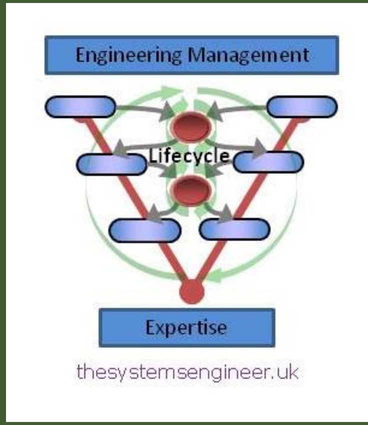
34th Annual INCOSE international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

Paul Davies
paul@thesystemsengineer.uk
p.davies@lboro.ac.uk



An MBSE group project challenge as a learning experience for Masters degree students

Presentation Flow

1. Introduction and Background
2. Module Outline
3. Conduct of the Project
4. Results
5. Observations and Conclusions

Background and Introduction

The Module

- Part of Level 7 (Masters level) Apprenticeship
- Students mostly in full-time employment, on block release to improve SE pipeline
- 15 credit module to participate in a group project
- Prior modules:
  - Systems thinking
  - System architecture |
  - System design | MBSE-based
  - Verification & Validation |
  - Holistic engineering
  - Managing capability *+ Options*

The learning objectives

- Improved knowledge & understanding of SE methods and processes
- Demonstrable cognitive and practical skills to achieve an outcome
- Personal research on potential solutions and improvements
- Individual reflection on transferable skills to take back to the workplace

All assessed – 50% group work, 50% individual report

1. Introduction and Background
2. Module Outline
3. Conduct of the Project
4. Results
5. Observations and Conclusions

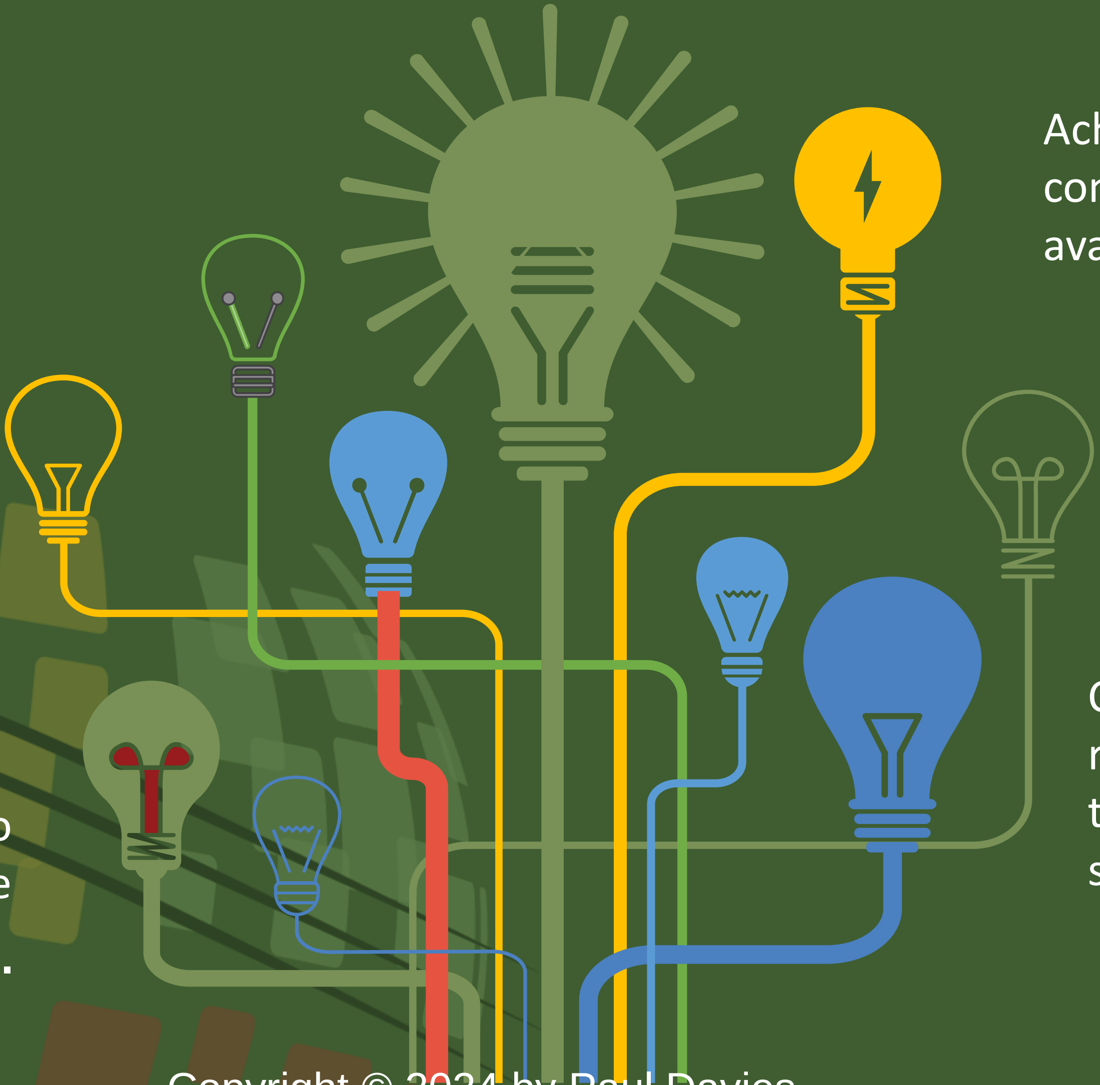
Project / Module / Challenge Constraints

Realizable in four days of dedicated collaborative working, by a group of 4-6 students with prior planning

Demonstrate a wide range of SE and engineering management methods.

Build on previous module content

Allow the students to choose their own case study system.



Achievable within the constraints of the toolset available to the University

Represent realistic challenges encountered in modern systems development

Consistent with University regulations, and amenable to an approved marking scheme

The Challenge

3 of the prior modules had strong MBSE content, but not strung together to make a project timeline. Why not join them up and put them in a realistic setting?

Plan the work, and incorporate tutor feedback

Design, review, implement, test, demonstrate

- Choose a Case Study System (CSS)
- Pick a variety of models to represent the CSS
- Imagine various change scenarios
- Create a Meta-Model System (MMS) to link all the models together
- Make the MMS animate through the change scenarios

Model types

"How do we know that each model represents the same functional and physical design? How do we verify that each model simulates the system realistically?"

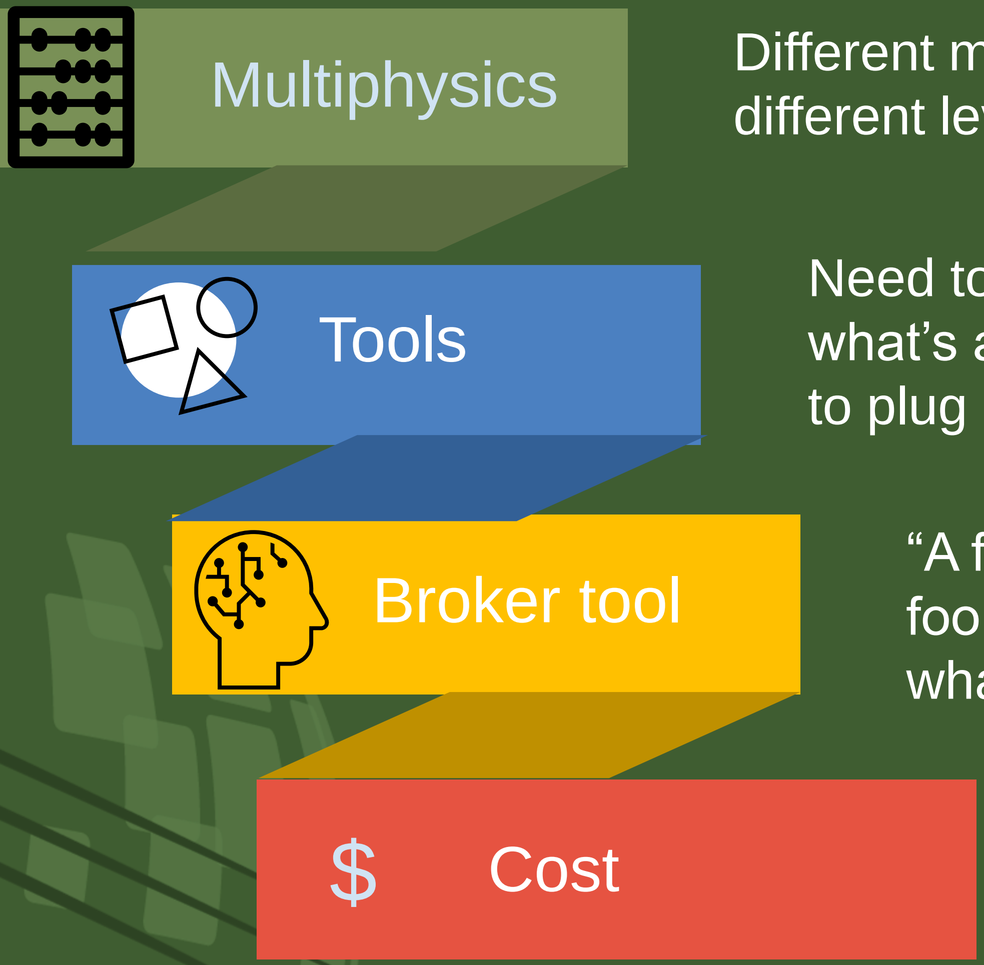
- 01 Logical structure (SysML)
- 02 Reliability, Availability and Maintainability (RAM)
- 03 Failure Modes and Effects Criticality Analysis (FMECA) and safety analysis
- 04 Geospatial location, both static and dynamic, and physical structure, Mechanical stresses
- 05 Security and cybersecurity analysis
- 06 Thermal and aerodynamic flows
- 07 Cost estimation
- 08 Parametric performance, e.g. sensitivity, bandwidth, throughput or flow rate, speed, response time, accuracy, probability of mission success…

=> An MMS to link them all together

Rationale

Why not use a Multiphysics model supported by a high-end tool?

Why not give them a ‘broker’ (integrated data management) tool?



Different models need different levels of granularity

Need to learn to cope with what’s available, and how to plug gaps

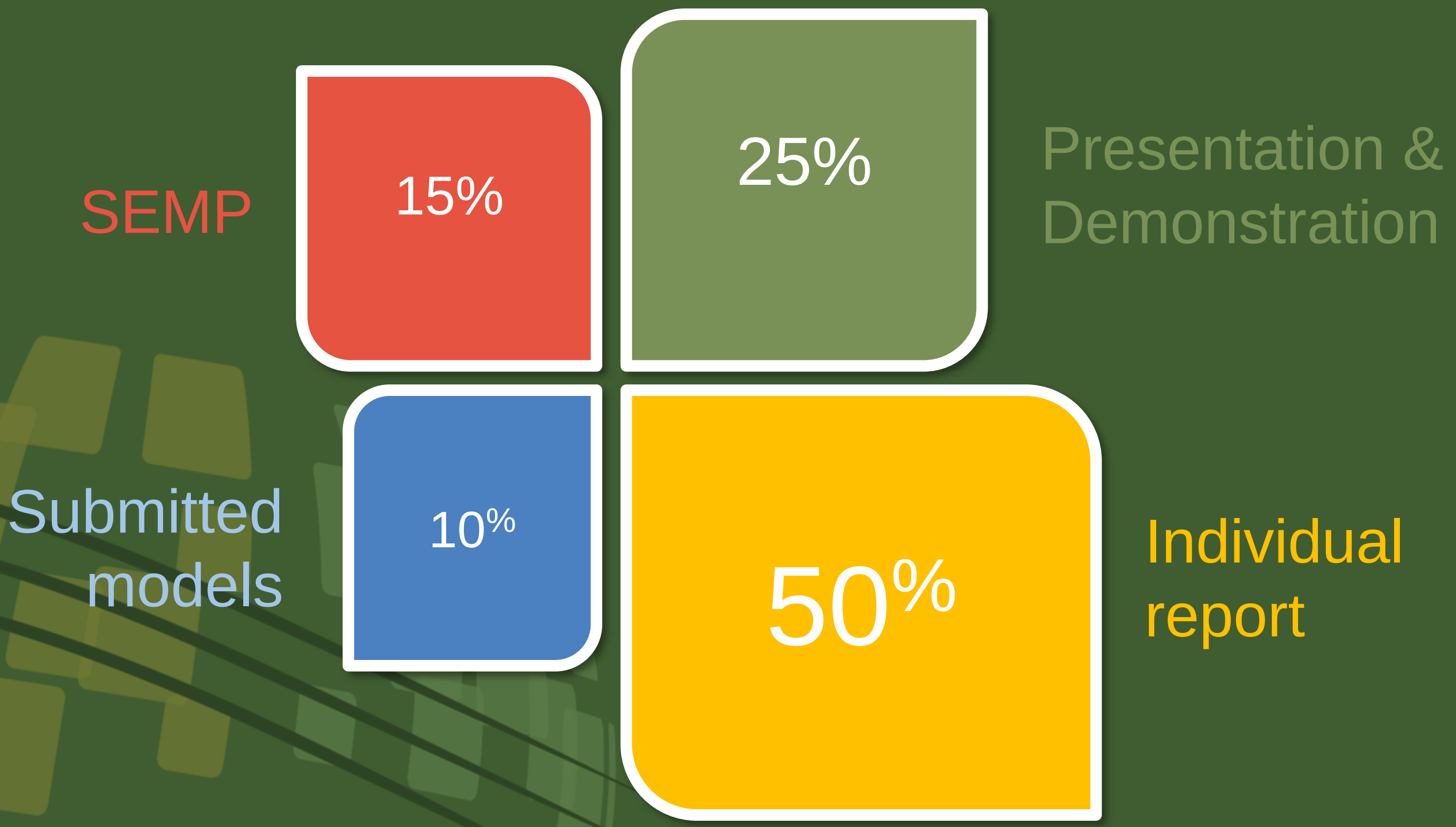
“A fool with a tool is still a fool” – need to understand what and how to use it

These tools are <u>very</u> expensive; need ROI proving by test

Requirements

1. The Design shall be illustrated with, and linked to, a basic model of a Case Study system. The central representation of both Case Study and MMS shall be in UML (minimum) or SysML (preferable).
2. The Design shall include representations of, or links to, as many of the models outlined in the Background section as practicable in the time allowed.
3. The Design model representations shall be linked documents or drawings as a minimum; referenced calculations (preferable); or hyperlinked models (ideal).
4. The Design shall show how version control of all embedded or linked models may be achieved. This shall include a whole-system version control mechanism, enforcing simultaneous consistency between constituent models.
5. The Design shall show how verification evidence for each model is supported and linked. Such evidence may be documentary argument (minimum), or shown explicitly via a UML/SysML model (for example).
6. .... (10 requirements)

Marking Scheme



1. Introduction and Background
2. Module Outline
3. **Conduct of the Project**
4. Results
5. Observations and Conclusions

Tools

MMS to link them all together, in:

- SysML?
- VB?
- Python?
- LabView?

01 Cameo Magic SOS Architect(SysML)

02 Excel or MagicDraw(RAM)

03 ? (FMECA etc)

04 Siemens NX or Solidworks (Geospatial, Structure, Mechanical)

05 ? (Security and cybersecurity analysis)

06 ? (Thermal and aerodynamic flows)

07 Excel or CoSysMO model in MatLab (Cost estimation)

08 MatLab / Simulink, Maple, CST Studio, Python (Parametric performance)

Lectures

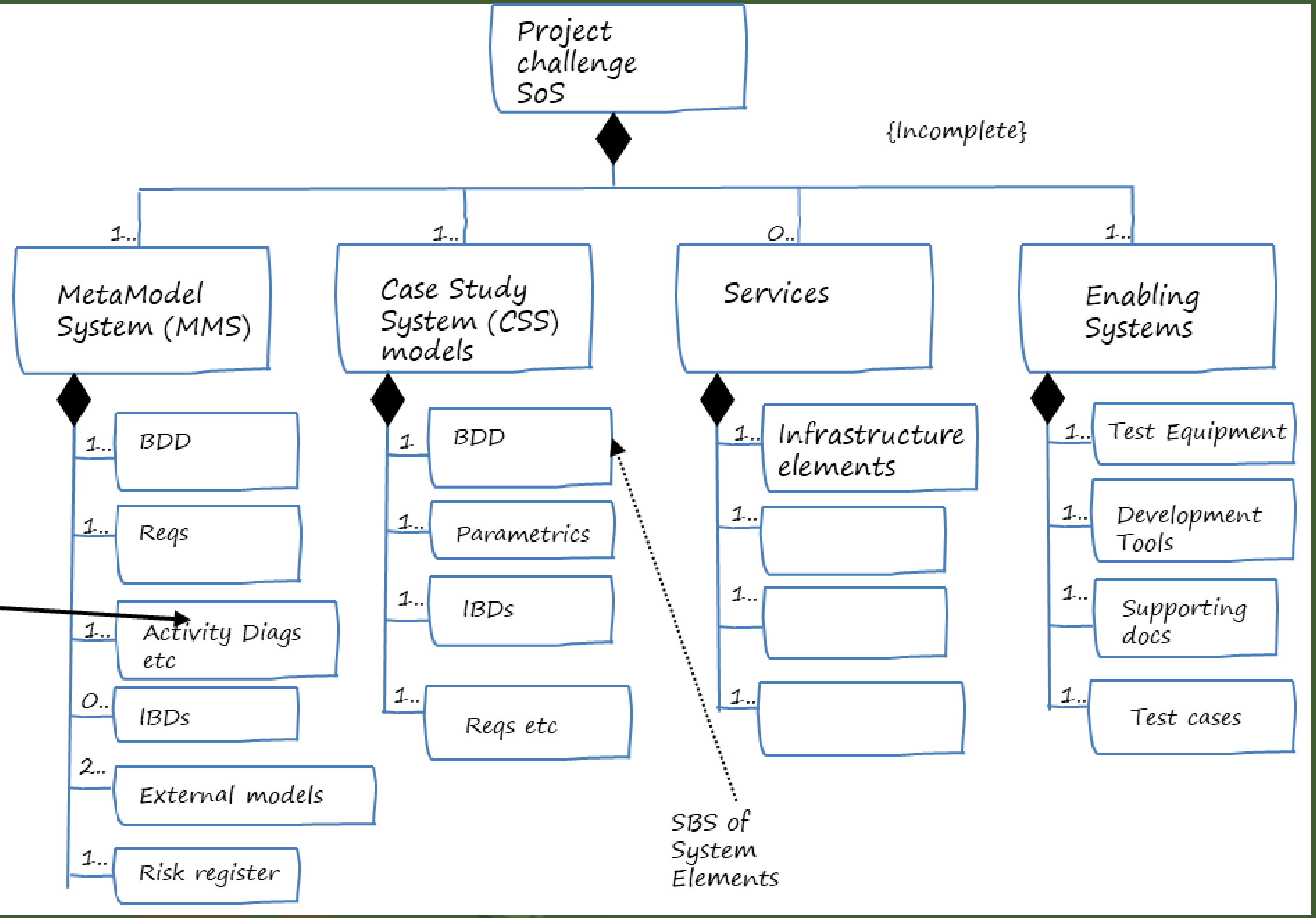
Preliminary

- Group Project Introduction
- Context and Project Challenge
- The Creating System
- Project Planning
- Coursework Deliverables
- Recommended Reading

Project Week

- Challenge Refresher + feedback on SEMP misunderstandings
- System Interfaces
- SysML – basics, usage, linking
- Tools overview
- Group demonstration requirements
- Individual Report

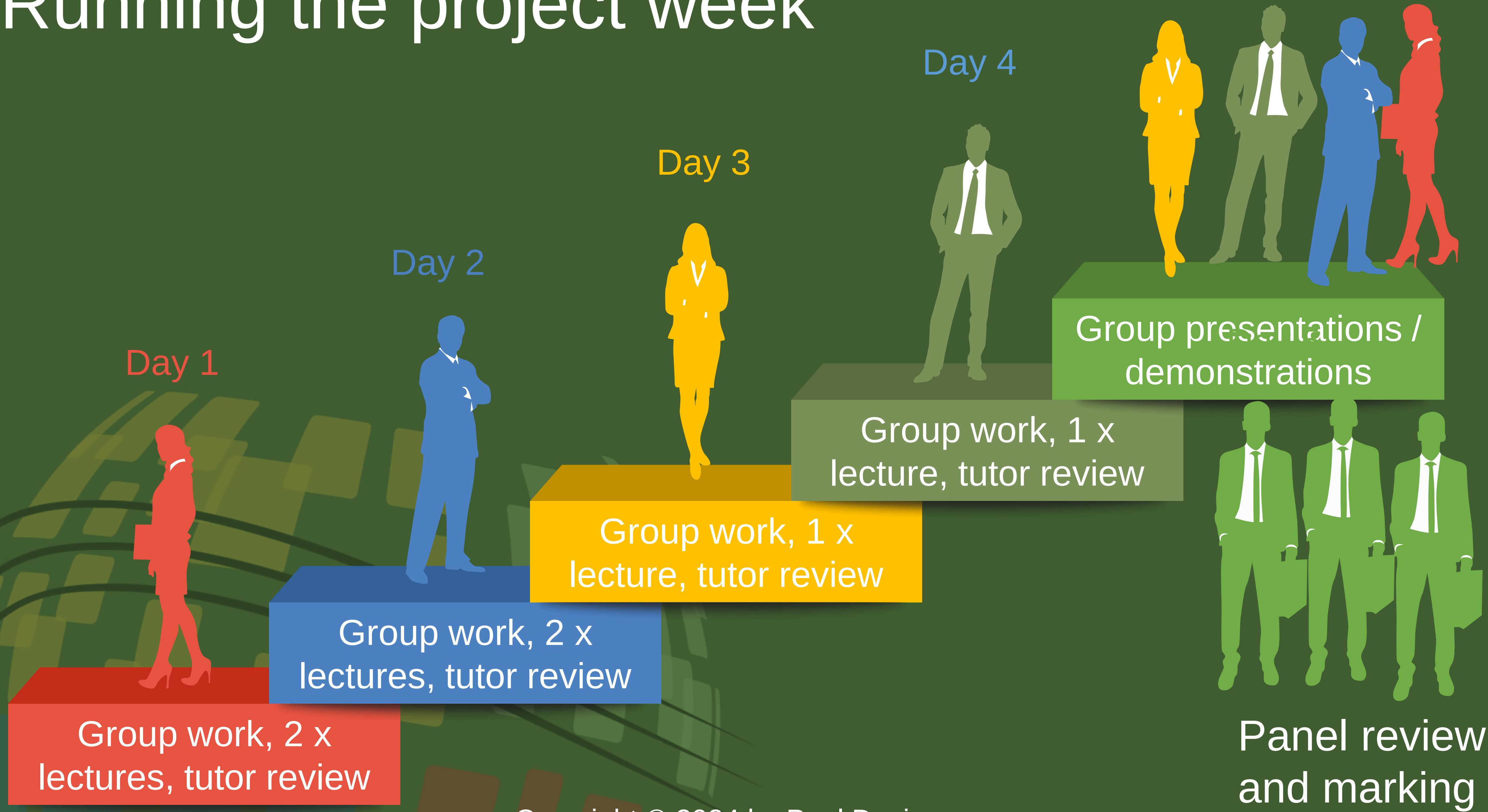
Initial System Breakdown Structure



Operations:

- Change mgmt.
- Version control
- Access control
- Manage refs to models

Running the project week



1. Introduction and Background
2. Module Outline
3. Conduct of the Project
4. **Results**
5. Observations and Conclusions

Overview

Expected solutions

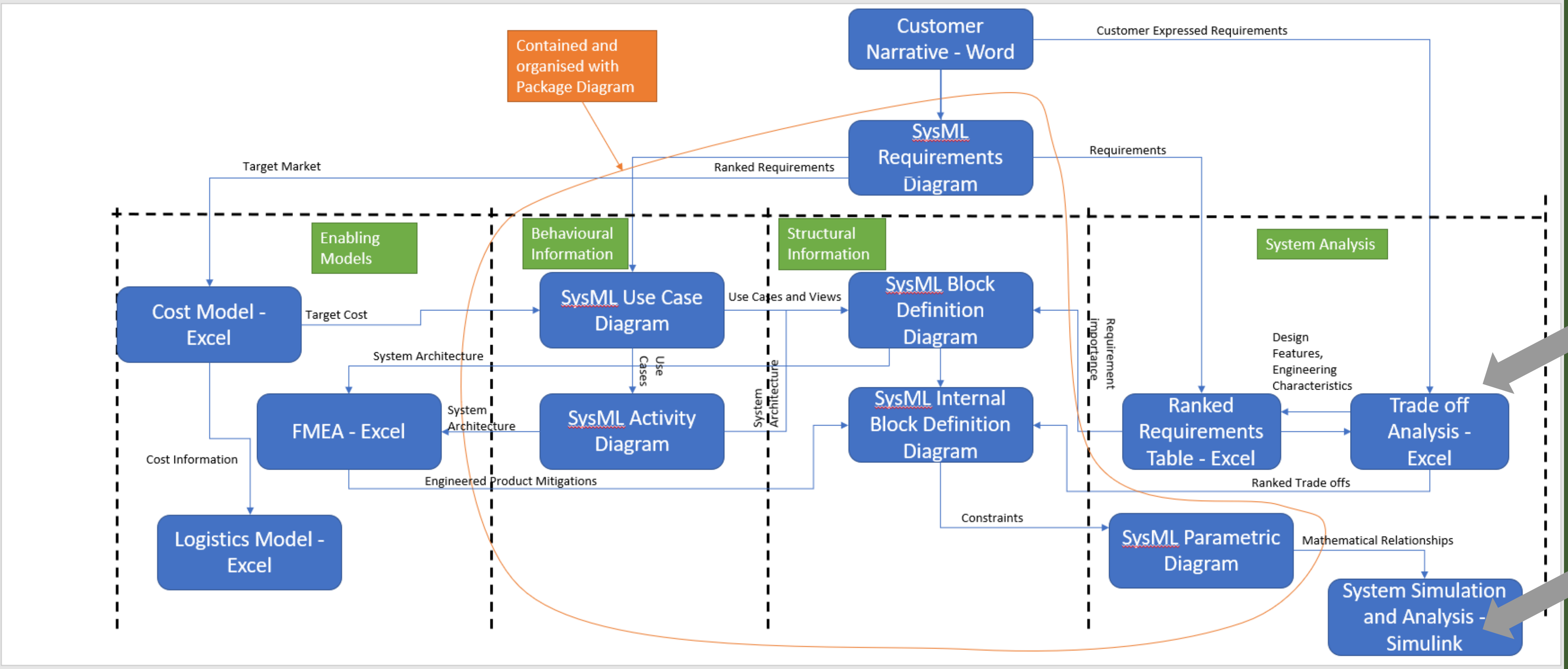
- Common problem setting
- Common tool suite
- Common marking scheme
- Common MMS requirements
- => similar solutions??

Actual solutions

- 20 different designs from 4 years' groups
- Wide variety of metamodels
- Several different linking mechanisms
- Very different Change Management 'dashboard' implementations
- None fully executable, several very close
- Many interesting workarounds to gaps in interoperability

Solution 1

Use cases for Change Management OK, but mandraulically implemented; Master Configuration Index implemented in nested self-populating BDDs

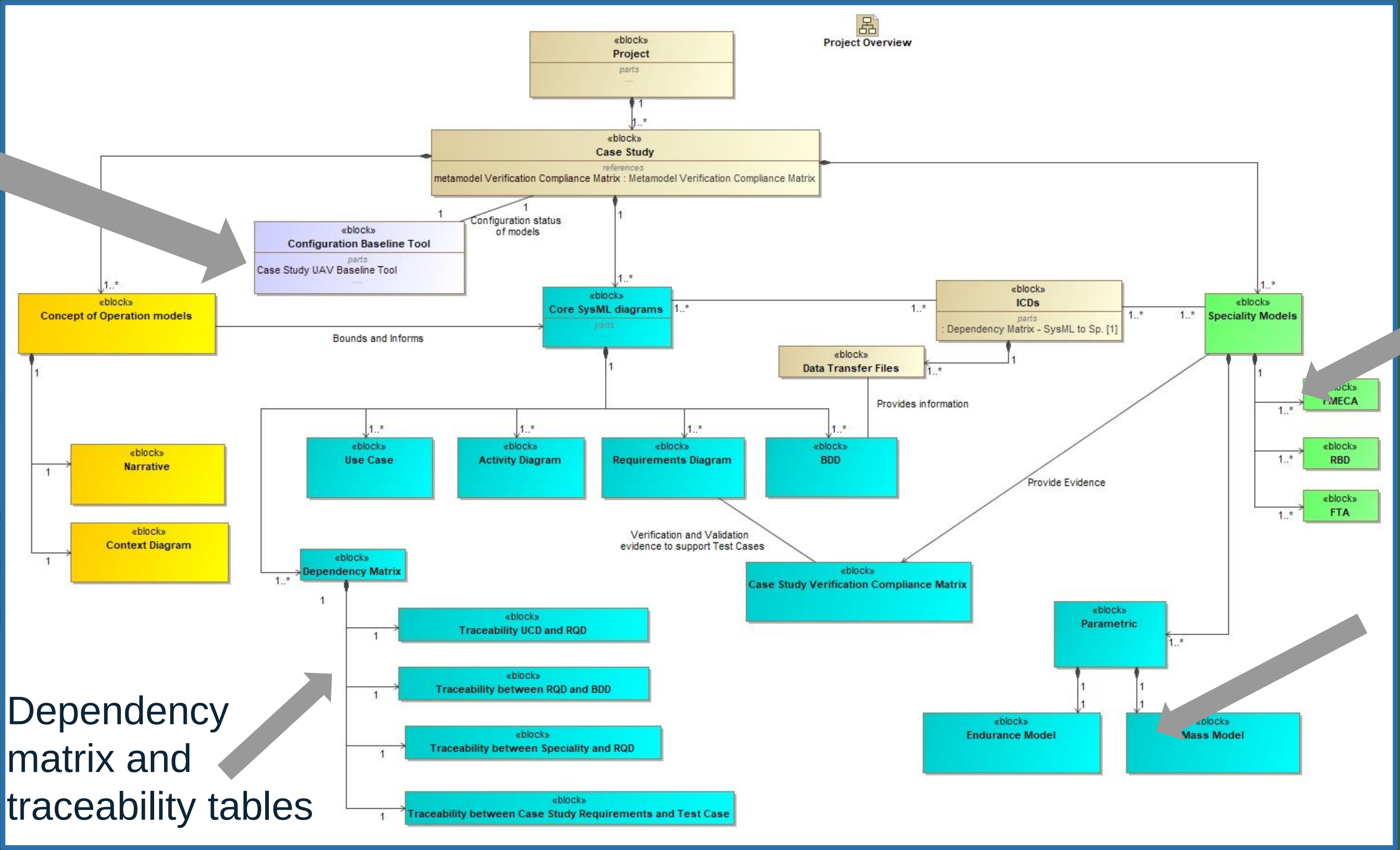


Interesting performance trade study in Excel, fed from model re-runs

Correct Para diagram linked to Simulink model

Solution 2

Excel config tool populated from RVM



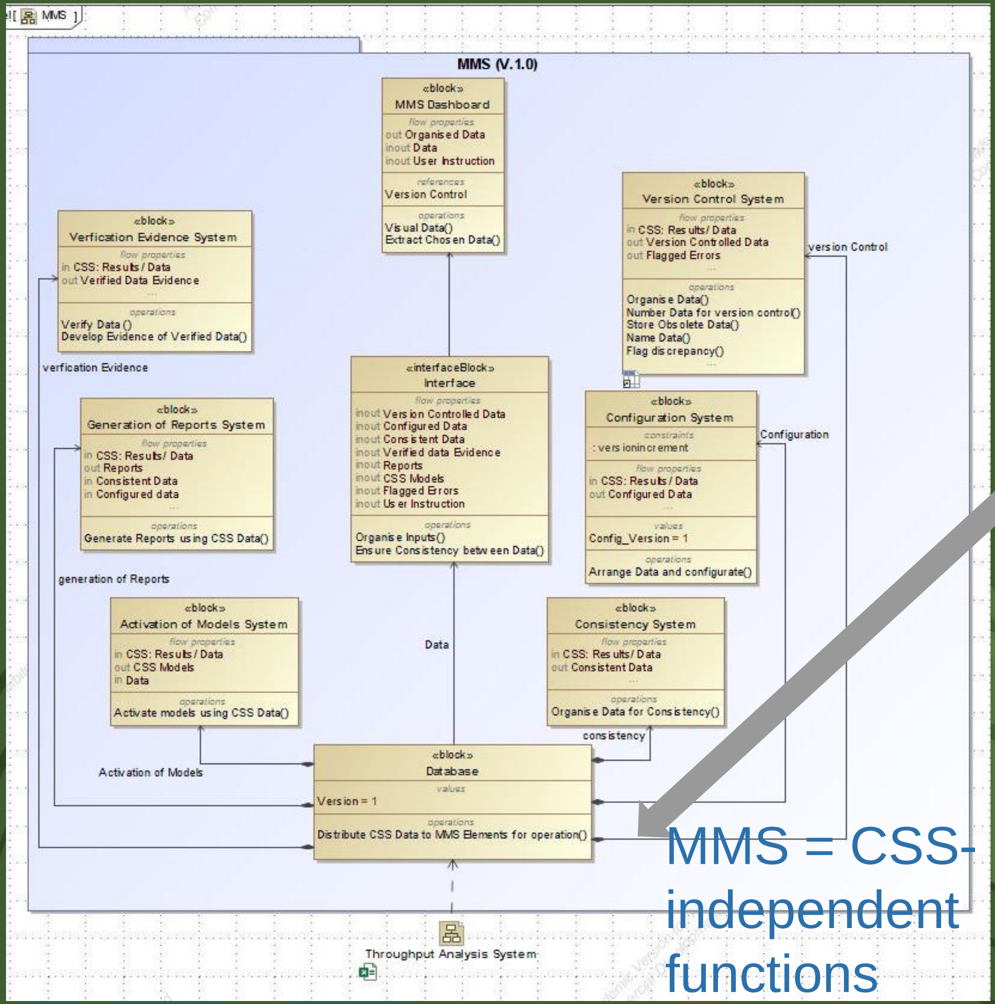
Excel and ext. tool models

Correct Para diagram linked to Simulink models

Dependency matrix and traceability tables

Solution 3

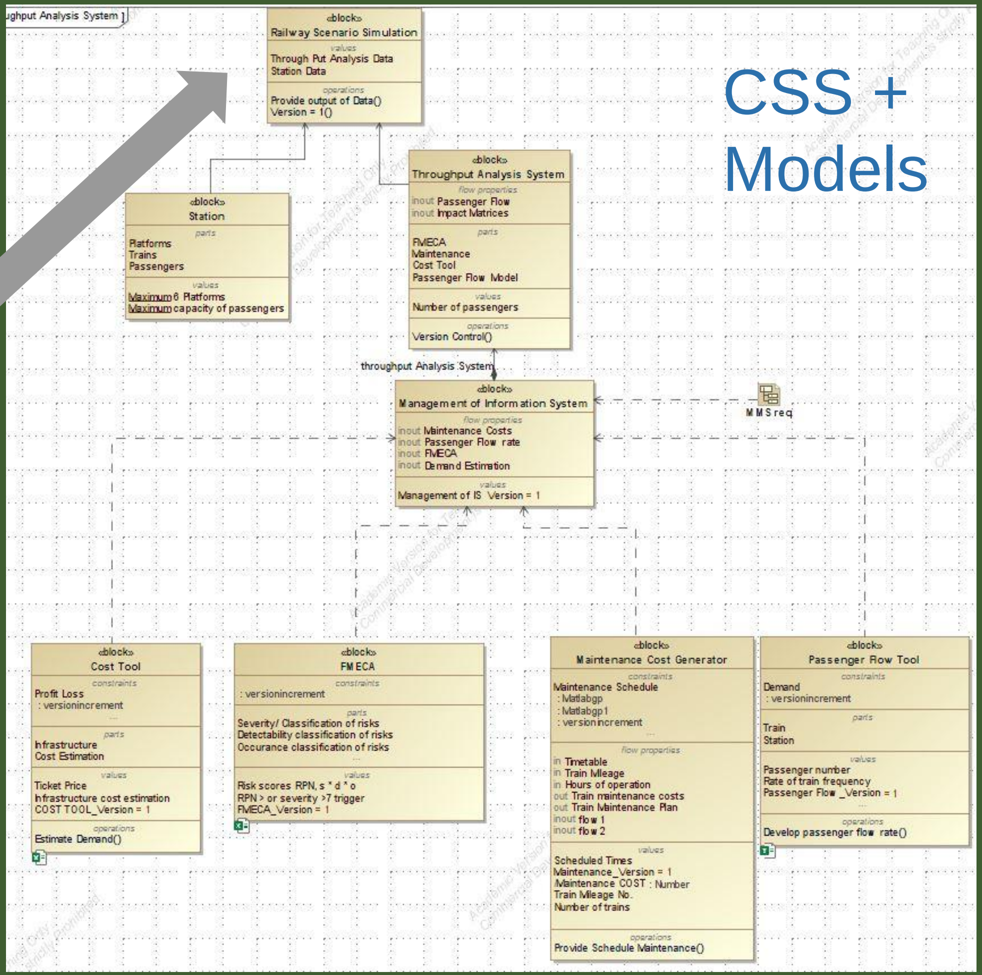
An attempt to be fully modular



MMS = CSS-independent functions

Single linking tables, self-describing

But it didn't work

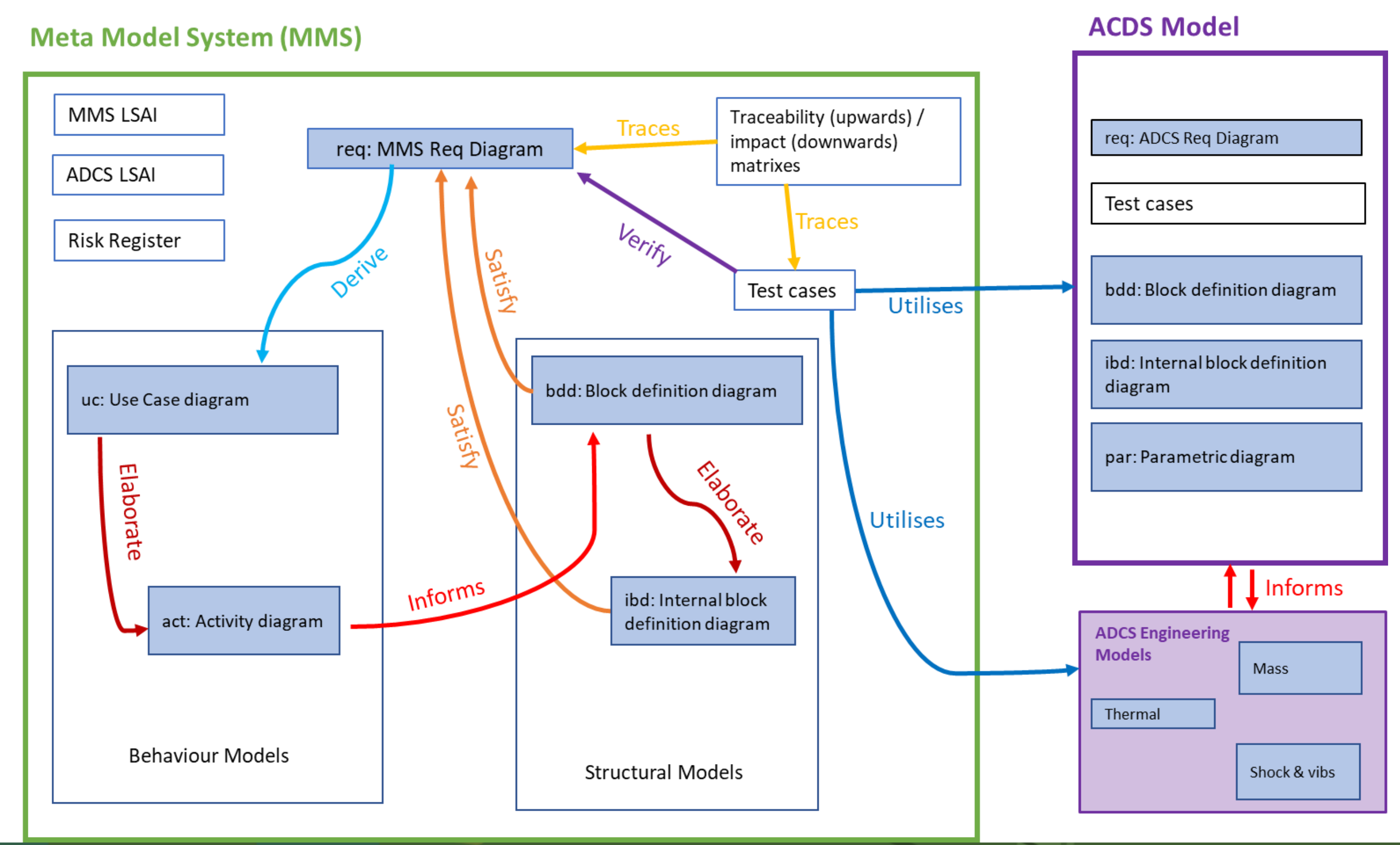


CSS + Models

Solution 4

- Latest State of Amendment Indix (LSAI) nicely constructed in Excel with multiple tabs
- Comprehensive V&V, with everything glued together via SysML Test Cases and IBDs

Requirements compliance checking using Constraint blocks



Initial MMS concept generated in PowerPoint.

Use case and activity diagrams showing behaviour of MMS.

Block definition and internal block definition diagrams detailed structure.

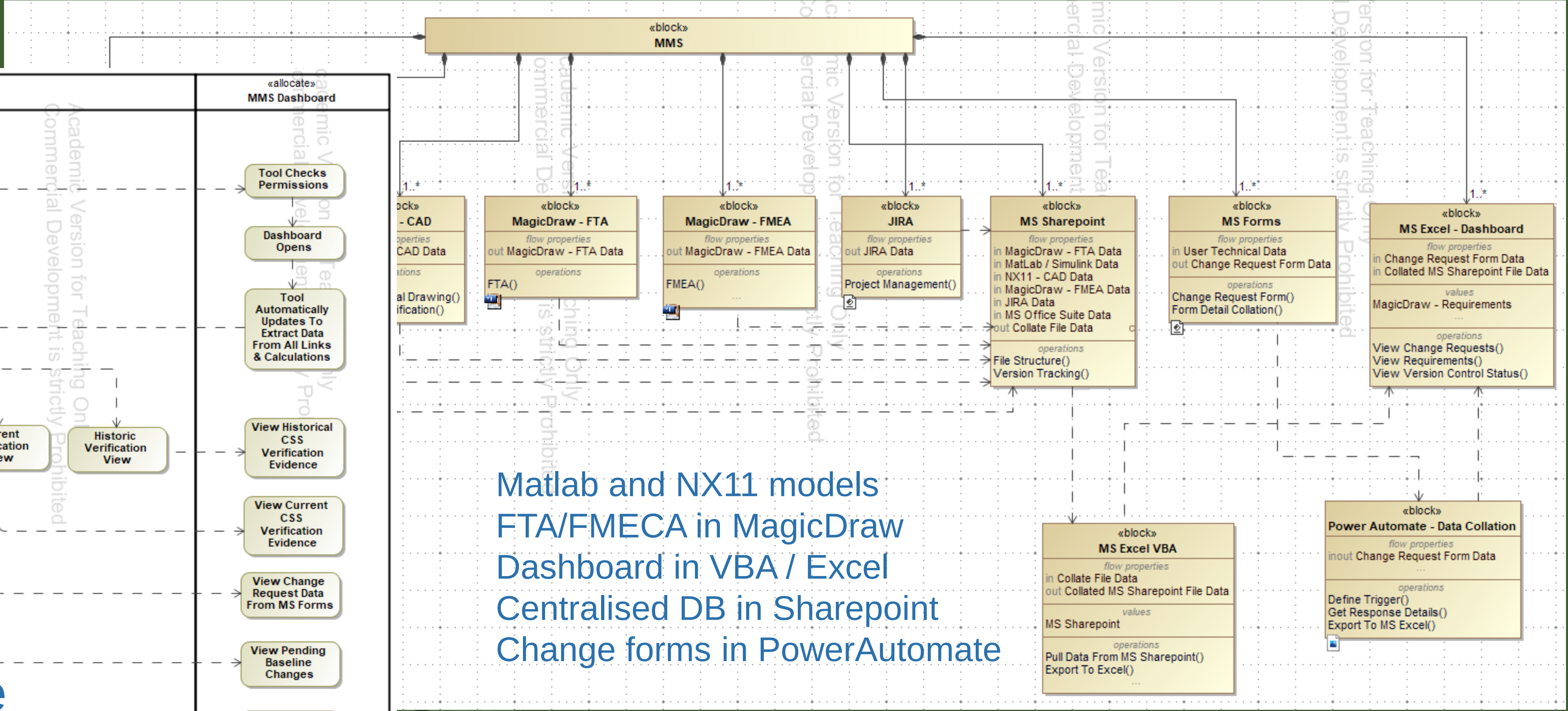
Relationships between all models shown.

ADCS Model separate but related to MMS.

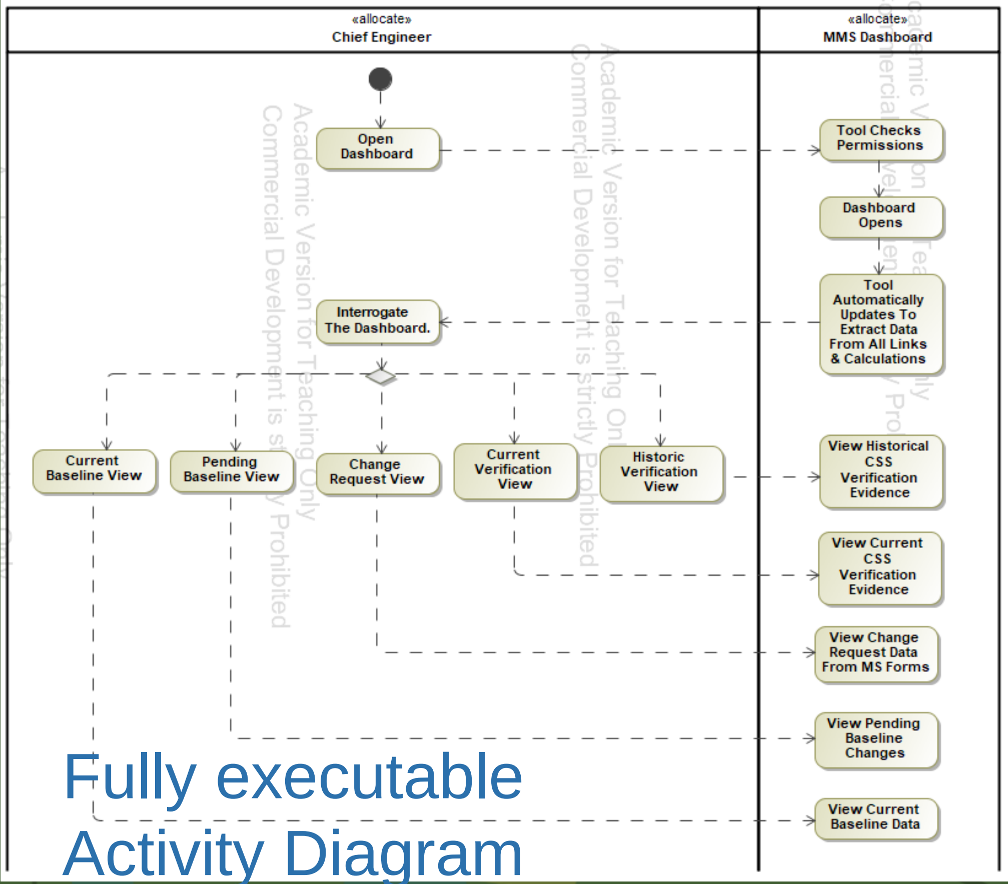
MMS LSAI created for document and model configuration control.

Engineering models maintained outside of MMS.

Solution 5

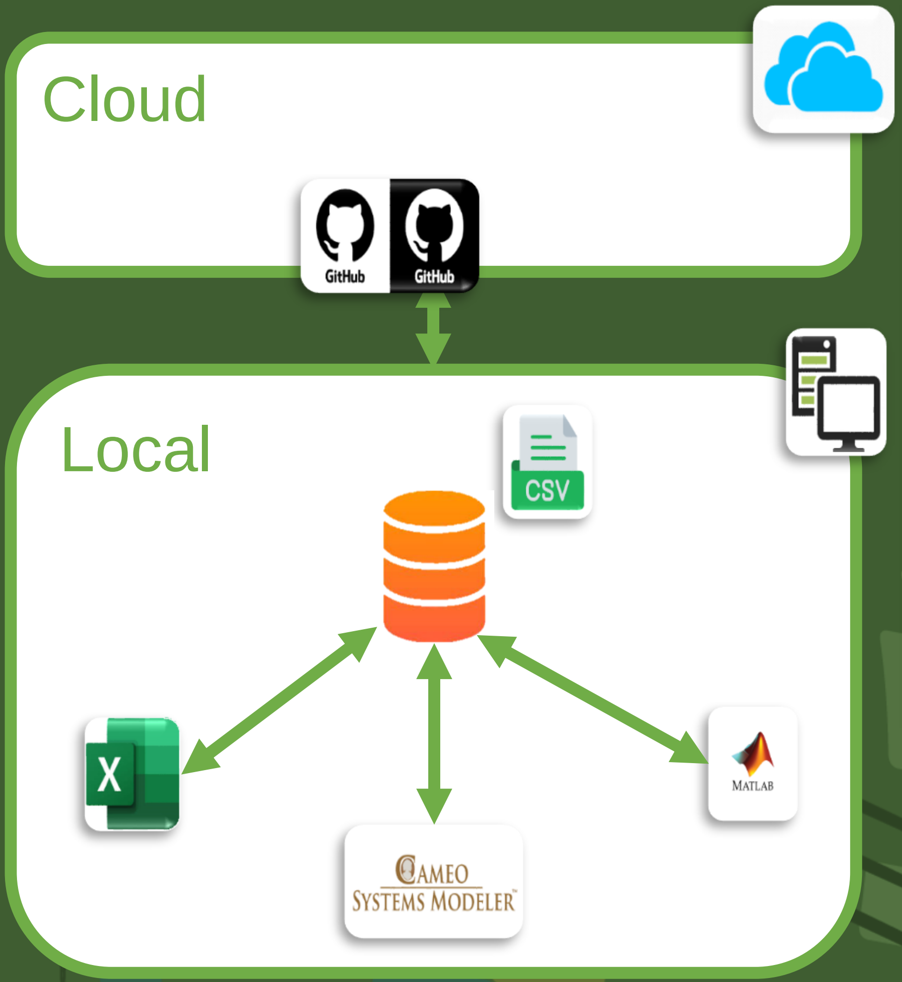


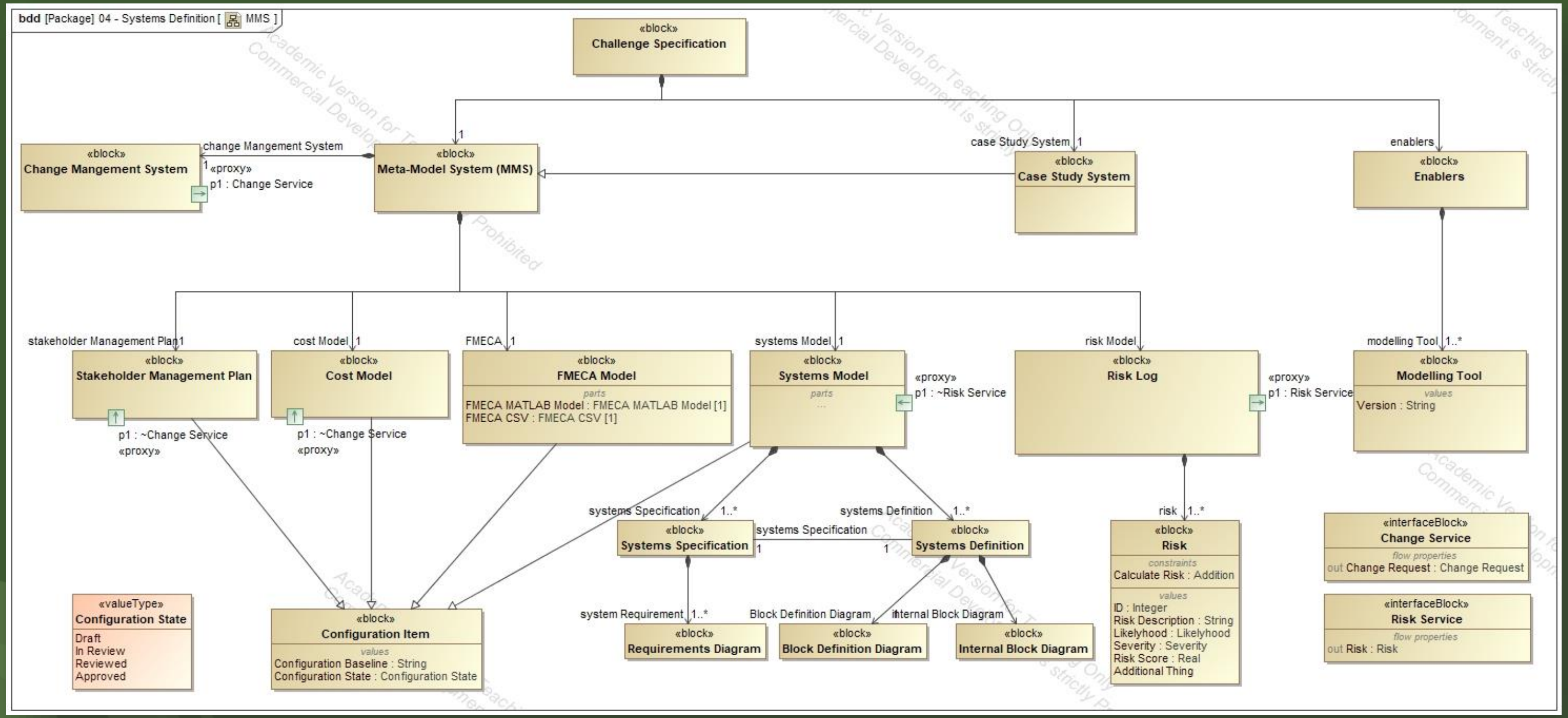
Matlab and NX11 models
FTA/FMECA in MagicDraw
Dashboard in VBA / Excel
Centralised DB in Sharepoint
Change forms in PowerAutomate



Fully executable
Activity Diagram

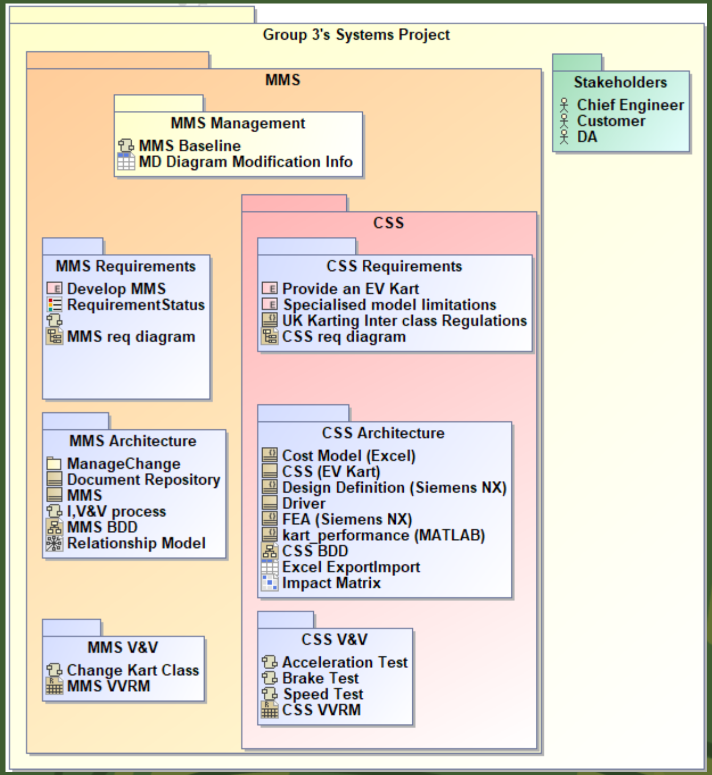
Solution 6

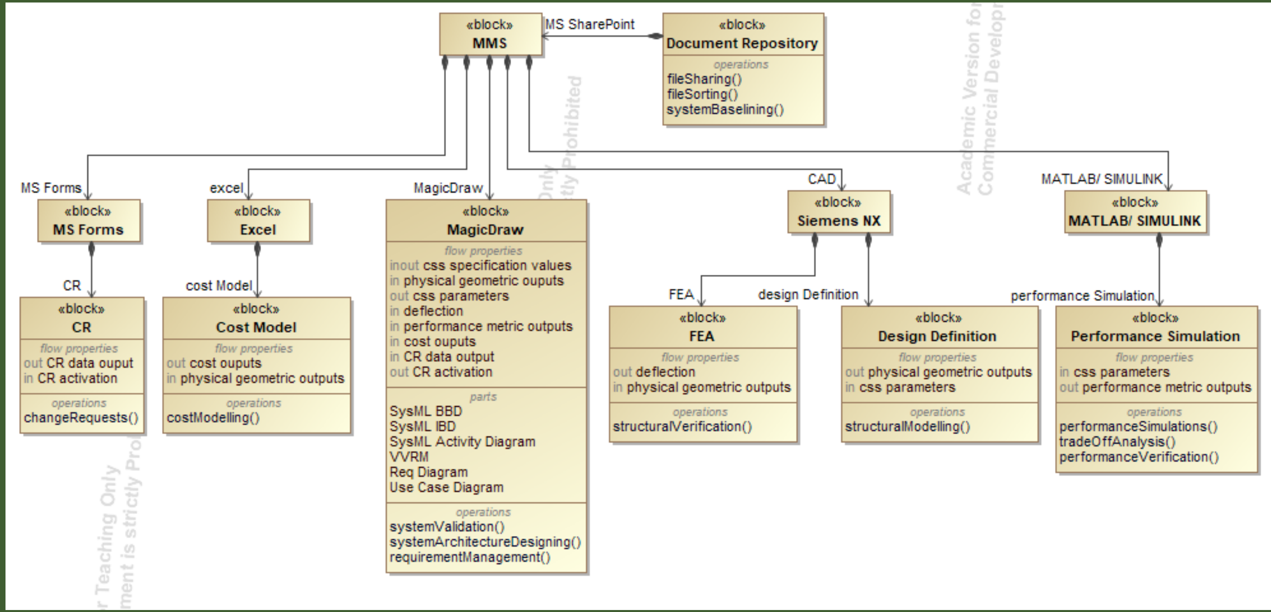




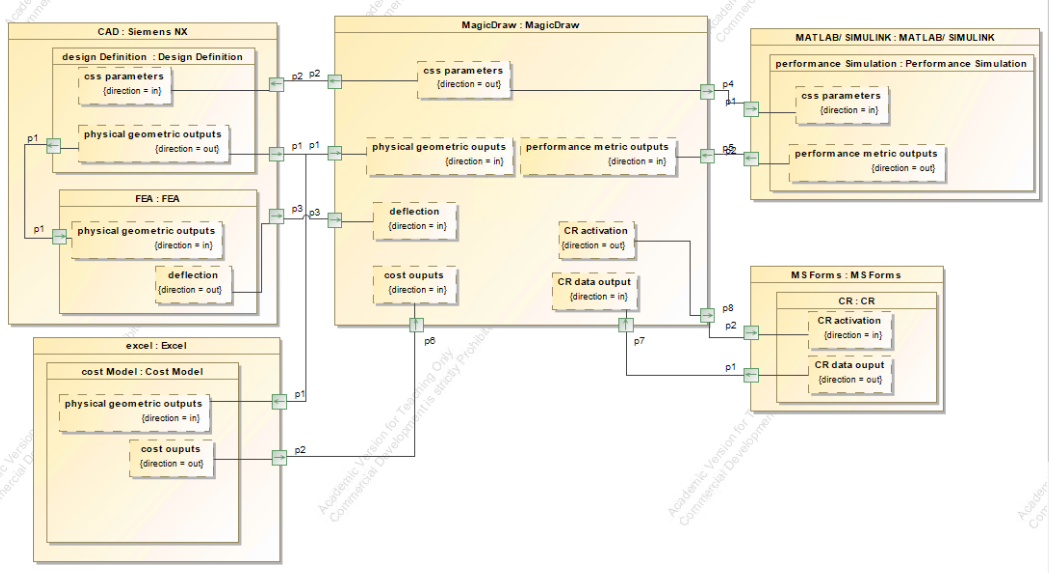
Config Management in GitHub
FMECA model in Matlab, Cost model (CoSysMo) in Excel
Risk Model and Change Management implemented as Services,
Implemented as a Service Oriented Architecture

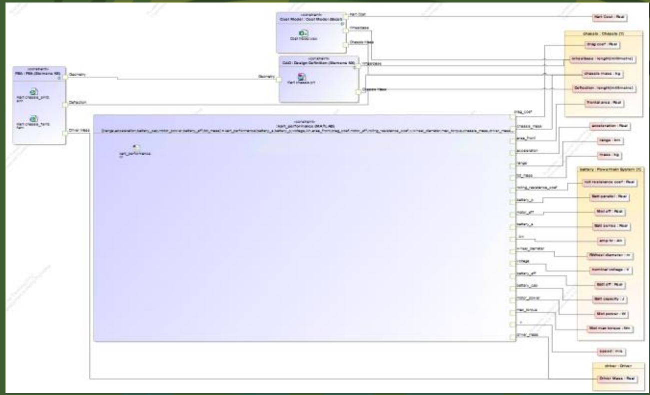
Solution 7





Specific model parameters tightly coupled via a tailored Internal Block Diagram (IBD); flow properties for model calls; embedded Parametric Diagrams and Constraint Blocks for compiled model code.



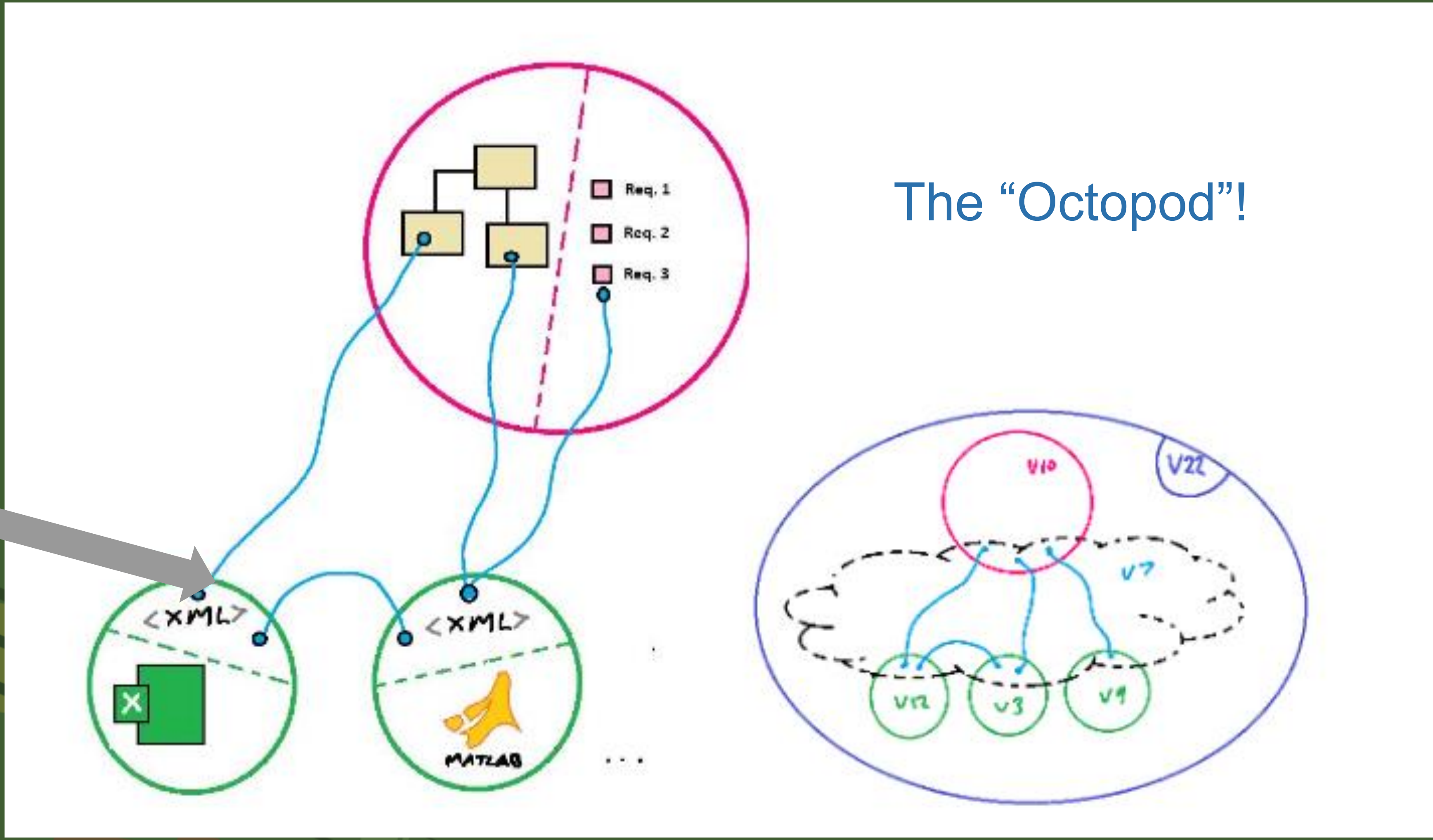


Not modular!

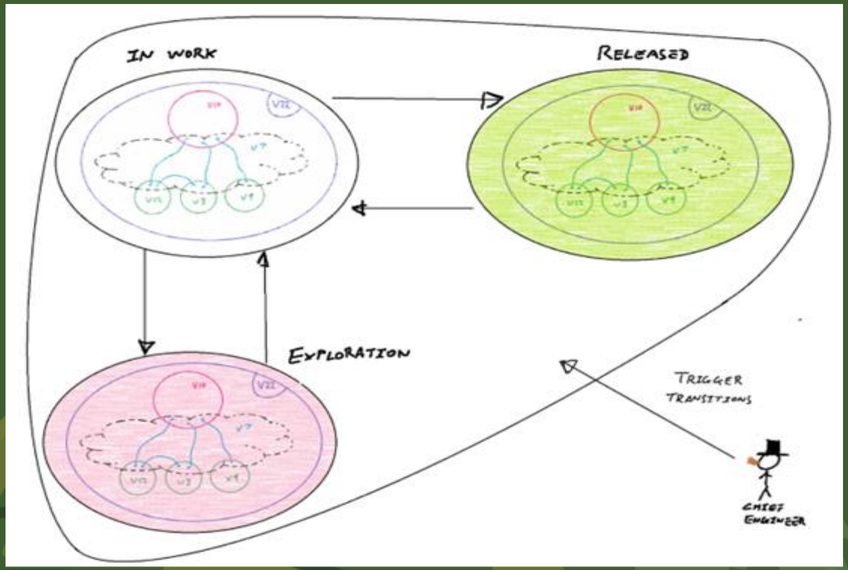
Solution 8

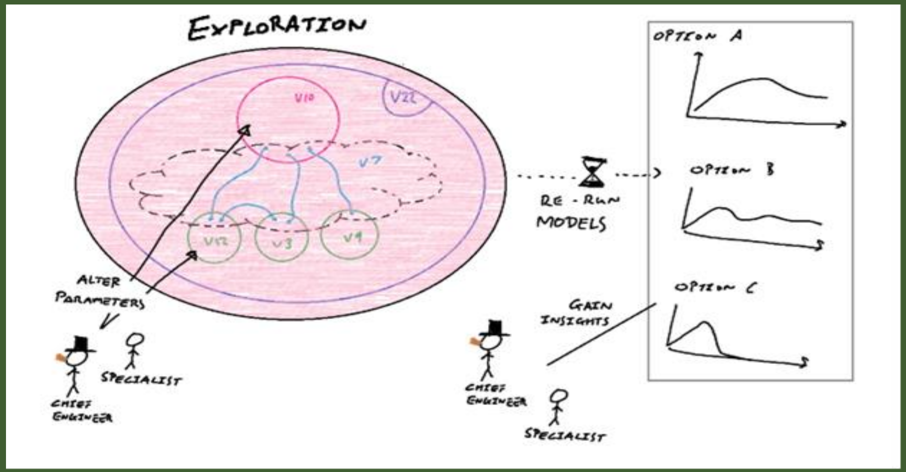
The right idea for version control

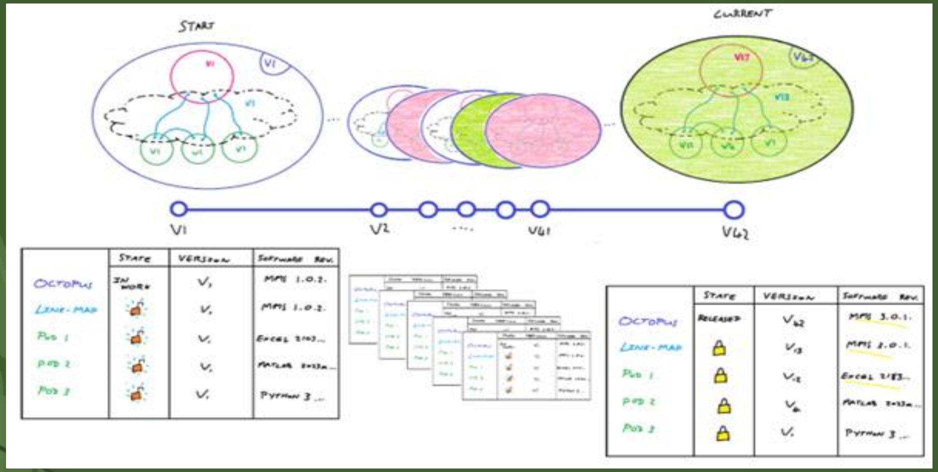
Bridging link modules



Solution 8 (2)





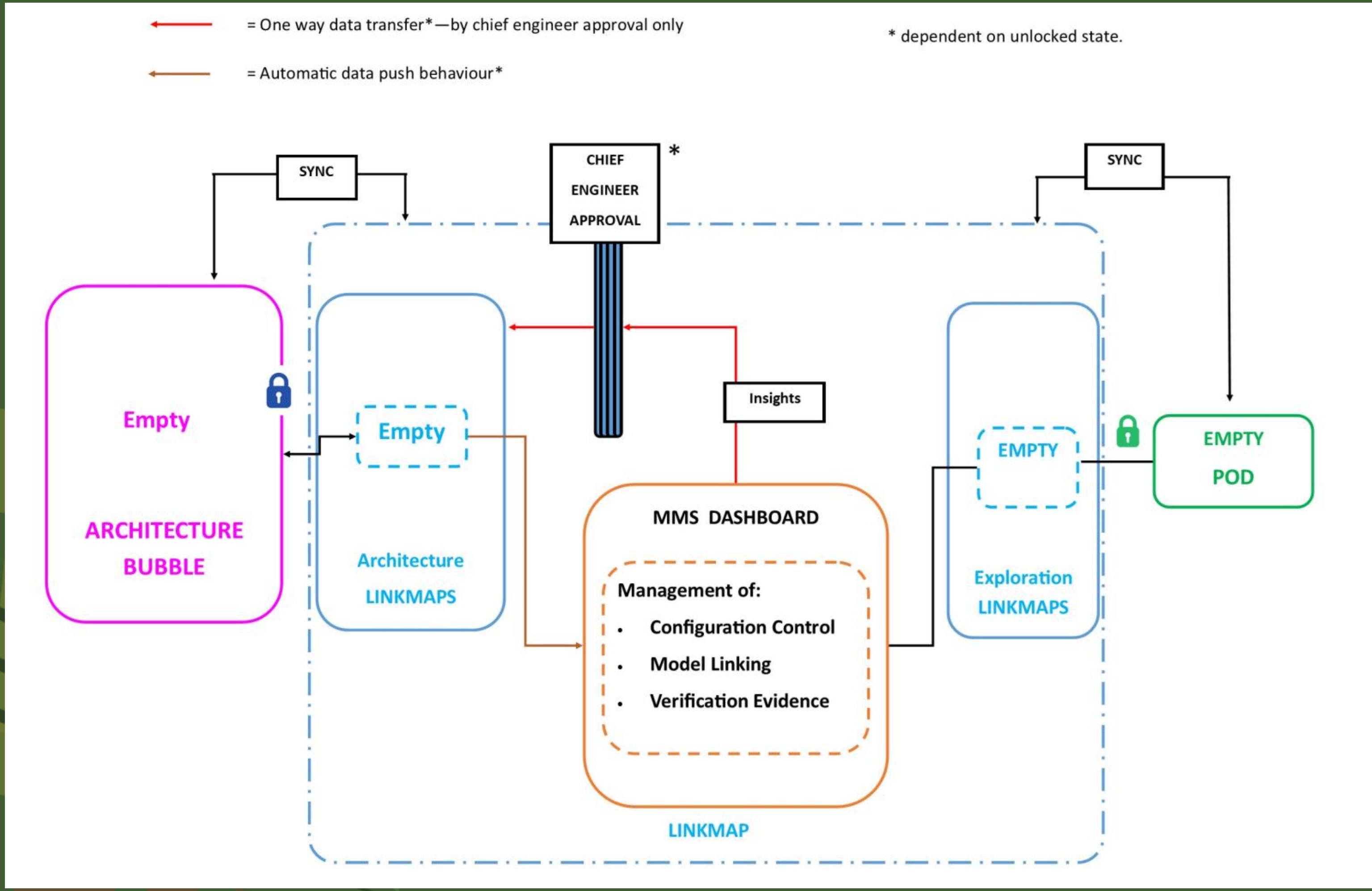


Rich picture for
Exploration Use Case

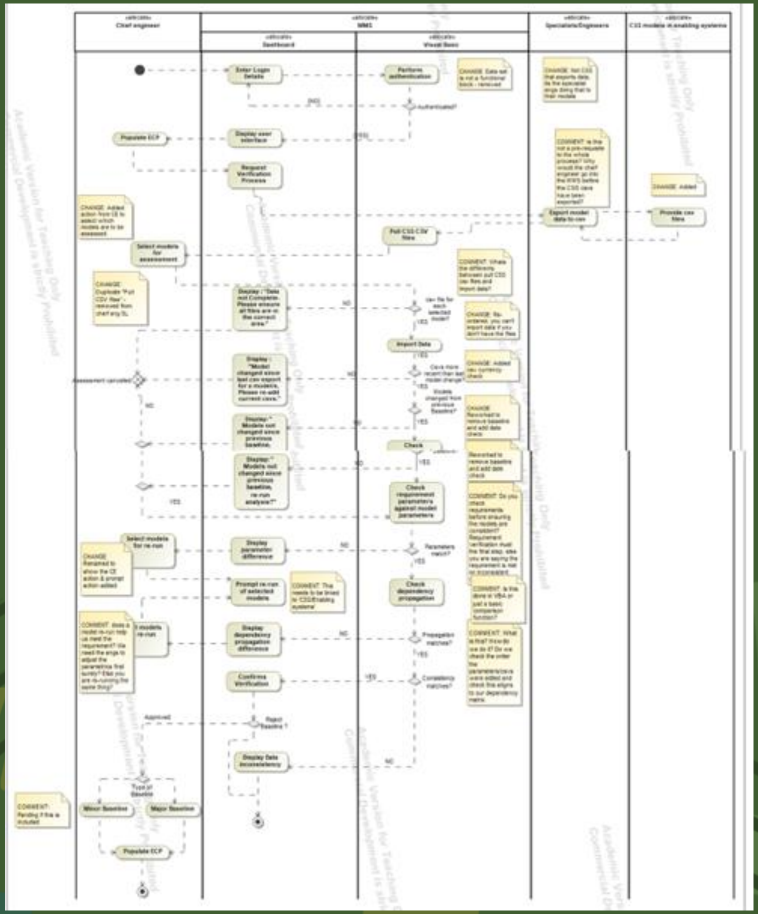
Solution 8 (3)

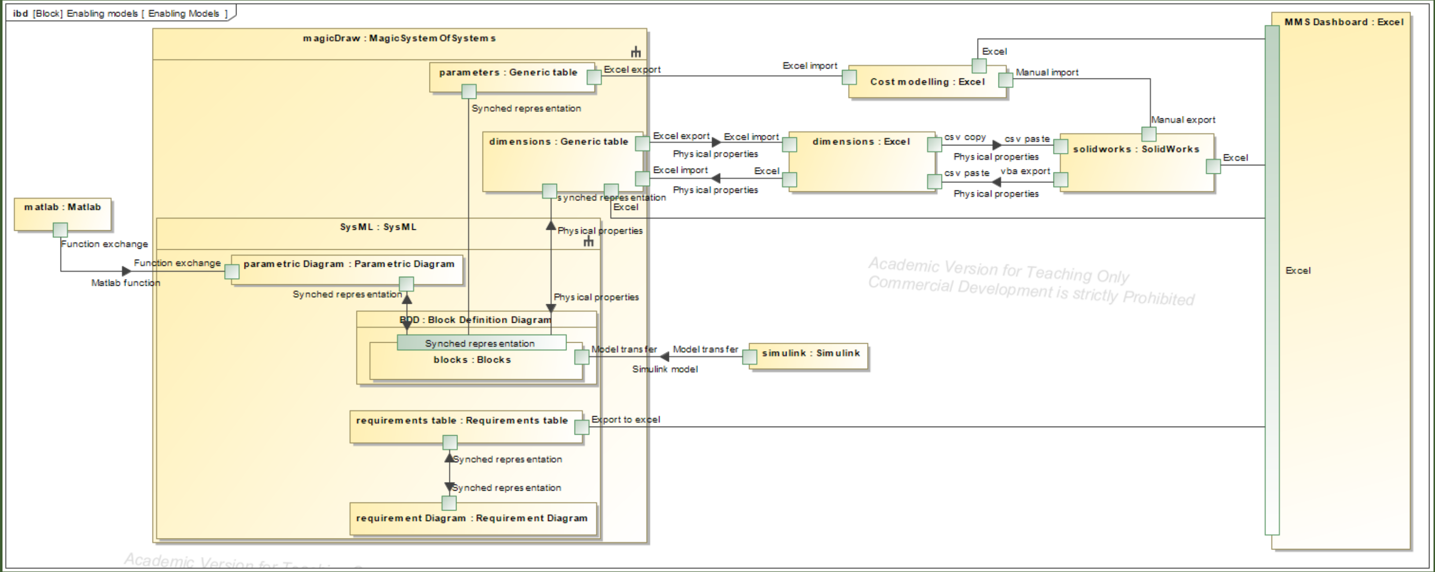
Implementation Schema

Fully functional
Modular
Intuitive



Solution 9



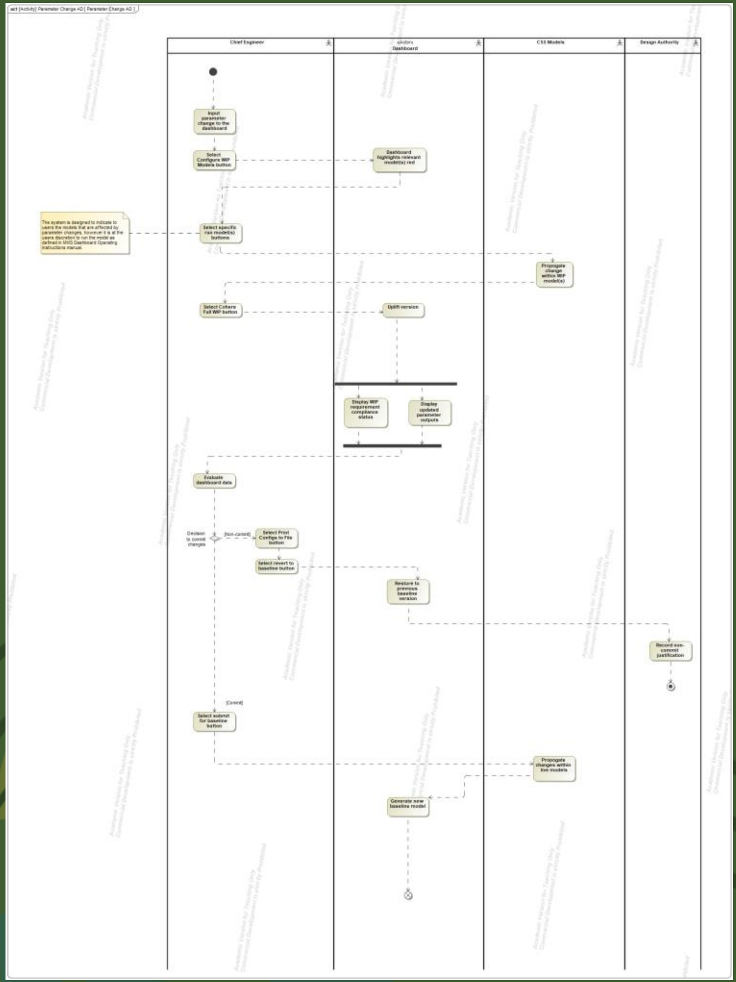


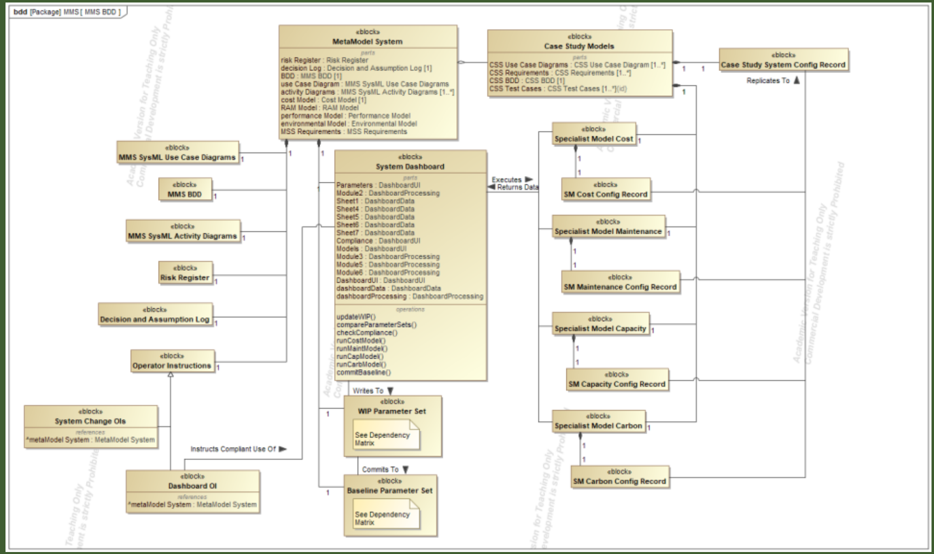
Produced Activity Diagrams and IBDs to confirm correct design…

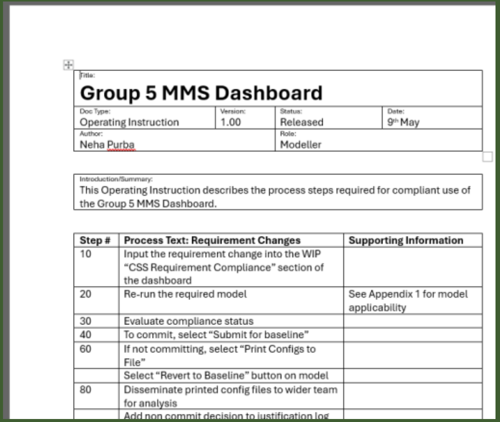
But then implemented all the linking and the Dashboard in Python!

With excellent and compliant results

Solution 10







Title:

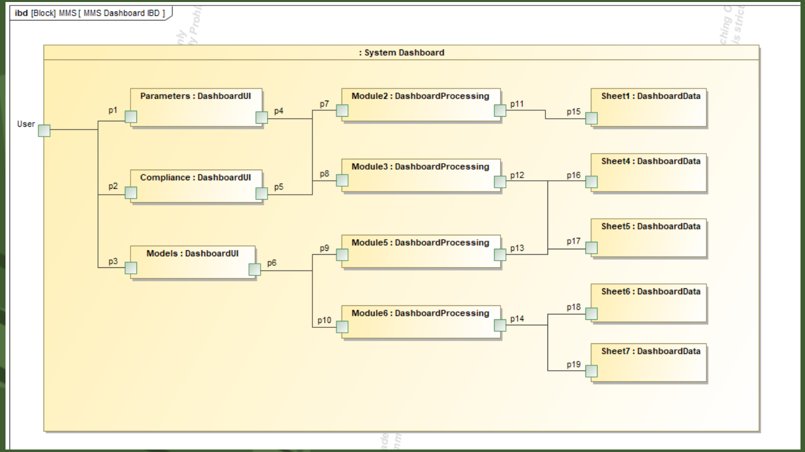
Group 5 MMS Dashboard

Doc Type:	Version:	Status:	Date:
Operating Instruction	1.00	Released	9th May

Author:	Role:
Neha Purba	Modeller

Introduction/Summary:
This Operating Instruction describes the process steps required for compliant use of the Group 5 MMS Dashboard.

Step #	Process Text: Requirement Changes	Supporting Information
10	Input the requirement change into the WIP "CSS Requirement Compliance" section of the dashboard	
20	Re-run the required model	See Appendix 1 for model applicability
30	Evaluate compliance status	
40	To commit, select "Submit for baseline"	
60	If not committing, select "Print Configs to File"	
	Select "Revert to Baseline" button on model	
80	Disseminate printed config files to wider team for analysis	
	Add non commit decision to justification log	



Similar method to Solution 9, but different models and Use Cases

Dashboard etc this time in VBA / Excel

And just as good!

1. Introduction and Background
2. Module Outline
3. Conduct of the Project
4. Results
5. **Observations and Conclusions**

Observations on Solutions

1. A common failing – not reading the requirements properly. Many SEMPs focused on the Case Study System, not the MetaModel System
2. Everyone leaps to the Architecture solution space before understanding the context correctly. Early tutor reviews focus on getting correct Use Cases and Activity Diagrams
3. Requirements were deliberately incomplete, to represent real projects. This tests groups' ability to challenge, refine and interpret the real CM use cases.
4. Everyone struggles with IBDs, Constraint Blocks and Parametric Diagrams.
5. Very few groups managed to automate their Activity Diagrams fully, but most were able to show how it should work with better tool suite interoperability.
6. Coupling between CSS, MMS and specialist models varied enormously – leading to trades between efficient, customer-specifc operation or modularity of the MMS.

Observations on Groups

1. Most teams started well, by comparing notes on individual strengths and weaknesses, plus relevant modules previously undertaken, before assigning project roles.
2. Not all groups listened properly to the guidance given by the course tutor. There was a strong correlation between those who listen well and high-scoring solutions.
3. Not all team members 'got' the concept of the metamodel, and there was a marked difference in speed of compilation of a mental map of the relationships between the MMS, the CSS and the animations of the relationships between them. The best system architects managed to convey this mental model to their group.
4. All teams collectively underestimated the effort required for architecting and integration – again, representative of real project life!

Students' Individual Reports

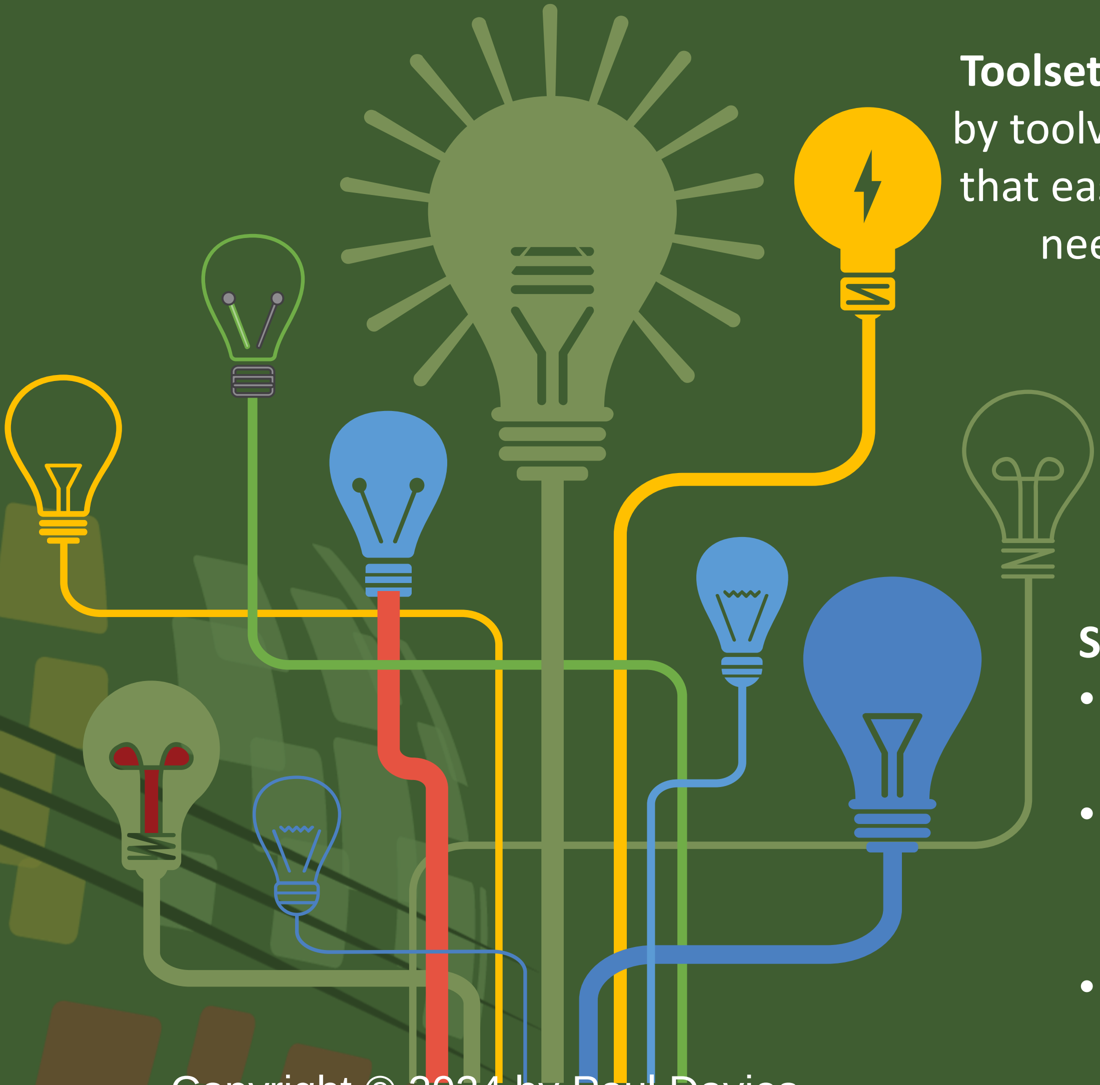
- The final required piece of coursework:
  - Analysis of group achievement of the learning objectives
  - Their solution, the processes used, and the team dynamics
  - Suggested improvements based on assessor feedback and further personal research
  - Lessons learned and portability for future work benefit
- The final project week lecture suggested more advanced tool suites, e.g. Intercax Syndeia, Cameo Systems Modeler and Ansys ModelCenter:
  - A big discriminator between students was their motivation and ability to follow up the references and apply them to the project challenge context

Conclusions

Realism – students appreciate the opportunity to put together their learning in a realistic project context

Dealing with Ambiguity – the gaps in the requirements lead to productive group debate on resolution – again, realistic

Tailoring – SEMPs are useless if they are generic repeats of standards. Every project is unique



Toolset Integration – often claimed by toolvendors, in real life it's never that easy. Students learn what they need to do, then how to do get towards a real ASOT.

Variety – the range of solutions shows that MBSE does not reduce SE to simply following a recipe

Successes of the Module:

- Cohorts of students better armed to deal with projects
- Proof that MBSE integration can be achieved in 4 days by a small group with help
- Positive reviews by external assessor, and by students themselves

34th Annual **INCOSE**
international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024