ECSE-211
Design Principles and Methods

Lecture 5: From Requirements to a System Model and beyond…

Date: 25 January 2023

1

Last Lecture Review

- Communication in a design team – why it is important
- How the company organization can impact the EDP
- Information – recording and transmission
- The concept of a document and a rationale for effective documentation
- The Requirements Document

2

Document Structure

- To be useful a document needs to convey basic information about itself
  - This is required for context
- Without structure, a document is of little use
- Typical header information common to all documents :
  - A Title – what is the document about?
  - Who is responsible for it?
    - An individual who can be contacted in the event of questions related to information in the document
  - Who has edited it?
    - Who has been involved in changing the information

3

Document Structure

- Typical header information common to all documents (cont'd):
  - Date: When was the document created?
  - Revision Date: When was the document last revised?
    - When put in the context of the EDP, this date can indicate when certain decisions were made
  - Version Number:
    - The current version number
  - Edit History:
    - A list of changes that were made
    - For each change:
      - Why was it made
      - Who made it
      - When was it made

4

Document Structure

- Following the Header, the main body of the document addresses the information related to the title
- For example, for the Requirements Document, the main body should include:
  - What is the system meant to do (Purpose and Scope)?
    - List any performance data you have and desired capabilities
  - What can you use to solve the design problem (Constraints)?
    - List any items that are explicitly specified, or limitations imposed by the client
  - Are there tolerances on performance or limits on user interaction?
    - List them
  - Is there a deadline?
    - List it
  - Do you know everything? (Unknowns)
  - …
  - When is the Requirements Document complete? What is the next step?

5

Question 1

- When is the Requirements Document complete? What is the next step?

At the end of the project.

Systems model

6

Question 2

- What went wrong in the Tree Swing design process?

No communication

7

Question 3

- Why can the Company Organization have an impact on the EDP?

information flow.

8

Question 4

- How is the Design Process controlled?

Communicati feedback
inter group information transfe

9

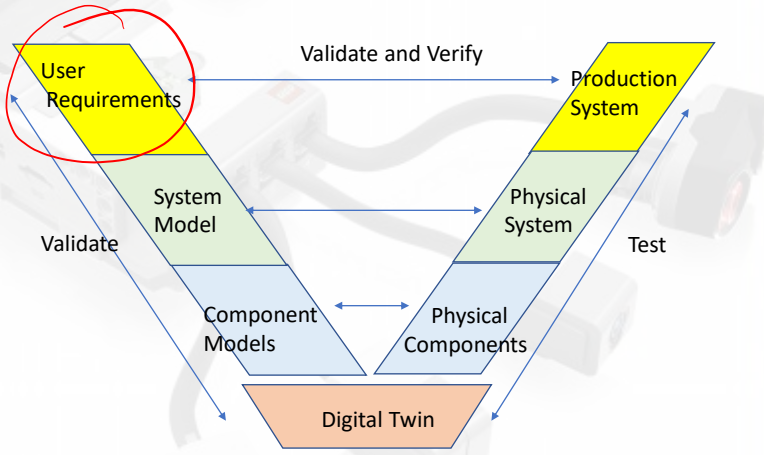
Contents

- What is a System Model?
- Inputs to the System Model Creation
- The Team Capabilities
- Generating the System Model
- Identifying Possible Design Implementations

10

What is a System Model?

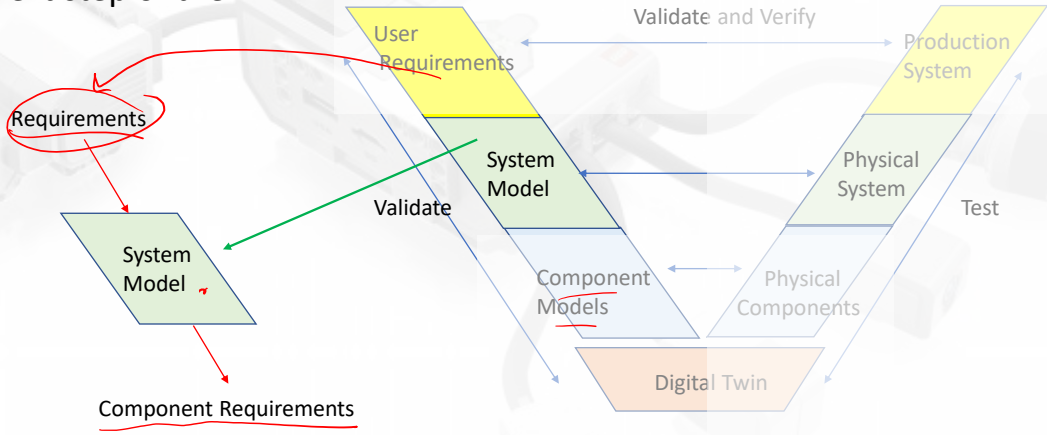
The next step of the EDP



11

What is a System Model?

The next step of the EDP



12

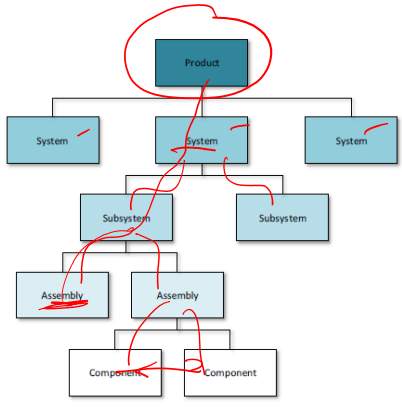
What is a System Model?

- What is the goal of the Model?
- Models provide a way of testing (or validating) ideas…
  - Simple algebraic models might allow the prediction of the navigation performance of a robot
  - A free body diagram might allow a prediction of how the various mechanical forces will behave
  - …
- So – a System Model should allow a validation of a possible structure that would meet the user description expressed in the Requirements Document

13

The System Model

- The Requirements lead to the development of a System Model
- What is a System?
  - *A collection of interdependent functional elements that, when brought together as a single unit combine to meet a set of common objectives.*



Example of System Hierarchy

Image courtesy of Morgan Jenkins, Siemens DISW

14

The System Model

- Requirement:
  "I want the highest electric range for a car in this class"

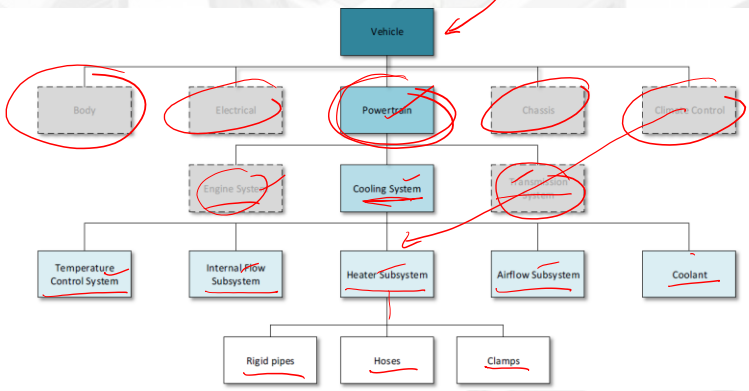


Image courtesy of Morgan Jenkins, Siemens DISW

15

The System Model

- A sketch showing how the subsystems address the requirements
- Combined with the previous diagram, we now have a System Model

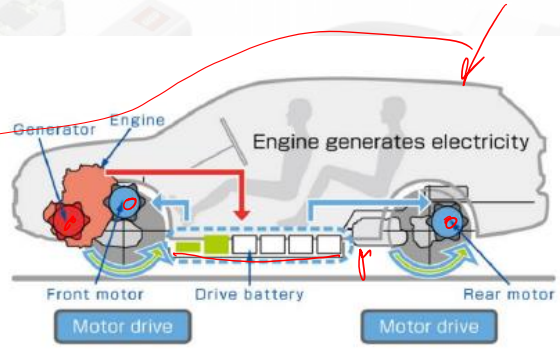
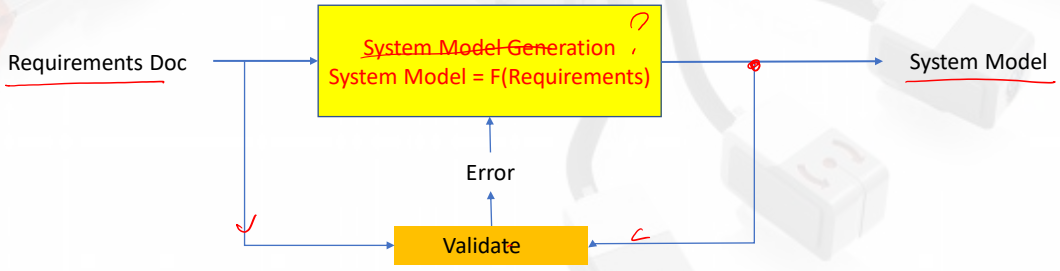


Image courtesy of Morgan Jenkins, Siemens DISW

16

Inputs to the System Model Creation

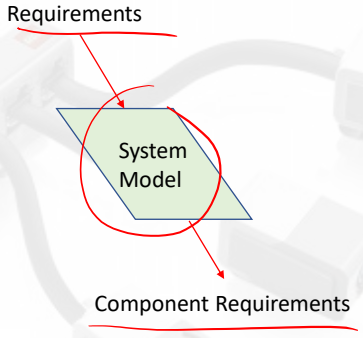
- The V-Cycle illustrates the *States* of the design process and the validation that occurs for each State.
- Moving from state to state requires a process
- Arriving at the System Model requires a process which acts on the Requirements Document and "transforms" it



17

Inputs to the System Model Creation

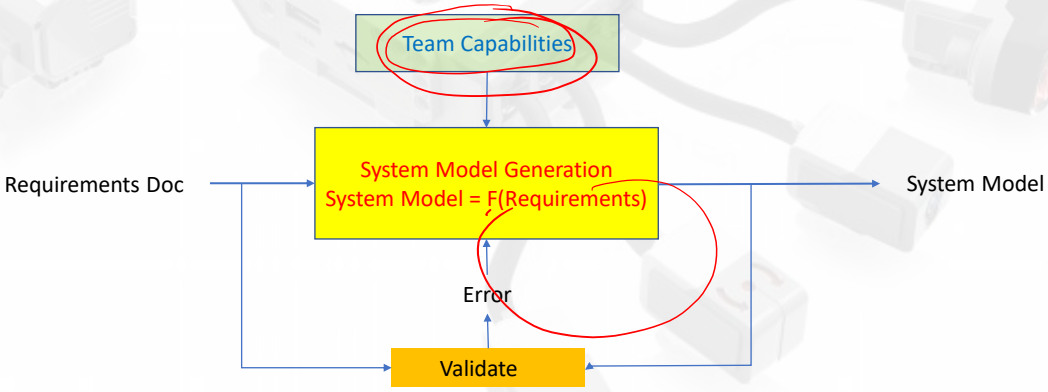
- The previous slides showed examples of a System Model, i.e. the outputs of the model generation process.
- What are the Inputs?
- Are the User Requirements the only inputs?
- What else might contribute to the generation of a System Model?



18

Inputs to the System Model Creation

- The Process is implemented by the Design Team
- The Design Team has Capabilities



19

Team Capabilities

- The process to generate the System Model is implemented by the Design Team
- Each member has knowledge which can be used to map the Requirements onto a System Design
- There is a need to know who can do what
  - Who is a software engineer?
  - Who understands mechanics?
  - Who has knowledge about systems?
- This information needs to be acquired before the system generation process can happen

20

The Team Capabilities Document

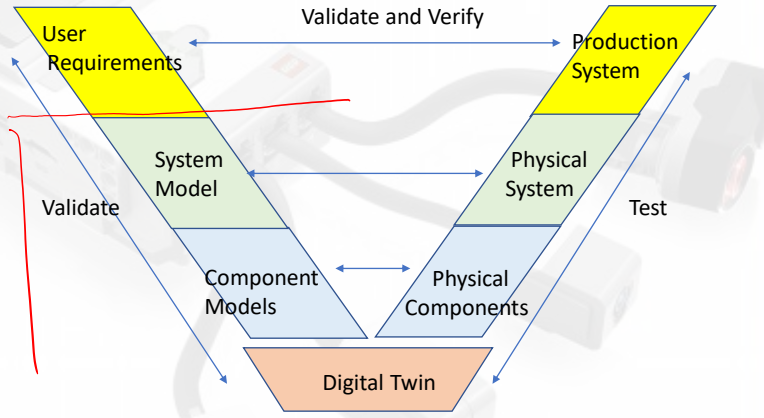
- What is the team capable of? What is your knowledge/capability base?
  - Create an inventory of capabilities – these may constrain your solution..
  - Document what you can do:
    - E.g. John has worked with Mindstorms before – he knows how to program it
    - E.g. Mary has been involved in a robotics project before McGill and understands the concept of a System

Create a document – Write it down – who has expertise in what?

21

The Capabilities Document

- This document needs to be generated before the system model can be created
  - In fact, it is required for the entire design process – the team is the entity that implements each of the processes necessary to move through the states of the V-Cycle



22

The Capabilities Document

- This is the knowledge/skill base of the team
  - Who can program?
  - Who understands mechanics?
  - Who understands systems?
  - Who can manage?
  - …..
- As with the Requirements Document, the Capabilities Document has a similar Header block and a structured contents.
- It may also be modified during the Design Process since it may also contain information such as availabilities

23

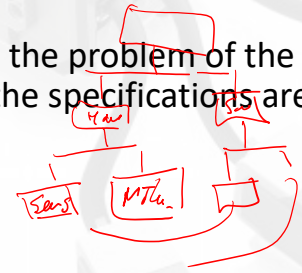
The Design Team

- The Capabilities Document is relatively unique to DPM
- In an industrial environment, the Design Team might be generated as a result of the Requirements Document
  - For example, the Requirements suggest that there will be electrical systems in the device – so include an electrical engineer in the team
  - Maybe there is to be a software component – so a software engineer would be needed
- So – the Design Team is often structured specifically to solve the given problem… - the jobs/roles are defined and then filled.

24

The System Model

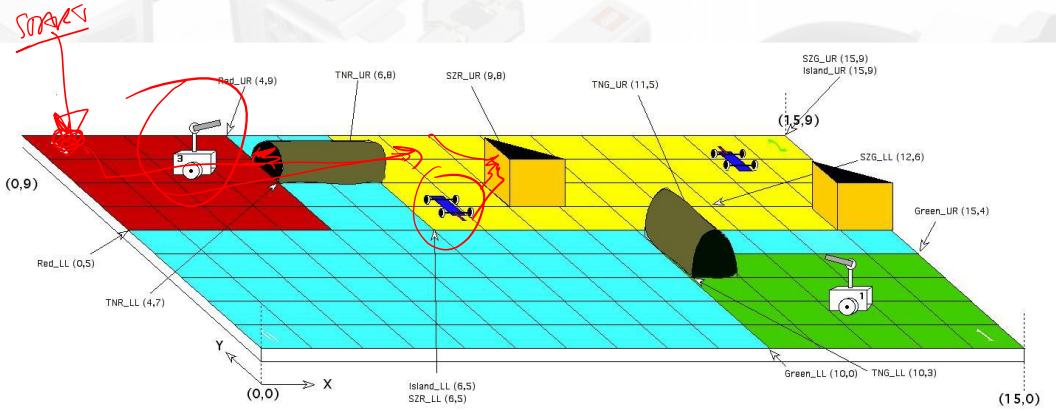
- For a Lego Robot, the "Product" is the robot
  - The System is the Hardware and Software structures and their interconnections that implement the Requirements
  - The subsystems are the sensors, the motor drives, the Lego structures, the software components, etc.
- In the following slides, consider the problem of the stranded vehicle discussed a few lectures ago – the specifications are on MyCourses



25

The Stranded Vehicle Problem

- The game is played in a constrained space:



26

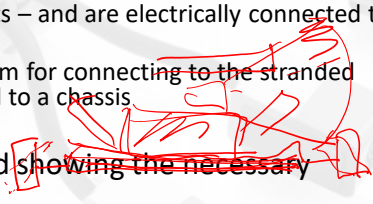
Generating the System Model

- So – for the DPM problem – A State Diagram
  - The System model could start as a simple diagram derived from the Requirements showing the various states that the system could find itself in
    - Each state could lead on to one or more states depending on the inputs to the system
    - For Example –
      - once the system is started, the first state would be to wait for information from the server
      - The next state would be to localize
      - Then navigate
      - From navigation, it could move to obstacle avoidance or it might reach its destination
    - This set of states can be validated against the Requirements Document

27

Generating the System Model

- For the DPM Problem – A Mechanical-Electrical Diagram
  - To implement each state will require a physical structure so a simple mechanical and electrical system can be described
    - Wheels and electric motors are needed to drive
    - These are electrically connected to the control brick
    - Sensors are needed to identify lines, walls, objects – and are electrically connected to the brick
    - In the case of the problem described, a mechanism for connecting to the stranded vehicle is needed.. this is mechanically connected to a chassis
    - All the components are connected to the chassis
  - From this list, a diagram could be constructed showing the necessary mechanical and electrical structures
  - Note that the actual implementations have not been described here – we are still at the Systems Level



28

Generating the System Model

- For the DPM Problem – A Software Diagram
  - The control system for the robot is to be implemented in software
  - The software blocks can be identified and the data flow from inputs (sensors) and to outputs (motors) can be detailed
- Note – there is no implementation of the software here – just a list of the blocks needed
- The three components of the System Model can be used to generate a list of requirements for each area
  - A new set of more specific Requirements Documents

29

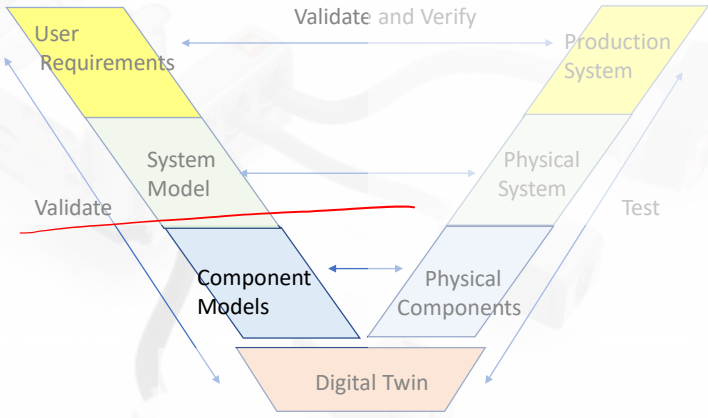
Increasing the Detail

- The System Model provides an overview of the proposed solution to meet the Requirements
- Once it has been verified, the subsystems can be defined
- The requirements of each subsystem can be specified
- Each subsystem can be implemented a set of components
- The requirements for each component can be specified
- *Now – and only NOW – each component can be designed and verified (tested)*

30

Identifying Possible Design Implementations

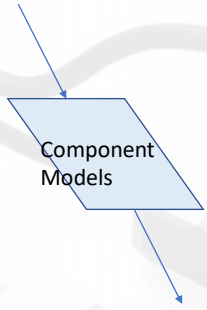
- Now that there are Requirements on the Components of the System
  - Can we start with a conceptual design of the components?
  - Do we have all the inputs?



31

Identifying Possible Design Implementations

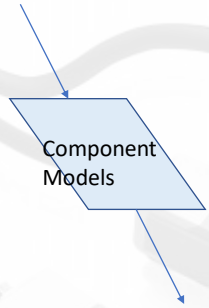
- What are the inputs to this process?
- First – what is it we want to achieve?
  - *Possible implementations of components to satisfy the System Model*
- Inputs
  - Requirements
  - Team Capabilities
  - Anything Else?



32

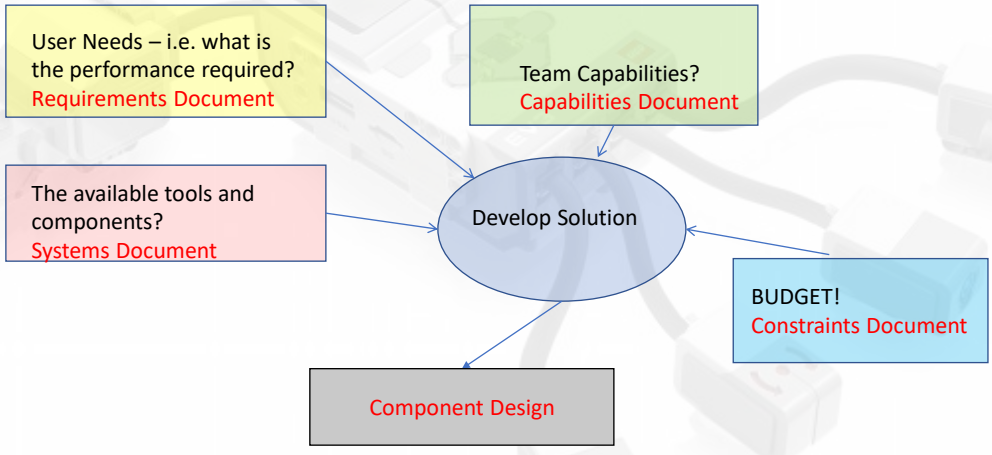
Identifying Possible Design Implementations

- These will be Physical Structures (mechanical and software)
- The conceptual design will need two more inputs…
- Input 1:
  - What can the solutions be constructed from?
    - Lego Mindstorms components
    - Java/Python based code
    - …
- Input 2:
  - Are there constraints (other than those listed already)?
    - BUDGET!
    - Delivery date
    - …



33

Inputs to the Component Conceptual Design



34

Inputs to the Component Conceptual Design

- So
  - Two more documents need to be created….
  - 1. The Systems Document
    - Identify the capabilities needed to construct the system defined in the System Model
    - Determine the tools available
      - Software and Hardware
    - Identify the basic building blocks that can be used
    - Determine if existing components (e.g. from the labs) can be re-used or re-purposed

35

Inputs to the Component Conceptual Design

- So
  - Two more documents need to be created….
  - 2. The Constraints Document
    - Is there a budget involved with the project?
    - Is there a delivery deadline?
    - Can components outside of those provided directly (e.g. Lego) be used?
    - Are there limitations on tools such as shared files, authoring systems, etc., that can be used?
    - …
- As before, these documents must have the standard header block and then content related to their purpose.

36

Summary

- We have:
  - Reviewed the concept of a System Model and how it is a critical first step in the EDP
  - Considered the System Model generation process and the required inputs
  - Discussed how the capabilities of the Design Team can affect both the EDP and the process outcomes
  - Stepped through the process for the generation of a System Model
  - Considered the inputs needed to enable to creation of basic component designs
  - Described 4 main documents which form the inputs to the EDP for DPM

37

Questions?

38