ECSE-211
Design Principles and Methods

Lecture 6: Ideas, Components and Models

Date: 30 January 2022

1

Questions from Lecture 5 (Requirements to System Model)

- Question 1

- What is the reason for generating a System Model?

- need a blueprint to build something –
- test and validate ideas –
- interactions between functional blocks –

2

Question 2

- What are the inputs to the creation of the System Model?

1. Requirements
2. Constraints
3. Team Capabilities

3

Question 3

- Why are the capabilities of the Design Team important?

because they constrain the solution

4

Question 4

- What kind of information can be included in the Capabiities Document for DPM?

experience? — level in program
elec/comp/soft
courses taken

availability

5

Contents

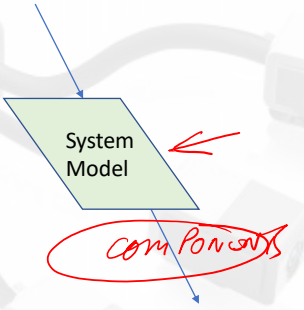
possible

- Identifying Design Implementations
- Review of the Documents
- Developing from the System Model
- The Concepts of a Design Space, Tolerances and Tradeoffs
- Identifying Possible Design Implementations
- Evaluating Proposed Designs

6

Identifying Possible Design Implementations

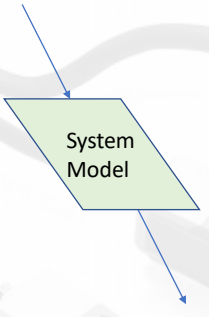
- What are the inputs to this process?
- First – what is it we want to achieve?
  - *Possible implementations of components to satisfy the System Model*
- Inputs
  - Requirements
  - Team Capabilities
  - Anything Else?



7

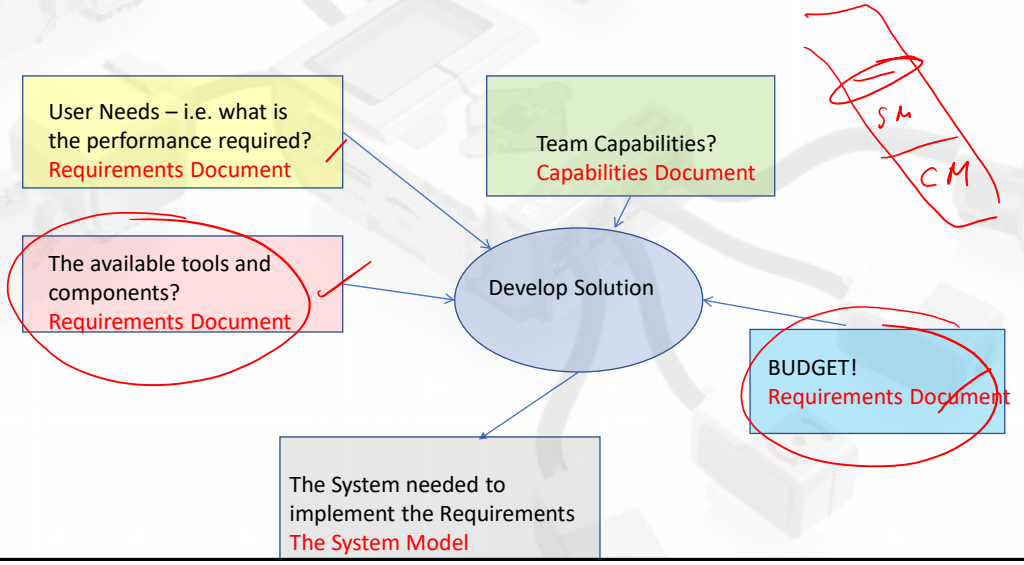
Identifying Possible Design Implementations

- 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
    - Python or Java based code
    - …
- Input 2:
  - Are there constraints (other than those listed already)?
    - BUDGET!
    - Delivery date
    - …



8

Inputs to the Solution Development



9

Inputs to the System Model

- So
  - Two more sections for the Requirements Document need to be created….
  - 1. The Systems Availability
    - 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

10

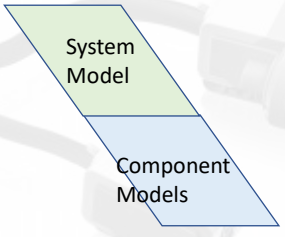
Inputs to the System Model

- So
  - Two more sections for the Requirements Document need to be created….
  - 2. The Constraints Section
    - 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?
    - …

11

Outputs from the System Model

- The system model has two functions
  - Translate the Requirements (inputs) into a possible system which meets the requirements
  - Output (describe) the list of components needed to implement the functions in the System Model – together with the function of each component
- Additionally, the System Model shows the interactions between components and this is critical in the final implementation
  - The interactions will happen during system (component) integration and can lead to failures



12

Information from the System Model

- The System Model
  - It is usually a set of figures – easier to understand than a set of words..
  - In a sense similar to architect's plans
    - These are usually drawings NOT a textual description
  - It could comprise any, or all, of the following
    - A state diagram
    - A system hierarchy
    - A relationship between hardware components
    - A flow chart for data indicating the software functional blocks
    - …

13

Information from the System Model, etc.

- The System Model describes
  - What the system must do
  - The components needed to achieve the goals
- The development of physical components is achieved by mapping the system model onto the materials and tools provided
- So
  - The solution is limited by the components in the Lego system
  - The solution may be constrained by the processing ability of the brick
  - …

14

Information from the System Model, etc.

- The Requirements Document and the System Model may also provide extra information:
  - e.g. tolerances that can be accepted
- But… what is a "tolerance"?
- A Requirement may state:
  - The output of a particular component must be X +/– δ
  - For example:
    - A robot is navigating to cross the bridge – the bridge is 20cm wide, the potential robot design is 18cm wide
    - To be acceptable, the navigation function of the robot MUST get the robot to the center line of the bridge + or – 1 cm.
- δ (or +/– 1 cm) is the tolerance (acceptable error) on the performance of the navigation

15

Errors in Component Operation

- All physical components have some inherent error or randomness in their transfer function…
  - So, for a given set of inputs (which may include errors), the outputs of a component will have a nominal value but also a random variation
- Consider a resistor
  - The manufacturing process generates nominally 10 ohm resistors but the value could range from 9.95 to 10.05 ohms
  - The input supply is nominally at 10 volts but it could be in a range from 9.9 to 10.1 volts
  - What is the output current?
- So – the model is Ohm's Law and one way to estimate the output is to look at the extremes

16

Errors in Component Operation

- Ohm's Law states:

$$V = IR \text{ or } I = V/R$$

- So – maximum current occurs when V is max and R is min:

$$I = \frac{10.1}{9.95} = 1.015\ Amps$$

- Minimum current occurs when V is min and R is max:

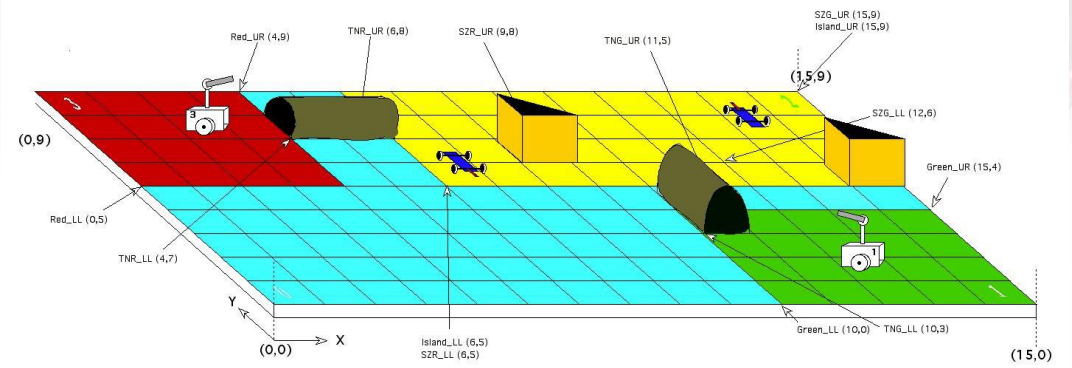
$$I = \frac{9.9}{10.05} = 0.985\ Amps$$

- So – the errors in input and the component itself result in an error in the output – *Can this be acceptable?*
- *If the tolerance on the current was set to nominal +/- 1%, then it isn't…*

17

A Typical Project Description

- The game is played in a constrained space:

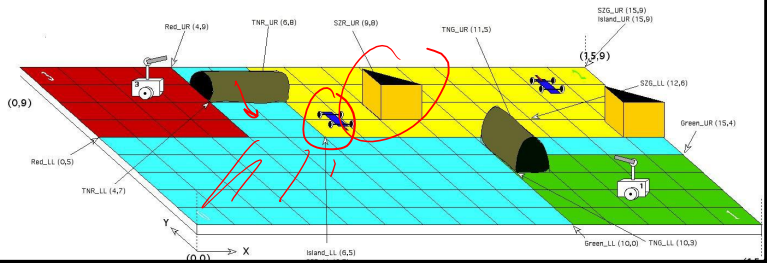


Let's look at design solutions

18

A Typical Project Description

- The colored areas do not exist – they are defined by coordinates
- The lines do exist and can be used for navigation
- The tunnel positions move and are defined by coordinates
- The obstacles exist but no coordinates are given
- The location of the breakdown is given



19

The Design Space

- The Requirements, etc., define a set of design parameters
- For example, consider the requirement for a navigation function
- For example, the need to move around is most easily implemented by using wheels but in what configuration? How many?
- You could have a 2 wheeled system
  - Minimum motors
  - But ability to move an object is limited by friction – leads to wheel spin
- You could have a 4 wheeled system
  - More motors
  - But increased friction – less wheel spin – can move larger loads
- You could use a tracked system
  - More friction
  - But turning is more difficult

20

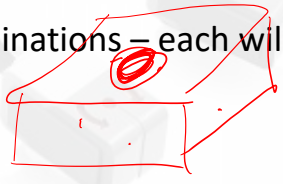
The Design Space

- So – the "wheel" parameter can have several values – all with positive and negative aspects
- Similarly, how many sensors might be used?
  - Two colour sensors might make line detection more accurate but the software is slower
  - Two ultrasonics might enable improved object detection
- So – there are multiple variants on the sensor types, numbers and positions – there could be 4 or 8 variations…
- Combining wheels and sensors gives a choice of maybe 24 different systems all of which might have the potential of meeting a navigation requirement…

21

The Design Space

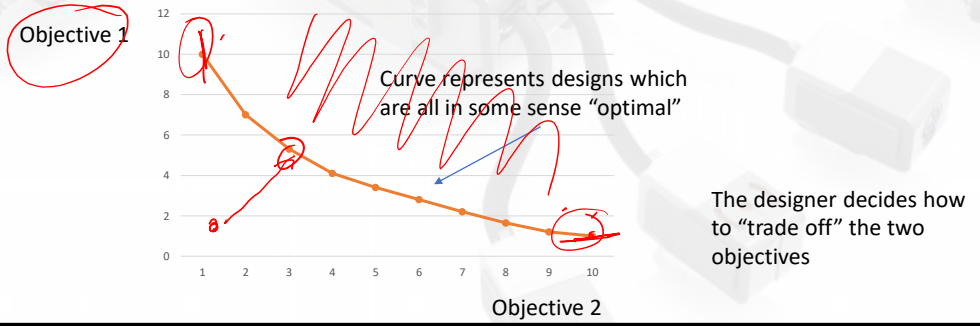
- Each of the navigation options might be combined with several possible chassis designs…
- So each parameter for the design of a component or system has multiple values
- The design space is the set of all valid combinations of the parameters
  - Note – some combinations may not be feasible
  - For example, for a given chassis structure, it may not be possible to have 4 wheels…
- However, there will be a large number of combinations – each will have positive and negative aspects
- No single solution will be "optimal"



22

Tradeoffs

- In a real design problem, there are usually several performance objectives and one design may be great at one objective but weaker at another
- In technical terms, this is referred to as “Pareto Optimization”



The designer decides how to “trade off” the two objectives

23

Tradeoffs

- Tradeoffs exist everywhere
  - In the implementation of the System
  - In the choice of the components
  - In the implementation of software functions
  - For example – there may be two possible algorithms for implementing a particular function in navigation
    - The first is fast, needs one sensor but has a potential error in output due to the filtering needed
    - The second uses two sensors, the filtering is easier, the errors are reduced but it is slow
    - So – the tradeoff is between execution speed and accuracy of result..

24

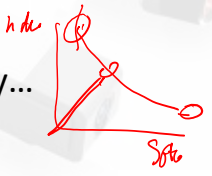
The Design Space, Tolerances and Tradeoffs

- Designing components is a phase of the EDP sometimes referred to as "Idea Generation"
- While the System Model may be reasonably well defined, its implementation is not unique
- It will depend on the combination of subsystems
- Each subsystem has many possible forms and benefits
- The "Ideas" are constrained by the Design Space, the Tolerances on the performance and the Tradeoffs
- Depending on the Team Capabilities – different solutions might be chosen

25

The Design Space, Tolerances and Tradeoffs

- A Team which is strong in software but weak in hardware might choose a minimal hardware system and try to address issues through complex software
- A Team which is strong in hardware but limited in its software abilities might choose a sophisticated hardware system and simplify the software requirements
- Both solutions might address the design problem effectively…



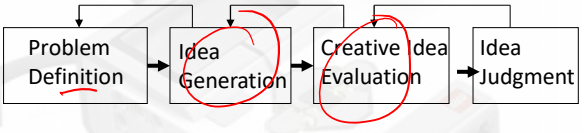
26

Multiple Design Choices

- The size of the design space implies there are many possibilities for solving the design problem
  - At the component level
  - At the system level
- It is not always clear at this stage of the design process which possibility is "optimal"
  - The Team is working with a high level set of parameters
  - Some of the designs may not work out due to issues that were not seen initially (in effect, dimensions missing from the design space)
- So – at the early stage, about 3 different designs for the system need to be developed
  - These should be significantly different – e.g. a 4 wheel system, a 2 wheel system and a tracked vehicle..

27

Summarizing the Process



- The "Problem Definition" comes from the various Requirements
- Ideas lead to several possible candidates for subproblem solutions and the complete system..
- We want a solution – delivered on time and on budget
  - which candidate idea gives us the best chance of achieving it?
- Look at the trade-offs
  - E.g.
    - Solution looks elegant but too many unknowns
    - Solution could perform the best but a time estimate suggests it cannot be done for the delivery deadline
    - Solution will need more software than can be built within the budget
    - …
- Choose a potential winner…

28

The Rationale for more than One Candidate

- Having multiple possible choices is the basis of “Set Based Design”
- With 3 potential candidates, each can be evaluated against the System Model and the Requirements Document
- The evaluation should include estimates of the “best chance of success” in the EDP
- One candidate is then chosen, both the system and the components, for further development
- NOTE – this process and the decisions together with the rationale for the decisions and any test results MUST be DOCUMENTED

29

Questions?

30