Teaching Systems Engineering to Undergraduates; Experiences and Considerations

by Gerrit Muller  Buskerud University College and Embedded Systems Institute
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract
Undergraduates need a teaching style that fits their (lack of) experience. Especially in systems engineering this is an issue, since systems engineering connects to so many different stakeholders with so many different concerns. Students need to get aware of the inherent ambiguities, uncertainties and unknowns in the systems world, in contrast to the focused world of mono-disciplinary engineering.
Well-defined problems

Well-defined methods, techniques, formalisms

Questions have the right answer

Dominant teaching style: knowledge transfer and skill training

Subject of interest is technical

strength of bridge

Bode plots in control

bridge will sustain upto x ton

control is converging and stable

no unpredictable humans

no inconceivable nature
Goals of Teaching SE at Undergraduate Level

to make students aware of
+ the impact of working in a large organization, e.g.
    processes, organizations, roles, responsibilities, economy and financials
+ the communication challenges between:
    various technical disciplines
    various less technical stakeholders
+ the ill-defined and multi-dimensional nature of system problems
    uncertainties, unknowns, ambiguities, dynamics, conflicting needs and goals
+ the impact of external conditions on the system and its design
    human behavior, natural phenomena
+ system life cycle

to provide insight in available methods, techniques, and concepts
The Context that a Student will Enter

- **tools**
- **procedures**
- **processes**
- **colleagues**
- **managers**
- **time**
- **pressure**
- **politics**
- **components**
- **organization**
- **documentation**
- **cost**
- **pressure**
- **project leader**
- **BoM**
- **WBS**
- **schedules**
- **sales price**
- **margin**
- **cost**
- **Quality ass.**
- **logistics**
- **production**
- **sales**
- **service**
- **customers**
- **users**
- **legacy**
- **installed base**
- **problems**
- **changes**
- **process/organization**
- **systems engineering**
- **project**
- **finance**
- **business functions**
- **customer+life cycle**
- **"hard" technology**

Teaching Systems Engineering to Undergraduates

version: 0.1
March 6, 2013
IMWEcontextExperience
The Challenge of Teaching Systems Engineering

Systems Engineering processes, methods, and techniques have a high "common sense" level.

Without experience they can easily be perceived as open doors.

How to get inexperienced students in a perceptive mode, such that they appreciate the subject matter?
Role Play as Teaching Paradigm

**management team**
- process
- organization
- people
- business, finance

**design team**
- customer understanding
- requirement specification
- concept selection
- partitioning, interfaces, functionality
- qualities, e.g. performance, cost, reliability
- technology choices

**teams of 3 to 5 students per team**

**guided process:**
- some theory
- apply in role play
- brief reporting and discussion

**case that relates to their knowledge**
A time-box is a fixed amount of time allocated to perform one activity.

We iterate many times over different viewpoints. Every viewpoint is addressed multiple times with new insights from other viewpoints.
Didactic Model; Homework

- **class room** -> flip charts -> homework -> ppt
- **class room** -> old ppt flip charts -> homework -> new ppt

1/2 day -> few days or weeks -> 1/2 day -> time

This didactic model is very intense. Students are exhausted after 1/2 day.

**class room**

Small steps on flip charts (or paper)

**homework**

consolidate results in PowerPoint or Visio
Case Requirements and Example

*case requirements*

multi-disciplinary aspects
original discipline of students should be clearly present
students must have some affinity with the application
open definition: unclear problem, large solution space

**Example: Tree Cutting Robot for mechanical engineering students**

*mechanics is dominant*
*vision and control require*
*electronics and software*

background:
Less young people are willing to work in the wild and mountainous areas in Norway, Canada, or USA to cut trees for wood production.

product:
Robot that supports the cutting and processing of trees so that less people are needed.
Example Designs of Tree Cutting Robot

[Diagram of tree cutting robot]

Cutter/de-brancher
Flexible arm
Cockpit
Independent terrain wheels x6

[Diagram showing parts of the robot]

[Diagram showing the interaction with a tree]

[Diagram showing the de-barker and cutter]
Example Stories of Apple Plucking Robot
Class Room Fills itselfs with Flip Charts

Teaching Systems Engineering to Undergraduates

version: 0.1
March 6, 2013
TSEUexampleClassRoom
Material for Design is Based on CAFCR+

---

**What** does Customer need in Product and **Why**?

- **Customer**
  - What
  - Customer Objectives
- **Application**
- **Product**
  - What
  - Functional
  - Conceptual
  - Realization

---

- drives, justifies, needs
- enables, supports

---

Teaching Systems Engineering to Undergraduates

version: 0.1
March 6, 2013
CAFCRannotated
CAFCR+ Model

Customer objectives
Application
Functional
Conceptual
Realization

Life cycle
operations
maintenance
upgrades

development
manufacturing
installation

sales, service, logistics, production, R&D
Steps for First 2 Sessions

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
Steps for Second 2 Sessions

**Customer objectives**

1. make core spec
2. why are these specifications needed
3. describe usage
4. make key driver graph
5. make story
6. make use case(s)
7. analyze design impact
8. assess story based on 5 story telling criteria
9. improve story
10. improve key driver graph
11. make cost of ownership model
12. explore alternative designs
13. update specification
14. make draft management presentation

**Application**

**Functional**

**Conceptual**

**Realization**

Teaching Systems Engineering to Undergraduates

version: 0.1
March 6, 2013
BSDCexercisesDay2

Gerrit Muller
Last Session

1. Customer objectives
2. Application
3. Functional
4. Conceptual
5. Realization
6. Life cycle

1. Specify life time
2. Draw dev. life cycle
3. Describe logistics and manufacturing
4. Describe installation and acceptance
5. Describe maintenance
6. Update specification
7. Analyze design impact
# Summary of all Steps

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
<th>Life cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. select case to work on</td>
<td>2. discuss possible solutions</td>
<td>3. discuss specification</td>
<td>4. make design</td>
<td>5. make construction decomposition</td>
<td>6. make functional design</td>
</tr>
<tr>
<td>7. make presentation of specification and design</td>
<td>8. make second and third design</td>
<td>9. compare three designs</td>
<td>10. make list of design criteria</td>
<td>11. make list of design choices</td>
<td></td>
</tr>
<tr>
<td>12 update specification</td>
<td>13 define performance use case</td>
<td>14 specify performance</td>
<td>1 make core spec</td>
<td>15 make performance model</td>
<td></td>
</tr>
<tr>
<td>2. why are these specifications needed</td>
<td>3. describe usage</td>
<td>4. make key driver graph</td>
<td>5. make story</td>
<td>6 make use case(s)</td>
<td></td>
</tr>
<tr>
<td>7. analyze design impact</td>
<td>8 assess story based on 5 story telling criteria</td>
<td>9 improve story</td>
<td>10. improve key driver graph</td>
<td>11 make cost of ownership model</td>
<td></td>
</tr>
<tr>
<td>13 update specification</td>
<td>12 explore alternative designs</td>
<td>14 make draft management presentation</td>
<td>1 specify life time</td>
<td></td>
<td>2 draw dev. life cycle</td>
</tr>
<tr>
<td>3 describe logistics and manufacturing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 describe installation and acceptance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 describe maintenance</td>
<td>6 update specification</td>
<td>7. analyze design impact</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Teaching Systems Engineering to Undergraduates**

Gerrit Muller

version: 0.1
March 6, 2013
BSDCsummary
Stretching from Comfort Zone into Unknowns

Customer objectives

Application

Functional

Conceptual

Realization

Life cycle

1. select case to work on

2. why are these specifications needed

3. describe usage

4. make key driver graph

5. make story

6. analyze design impact

7. assess story based on

8. improve story

9. improve key driver graph

10. explore alternative designs

11. update specification

12. define performance use case

13. make performance model

14. specify performance

1. specify life time

2. start in well known territory

3. draw dev. life cycle

4. multi-disciplinary

5. multiple concepts

6. keep iterating

7. keep iterating

8. keep iterating

9. keep iterating

10. keep iterating

11. keep iterating

12. keep iterating
Guides the students through a journey.

Stretches students one step at a time.

Regularly forces students out of their comfort zone.

Provides feedback on their intermediate deliverables.

Helps students to reflect on their experience.

Provides theory JIT (Just In Time: appreciation and application).

Illustrates theory with examples from practice.

Keeps the pace high.

Initiates frequent breaks (this approach costs lots of mental energy).

Unfreezes students: let them sketch, stimulate creativity and imagination.
### Experiences of Teaching in this Way

<table>
<thead>
<tr>
<th>Time-boxes can vary from 5 to 20 minutes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sometimes a few steps have to be skipped.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Too funny concepts or stories distract.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teams that get stuck in unrealistic proposal.</td>
</tr>
<tr>
<td>Students that miss sessions; participation is mandatory.</td>
</tr>
<tr>
<td>Teams that stick to the initial solution</td>
</tr>
</tbody>
</table>

The sessions can be a lot of fun for students and teacher.
Some interesting concepts pop-up.
The increase of awareness can be observed.
Some nice visualizations or animations are shown.
Links

Course slides:

http://www.gaudisite.nl/BachelorSDallSlides.pdf

Background CAFCR model:


Short introduction course in Systems engineering:

http://www.gaudisite.nl/ShortIntroCourseSESlides.pdf