TERMA
SPACE DIVISION

# SYSTEM ENGINEERING WORKSHOP

## How to come to a design in a systematical way

## Application:
## an Autonomous Snail Mail Delivery System

Ron Noteborn, MSc.
TERMA A/S, Birkerød
ron@terma.com

# WHAT IS THIS WORKSHOP ABOUT?

- **Common Design Problem:**
  - run for a design,
  - but design did not do what it was supposed to do
  - discovered late, too late for conceptual change

- **Some Observations:**
  - designer goes for first idea he gets
  - rejecting designs in group discussions
  - patching of designs to cover up for non compliances
  - unnecessary constraints from superiors or customers

# WHAT IS THIS WORKSHOP ABOUT? (continued)

- **Example: Satellite Structural Design**
  - suppose structures engineer started on a design much too early
  - he/she is buried in detailed analyses (FEM, thermal etc.)
  - design focussed around specialty of this engineer
  - users of satellite and other subsystems start to complain

- **Root Cause**
  - engineer had not looked into users needs sufficiently
  - engineer had not considered other options sufficiently

→ Need for Systematic Design

# WHAT IS THIS WORKSHOP ABOUT? (continued)

- **In this workshop**:
  - a three step approach to a systematic design process
  - not unique, not solving all problems, but a possible tool for the engineer

- **Origin of the tools**:
  - papers/lectures of other industries (Fokker Space, Alenia)
  - my own application in every day practice in space engineering work
  - not unique to aerospace:
    - industrial designers are used to requirements, alternatives and trade-offs (coffee-machines, vacuum cleaners : not quite a satellite!)
    - managers designing human processes in factories or maintenance stations

TERMA
SPACE DIVISION

# GOAL OF THE WORKSHOP

- **Goal:**
  - experimenting with a systematic way of working
  - find out for yourself what you can do with it
  - discover that by starting at the user and working your way up to design, you get a design with a good foundation

- **Working in groups:**
  - each group goes to the same process
  - will the results be different?
  - cross fertilization during breaks ?

# STRUCTURE OF THE WORKSHOP

- **Introduction**
- **3 Sessions:**
  - Explanation
  - Assignment
  - Student Application
- **Discussion**
- **Conclusion**

# APPLICATION:
# Autonomous Snail Mail Delivery System

- **The application:**
  - a system that distributes mail (packets and letters) in a big office building
  - system picks up the mail at the front door when it arrives
  - and then *autonomously* does it job

- **The idea:**
  - you get some *Requirements* and *Constraints* from a customer
  - your task: come up with a concept that can do this job (delivering mail)

- **On purpose:**
  - top level systems
  - easier to work with, more common sense

# SYSTEM REQUIREMENTS & CONSTRAINTS

- **Main Function**:
  - to deliver snail mail in a large office building to mail boxes
    - pick up mail at entrance of building
    - mail comes as letters and packets
    - mail delivery time at entrance uncertain
    - undeliverable mail to be collected centrally
- **Constraints:**
  - autonomous delivery (no personnel)
  - no major adaptations to building
  - shall not interfere with personnel
  - various floors in building
  - low cost
  - fast delivery of mail

- **Assume**:
  - mail contains identification codes
  - building has elevators, stairs
- **Focus on**:
  - how to get mail from the entrance into the right mail box
    - manipulating the mail
    - transporting it through the building

TERMA
SPACE DIVISION

# REQUIREMENTS

- **First half of the workshop concentrates on *Requirements*.**
  - We'll get to the design later…

- **Requirements communicate:**
  - What should it do?
  - How well should it do that?
  - What kind of interfaces are there?

- **Real Requirements are Design Independent!**
  - They do not specify what the system design shall *be*, *but*
  - what that design shall *accomplish*

# SESSION 1: Functional Analysis

- **Example: magnetometer and science instrument**
  - "how should I make the requirements, I don't know the design ?!"
  - "I can't come up with specs until I have designed the whole thing!"

- **Difficult to start NOT designing right away**
  - Designing is fun!
  - What are those requirements about anyway?

- **How to get then to Requirements?**
- **Solution is the functional analysis:**
  - specify the functions of your system
  - top-down approach
  - design independent

TERMA
SPACE DIVISION

# SESSION 1: Functional Analysis (continued)

- **Starting Point**
  - main functional description of the system
    - “transport three people into low earth orbit”
    - “measure the magnetic field”
    - “communicate with system operator and report back”

- **Split the functional description out in sub-functions**
  - Functional Decomposition
  - AND tree diagram: each block is composed of all the branches below it

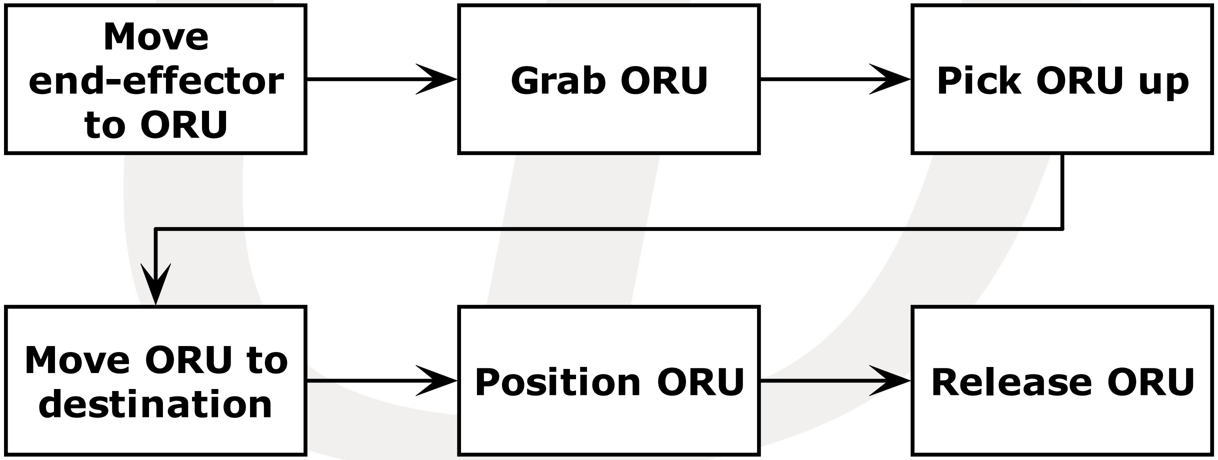
- **Try to be complete!**
  - difficult
  - try functional flow diagrams to identify new functions

# SESSION 1: Functional Analysis (continued)

## Example of a Functional Flow Diagram

Space Robot Manipulator

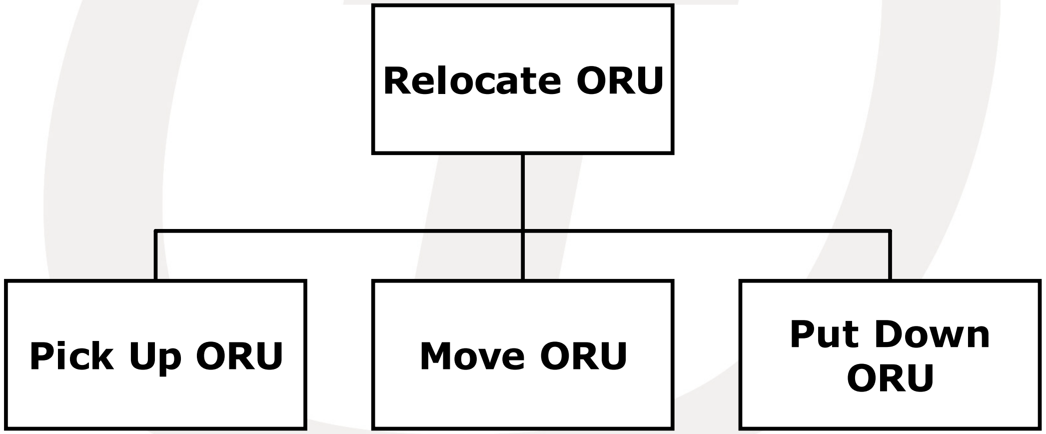
Main Function: Relocate an ORU



# SESSION 1: Functional Analysis (continued)

## Example of a Functional Tree

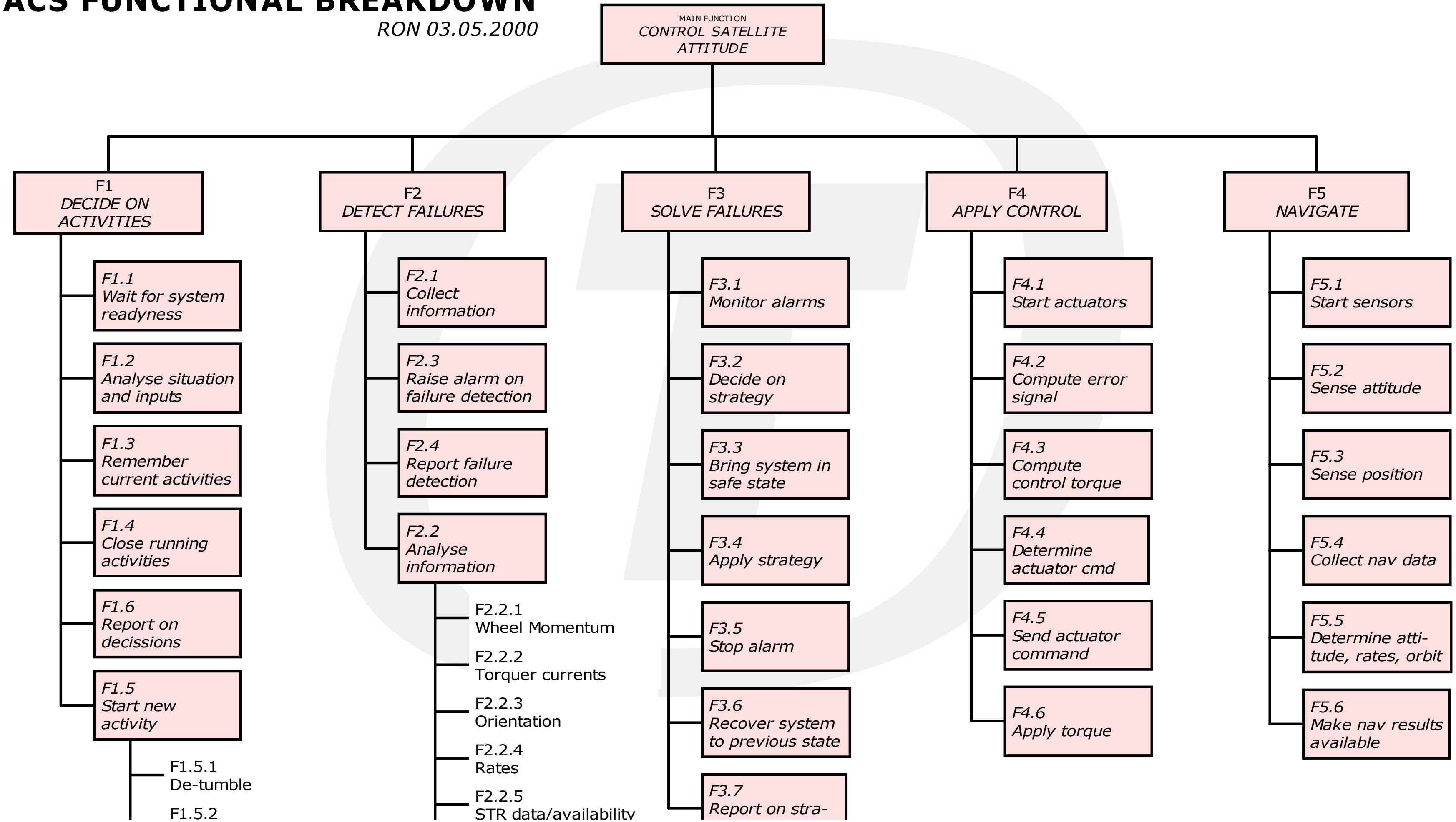
Space Robot Manipulator



# SESSION 1: Functional Analysis (continued)

## mACS FUNCTIONAL BREAKDOWN

*RON 03.05.2000*



TERMA
SPACE DIVISION

# SESSION 1: Requirements Derivation

- **Functional Analysis has given us the Step Up to Requirements Specification**

- **Common Mistakes**
  - making requirements after designing the system
  - making requirements by just writing down whatever comes up in your head
  - making design specific requirements

- **Function of the Spec is to:**
  - drive the design (not to define it)
  - have a set of rules to see what the system should do
  - have a set of test objectives for validation

- **Use the Functional Tree as an input for Requirements Specification**

TERMA
SPACE DIVISION

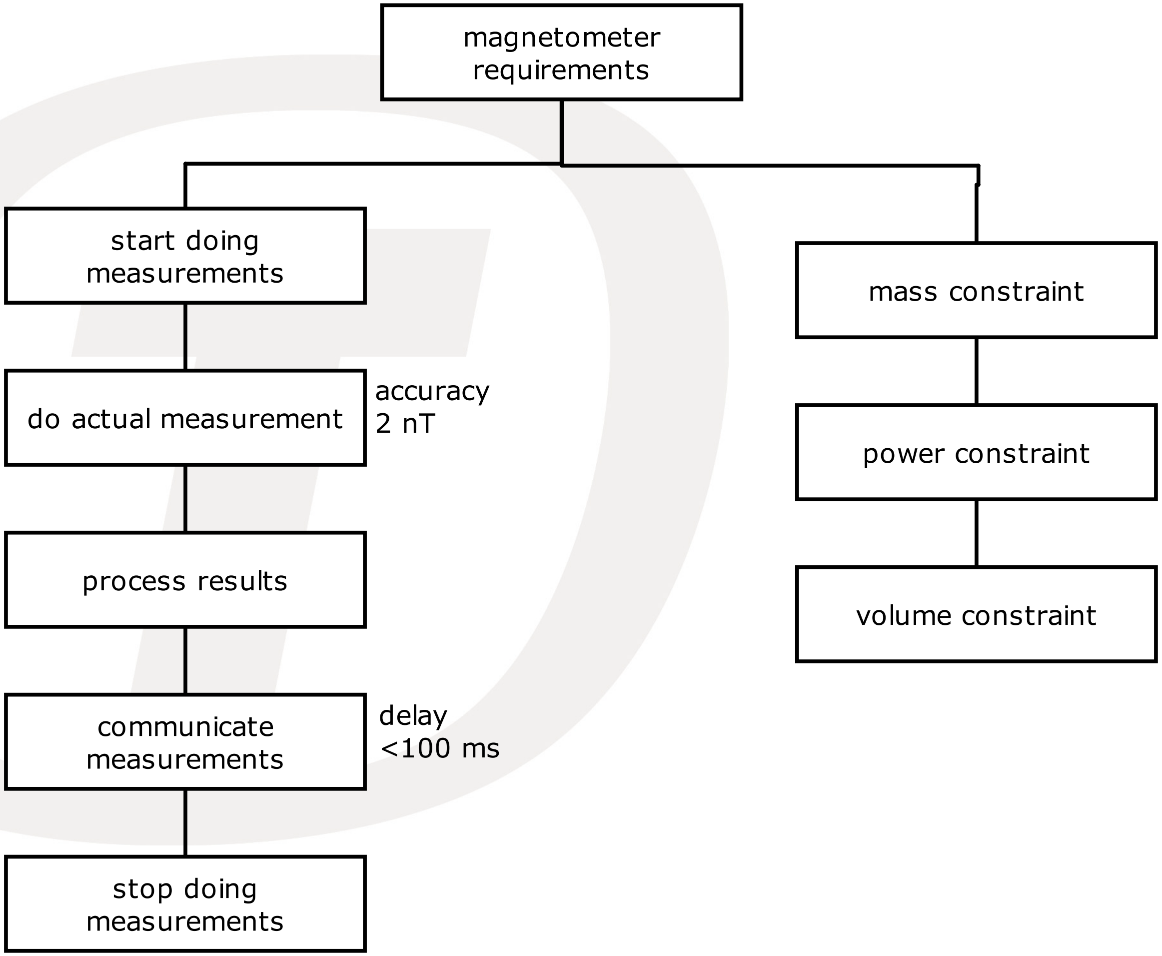
# SESSION 1: Requirements Derivation (continued)

- **Tool: Requirements Discovery Tree**
  - this is the functional tree augmented with
    - constraints
    - performance specs

- **Example: Magnetometer**
  - initial requirement only specified the measurement performance
  - with a functional tree, all sorts of things could have been specified

# SESSION 1: Requirements Derivation (continued)

- **Example: Magnetometer**
  - Limited Tree
  - Requirements for enabling and disabling measurements have been identified
  - A delay in communication is introduced
  - Functional requirements
  - Constraints on mass, power and volume



TERMA
SPACE DIVISION

# SESSION 1: Requirements Derivation (continued)

## Magnetometer Requirement Spec could now look like:

R1: The magnetometer shall measure the magnetic field in three axes.

R1.1: The accuracy of the field measurements shall be 2 nT RMS.

R2.1: The magnetometer shall only start measuring after it has been enabled.

R2.2: The magnetometer shall stop measuring after it has been disabled.

R3: The magnetometer shall perform the necessary pre-processing of the measurements.

R4: The magnetometer shall transmit the measurements through its interface.

# SESSION 1: Requirements Derivation (continued)

- **Information Sources:**
  - Functional Behavior comes from Functional Tree
  - Constraints come usually from above (customer!)
  - Performance Requirements:
    - customer specified
    - other subsystems
    - self derived

# SESSION 1: Assignment

- **Identify Main Function of your System**
  - Use this as the starting point for your functional tree

- **Create a functional Flow Diagram of your System**
  - concentrate on high level
  - work out lower levels later
  - time is often a good parameter to use for the flow

- **Create the Requirements Discovery Tree of your System**
  - derive functions from functional analysis
  - augment this with the constraints you can identify
  - give id numbers to each function
  - end product is the requirements discovery tree

# SESSION 2: Design Options

- **Often: designer has some idea of what his design will look like**
  - designers specialty
  - focussing on a specific aspect of the design because of good ideas in that area
  - customer or boss prescribes a certain solution

- **It is questionable though if you get the best design solution**

# SESSION 2: Design Options

- **Example: Small Satellite with electric propulsion**
  - aerospace company wanted to build a national communications satellite (LEO, smallsat, low cost)
  - company had a new business development: **electric propulsion**
    - needed a flight opportunity
  - satellite attitude control had to be done with electric thrusters,
    - according to management
  - by far not the best option,
    - led to a power driven design (large solar arrays)
    - large costs and risk
  - more applicable design had been **gravity gradient stabilisation**
    - simplicity, cheaper, less risk

TERMA
SPACE DIVISION

# SESSION 2: Design Options (continued)

- **Results of too early start with design**:
  - not all *user needs* are identified and thus not in the design
  - so, design needs patching, becoming less optimal
  - designer does not want to change concept because he is married to it by now
  - a change of concept costs lots of money and schedule delays
  - another design that was more optimal and better is available but was never discovered

- **Solution:**
  - come up with concurrent alternatives: **Design Options**
  - these have to be tested against requirements before adoption of a concept

# SESSION 2: Design Options (continued)

- **Example: Star Tracker Temperature Effects**
  - New contract required large temperature differences on Star Tracker
  - T has an effect on background signal of a CCD image of the sky
  - Software was to cope with this

  - Initially, it was difficult to find a solution

  - Application of the design option tree gave about 10 different solutions
  - Final selection was simpler than anyone could have imagined

# SESSION 2: Design Options (continued)

- **Developing different design options**
  - gives you the chance to sort out pros and cons of all options
  - gives you a head start for the review
    - when reviewer has alternative, it has probably come up in your design option tree already, and you know why it is less optimal!
  - You will be able to find simpler solutions than what you initially thought of
  - It structures the discussion

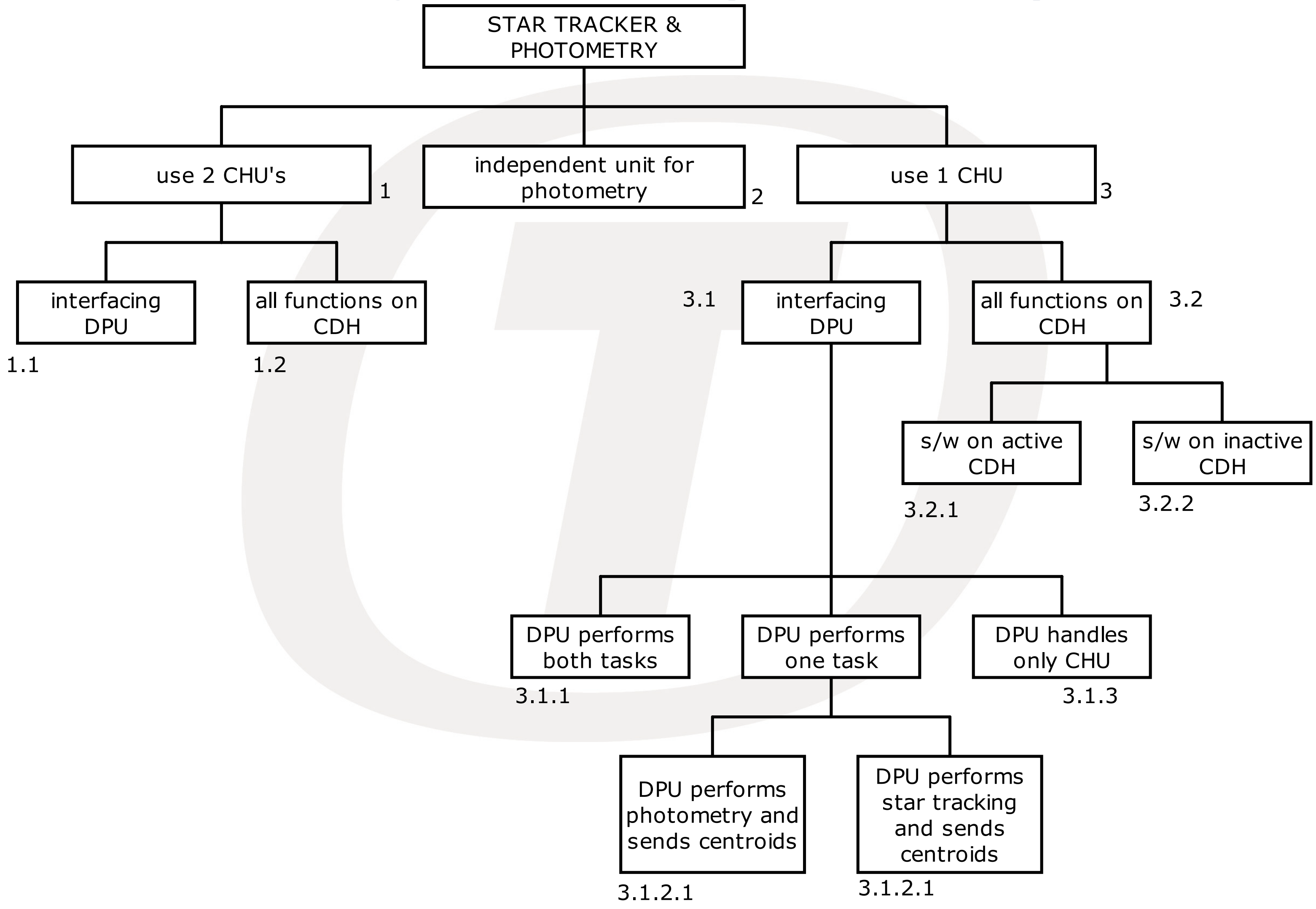
TERMA
SPACE DIVISION

# SESSION 2: Design Options (continued)

- **Design Option Tree**
  - yet another tree!
  - it lists different options for the design implementation
  - do not try to kick out strange or obviously wrong solutions
  - start out with a brain storming session
  - try to find common elements and structure the options in a tree

TERMA
SPACE DIVISION

# SESSION 2: Design Options (continued)



# SESSION 2: Assignment

- **Assignment:**
  - Create the Design Option Tree
  - Do a small brain storming session on design options
    - do not kick out out bad options
    - be creative, think about strange and impossible solutions
  - Give ID numbers to all end options
  - End product is the Design Option Tree

TERMA
SPACE DIVISION

# SESSION 3: Trade Off Map

- **Last step is the final selection of the Design**
  - We have different solutions
  - We also have requirements

- **Combine these in the Trade Off Map:**

| | Weight | Criterium 1 | Criterium 2 | Criterium 2 |
|---|---|---|---|---|
| Design 1 | | | | |
| Design 2 | | | | |
| Design 3 | | | | |
| Design 4 | | | | |

TERMA
SPACE DIVISION

# SESSION 3: Trade Off Map

- **What is the Trade Off Map?**
  - Select Trade Off Criteria from the requirements and constraints
  - List the Design Options horizontally
  - List Criteria vertically
  - Assign Weights to the Criteria
  - Fill in relative scores of each design option to each criterium
  - Total the scores for each design, with help of the weights
  - The best option will have the highest score
  - If no best option, add more criteria

# SESSION 3: Trade Off Map

- **Do not include ALL design options**
  - There are always clearly infeasible design options in the tree
  - Eliminate those
  - Options that you can not analyse, should be set aside.
- **Design Criteria**
  - Design criteria come from requirements and constraints
  - Constraints are most important

# SESSION 3: Trade Off Map (continued)

*Table 1: STR and Photometry Trade Map.*

| Criterium | W | 1.1 | 1.2 | 3.1.1 | 3.1.2.1 | 3.1.2.2 | 3.1.3 | 3.2.1 | 3.2.2 |
|---|---|---|---|---|---|---|---|---|---|
| Bus load | 3 | - | - | - | - - | - - - | - - - | - | - |
| Mass | 3 | - - | 0 | - - | - - | - - | - - | 0 | 0 |
| Power | 2 | - - - | - | - - | - - | - - | - - | 0 | - |
| Volume | 1 | - - | 0 | - - | - - | - - | - - | 0 | 0 |
| CDH Memory Load | 2 | - | - - | - | - | - - | - - | - - - | - - |
| CDH Processor Load | 3 | + | - | + | - | - - | - - | - - - | - |
| Extra H/W Work | 1 | - | - | - | - | - | - | - | - |
| Extra S/W Work | 2 | 0 | - | - | - | - - | - - | - - | - |
| DPU Processor Load | 1 | - | 0 | - - | - | - | + | 0 | 0 |
| ACS STR Redund. | 2 | + | + | - | - | - | - | - | + |
| | | -16 | -13 | -21 | -29 | -39 | -37 | -25 | -13 |

# SESSION 3: Assignment

- **Assignment:**
  - Select the Trade Off Criteria
  - Select the Weights
  - Select the Design Options (from Design Option Tree)

  - Create the Trade Off Map
  - Fill In the Trade Off Map

  - Select the best design

  - End product is the Trade Off Map with selected Design(s)

# CONCLUSIONS

- **Three steps to a design**
  - Functional Analysis and Requirements Derivation
  - Design Options
  - Trade Off Map

- **Gives a good idea about Requirements**

- **Gives better chances for Optimal Design**

- **Forces to think about Alternatives**

# CONCLUSIONS (continued

- **No guarantee for success**:
  - You can still mess it up!
  - Discipline is needed to have a chance

- **Use these tools as you please**
  - Their use is not always justified, sometimes it is a burden
  - Bend the tools to your own needs
  - Sometimes, only some of them apply
  - Think about what you do: don’t just follow the standard. You never know if it applies!