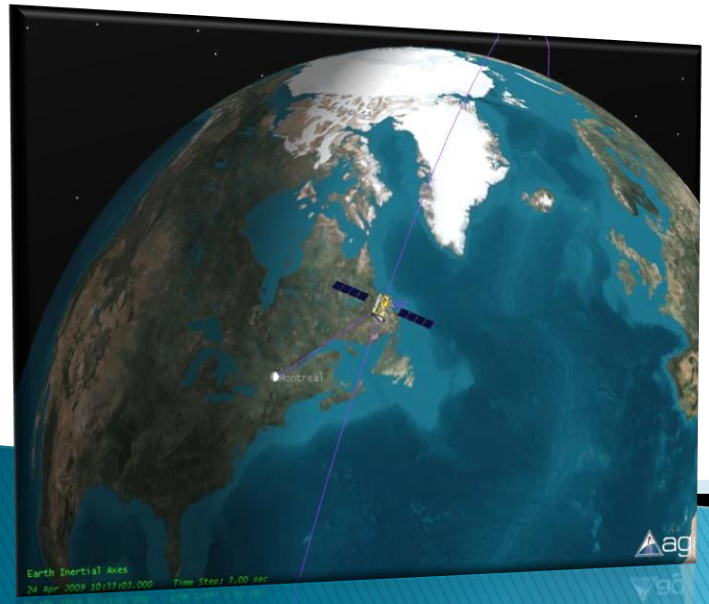
Modelica Library for Spacecraft Resource Budgeting



Niccolo Cymbalist
Chahe Adourian (supervisor)

Introduction

- Resource budgeting is one of the preliminary design phases in which system level requirements are specified.
  - Efficiently achieved using rapid prototyping
- We implemented the library in Modelica because it is well suited to hybrid, multidisciplinary modeling, it is modular and **easy to use**

Presentation Overview

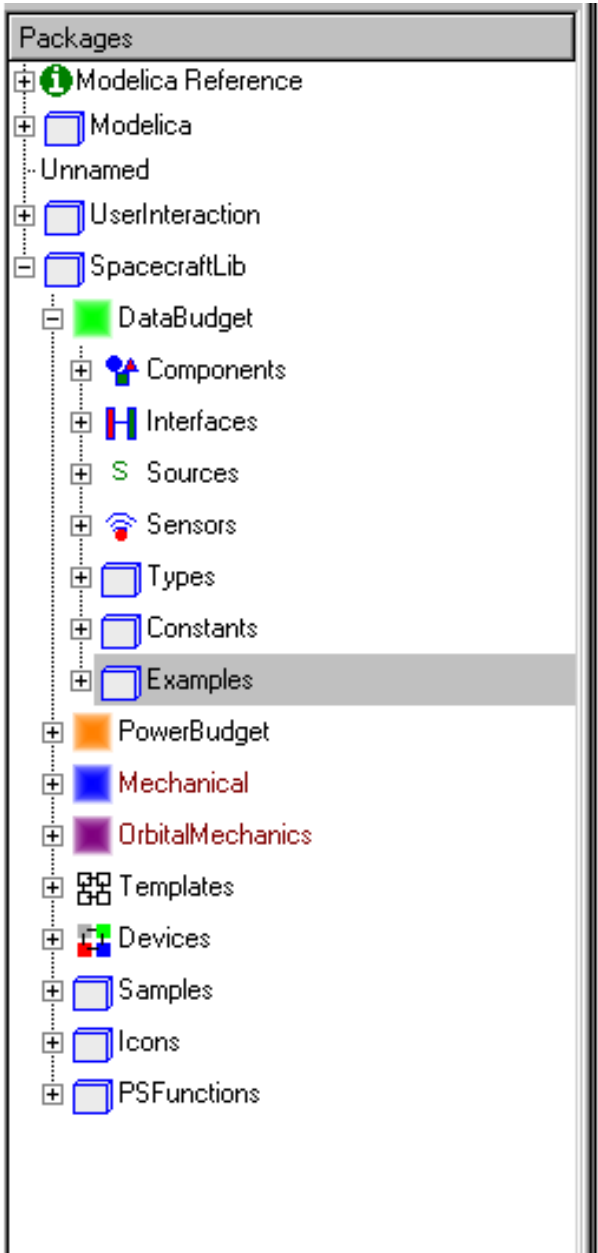
- Why?
  - I will demonstrate why this tool is needed and the advantages it offers
- What?
  - I will explain what the tool consists of
- How?
  - I will show how parts of this tool were built and examples of complete spacecraft built using this tool

Why?

- Rapid prototyping offers (some of) the advantages of a full simulator while remaining flexible and easy to use
  - Time dependant simulation
  - Accuracy
  - Ease of assembly, use and modification
- SpacecraftLib allows the user to easily customize the level of complexity of the model to suit the task at hand
- There are currently no widely available, easy to use rapid prototyping libraries for spacecraft

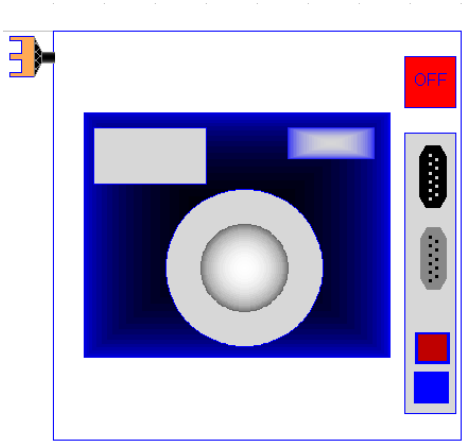
What? (1)

- Multidisciplinary tool, 4 sections dealing with:
  - Power
  - Command and data handling
  - Mechanics
  - Orbital mechanics
- The user can follow the interactions between the different subsystems



What? (2)

- *Device* concept. Multiple subsystems are modeled together in a *device* which:
  - Consumes power
  - Generates data
  - Responds to commands
  - Has mass and inertia
- The idea behind this was to model the physical device as opposed to the behaviour of the device



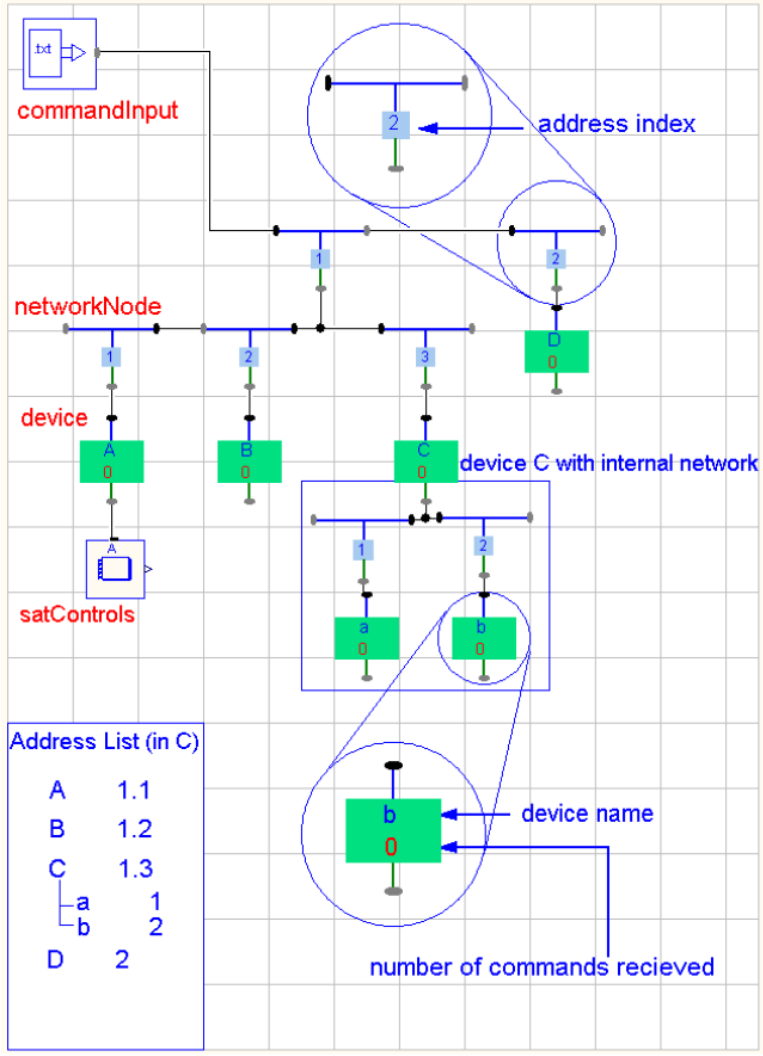
What? (3)

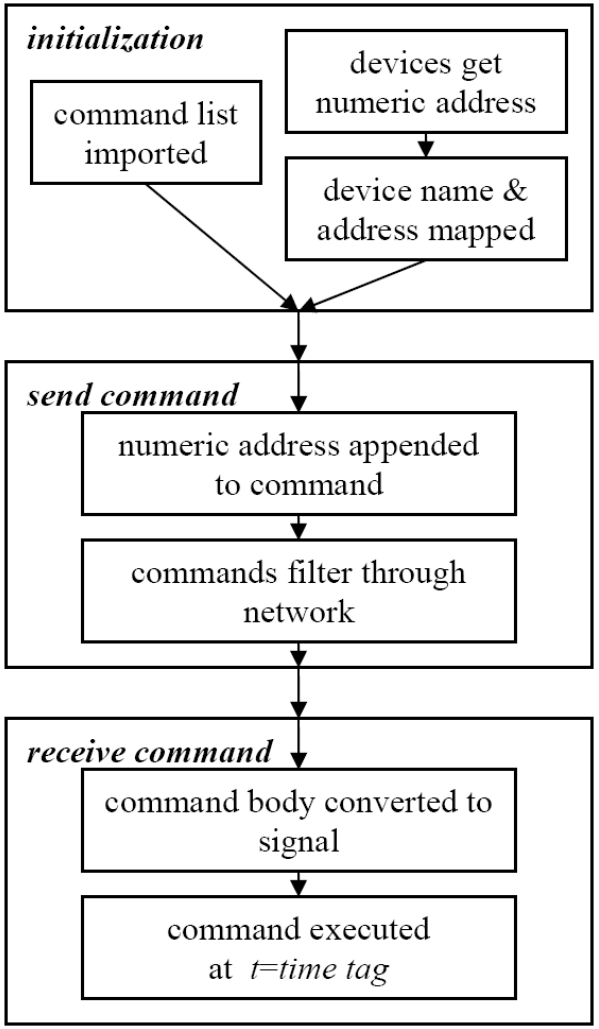
- Devices are combined into a complete spacecraft which:
  - Is initialized into an orbit
  - Can be commanded to take pictures, transmit data, change attitude and/or orbit etc…
  - Interacts with the ground station
- The spacecraft and GS models can be easily modified in order to test and compare different designs/specifications

How?

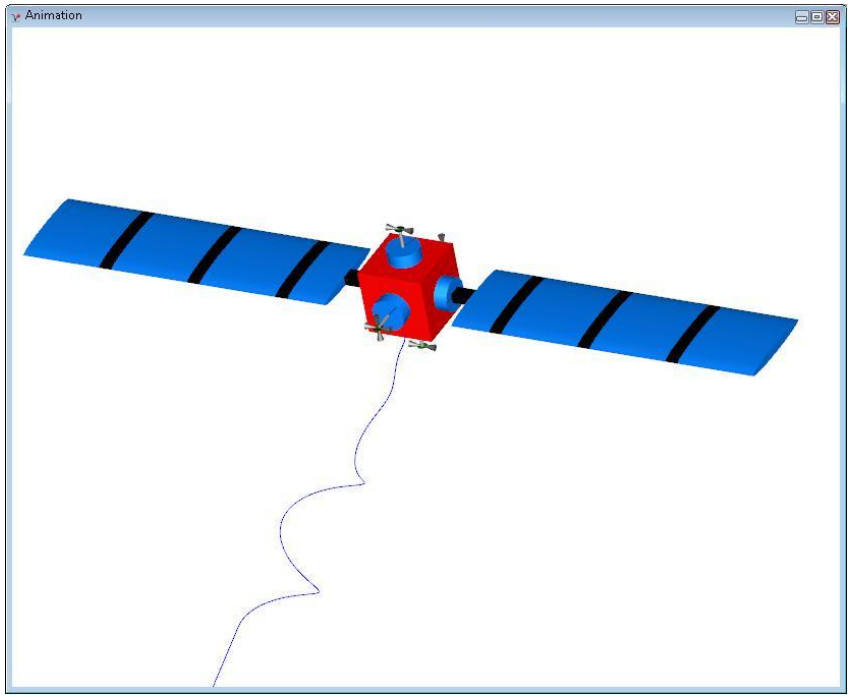
- DataBudget section
  - Data treated as a sort of physical quantity
  - Data may be generated, stored, compressed, transmitted
  - Bit rates, memory capacities are set by the user
- Command Network
  - ‘Plug and Play’ behavior
  - Built with a combination of C and Modelica
  - The user writes a list of time tagged commands in a text file which are executed during the simulation (we will upgrade this system to a GUI eventually)

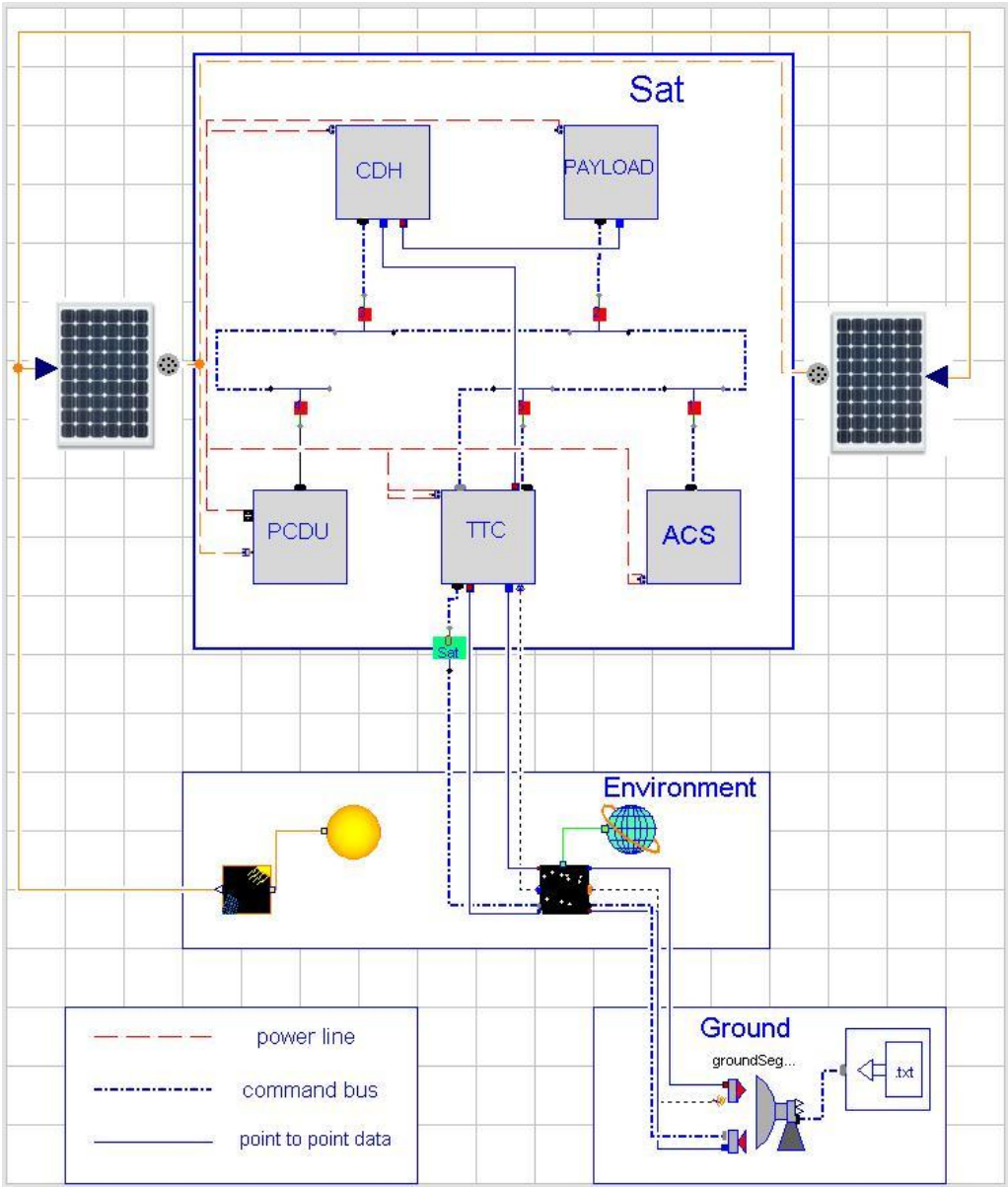
How? (command network)





Examples of spacecraft built using SpacecraftLib





Thank you for attending!