Nanospace, an open source tool to help concurrent engineering teaming in cubesat preliminary design

Thibault Gateau, Lucien Senaneuch

OSCW 2020 - December 11, 2020
Phase 0/A: Preliminary Design

Specialized skills
- Sub-systems
- Budgets
- Sub-systems architectures

Team work
- Communication
- Management

Eyesat synaptic view
Expectations vs reality

https://www.kerbalspaceprogram.com/ - January 2018
Expectations vs reality

Preliminary 1U design by Anton Poltoradnev - ISAE-Supaero
Software Tools and libraries

Candidate Software List

**Specification for Nanostar**

**LOS :**
- Stella

**Power budget :**
- IDM-CIC
- Home made scripts

**Structure :**
- IDM-CIC
- Catia [X]

**Dynamic database :**
- IDM-CIC (ECSS Standard)
- Valispace database
- Home made database

**Mission analysis :**
- GMAT
- Satorb
- DOCKS
- libs :
  - Celestlab
  - Poliastro
  - Orekit

**Visualization Tools :**
- Home made
- VTS
- Celestia
- Cosmographia

**Thermal :**
- Systema thermica
- Home made scripts

**Link Budget :**
- AMSAT
- Home made

**LOS :**
- Stella
Trendy way: Concurrent Design Engineering

Efficient data management and exchanges
- create data models
- shared common models
- data update should be propagated in each expert tool

Strategy?
- take an existing CDE?
- adapt from libs?
- adapt from MBSE?

[Di Domizio and Gaudenzi, 2008]
Review [Knoll et al., 2018]
Goal: Concurrent design engineering with some requirements

- Student challenges on cubesats preliminary design
- Multiple access from multiple sites (5 institutions)
- Database rather than spread-sheets [Gordon, 1999]
- Allow users to keep their favorite expert tools
- Student friendly & Open-source
NSS: Nanostar Software Suite

- **Dissipation Budget**
- **Structure**
  - Mass Budget
- **Mission Analysis**
- **Link Budget**
  - Dosa (AGPL v3)

- **Launcher Choice**
- **Data Budget**
- **Activity Profile**
- **Centralized Database**
  - Neo4j (GPL v3)

- **Power Budget**
  - Script Nanopower (AGPL v3)
    - Author: Guillaume Roux (ISAE-SUPAERO)
    - https://sourceforge.isae.fr/projects/nanopower

- **ADCS study**
  - Script NSS_ADCS (AGPL v3)
    - Author: Pedro Miguel De Aguiar Coelho (IST)
    - https://sourceforge.isae.fr/projects/nss_adcs

- **Radiation Budget**
- **CSV file editor**
- **Data Budget**
- **IDM-CIC (CNES)**
- **Neo4j.com**

- **LOS check**
  - Stela (CNES)
    - https://logiciels.cnes.fr/fr/content/stela

- **Nanospace UI (AGPL v3)**
  - https://sourceforge.isae.fr/projects/nanospace
  - https://sourceforge.isae.fr/projects/nanospace_ui

- **Centralized Database**
  - Neo4j
    - https://neo4j.com

- **Power Budget**
  - Script Nanopower (AGPL v3)
    - Author: Guillaume Roux (ISAE-SUPAERO)
    - https://sourceforge.isae.fr/projects/nanopower

- **Launcher Choice**
- **CSV file editor**
- **Activity Profile**
- **Data Budget**

**Contact:**

thibault.gateau@isae.fr

Nanospace - OSCW2020 - December 2020
1 Nanospace - Demo
   - Demo: First connection
   - Nanospace-UI
   - Demo: First interactions

2 Nanospace - Architecture
   - Technical choices
   - Running Architecture
   - Interaction with third party applications
1 Nanospace - Demo
  - Demo: First connection
  - Nanospace-UI
  - Demo: First interactions

2 Nanospace - Architecture
Demo: First connection

Using your favorite Web Browser

- Go to https://dcas-nanostar.isae.fr
- You can Subscribe and Login
- Import an example project json project file: https://gitlab.isae-supaero.fr/nanostar/nanospace/nanospace-user
View of a project in a browser
First interactions

View of a project in a browser
First interactions

View of a project in a browser
First interactions

View of a project in a browser
First interactions

View of a project in a browser
First interactions

View of a project in a browser
First interactions

View of a project in a browser
Nanospace - Demo

Nanospace - Architecture
- Technical choices
- Running Architecture
- Interaction with third party applications
Requirements

- Web App  
  <platform independent>
- REST API  
  <ease third party connection>
- ACID\(^1\) property  
  <concurrent access>

Technical choices

- Database: Neo4j
- Front-end: Angular
- Back-end: Spring Boot  
  (Neo4j direct compatibility)

1: Atomicity, Consistency, Isolation, Durability
Running Architecture

**Client browser**
- Client Web Page (Angular JS)

**Server Side**
- Web Server (Nginx)
- Nanospace Service (Java)
  - Access to data through HTTP
  - Provide basic distributed methods
- Database (Graph database Neo4J)
  - System and Subsystems data

**Third Party Application(s)**
- Manual Interface
- REST Interface

**Third Party Script(s)**
- Embedded code
- Embedded Interface

**Third Party Application(s)**
- REST Interface
- Embedded Interface
REST interface

Easy to run your own scripts

- Direct connection to the REST interface
- Python API provided (nanospace.py)
- You can use your favorite libs (poliastro, orekit, celestlab...)

Python example

```python
from nanospace import Nanospace
nanospace = Nanospace(srvAddr, usr, pw)
altitude = getNanospaceString(nanospace, altitudeID)
d = getMarginDown(altitudeKm, dataJsonFile)
EB_N0 = round(d['Eb N0 Down'], 1)
nanospace.update_string_value(Eb N0 DownID, 'EB N0', str(EB N0)+" in [dB]")
```
Adding an Angular component

Angular component example available on-line

- Angular Component (3 lines of code)
- Available online:
  www.npmjs.com/package/ngx-nanospace-client-lib

Embedded component example

/*app.component.html*/

```html
<nano-input-id [(ngModel)]="idImported"></nano-input-id
<nano-input-value [(ngModel)]="valueImported"></nano-input-value
<nano-import-export-value [(ngModel)]="valueImported"></nano-import-export-value
```
Example with Angular component

Pedagogical link budget module view
Example with Angular component

Pedagogical link budget module view
Example with Angular component

Pedagogical link budget module view
Targeted Architecture

Client browser

- Client Web Page (Javascript)
  - HTTP REQUEST
  - Data subscription
  - HTTP REQUEST

Server Side

- Web Server (Nginx)
  - Provide web application
- Nanospace Service (Javascript)
  - Access to data through Http
  - Provide basic distributed methods
- Message Broker (RabbitMq)
  - « Events Bus »
- History Service (Javascript)
  - Manage data access through Http
  - Subscribe

- Database (Graph database Neo4J)
  - System and Subsystems datas
- Database (MongoDB)
  - Store Data events (creation, delete and update)

Interface available from front-end and domain specific applications

http
Take Home Message

- Easy to integrate to third party application
- Concurrent access, remote-located team context
- Available source code (AGPL v3):
  https://gitlab.isae-supaero.fr/nanostar/nanospace
- Available test server:
  https://dcas-nanostar.isae.fr/
- Web-service - Docker Version

Future Works

- Event management
- Formal pipeline management, top bottom approach (MBSE?)
- Life cycle beyond phase 0/A (up to C...)
- Check resilience (interaction with DOCKS? GMAT?)
Thank you for your attention!

Any question?

Special thanks to:
Marie-Carmen Fauré, Frédéric Fal, Maxime Syidalza,
Jacques Villedumur, Ludovic Bosseaux
A model for preliminary design procedures of satellite systems.
Concurrent Engineering, 16(2):149–159.

DLR (2019).
Virtual satellite.

ESA (2019).
Ocdt.
https://ocdt.esa.int.

Gordon, K. J. (1999).
Spreadsheet or database: Which makes more sense?

Review of concurrent engineering design practice in the space sector: state of the art and future perspectives.
In 2018 IEEE International Systems Engineering Symposium (ISSE), pages 1–6. IEEE.

Idm-cic.
https://www.clever-age.com/fr/case-studies/cnes-une-application-de-modelisation-3d/.

NASA.
GMAT.
http://gmatcentral.org/.

Rheagroup (2019).
Cdp4.