An Eclipse Based Software Solution for Space Mission Design

EclipseCon Europe 2011 – Philipp M. Fischer
Today’s Mission…

➡ A story about building spacecrafts…

➡ Building spacecrafts using Eclipse…

➡ A data model for spacecraft design…

➡ The good and the bad using Eclipse…

*Source: www.media.dlr.de (accessed 31.10.2011)
Building a Spacecraft is Complex…

- Knowledge of various experts
  - Power
  - Structure
  - Thermal
  - Propulsions

- Long term projects
  - Space shuttle
  - Mars rovers
  - ISS
Working in Parallel – The CEF

- Built in 2008 in Bremen

- Bringing the experts together
  - Estimating…
    - Weight
    - Power
    - Costs

- Roughly 14 days
  - 2 Sessions per day
  - Feasible results: +/- 15%
From Excel Sheets to Model Driven…

- Data provided as parameters
- The old way…
  - Modeled in Excel
  - Done by DLR, ESA, NASA
- The new way…
  - Modeled in Step – ECSS-E-TM-10-25
  - Modeled in EMF – Virtual Satellite
A Model Beyond the Early Design…

- It is not just parameters
- Simulations
- CAD
- Commands

- Vision: Supporting the whole lifecycle
- ECSS-TM-10-23 VSD
- DLR Virtual Satellite
Virtual Satellite an Aerospace Design Tool

- Developed together with CEF
- Tool for spacecraft lifecycle
  - Phase A/B - Planning
  - Phase C/D - Building
  - Phase E/F - Operating
- Development in progress
  - Phase A/B in progress
  - Phase B/C just started

<table>
<thead>
<tr>
<th>Activities</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Phase 0</td>
</tr>
<tr>
<td>Mission/Function</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
<tr>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>Verification</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td></td>
</tr>
<tr>
<td>Utilization</td>
<td></td>
</tr>
<tr>
<td>Disposal</td>
<td></td>
</tr>
</tbody>
</table>

*Source: ECSS-M-ST-10C Rev. 1 ESA-ESTEC Standards Division

An Eclipse Based Software Solution for Space Mission Design > Philipp M. Fischer > 31.10.2011
<table>
<thead>
<tr>
<th>Name</th>
<th>Current Value</th>
<th>Unit</th>
<th>Shared</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>visColor</td>
<td>1671680.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visMassDry</td>
<td>7.200</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visMassWet</td>
<td>7.200</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visPositionX</td>
<td>0.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visPositionY</td>
<td>0.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visRotationX</td>
<td>0.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visRotationY</td>
<td>0.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visShape</td>
<td>1.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visSizeX</td>
<td>0.200</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visSizeY</td>
<td>0.400</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visSizeZ</td>
<td>0.400</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visTransformationOrder</td>
<td>0.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
<tr>
<td>visVisibility</td>
<td>0.000</td>
<td></td>
<td>no</td>
<td>N/A</td>
</tr>
</tbody>
</table>
And How it is Made…

- A productive software – not just science
- 2 Engineers full-time
- 1 Student part-time

- Agile project – DLR Software Process
- Quality procedures
  - Junit and Jubula
- Build Process – Continuous Integration
  - SVN, RepoGuard, Maven, Mantis, Jenkins …
Virtual Satellite’s Architecture

- Eclipse widely spread in
  - Academia
  - Industry

- Build on Eclipse RCP using
  - DLR-RCE
  - EMF
  - EMF.edit
  - Xtext

Virtual Satellite / IDE Workbench

- EMF / EMF.edit
- Xtext

Eclipse/RCP

*Source: sourceforge.net/projects/rcenvironment/ (accessed 31.10.2011)
One Data Model for All

- One central data model
  - Common place of knowledge

- Modeling
  - Decomposition of spacecraft
  - Parameters for phase A/B
  - Calculations
  - QUDV

- Repository as central data management unit
  - Backend Interchangeable
How to Get it Distributed?

- One data model
  - But multiple engineers

- Engineers used to SVN
  - Commit – Provide changes
  - Update – Obey changes

- Organizing read and write

- Avoiding merge conflicts
First Step – Who is Allowed to Modify the Data?

- **Modeling Users**
  - A Discipline
  - One User Name

- **System Component**
  - Applying one User
  - Sharing data areas

- **Asking RCE LDAP for authorization**
  - Granting permissions
Second Step – Serialization to Store the Data...

- Text for SVN Backend
- Small changes in one session
- Using XMI
- Provided by EMF
- So far writing as one resource
- Multiple resources intended
Third Step – Separation for Mutual Units…

- Multiple files for SVN
- Separate files for System Components
- No merge conflict since
  - One engineer per component
  - Each component one file
- Result: Just update / commit needed
Fourth Step – Storing it Version Controlled

- Using SVN
- Allows for version history
  - What happened yesterday
- Trunk/Tag/Branch
  - Not sufficient for us
- Centralized control needed
  - SVN internal copy as new Head-Revision
Why not Other Alternative Tools?

- What about CDO, Hibernate …?
- So many good eclipse projects
  - But which one to use?
- There is little time for evaluation!
- Question of acceptance…
  - SVN is simple for engineers
  - SVN is simple for development
Eclipse the Bad Side…

- Sometimes poor documentation
- Non appropriate examples

- EMF compare framework
  - Simple comparisons needed
  - 2 days of effort – No results

- Developed our own comparison
- Not even 1 day of effort !!!

*Source: wiki.eclipse.org/EMF_Compare (accessed 31.10.2011)
A Docking Maneuver to E4...

- Our current Eclipse version
  - 3.7 Indigo

- Our way to E4
  - Not many trouble expected
  - Changed to Command Framework

- Good experience with Command Framework but…
  - What about Ctrl-S - Still an Action…
  - What about Toggle-Buttons…
Eclipse the Good Side…

- A good tool for SW development…
- Many good tools for applications
  - Modeling Framework
  - Xtext
  - …
- Many other projects
  - Integration on RCP level
Taking software engineering to space?

- Open source software engineering is concurrent…
  - It seems years ahead of the classical engineering disciplines…

- We already use Subversion as data storage backend!

- Why don’t we use…
  - Mantis/Mylyn for traceability in the CEF design sessions?
  - Junit Test Runner for verifying simulation modules?
  - Equinox for component based simulations?
What Have we Seen Today…

- A little story of building a spacecraft. We have seen that a lot of different experts are contributing to such a design.

- An application to design spacecraft and to go beyond the early planning phase. All built on top of Eclipse RCP and EMF.

- Some good and bad aspects on using Eclipse. In particular the broad variety of projects and poor documentation is a main blocker for our developments.
See you soon...

Philipp M. Fischer
German Aerospace Center DLR
Software for Space Systems and Interactive Visualization
Philipp.Fischer@dlr.de