Satellite System Engineering

-- Requirements

Prof. Dr. Shufan Wu
Chinese Academy of Science (CAS)
Shanghai Engineering Centre for Microsatellite
Haike Road 99, Shanghai, China

Email: shufan.wu@mail.sim.ac.cn

Requirement Engineering

Requirement engineering is defined on ECSS-E-ST-10-C and ECSS-E-ST-10-06C and includes:

– Requirement statement
– Requirement allocation
– Requirement validation
– Requirement maintenance
Requirements

Formal statements of needs

3 key points to remember about requirements:

- define “what” is to be done ⇒ function
- define “how well” it has to be done ⇒ performance
- define (sometimes implicitly) how the requirement should be confirmed ⇒ verification

Requirement statement/allocation

- Requirements shall follow a hierarchy
- Requirements shall be linked to the mission/product tree
- Requirements need to be traceable: it must be possible to track a requirement for a subassembly up to higher levels by a logical/numerical link
- Requirements shall always be comparable to design/mission parameters
- Interface requirements shall be identified
- Requirements shall be maintained (docs and tools) (i.e. Please don’t use MS Word for req. engineering)
Requirement formulation

• Good requirements are:
  – Short/Synthetic
  – Definite/Unambiguous
  – Verifiable
  – Traceable
  – Formulated using terms that have been properly defined earlier

• Examples:

The system shall provide a throughput of 250 kbits/s

The system total wet mass at launch shall be less than 1300 kg

Requirement formulation – 2

The system shall be designed to achieve the required performance maximising the spectral efficiency

The FDIR function shall be implemented at the lowest possible functional level

SCIH-001  During the prime mission from 3.25 AU through perihelion passage high resolution mapping of the nucleus (>80% of the surface) should be possible; i.e. each point of the surface (between local noon, over the terminator, to local midnight) shall be observable at highest possible resolution at any time by the remote sensing suite of instruments - ALICE, OSIRIS, VIRTIS and MIRO - and the Neutral Gas and Ion Mass Spectrometer, ROSINA. For the Narrow Angle Camera of the Imaging System, e.g. this means that most of the nucleus surface will be mapped with a resolution better than 10 cm. The instrument line of sight shall be independently selectable to any point of the comet.
Requirement formulation – 3

A TBC figure in a requirement means that the Author is proposing that figure but is asking the Contractor to perform its own analysis to confirm it: Violating a TBC req. is not a non-compliance.

A TBD figure indicates that the Author has identified this as a relevant requirement but precise quantification is missing. It is left to the Contractor to propose and justify its own value for agreement with the Author.

A “should” requirement is a goal; meaning that first the “shall” req.s must be used to design the system and then, the impact of the additional “should” req. analysed. In case this impact is affordable, the “should” may be later turned into a “shall” (ECSS says “should” is a recommendation).

Requirement formulation – Restrictions according To ECSS

Terms not to be used:
“As necessary”
“Appropriate”
“As far as possible”
“Optimise/minimise/maximise”
“Sufficient”
“Best possible”
“State-of-the-art“
Etc...
### Requirement Hierarchy

<table>
<thead>
<tr>
<th>Requirement types</th>
<th>Respon.</th>
<th>Content</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Requirements/ Mission Objectives</td>
<td>User</td>
<td>Non-technical, high level, general</td>
<td>URD or Mission Objectives Document</td>
</tr>
<tr>
<td>Mission Requirements</td>
<td>Customer/Sponsor</td>
<td>Functional, technical, overall performance. Applies to the Mission</td>
<td>MRD or MSRD</td>
</tr>
<tr>
<td>System Requirements</td>
<td>Customer/Sponsor</td>
<td>Functional, technical, overall performance. Applies to the System</td>
<td>SRD</td>
</tr>
<tr>
<td>System Requirements Specification</td>
<td>System Developer</td>
<td>Detailed, technical, reflects the design. Represents the interpretation of the customer req.s from the developer</td>
<td>System Specification</td>
</tr>
<tr>
<td>Lower level specifications</td>
<td>Lower Tier Supplier</td>
<td>Very specific and detailed, one-to-one correspondence to system requirements</td>
<td>Element, Subsystem or unit specification</td>
</tr>
<tr>
<td>Interface Requirements</td>
<td>Customer</td>
<td>Allows connecting the system with other systems</td>
<td>IRD and ICD</td>
</tr>
<tr>
<td>Operations Requirements</td>
<td>Operator</td>
<td>Constraints due to operations</td>
<td>OIRD</td>
</tr>
</tbody>
</table>

### Requirements’ Tree and Flow-down

Mission requirements and constraints are used to generate system requirements (added value from the design team). They are in turn used to generate the spacecraft system and subsystem design and fix the interfaces with the launcher, the payload and the ground segment.
### Requirements Evolution

<table>
<thead>
<tr>
<th>Requirement definition phase</th>
<th>Evolution and detail of the requirements; Lead to Preliminary Requirement Review by the end of the phase</th>
<th>Requirement are frozen by the beginning of the phase (System Requirement Review - SRR)</th>
<th>System and Subsystem requirements are translated into specifications</th>
<th>Any requirement modification has a heavy cost impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 0</td>
<td>Phase A</td>
<td>Phase B</td>
<td>Phase C</td>
<td>Phase D</td>
</tr>
</tbody>
</table>

#### Design Cycle

2016.10.24, Thailand

### Requirements’ iteration

Link between req.s and design in initial mission phases:
- req.s are allocated top-down but bottom-up design verification is used to confirm them

```
Mission requirements

NO

System requirements

YES

Design

Mission req.s achievable?

Confirm/freeze req.s
```
The V-model

Constraints

Restrictions imposed to the possible design choices ("negative requirements")
Normally of programmatic/politic/cost nature.
The most important are:
- Latest acceptable launch date
- Lowest level of technology development which is acceptable to the project (linked to cost and risk considerations) - TRL
- Launchers to be excluded (politics)
- Re-use or not of existing platforms or units (industrial policy/competition)
ECSS Standards

A set of documents that capture standard/best practice in space project management, engineering and quality assurance

Painstakingly prepared in cooperation with national space agencies and industry

Mandatory to use in any ESA project (satellites, launchers, inhabited vehicles, etc.)

Generic in nature. Always require tailoring

Should not be duplicated within project specific documents (e.g. MRD, SRD, etc.)

---

How to deal with interfaces?

Each side of the interface shall define the req.s which need to be placed on the other end. These are stated in IRDs

Once the interface is designed (based on the above req.s) the result shall be reported in an ICD (Interface Control Document). This is “signed” by both ends of the interface and allows for a clear definition of the physical and logical interface.

Responsibility for the different element of the interface shall also be clearly assigned

Example of important interface req.s:

- Maximum allowable payload mass (interface between SC and P/L)
- Mechanical and thermal loads at interface (interface between SC and any onboard unit)
- Voltage provided by spacecraft bus (interface between SC and any onboard unit)