# Requirements-Driven Design of Small Satellites: TET and AsteroidFinder

Sergio Montenegro, Frank Dannemann

German Aerospace Center (DLR)
Institute of Space Systems - Core Avionic
Robert-Hooke-Str. 7
28359 Bremen, Germany

sergio.montenegro@dlr.de, frank.dannemann@dlr.de

## Introduction

The design and implementation of a satellite mission is divided into several different phases, normally ranging from phase A (feasibility) to phase E (utilization). Parallel to these phases an evolution of requirements will take place starting with the user requirements. The whole development process has to be traceable back to requirements until the system engineers can achieve the final state, which resides in the actual specification of the system. Therefore, the implementation of a satellite can be defined as a continuous process with the requirements at its center.

One of the biggest problems faced is the handling of the requirements, which are edited mainly with WYSIWYG word processors (such as MS Word or OpenOffice). During the process of manually copying the actual requirements (or references to them) between documents (e.g. from the functional to the technical specification) a lot of errors may arise while attempting to achieve backwards traceability, if not in the initial stages of requirement creation or tracing, then certainly in the later project phases when the possibility exists of and increase in the number of inconsistencies. Such critical points in time can occur after a review process like a detailed/critical/… design review, or after major changes in one of the subsystems of the satellite.

In this paper we present the approach which we are using for both the TET [1] and AsteroidFinder [2] missions, where the design of the satellite missions is driven by a tool-based requirements management process. Using Telelogic DOORS [3] as the tool of choice, all requirements will be stored in a central database which is accessible by all project partners regardless of their location. Using features such as baseline management and change requests the software supports the typical development phases of a satellite in an ideal way, also making the time-consuming process of requirements reviewing and RID-processing a lot easier. The usage of the web interface DOORSnet and e-mail notification completes the projects' goal of allowing internal and external users to stay informed of all changes in the satellite's database.

In our paper we will show that, especially for small satellite projects, the usage of a requirements management tool can save valuable time and money, enhancing the verification and validation process and eliminating many errors during the process of requirement definition.

## Requirements Network

The requirements are structured and linked into a network and are intended to be more than just a list of to-do items. This structure clarifies any decisions taken and can be considered as a part of the design justification folder. Following the links the reader can get answers to the hows and whys (e.g. why X-band? How to transmit 80 gigabits per day?), so increasing the understandability of the requirements and providing a means for traceability beginning with the most abstract requirements through to the implementation requirements and down to the implementation itself. The technical requirements are linked from goals/abstract/concern requirements, going through some intermediate steps, finishing with the implementation requirements (what the system designer has to consider), like it is shown in the follwowing figure:

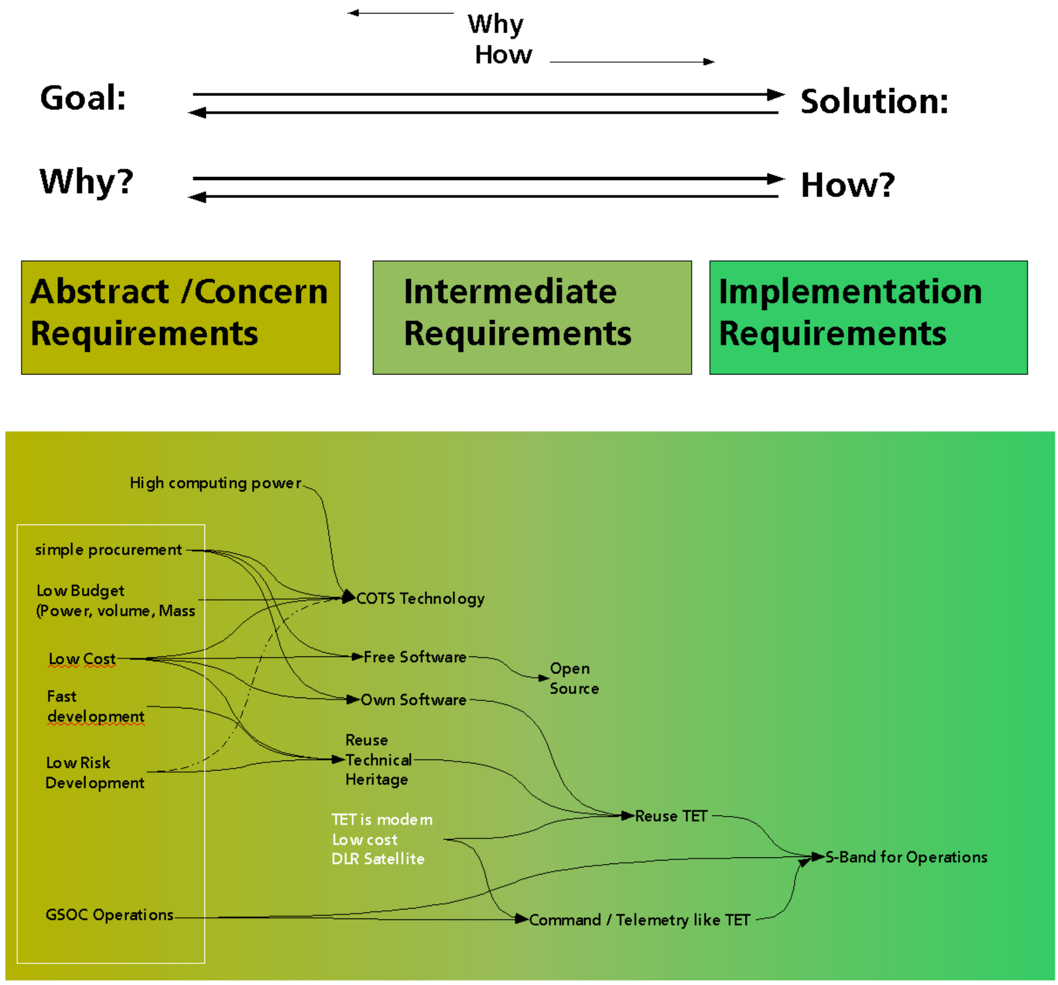


**Figure 1: Network arrangement of requirements (hypothetical example)**

The abstract requirements are mainly functional requirements (what to do) and the implementation requirements are mainly structural ones (how to do it). This produces a graphical network of goals and sub-goals. Following the arrows (mostly from left to right) in the network we answer the question how? The opposite direction (right to left) answers the question why?

The overall network of requirements does not have a graphical representation. The picture above is solely to clarify the concept. Nevertheless DOORS provides the possibility to visualise all links between <u>two</u> formal requirement modules. The picture below gives an example of such link graph, showing some of the connections between funtctional and technical requirements for the AsteroidFinder satellite:

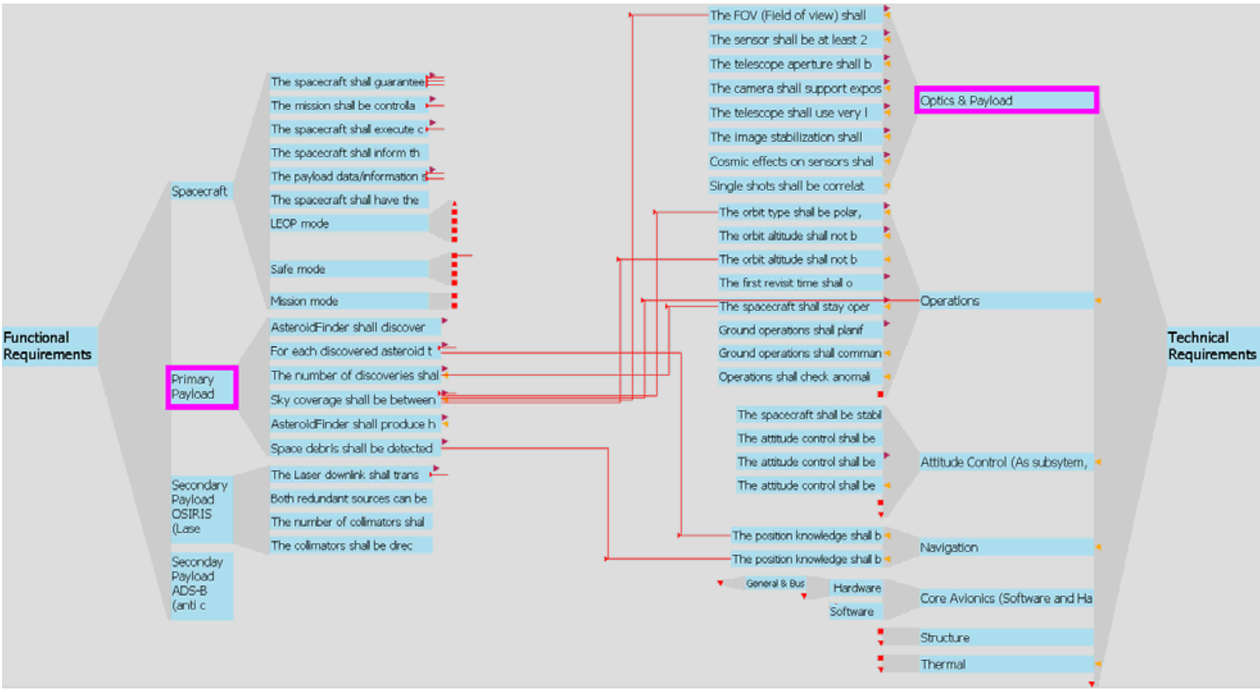


**Figure 2: Link graph between two requirement modules.**

A requirement without an output arrow implies that it has no further consequence and it can be considered as a final requirement. Many output arrows mean this requirement has many implications to the system. Many input arrows mean this requirement is a solution for many other implications. Only one input arrow would be enough to justify this decision.

## Requirement Types

Not all requirements are the same. They are classified into different groups which are interconnected into a network as shown in the following figure:

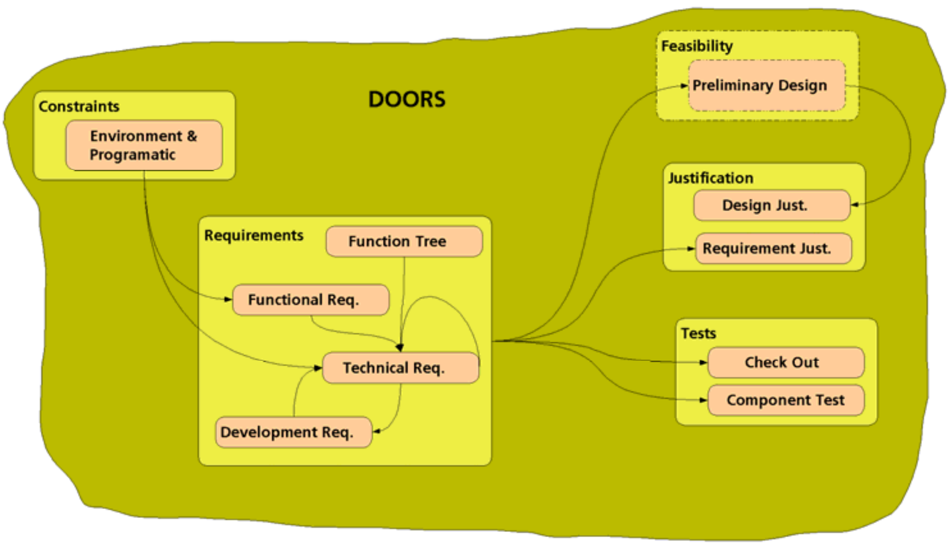


**Figure 3: Types of requiremts**

We can see that not only requirement modules influence the requirement graph. Modules containing constraints also have an important impact, like for example: “Space is cool” or “Reuse TET”. These constraints are facts we cannot change and have to accept. Constraints are grouped into environment constraints (e.g. “Space is cool”) and programmatic constraints (e.g. “Reuse TET”). Constraints influence directly the functional requirements (e.g. “Find at least 10 asteroids!”) and the technical requirements (e.g. how to survive in space).

There is one additional requirement module which has no input, but a lot of output connections: the Function Tree is based on the experiences of past satellite missions and contains a list of typical functionality which has to be implemented in order to operate the satellite and to use the payload. This Functional Tree has implications on technical requirements which describe how to implement this functionality. The following picture shows how some of the typical activities to operate a satellite are influencing the underlying technical requirements:

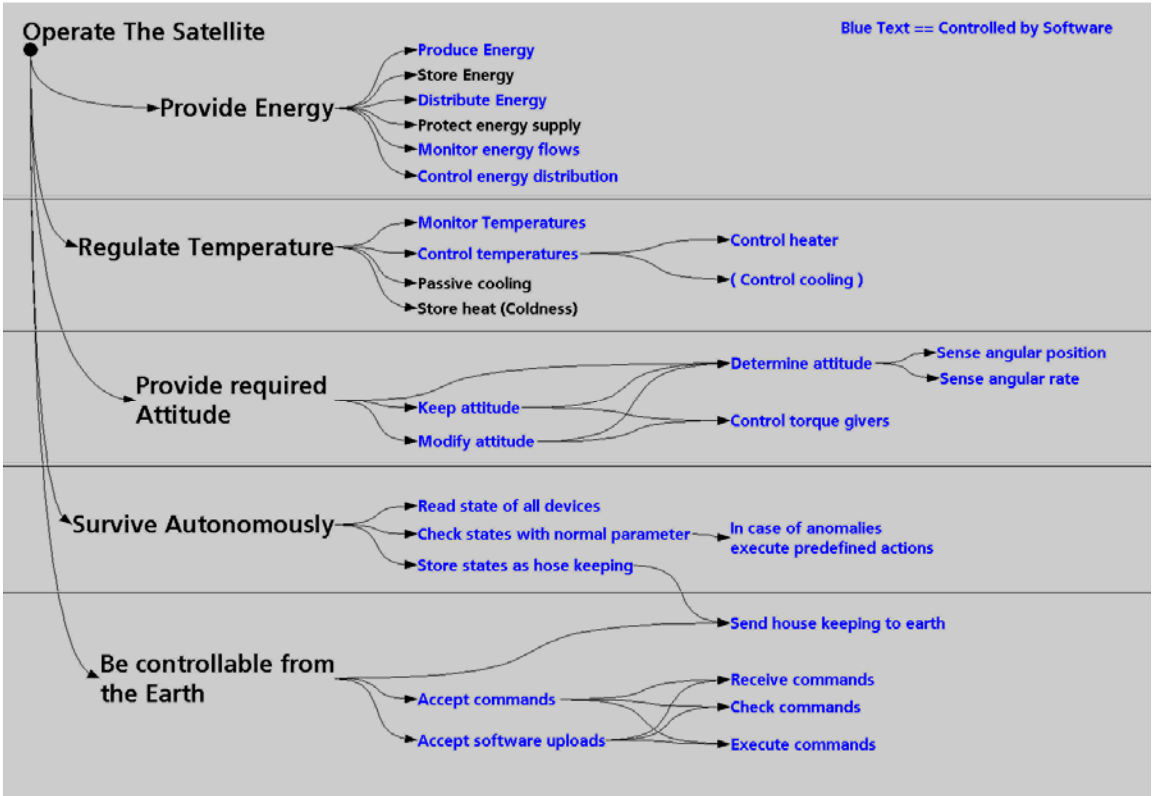


**Figure 4: Functional tree to operate the satellite**

## References


[1] *Technologieerprobungsträger TET*, http://www.dlr.de/rd/desktopdefault.aspx/tabid-2274//3396_read-5085/

[2] Mottola, S., Börner, A., Grundmann, J.T. , Hahn, G., Kazeminejad B., Kührt, E., Michaelis, H., Montenegro, S., Schmitz, N., Spietz P.: *AsteroidFinder: Unveiling the Population of Inner Earth Objects*, 59th International aeronautical congress, 29th September to 3th October 2008, Glasgow, Scotland.

[3] http://www.telelogic.com/products/doors/