Implementation of an automated Systems Engineering Toolset in the DST project

MSc. Thesis

R.J. de Vries

Technische Universiteit Delft

Implementation of an automated Systems Engineering Toolset in the DST project

MSc. Thesis

by

R.J. de Vries

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Monday January 17, 2022 at 09:30 AM.

Student number:	4278933	
Project duration:	February 1, 2020 – January 17, 2022	
Thesis committee:	Dr. Ir. J. Bouwmeester,	TU Delft, supervisor
	Prof. Dr. E. K. A. Gill,	TU Delft, chair
	Dr. S. M. Cazaux,	TU Delft, examiner
	Prof. J. J. D. Loicq,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

I would like to thank Dr. Ir. Bouwmeester for the opportunity to work on the DST project and for his supervision. His input has been very helpful throughout my thesis. I would also like to thank the team members of the DST project for their input for the systems engineering approach and SE Toolset.

I would also like to thank my family and friends for their support and help throughout the last year. Without them, it would have been a lot more difficult. Furthermore, I would like to thank T.J.G Roepman for our online Vrijmibo's and discussions about the SE Toolset. This was really appreciated. I now look forward to a new challenge to start my engineering career in the field of Systems Engineering.

R.J. de Vries

Delft, December 2021

List of Abbreviations

AHP	Analytic Hierarchy Process
CIT	Configuration Item Tree
DOT	Design Option Tree
DST	Deployable Space Telescope
ISB	Integrated Spacecraft Bus
IQR	Inter Quartile Range
M1	Primary Mirror
M2	Secondary Mirror
MBSE	Model Based Systems Engineering
MWIR	Mid Wave Infrared
NIR	Near Infrared
NLP	Natural Language Processing
RGB	Red Green Blue
SE	Systems Engineering
SMSS	Secondary Mirror Support Structure
SWIR	Short Wave Infrared
TBD	To Be Determined
TIR	Thermal Infrared
TRL	Technology Readiness Level
VIS	Visual

Contents

List of Figures

List of Tables

1 Introduction

In fast changing and innovative projects with dynamic resources it can be difficult to keep track of the changes and to transfer information between team members. Systems engineering (SE) is a necessity for the success of these projects. However, in most projects the classical document based form of systems engineering is still used. This requires a systems engineer to keep track of all the changes and to cascade updates. With dynamic resources this is difficult. In particular if there is no dedicated systems engineer, which is the case for the Deployable Space Telescope (DST) project of the TU Delft. Therefore, a more automated SE system which limits the need for a systems engineer is desired.

In the DST project a deployable space telescope is designed by MSc students, PhD students and TU Delft Staff. The design of deployable optics is challenging, especially with the dynamic resources. For such a project a Model Based Systems Engineering (MBSE) approach would be better suitable than classical SE as this has a higher degree of automation, is centralized and is easier to transfer to succeeding team members. However, as the layout and architecture of the design were not known yet and setting up a MBSE model takes considerable time, it was deemed to difficult to implement for the DST project. This led to the development of an automated SE Toolset, which supplements the classical SE to resemble more of a MBSE approach. The SE Toolset automates most of the SE which does not require user input, such as identifier creation and version control, while not being as complicated as an MBSE model. The team members can focus more on the contents of the requirements while the role of systems engineer is limited to a checking role rather than bookkeeping every change or update. Since the SE Toolset maintains all requirements and version control, the systems engineer only has to check the contents of new requirements or new versions. Apart from only maintaining requirements the SE Toolset can also maintain budgets and makes use of variables to link requirements and budgets. The functionality can even be expanded to handle different versions or iterations of the design on multiple levels. Commenting functions and update notifications are also possible with the SE Toolset.

In this thesis the needs of the DST project on the top-level systems engineering (SE) are investigated and a suitable SE approach is designed. The research objective of this thesis is formulated as:

> *The objective of this research is to develop a systems engineering approach for dynamic and innovative projects by combining existing systems engineering methods into a centralized environment that will be implemented on the TIR DST project as a running case study.*

In order to successfully develop a new SE approach it is important to first investigate the needs of the DST project on the top-level SE. Therefore, the first research question is formulated as:

1. *What are the most important needs for top-level systems engineering for the TIR DST project?*

Once the most important needs are known, the most suitable SE approach for the DST project can be determined. It is furthermore important to investigate how it can be proven that it is the most suitable SE approach. The second and third research questions are therefore formulated as:

2. *What is the most suitable systems engineering approach for the TIR DST project?*
3. *How can the suitability of the systems engineering approach be proven?*

The SE approach developed during this thesis, will be tailored to the specific needs of the DST project. It is however very interesting to investigate how this approach can be adapted and applied in other projects. Therefore, the results of using the SE approach in the DST project are analyzed and recommendations for further development are made. The last research question is formulated as:

4. *For what kind of projects is the systems engineering approach applicable?*

First, in Chapter 2, a more detailed analysis is presented on the state of the SE in the DST project at the start of this thesis. Next, in Chapter 3, the results of the analysis are discussed, improvement points and preconditions are identified and requirements on the SE approach are formulated. In Chapter 4 the options regarding programs for the creation of the SE toolset are discussed, a trade-off is performed and the implementation and set-up of the SE toolset is described. After the implementation of the SE toolset, a top-level SE analysis and comparison to reference satellites was done, which is discussed in Chapter 5. Next, in Chapter 6, the SE toolset is evaluated on it's performance in the DST project and is evaluated in a broader perspective to elaborate on the scientific value and implementation in other projects. Lastly, conclusions are drawn in Chapter 7 and recommendations for further development are made in Chapter 8.

2

Deployable Space Telescope Project

In this chapter an overview of the Deployable Space Telescope (DST) project is given. First, in Section 2.1, it is described how the DST project started and how the project developed over time. The change towards a thermal infrared (TIR) design is also explained. Next, in Section 2.2, the systems engineering (SE) documents from the visual (VIS) design are analysed and discussed. In this section the bottlenecks in the VIS DST SE are also identified and discussed.

2.1. Deployable Space Telescope Overview

The Deployable Space Telescope, hereafter referred to as DST, is a project at the TU Delft to design a low cost deployable space telescope. The goal is to develop and prove that deployable telescopes can have high spatial resolution with respect to their stowed volume. The deployable optics are also a mean to do this at low cost. The team consists of MSc students, PhD students and TU Delft staff. The first design was based around the optical bands, wavelength range from 450nm to 692nm, which is referred to as the Visual (VIS) DST. In Figure 2.1 the VIS DST design is shown.

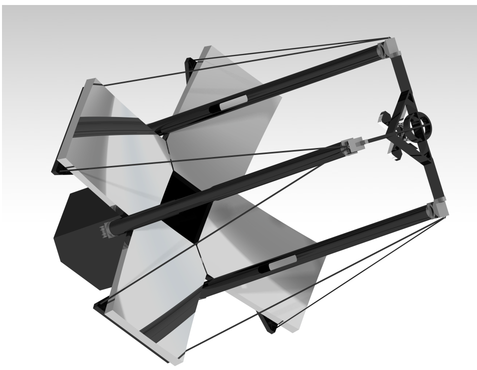


Figure 2.1: Render of the VIS DST design.

The project started in 2015 after the thesis of D. Dolkens. He concluded that the development of deployable optics could provide high ground resolution at a lower stowed volume. [10] In the following years more and more master students started working on the project and by 2020 most of the design was in the detailed design. With V. Nagy [18] working on a third iteration of the baffle and I. Akkerhuis [2] completing a detailed design of the secondary mirror (M2) deployment. The organisational structure of the VIS DST project is shown in Figure 2.2.

In 2020 the decision was made to focus on a thermal infrared (TIR) DST, because the interest of third parties shifted from the visual (VIS) spectrum towards the TIR spectrum. Thesis students that where close to the end of their thesis continued to work on the visual design, while new thesis students started on the TIR design. In some cases, like V. Nagy [18] and F. Hu, team members started their thesis on the visual design and switched

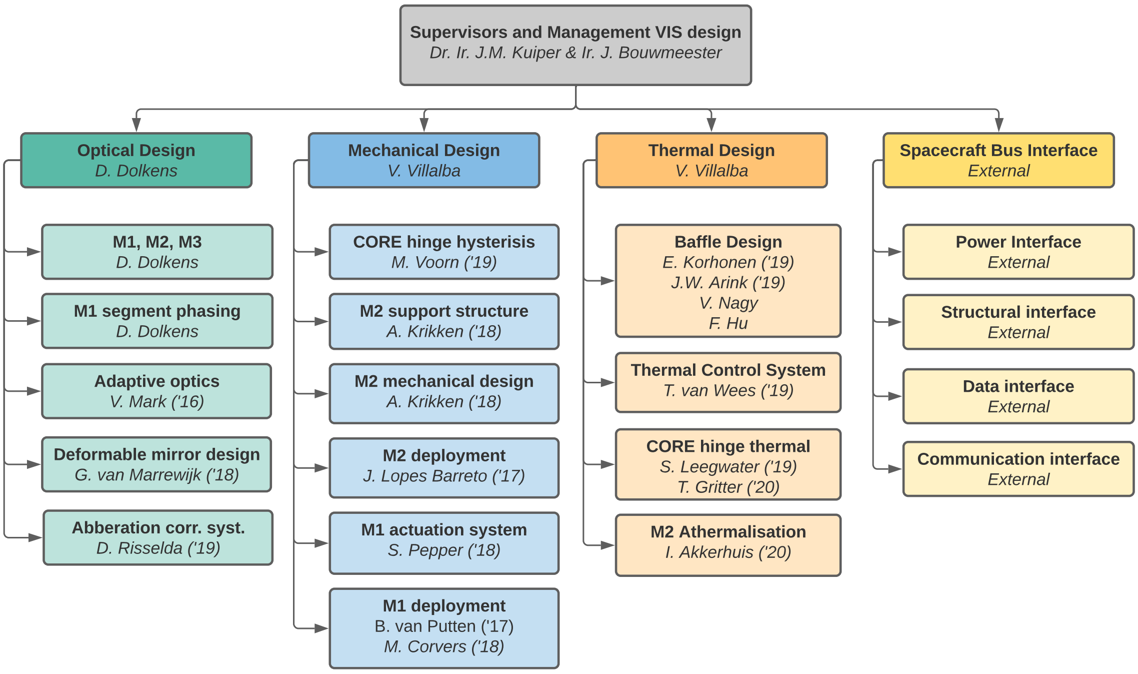


Figure 2.2: Organisational structure of the VIS DST project.

halfway to the TIR design. An overview of the current organisational structure of the TIR DST project is shown in Figure 2.3. To accommodate the switch to a TIR design there was a need for good systems engineering. Especially since the last top-level systems engineering in the project was done in 2018. As a result, there was no clear overview, dependencies were unknown, there was limited to no communication between team members and requirements where not up to date. All team members worked compartmentalized, also known as working in silos, which led to discrepancies between designs and requirements. To get the systems engineering (SE) back on track and to prevent a similar situation happening in the future, a new SE approach was needed. To identify the specific problems in the DST project and to identify improvement points, an analysis of the VIS DST SE documentation was performed. This analysis is described in detail in the following section.

2.2. Visual DST Systems Engineering Analysis

At the start of the thesis an analysis was done on the performed systems engineering in the project. From this analysis, it was clear that in the first phases of the project systems engineering (SE) was done to start up the project. The following need and mission statement were found:

Need Statement
There is a need for a dramatic decrease in launch cost of high-resolution Earth observation telescopes to provide data with a higher temporal resolution and at a lower price than is currently available.

Mission Statement
The goal of this project is to design and develop a Deployable Space Telescope (DST) that is capable of achieving the same GSD as state-of-the-art Earth Observation satellites for a fraction of the costs, by making it able to achieve a relatively very low stowed volume and mass.

Besides the need and mission statement a functional flow diagram, N2 diagram, requirements discovery tree, requirements list and a mass, volume and alignment budget were found. Furthermore, it was found that multiple options to perform the desired functions had been investigated and a trade-off was done. The analysis of the VIS DST SE documents is described in more detail in the following section.

2.2.1. Systems Engineering documentation

The following systems engineering (SE) documentation was found:

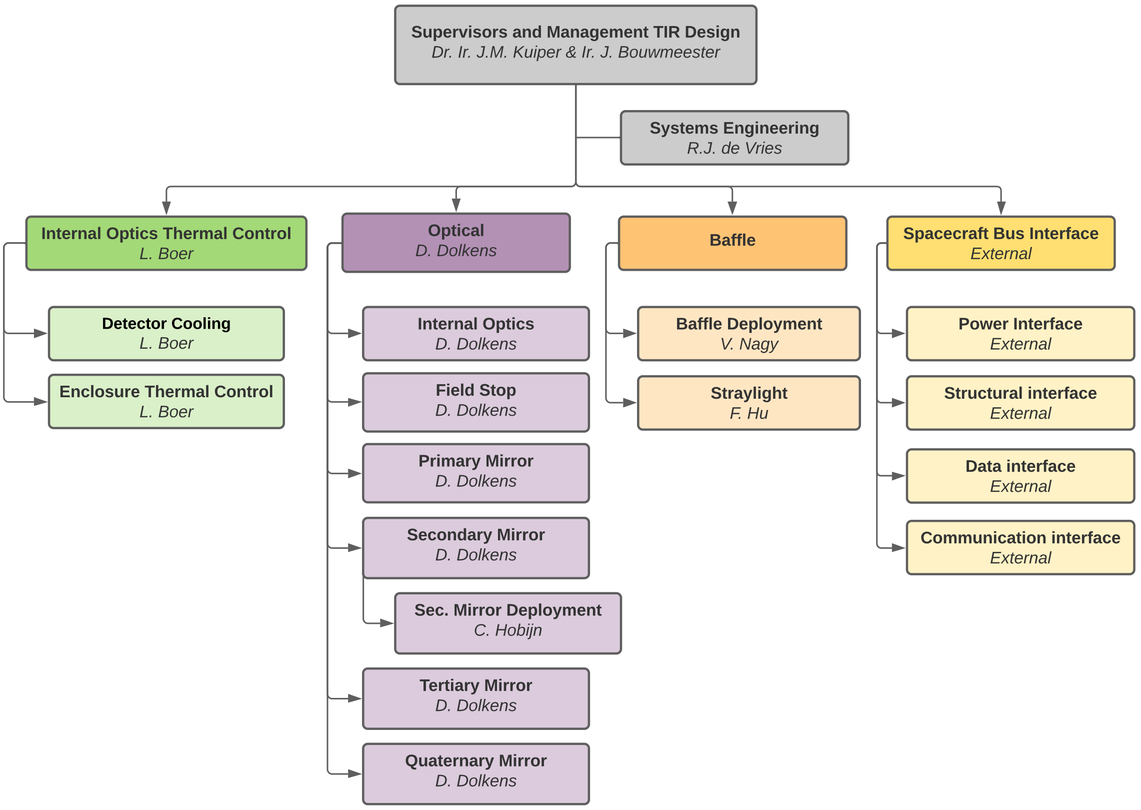


Figure 2.3: Organisational structure of the TIR DST project.

Systems Engineering Excel File
Documentation of the overall systems engineering performed in the VIS DST project was found on the "Deployable Telescope Project" shared drive. In this folder multiple excel sheets were present with similar names. This makes it quite confusing to know what the latest version is. Looking at the date and name and comparing the different versions the most recent version appears to be "20181121 - SE document - Adapted". In the document the following was found: organogram, need and mission statement, functional flow diagram, project decisions, N^2 chart, requirements discovery tree, requirements matrix, compliance matrix, risk maps and a mass budget. Most of the sheets were worked out pretty well. However, part of the documentation was incomplete or missing. For example, the references in the projects decisions sheet were not clear. Page numbers or chapters were not mentioned for the relevant thesis or the reference was missing completely. For the requirements the identifiers where not clear and it was found that they were last updated in 2018. The compliance matrix and risk maps were empty.

Engineering Budgets Word Document
A word document, "DST Systems Engineering revision December 2018", which contained additional budgets was also found. The performance budget was left empty and the power and cost budget were still to be determined. The alignment budget and volume budget lacked sources. For the volume budget it was only mentioned that the calculation was done using a parametric estimation relationship developed by the Aerospace Engineering staff of the TU Delft. The instrument mass estimates were done through a preliminary bill of materials. This was done by estimating the mass of each component from previous design work or from historical data with an additional safety margin of 20%. The resulting masses appear to be adequate, although the last one is from 2017.

Engineering Budgets Older Word Document
In an older word document, called "20180201_DST_SysEn_document_v01", some extra information was found. In this version of the document, the names of the persons responsible for updating the different budgets are present and the correct references are also mentioned. Furthermore, the alignment budget in this document contains a line which mentions that the alignment budget comes from the most recent paper of Dolkens. However, which paper this exactly is, is not clear.

Requirements Excel Sheet
In one of the older folders on the drive the "requirements_revI" was found. In this excel sheet the requirements are stated. This is an older version than the "20181121 - SE document - Adapted" excel sheet. However, this excel sheet contains a sheet with the requirements hierarchy which clarifies the requirement identifiers a bit more. However, the abbreviations are also not explained here.

Requirements Word Document
In the same old folder, a word document was found, "DST_requirements_document_revC", which contains more information on the requirements hierarchy and the abbreviations used. Furthermore, it explains for some requirements how they were obtained and how they should be verified.

SMSS Document
For the subsystems there was only one excel file present, which was for the Secondary Mirror Support Structure(SMSS). The file is called "20181121 - SMSS SE document" and contains the N^2 chart for the SMSS. There was no further context or documentation present in the document.

2.2.2. Bottlenecks in Visual DST Systems Engineering

From aforementioned documentation and from interviews with the team members the bottlenecks in the project were identified. The interviews can be found in Appendix A. First, a small overview is given as to what led to the lack in SE. Next, the bottlenecks are identified and explained.

Overview
The project has always been done in a bottom-up fashion with the resources that were available. However, there has never been a complete team in which all aspects were covered. During the DST project the team working on the project became larger. More MSc students started working on the project and PhD students and TU Delft staff got involved. However, MSc students graduate and mostly leave the team and TU Delft. The extended knowledge they had gathered in their subject had to be handed over and a part of their knowledge got lost. The involvement of PhD students and staff is beneficial to carry this knowledge on after the MSc students leave, which provides some continuation. Nonetheless, the project got compartmentalized. Each part of the project, mostly divided by subsystem, was working on it's own. There was no clear interface management and not much consultation between team members from different parts of the project. This was mainly due to the lack of clear overview on the project and poor interface management. There was no centralized documentation in which all the adaptations to the design and changes in requirements were kept with a dedicated person to keep track of this.

The initial systems engineering performed was not adequately maintained and evolved in an ad-hoc and incomplete fashion. Meaning that in each subsystem design systems engineering was still used as it is necessary to come up with requirements, different design concepts and to do a trade-off. However, the overall systems engineering to keep an overview over the project and to make sure all the different parts interface with each other was not updated. This is an important aspect which needed to be improved in the new SE Approach.

Bottlenecks
The bottlenecks which were identified are first listed altogether, after which they are explained one after each other. The following bottlenecks were identified:

- No clear documentation on systems engineering
- No clear overview over the complete project(interfaces and dependencies)
- Higher need for interface management when the project became larger
- Top-level requirements were outdated

- It was not clear what the mass requirements were based on
- No risk mitigation performed

No clear documentation on systems engineering
In the documentation it was found that there was no clear documentation naming, referencing was lacking or incomplete and some older documents contained more information than the newer version. For new team members it is very difficult to make sense out of the documents. This is very confusing and even leads to team members creating their own estimates, as was also mentioned by I. Akkerhuis in his interview. For future team members the different estimates and documentation are confusing, which creates a snowball effect and eventually leads to an even more compartmentalised project. This is also related to bad interface management which is discussed later.

No clear overview
There was no clear overview over the complete project. On the DST drive there were no files which provide a clear overview of the project or give the interactions between the different parts. There was a N^2 chart present, however it was not clear when it was last updated. Having a document which is up to date with a clear overview would help the team members to identify who they should inform in case they make adjustments to their design. This also makes it more clear what the impact of a change in one part is to the rest of the system. Having a dedicated person who is responsible for this document and who is keeping track of all the changes would greatly help in keeping a better overview over the project. Furthermore, the requirements which were generated on subsystem level were not updated in the overall systems engineering documentation. This leads to different sets of requirements used by different team members. This ultimately leads to a non functioning design.

Interface management
According to multiple team members, D. Dolkens and V. Nagy, the interface management was not adequate anymore. Especially when the project became larger there was a higher need for interface management. V. Nagy wrote that the important role of central systems engineer was missing in the project and that the different parts of the DST were being designed separately. Furthermore, he wrote that there was minimal engineering communication between the team members from different parts of the project. Due to the missing role of systems engineer there was no clear overview and therefore also no proper interface management.

Top-level requirements outdated
The last time the requirements were updated was 2018. The outdated top-level requirements were also mentioned by I. Akkerhuis and T. Gritter in their interview. Furthermore, V. Nagy mentioned that he had a local list of requirements which he updated when necessary. This list was checked by many DST members, which is good. However, keeping these requirements locally means that they were not easily accessible for team members in case they need them.

Mass requirements
For the mass requirements it was not clear how they were found. Since the mass requirements and budgets were unclear some team members came up with their own. I. Akkerhuis mentioned that he discarded the mass requirements. The reason for this was that there was no explanation as to how they had been formed. Therefore, he came up with an educated guess for the secondary mirror (M2) deployment system. This proves again that there was no clear overview and that the project worked in silos.

No risk mitigation performed
In the systems engineering excel sheet both risk maps and the compliance matrix are empty except for some dummy values. It can be argued that in an innovative project, like the DST project, risk mitigation would lead to a conservative design since all the innovations are high risk and high impact. Mitigating these risks would kill the innovation and lead to a conservative design. Even though this is true, at least a risk identification should be performed to know which risks there are in the project. Furthermore, the risks which do not kill innovation should still be mitigated. For instance, a deployment mechanism risk can still be mitigated by redundancy.

3

New Systems Engineering Approach

From Section 2.2.2 it is clear that a different systems Engineering (SE) method was needed. In this chapter the new SE approach is discussed. First, the required changes regarding SE are discussed in Section 3.1. In Section 3.2 the main drivers for the SE approach are discussed and requirements are formulated. Then, in Section 3.3, the goals for the SE approach and SE Toolset are introduced and it is discussed that communication is very important to have a successful SE Approach. Lastly, the trade-off and set-up of a reference library is described in Section 3.4.

3.1. Changes Needed in Systems Engineering of DST

The dynamic setting of the human resources and the changing requirements for the DST ask for a different SE approach than the classical top-down methods. Furthermore, the DST concept and its underlying innovative technologies are beyond state of the art and cannot be predicted easily by first order analysis. In depth modelling, prototyping and testing are required before the high level trade-offs can be performed. Therefore, an agile SE method should be developed, which works both top-down as well as bottom-up. The top-down approach looks at the latest potential applications and the derived system requirements. The bottom-up approach looks at the current level of innovation and what could be achieved with this. Furthermore, this method should preserve the required coherence and overview of the project while being easily adaptable to the latest insights.

In order to preserve the required coherence and overview over the project, a dedicated person is needed who is in charge of the SE. This person should keep track of the SE documentation and make sure that everything is kept up to date. Furthermore, this person should ensure that adequate interface management is done. He should furthermore ensure that different team members from different subjects keep working together and consult each other in case of changes or questions. Having a dedicated person within the DST project for the SE will however be difficult due to cost restrictions. A staff member can be made responsible for the SE, as lead systems engineer, since a staff member is longer involved in the project than a master student. However, staff members are limited in their time available for the DST project. For this reason the SE approach should require minimal time and effort from both the team members and the lead systems engineer to keep everything updated and coherent. The SE approach should preferably save more time in the long run than that it takes to keep the SE updated. To improve the interface management, a team meeting on a regular basis is needed to make sure that all the team members are still on the same page.

For new team members it was very unclear what SE was done in the project and where it was based on. Therefore, they had the tendency to develop their own requirements and use their own SE methods. For coherency it would be beneficial to use a universal approach and method which can be used by everyone and for all subsystems. This would make sure that all the SE on subsystem level is performed in the same way. This would also help in a bottom up fashion, as all the SE on the subsystem level has the same structure. However, it should be clearly defined until which level this is used and the central SE should be clearly defined. At the lowest level the central SE should not become restrictive.

In order for the SE approach to work properly, the SE method should also be agile. This means that in case a change is made in a subsystem it is easily communicated and adapted by the other subsystems. This would also help in a bottom up approach, as new technologies could be easily implemented when the other subsystems can adapt to such a change in an easier fashion.

To improve the overall SE and to create one set of coherent requirements, it was important to perform the top-level SE. With the change to thermal infrared (TIR) there was no clear concept yet. Therefore, team members were struggling to continue their design and they were waiting for concrete input values. To improve the top-level SE of the TIR DST project a detailed top-level analysis was needed. To fulfill this need, a top-down concept trade-off was performed and is described in Chapter 5. The top-down concept trade-off was based on the goals and development of the DST project and for the DST TIR demonstrator mission specifically. The DST concepts were furthermore compared to reference satellites to check if they are state of the art and perform better than fixed telescopes. This is also described in Chapter 5.

3.2. Requirements for SE Approach

In this section the drivers and associated requirements for the systems engineering (SE) approach and SE Toolset are formulated.

First the communication between team members should be improved. Currently the communication is only on a need basis which leads to interface problems, discrepancies in requirements and a non effective team. Improving the communication will solve part of these problems already and will create a good foundation for further improvement of these problems. It is furthermore important that emails that are relevant to the team are stored in a central location. The following requirements are formulated for communication:

SE-COM-01: A central communication platform shall be used

SE-COM-02: Team meetings shall be held on a regular basis

SE-COM-03: Emails relevant for the team shall be stored in a central location

SE-COM-04: The SE Toolset shall support communication features

SE-COM-05: The SE Toolset shall support update notifications

Secondly, the interface management should be improved and current interfacing problems should be resolved. Aligning requirements and document them in a central document will, together with good communication, result in better interface management. Furthermore, a dedicated systems engineer is needed who makes sure all parts interface and responsible team members keep communicating. The following requirements are formulated for the interface management:

SE-INT-01: The SE Toolset shall always be available to team members

SE-INT-02: The SE Toolset shall support requirements

SE-INT-03: The SE Toolset shall support budgets

SE-INT-03-01: The budgets shall be linked to requirements

SE-INT-03-02: The budgets shall update automatically when requirements change

SE-INT-04: The SE Toolset shall have a clear overview of the sub-systems structure

Besides a good communication and interface management it is very important to have good information sharing. [1] Currently, the information sharing within the DST project is not sufficient. Due to the lack of communication there is hardly any sharing of information and documentation. Therefore, it is important to first solve the lack of communication, which will automatically improve the information sharing. To further improve the information sharing, it is important that the documents are in one central location and accessible by all team members. To be able to trace the documents and to properly reference them, it is important that they all have a unique identifier. Furthermore, the documents should contain the name of the author, date of creation and a revision history.

In order to speed up the search for relevant papers, articles and books, a DST library is needed. In this library all reference documents that are used in the project should be present. This makes it easier for team members to find the references mentioned in theses and can help in finding relevant information when a problem is faced. A search function should be created to help find the needed information. For the documentation the following requirements are formulated:

SE-DOC-01: Documents shall have a unique identifier

SE-DOC-02: Documents shall contain the name of the author

SE-DOC-03: Documents shall contain the creation date

SE-DOC-04: Documents shall contain the revision history

SE-DOC-05: The references made in the documents shall be traceable

SE-DOC-06: Documents shall be stored in a central folder accessible to all team members

SE-DOC-07: A library shall be created in a central location

SE-DOC-07-01: The library shall contain a search function

SE-DOC-07-02: The library shall contain the previous theses of the DST project

For the requirements it is important that they should be kept in a central location. It is furthermore important that they have a unique identifier which is not used again in case the requirement is omitted. To have good traceability, version control is important and documentation on which the requirement is based should be clear. All requirement information that does not require user input should be automatically filled in to prevent errors. Lastly, it is important that a systems engineer checks the requirements when a change is made. This leads to the following requirements on the requirement set-up in the SE Toolset:

SE-REQ-01: The SE Toolset shall document the requirements in a clear way

SE-REQ-01-01: The SE Toolset shall support version control

SE-REQ-01-02: The SE Toolset shall automatically generate a unique identifier

SE-REQ-01-03: Requirement identifiers shall not be reused after a requirement is deleted

SE-REQ-01-04: All possible requirement fields shall be automatically filled in

SE-REQ-02: Changes to the requirements shall be peer reviewed by a systems engineer

SE-REQ-02-01: The systems engineer shall be notified when a change to a requirement is made

SE-REQ-03: Requirements shall be traceable to documentation where they are based on

For decisions it is very important that they are easy to trace and clear. It is therefore important that minutes of meeting are made for every team meeting and stored in a central location. This leads to the following requirements for the decision making process:

SE-DEC-01: Minutes of meeting shall be made for every team meeting

SE-DEC-01-01: Minutes of meeting shall be easily accessible in a central location

SE-DEC-02: Design decisions shall be documented in central documentation accessible to all team members

SE-DEC-02-01: The team members involved in the design decision shall be stated in the documentation

SE-DEC-03: Design decisions shall be traceable

SE-DEC-04: Design decisions shall be communicated to all team members

Besides good communication and documentation it is also important that the SE approach is lean. Preferably the SE approach saves more time in the long run than it costs to keep it updated. Once it becomes a hassle to keep it updated, it is mostly abandoned. This should be avoided and should be taken in to account when setting up the SE approach. The following requirements are formulated:

SE-LEA-01: The SE Toolset shall have an intuitive interface

SE-LEA-02: The SE Toolset shall be easy to use

SE-LEA-03: The SE Toolset shall automate everything that does not require user input

SE-LEA-04: The SE Toolset shall reduce the systems engineer's work load to a minimum

SE-LEA-05: The SE Toolset shall minimize the SE work load for team members

Another driver for the SE approach is that it should be an agile system which supports both a top-down approach and a bottom-up approach. This is necessary to be able to easily adopt the latest insights. The following requirements are formulated:

SE-AGI-01: Adjustments to the low level SE shall be easily adopted by the top-level SE

SE-AGI-02: The SE Toolset shall support the creation of different versions of the design

SE-AGI-03: The SE Toolset shall support multiple iterations of the design

Lastly, the SE approach should support both the research purposes as well as the goal to have a fully working system. The difficulty with this is that in a company the end goal is to have a fully working system. While in a university project the goal is to research new technologies and in the end also to have a fully working system. This can however contradict each other as a new technology can arise which is marginally better than existing technologies, but it shows potential or is interesting to investigate. In a company the choice would go for the existing technology as it is proven to work and the fastest, and most likely the cheapest, way to achieve a working system. In a university project the choice could be made to go with the new technology and research it further. The SE approach should therefore be able to guide and implement this research with the rest of the design as well. Requirements SE-AGI-02 and SE-AGI-03 are therefore also applicable here. By supporting different versions the research version could be a branch off of the main design and investigated further.

3.3. SE Approach

In this section the goals for the Systems Engineering (SE) Approach are first summarized in Section 3.3.1. Next, the functions and support the tools need to offer to support the SE approach are discussed in Section 3.3.2. The improvements to the communication, which were needed to make the SE Approach a success, are discussed in Section 3.3.3.

3.3.1. SE Approach Goals

In this section the goals of the SE approach are discussed. They are first summarized after which they are discussed in more detail.

SE approach goals:

- A simple and clear set of tools in a central environment which are easy and intuitive to use
- Link the SE tools together to make them coherent
- Automate as much as possible
- Reduce the need for a systems engineer to a minimum
- Reduce the amount of manual labor for team members when working on systems engineering
- Keep the SE tools cost-effective
- Preferably be independent of third party granted licensing

3.3.2. SE Approach Toolset

In this section the required functionality and support of the SE Approach Tools are discussed. In Figure 3.1 this is represented in a flow diagram.

The first step in the Systems Engineering (SE) approach is to define the configuration item tree. The configuration item tree consists of the system and all sub-assemblies and components of the design. This provides a clear overview of the different components of the system and how it is structured. The configuration item tree is also used to come up with the identifiers for the requirements. The identifiers are tied to the configuration item tree to follow the same structure and to be coherent. Preferably the identifiers should be automatically updated if the configuration items are changed and new configuration items should be easily added. Next to that, it would be preferred to have the functionality to branch of different versions or iterations from the configuration item tree. Meaning that one program can handle different versions or iterations of the design to keep changes centralized.

After the configuration item tree is set up, the requirements can be created and linked to the configuration item tree. Filling in the identifier numbering, update date, updated by and version numbering should all be done automatically. It would be preferred to select the parent requirement also through the configuration item tree. The budgets, like mass, volume and accuracy budgets, should be linked to the requirements. In case a requirement changes, the budget should automatically be updated as well. Vice versa, when the budget is updated a new version of the linked requirement should be created.

To be able to trace decisions and information it is important to link the configuration item tree, requirements and budgets to the theses. It would be preferred to have the theses stored within the program. The most optimal would be to retrieve relevant sections of the thesis automatically, store this data in the program and link it to the configuration items, requirements and budgets.

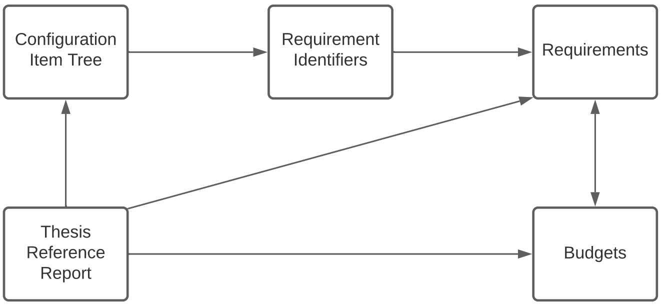


Figure 3.1: Flow Diagram of the Systems Engineering Approach.

3.3.3. Communication for Successful SE Approach

For a project to be successful communication is key. If the communication starts lacking, the effectiveness of the team decreases and the success of the project becomes at risk. In the past years of the DST project all the team members worked on their own part simultaneously. Therefore, the project followed a concurrent engineering approach. This is however not necessarily done on purpose, but happens often in university projects. The reason is the unpredictable and changing human resources. In concurrent engineering everyone works on his own part simultaneously and all parts are interconnected. Therefore, concurrent engineering relies heavily on good communication and information sharing. [1] From the literature study it was found that concurrent engineering is a suitable SE method for the SE approach. [23] Therefore, it is very important to improve the communication and lay a good foundation for the SE approach to be successful and effective. In the DST project the following improvements have been made to stimulate good communication.

The first step was to use Microsoft Teams more regularly. A DST team was created with several channels to host discussions and information sharing. An additional channel was added which serves as an email archive. The use of Microsoft Teams fulfilled requirements SE-COM-01 and SE-COM-03.

The second step was to use a scrum like method with sprints of two weeks. A short meeting, led by a dedicated systems engineer, was introduced on Tuesdays and Fridays. These meetings led to a better team bond, better team dynamics, quicker problem solving and peer reviewing each other's work. Besides the progress meetings, a bi-weekly meeting was organised, in which the thesis supervisors were also present, to focus more on the project status and overview. This meeting helps to keep the team updated on the available resources, changes in stakeholders, changes in the project plan or project goal and decisions regarding the design. It is furthermore a good update for the supervisors to hear all the latest progress from the team members. With these meetings requirement SE-COM-02 was also fulfilled.

The minutes of the two weekly meeting were stored on the DST drive. This provides transparency and gives the opportunity to retrace when certain aspects where discussed or decisions were made. To make the minutes of meeting easier to write, abbreviations for the team members were introduced. The abbreviation consists of three letters. The first letter of the first name and the first two letters of the last name. These abbreviations are both short and still descriptive. With the minutes of meeting, requirements SE-DEC-01 and SE-DEC-01-01 were fulfilled. Since most design decisions were made during the two weekly meetings, requirements SE-DEC-02 and SE-DEC-02-01 were also fulfilled. Design decisions which were made outside the two weekly meetings were send via email to the DST email archive. This fulfilled requirement SE-DEC-04. By documenting the design decision in the minutes of meeting or the email archive they also became traceable. Therefore, SE-DEC-03 was also fulfilled.

3.4. Reference Library

To fulfill requirements SE-DOC-07, SE-DOC-07-01 and SE-DOC-07-02 a reference library was created. The options for a shared reference library are first discussed in Section 3.4.1. Next, it is described how the trade-off was done in Section 3.4.2. Lastly, in Section 3.4.4, it is discussed how the chosen program was set up and used in the Systems Engineering (SE) approach.

3.4.1. Reference Library Options

The most straightforward option is to store all the references in a shared drive. The folders can be structured per subject or per team member. The files should all have a unique identifier and a descriptive title to make it easy to locate the right reference. A search function could be implemented by using an Excel sheet in which all references are stated with their information. Tags can be added to indicate what the references are about. For a small project this approach could work. However, it involves many hours to keep the references structured and to keep the Excel sheet up to date. Furthermore, it would still require manual labor to export the reference, to for instance LaTeX. Therefore, a more automated reference managing system is preferred, like a reference manager with sharing capabilities. PDF files of the references can be imported into the program after which the program automatically finds what kind of document it is and retrieves the relevant information. This is very useful as it saves time and prevents errors.

From the TU Delft Library website and a Reference Management Comparison on Wikipedia it was found that there are quite some Reference Managers.[1] [2] They mostly share the same functionality. From the comparisons done by The University of Chicago Library, Dartmouth Library and G2.com it was found that Mendeley, Endnote and Zotero are used the most and scored best.[3] [4] [5] Therefore, these three programs are further analysed in the next section and a trade-off is performed.

3.4.2. Trade-off

In this section the trade of between the Reference Management Software is done. First the availability is discussed after which the ease of use and sharing capabilities are discussed.

Availability

Mendeley can be used for free with a Mendeley Basic Individual license. However, for all TU Delft students and staff the Mendeley Institutional license is available. This license grants 100Gb of storage for both personal use and shared groups. Furthermore, the limit of having a maximum of 5 groups is removed and the amount of collaborators in a private group is increased from 25 to 100.[6] Since for Mendeley both a free and institutional license is available it scores positive. In case the Institutional License is dropped by the TU Delft the program can still be used with the free version, although with some limitations.

For Endnote the TU Delft has a license for students and staff. There is no free version.[7] In case the support for Endnote X9 is dropped by the TU Delft the program cannot be used anymore. In this case an alternative needs to be found. Therefore, Endnote scores nominal.

---

[1] https://tulib.tudelft.nl/managing-your-information/reference-management/, accessed 15 May 2021
[2] https://en.wikipedia.org/wiki/Comparison_of_reference_management_software, accessed 15 May 2021
[3] https://guides.lib.uchicago.edu/c.php?g=297307&p=1984557, accessed 15 May 2021
[4] https://researchguides.dartmouth.edu/citingsources/refs, accessed 15 May 2021
[5] https://www.g2.com/categories/reference-management?utf8=%E2%9C%93&order=g2_score, accessed 15 May 2021
[6] https://www.tudelft.nl/library/library-voor-onderzoekers/library-voor-onderzoekers/onderzoeken-experimenteren/reference-management-tools/mendeley, accessed 15 May 2021
[7] https://www.tudelft.nl/library/library-voor-onderzoekers/library-voor-onderzoekers/onderzoeken-experimenteren/reference-management-tools/endnote, accessed 15 May 2021

For Zotero no license is needed unless more storage is needed than the free 300Mb. Shared project libraries quickly require more space than 300Mb. Therefore, a paid version would be required which decreases the availability. For this reason Zotero scores nominal.

Ease of Use

It was found that Mendeley is easy to download and install. During the creation of an account you can sign in through your institution or organisation, which grants you automatically the institutional license. Both Mendeley Desktop and Mendeley Reference Manager have the same functionality and are easy in use. PDF files can be dragged and dropped into the program after which the program automatically retrieves the information. Even though it works most of the time very well, the references should always be checked if everything is correct. In some cases it was found that the citations contained wrong information or strange words.

The installation of Endnote X9 was more cumbersome and some problems were encountered during the set up of an account. The layout of the program is a bit more complicated than Mendeley or Zotero. Adding PDF's needs to be done through an import command which is more cumbersome. Furthermore, only the title was imported and no other information. Therefore, it was concluded that Endnote X9 is not easy to use.

The installation of Zotero and setting up an account is easy. The layout of the program is simple and clear and importing references is easy through drag and drop. A nice display is showed which provides the progress on the imports and if Zotero was successful in retrieving information from the file. Zotero was not able to retrieve information for all the references, while Mendeley was. Therefore, Zotero scores neutral for ease of use.

Sharing Capabilities

All three programs support the option to share the references using a group. Invitations to the group can be send through email. All three programs save the imported document which can be downloaded by group members from the program. This removes the need to store the documents also on a server.

Reference ordering

Mendeley and Zotero both have the option to create collections within a group. In this way the references can be ordered according to subject within a group. Multiple layers of collections can be created to sub divide a collection into smaller collections. In Endnote X9 this function was not found. A solution is to create a group set and to add groups in this group set. However, this is limited to two layers.

Commenting

Mendeley and Endnote have the option to preview the document in the program and to add comments to the document. This can be particularly useful to highlight interesting parts of an article or to ask questions to each other about a particular section. The comments contain a record which includes who made the comment and when it was made.[8] Mendeley has an annotations tab in which all comments from the document can be viewed. Endnote does not have this functionality. The commenting system of Endnote also works less intuitive. In Zotero commenting options were not found. Double clicking the reference opens the file locally.

Activity tracking

In Endnote X9 you can view the activity in the shared group. You can see for instance if someone added references or when references were deleted and when a new member joins the group. This can be very useful to track what is going on in the library and how it is used. However, it is not possible to restore references if it is seen in the activity feed that someone deleted them.[9] In Mendeley and Zotero such feature was not found.

Result

Mendeley and Endnote X9 both score positive on the sharing capabilities as shown in Table 3.1. Zotero scores neutral as it lacks the ability to share comments within the program.

3.4.3. Trade-off Table

In Table 3.1 the trade-off table is shown. It can be seen that Mendeley scores overall the best. Therefore, the choice is made to use Mendeley as reference management program.

[8] `https://www.mendeley.com/guides/private-groups/mendeley-reference-manager/02-working-together-on-shared-documents`, accessed 11 March 2021

[9] `https://support.clarivate.com/Endnote/s/article/EndNote-Shared-Library-Activity-Feed?language=en_US`, accessed 11 March 2021

Table 3.1: Trade-off table of the Reference Management programs.

Program	Availability	Ease of use	Sharing capabilities
Mendeley	+	+	+
Endnote X9	o	-	+
Zotero	o	o	o

3.4.4. Mendeley Implementation

In this section the implementation of Mendeley in the DST project is discussed. First the library structure is discussed after which the citation key and PDF name are discussed.

Library Structure

The references are stored according to subsystem or subject. This gives a good overview and the number of different folders stays limited. A downside could be that references are often about a subject and not a subsystem. Therefore, it is chosen to place the references in the subsystem/subject folder which they are related to.

To make it easy for team members to find their set of references a tag with their abbreviation is added. With a search for their abbreviation they get to see their own references. Furthermore, each reference is given a tag which is unique for the subsystem or subject they belong to. Next to that, key words which are relevant to the reference are added as tags and a description of the reference can be given if necessary. These points will improve the effectiveness of the search function.

Citation Key and PDF Name

To make the references coherent, the citation key should be according to a certain format. This format can be arbitrary but should be recognisable for everyone and be intuitive. It was therefore chosen to use an author - year citation key. This would be for instance "DeVries2021". In case multiple references are present from the same author and year a suffix of A, B, C etc. is used. This would lead to "DeVries2021A" and "DeVries2021B". In case the reference lacks an author, for example a spec sheet, the abbreviation of the organisation, year and suffix will be used. For example, "NXP-2012A".

To make it easy to find the correct reference document file in a local folder, a similar naming is used for the imported PDFs. The PDF name starts with the citation key followed by the document title. This leads to the following format: "DeVries2021 - Implementation of an automated Systems Engineering Toolset in the DST project". When the references are downloaded from Mendeley to a local folder, the name of the PDF is descriptive and traceable. This also prevents very strange file names from showing up in Mendeley and when they are downloaded by other team members.

DST Theses

To store the DST theses of previous team members in a central location, a separate folder is added to the Mendeley Library.

4

SE Toolset Trade-Off and Implementation

In Section 4.1 the available software to construct the Systems Engineering (SE) Toolset is investigated and a trade-off is performed. Next, in Section 4.2 the set-up and implementation of the SE Toolset in the DST project is discussed.

4.1. SE Toolset Software

In this section the software options to create the configuration item tree are first discussed in Section 4.1.1. Next, the software options to handle the requirements are discussed in Section 4.1.2. After the options are identified, possible combinations are made and the trade-off criteria and scoring are introduced in Section 4.1.3. In Section 4.1.4, the trade-off is described. Lastly, in Section 4.1.5, the trade-off table is shown and a conclusion for the best software combination is drawn.

4.1.1. Configuration item tree options

The configuration item tree shows in a graphical way how the system is structured and is an important part of the systems engineering (SE) approach. For the creation of the configuration item tree there are some different software options. First, there are the programs dedicated to creating diagrams and flow charts. Examples are Microsoft Visio, Lucidchart and Miro. These three options score best in the comparison of G2.com[1] and are therefore taken into consideration for the trade-off.

Another option is to use Microsoft Powerpoint. Although it is not specifically designed to create diagrams and flowcharts, it works quite well for simple and small diagrams. However, for the DST project it was found that with only 4 sub-assemblies and 14 components the maximum slide size was already reached. Next to that, there are no sharing capabilities, besides saving it in a shared drive, and there are no options for automation. Therefore, this option is discarded.

Further options are programs like Capella and Valispace, which are dedicated Model Based Systems Engineering (MBSE) programs. Capella is a free to use software with good functionality. However, to efficiently work with the program, extensive training is required. Valispace on the other hand, has quite high licensing cost, but has a better interface and is easier to work with. Both programs can handle multiple SE tools such as diagrams and requirements. Next to that, they have a high degree of automation which is a key element of the SE Approach. However, these programs could also be too extensive and detailed for the DST project and fail the goal of a set of simple and clear tools.

Microsoft Access is also considered. Even though it is not a program dedicated to creating diagrams and flow charts it is capable of creating these. Since this is a database program with Visual Basic for Applications (VBA) coding behind it, there are options to automate the creation of the configuration item tree.

Lastly, Python is considered as an option. With Python a fully customized program can be constructed, with good interfacing and automation options. It is freely available. A downside is that everything has to be programmed from scratch, which means initially more time and effort are needed.

[1] `https://www.g2.com/categories/diagramming?utf8=%E2%9C%93&order=g2_score`, accessed 20 May 2021

4.1.2. Requirements tool options

The first and most simple requirements tool would be Microsoft Excel. It is well known and is ideal for storing data in a table format. Even though it does not work well for large projects with many and complex requirements with lots of dependencies, it is still a good option for small projects.

A step towards a more automated program would be Microsoft Access, which is available in the Microsoft Office package. The database format suits requirements management very well and a nice interface can be created by using forms. In this way the requirements can be showed in a clear and concise way without losing information or traceability.

Other programs which are dedicated to requirements management are for instance Modern Requirements, Jama Software, Visure and ReQtest. These programs are very well suited for a requirements tool as they are specifically designed for it. The implementation of such tools with other software for creating configuration item trees and budgets is generally also good. However, the licensing cost for this kind of software is quite high.[2] As stated on businessanalystlearnings.com there are plenty of "try-before-you-pay-a-lot offers".[3] Free software or reasonable priced software with good performance is rare. Since requirement management software is expensive, it is discarded as viable option.

The last options considered are Capella, Valispace and Python. They are all discussed in the previous section.

4.1.3. Combinations

From the above discussed programs for constructing a configuration item tree and maintaining requirements, combinations are made. Below, the combinations that are possible are stated and in the following sections they analyzed for the trade-off. For some combinations a SQL server or Sharepoint is needed to handle the integration of the tools and to allow for sharing.

Combination 1: Microsoft Visio - Microsoft Excel - Sharepoint

Combination 2: Lucidchart - Google Sheets/Microsoft Excel

Combination 3: Miro - Google Sheets/Microsoft Excel

Combination 4: Microsoft Access - SQL server/Sharepoint

Combination 5: Capella - Team for Capella

Combination 6: ValiSpace

Combination 7: Python - SQL server

Before the combinations are discussed in detail the trade-off method is discussed. A graphical trade-off is used in combination with Analytic Hierarchy Process (AHP) to determine the weights for the graphical trade-off. The weights are used to set the column width in the graphical trade-off. The following criteria are defined:

- **Availability** - The availability of a program is important to prevent high cost for licenses. If multiple different functions, such as handling requirements and a configuration item tree, can be created in the same program than license costs can be acceptable.
- **Simplicity and Clarity** - Simplicity is important since the goal of the SE approach is to have a simple and clear set of tools, which are easy to use for all team members. It is also important that the program is easy to understand and maintain for future students.
- **Integration** - Integration of the separate tools/programs together is very important to make one set of coherent tools. The tools should support each other and preferably be in one central environment. If a tool can handle both the configuration item tree and the requirements then it has an excellent integration.
- **Automation** - Automation is a key element of the SE approach. One of the goals of the SE approach is to take away most of the manual labor in keeping the SE updated. Therefore, automation is very important in picking the software for the SE Toolset.

[2] `https://www.softwaretestinghelp.com/requirements-management-tools/`, accessed 16 June 2021

[3] `https://www.businessanalystlearnings.com/technology-matters/2017/7/4/a-list-of-free-requirements-management-rm-software`, accessed 16 June 2021

- **Sharing capabilities** - Sharing of the SE tool is important as everyone needs to work with the same data to prevent interface problems, different versions and discrepancies in the design. The sharing needs to be simple and with good functionality.

The resulting weights from the AHP trade-off for the criteria are shown in Table 4.1. The following scores are defined for the graphical trade-off:

- **Unacceptable** - The combination scores unacceptably low on the criterion. Deficits cannot be corrected.
- **Correctable** - The combination has deficits which can be corrected.
- **Acceptable** - The combination meets the requirements and expectations but can still have some deficits.
- **Good** - The combination exceeds the requirements and expectation. Small deficits can still be present.
- **Excellent** - The combination scores exceedingly well on the criterion and has no deficits.

Table 4.1: Table containing the results of the AHP trade-off for the SE Toolset criteria.

Criteria	Criteria Weight
Availability	5.4%
Simplicity and Clarity	10.0%
Integration	29.4%
Automation	30.5%
Sharing Capabilities	24.7%

4.1.4. Trade-off

In this section the graphical trade-off is performed.

Combination 1: Microsoft Visio - Microsoft Excel - Sharepoint
Microsoft Visio can be combined with Microsoft Excel through Sharepoint. A paid version of Microsoft Office is needed together with a separate license for Microsoft Visio. A Microsoft 365 Standard Business license costs 12.50 dollar per month per user.[4] The pricing for a Microsoft Visio pro license, which is needed to integrate Excel with Microsoft Visio, costs 15 dollar per user per month.[5] For larger projects with multiple team members this gets quite expensive. Sharing licenses could be an option, however, this is not preferred. Therefore, combination 1 scores correctable for availability. For simplicity and clarity combination 1 scores excellent. Both programs are simple in use, have a clear overview and different kinds of shapes and diagrams are easily constructed.

The integration of the programs together scores acceptable as multiple separate programs are still needed. Visual Basic for Applications (VBA) can be used in Excel for automation and links to Microsoft Visio are automatically updated. However, since the automation possibilities are not as extensive as other combinations, combination 1 scores acceptable. For sharing capabilities combination 1 also scores acceptable.

Combination 2: Lucidchart - Google Sheets/Microsoft Excel
Lucidchart is a program dedicated to creating diagrams and a free version of the program is available. However, to have more than 3 editable documents a paid version is required.[6] In case of multiple iterations and different configuration item trees this could be limiting. Combined with Google Sheets or Microsoft Excel the availability scores acceptable.

Lucidchart is easy in use and has a clear overview due to which it scores excellent for simplicity and clarity. Lucidchart combined with Google Drive or Microsoft Excel scores good on integration. Lucidchart can import data from a CSV file or Google Sheet and link it to a diagram. The diagram itself should still be shaped and new entries are not automatically put into the diagram. Therefore, combination 2 scores acceptable for

[4] `https://www.microsoft.com/en-ww/microsoft-365/business/compare-all-microsoft-365-business-products?rtc=1&market=af`, accessed 23 November 2021
[5] `https://www.microsoft.com/en-ww/microsoft-365/visio/microsoft-visio-plans-and-pricing-compare-visio-options?market=af`, accessed 23 November 2021
[6] `https://lucid.app/pricing/lucidchart#/pricing` accessed 16 June 2021

automation. The sharing capabilities of combination 2 are scored excellent. Collaborators can be invited to Lucidchart and if integrated with Google Drive the diagrams can be shared and modified through Lucidchart.

Combination 3: Miro - Google Sheets/Microsoft Excel
Miro is a similar program as Lucidchart. The interface is simple with good functionality but not as clear as Lucidchart. Combined with Google Sheets or Microsoft Excel the combination scores good on simplicity and clarity. For availability, combination 3 scores acceptable since a paid version is required when more than 3 editable boards are needed.[7] The sharing capabilities of Miro are good as unlimited team members can join the board. Integration with Google Sheets or Microsoft Excel works well.[8] Google Sheets can be for instance embedded in Miro. Therefore, combination 3 scores good for integration. The combination scores acceptable for automation as the diagrams are not updated automatically when new entries are added.

Combination 4: Microsoft Access - SQL server/Sharepoint
Microsoft Access is included in most of the Microsoft Office packages. For TU Delft students and staff Office 365 is available for free. Otherwise, Office 365 costs 12.50 dollar per month per user for a business license.[9] [10] The availability of Microsoft Access is scored acceptable as Microsoft Office is widely used. If a license needs to be bought, the expenses are acceptable as you get a set of useful programs.

For simplicity and clarity Microsoft Access scores acceptable. The program is easy to use even though some practice is needed to get started with it. There are lots of tutorials available online, which are very helpful. The program provides a clear overview and can show data in a clear way with forms. Using Visual Basic for Applications (VBA), the forms and processing of data can be automated. Therefore, combination 4 scores excellent on automation.

Data can be synced to Sharepoint or a SQL server, called the back-end. The front end is a local or shared Access file which retrieves the data from the linked table. The front-end can be updated with new versions and functionality while still being compatible with the back end and previous versions of the front-end. In Sharepoint, Power Automate can be used to automatically send an email to team members in case of a change or update. Power Automate can also be integrated with Microsoft Teams to post a message in a Teams channel. With a SQL server email notifications can be set if there is a change in the database. Therefore, the sharing capabilities of Microsoft Access score excellent. For integration, combination 4 scores excellent as Microsoft Access can handle diagrams, requirements and budget all in one tool.

Combination 5: Capella
Capella is a dedicated Model Based Systems Engineering (MBSE) software. It is available for free and therefore scores excellent on availability. On simplicity and clarity it scores correctable. The reason for this is that the program does not have a clear overview. It has multiple different tools and layers and is very extensive. To understand the program and to efficiently work with it, extensive training is needed or knowledge of MBSE and modelling language Arcadia/Capella. Even though this would be possible, it would be in conflict with the goal of the SE approach to minimize time and effort. Therefore, Capella scores correctable.

For integration, Capella scores excellent as everything can be done in one program. A model of the system can be created with links between components, functions and interactions. Diagrams of the system architecture and a configuration item tree can be created and can be linked to requirements and budgets. In addition, an add-on, called Requirements Viewpoint, is available which provides the ability to import requirements from a requirements management program and link them to the model.[11] Therefore, Capella also scores excellent on automation.

Sharing capabilities in Capella itself are limited. However, there are add-on options to share the project and model, called Team for Capella. This is an add-on created by ObEO to share and work simultaneously on the same project and model.[12] The downside of this is that it is a paid commercial add-on. Therefore, Capella scores acceptable on sharing capabilities.

[7] `https://miro.com/pricing/`, accessed 16 June 2021
[8] `https://miro.com/marketplace/category/task-automation/`, accessed 15 June 2021
[9] `https://www.microsoft.com/en-ww/microsoft-365/buy/compare-all-microsoft-365-products?market=af`, accessed 16 June 2021
[10] `https://www.microsoft.com/en-ww/microsoft-365/business/compare-all-microsoft-365-business-products?rtc=1&market=af`, accessed 16 June 2021
[11] `https://www.eclipse.org/capella/addons.html`, accessed 15 June 2021
[12] `https://www.obeosoft.com/en/team-for-capella`, accessed 15 June 2021

Combination 6: Valispace
Valispace is a similiar program to Capella but it is not free. Licenses range between 50 to 250 euro per user per month.[13] In comparison to the other programs discussed this is quite excessive. For large companies, that charge 150 euro an hour, this can be acceptable. Having one program with all documentation and tools likely saves at least 1 to 2 hours a month. However, for the DST project these amounts are unacceptable. For educational purposes there are some options to work together with Valispace which will have lower to no cost. However, in that case the DST project would become dependant on Valispace, which is not preffered. Therefore, Valispace scores unacceptable for availability.

For sharing capabilities Valispace is scored excellent. It can be used as a web based application and notifications can be set to see updates of team members.[14] For integration and automation it also scores excellent as requirements and budgets can all be done in the same program and linked together or to the configuration item tree.

For simplicity and clarity the program is scored good. The program has a clear overview and is intuitive to work with. However, no option was found to make a graphical representation of the configuration item tree. The systems and all sub-assemblies and components of the design are structured according to a folder structure. A graphical way shows the relationships and identifier structure better. Therefore it is not scored excellent.

Combination 7: Python - SQL server
With Python a program can be created which shows the requirements and budgets in a structured way and generates the configuration item tree. Since almost everything can be programmed in Python it scores excellent on both automation and integration. A SQL server can be used as back-end which gives it excellent sharing capabilities.

With regards to availability Python scores excellent as the program is available for free and many additional packages are available for free. Regarding simplicity and clarity Python scores correctable. Even though the program performs very well in the other fields, it all needs to be build from scratch. All buttons, forms and interfaces have to be created which takes quite some time. In for instance Microsoft Access, these parts and options are already present and can simply be implemented and customized. Next to that, it would become quite an extensive program with lots of coding. To transfer this to future students and for them to understand, it would take quite some time. For these reasons Python scores correctable for simplicity and clarity.

4.1.5. Trade-off table
In Table 4.2 the graphical trade-off table can be seen.

Table 4.2: SE Toolset trade-off table. Avail = Availability, Simpl = Simplicity

Option \ Criterion	Avail. 5.4%	Simpl.& Clarity 10.0%	Integration 29.4%	Automation 30.5%	Sharing capabilities 24.7%
1: *Microsoft Visio - Excel - Sharepoint*	Corr.	Excell.	Acceptable	Acceptable	Acceptable
2: *Lucidchart - Google Sheets/Excel*	Acc.	Excell.	Good	Acceptable	Excellent
3: *Miro - Google Sheets/Excel*	Acc.	Good	Good	Acceptable	Excellent
4: *Microsoft Access - SQL server/Sharepoint*	Acc.	Accept.	Excellent	Excellent	Excellent
5: *Capella*	Excel	Correct.	Excellent	Excellent	Acceptable
6: *ValiSpace*	Unac	Good	Excellent	Excellent	Excellent
7: *Python - SQL server*	Excel	Correct.	Excellent	Excellent	Excellent

Valispace scores an unacceptable which makes it not suitable for the SE Toolset. Although combinations

[13] `https://www.valispace.com/pricing/`, accessed 16 June 2021
[14] `https://valispace.zendesk.com/hc/en-us/articles/360002130918-Introduction-to-Valispace`, accessed 16 June 2021

1, 2 and 3 score quite well, they score lower on integration and automation than combinations 4, 5 and 7. Therefore, combinations 1, 2 and 3 are also not considered the optimal combination for the SE Toolset. Combinations 4, 5 and 7 score the same for integration and automation, while Capella scores lower for sharing capabilities. Together with the required training to work with the program, it is deemed that Capella is not the most optimal solution for the SE Toolset. Combination 4 and 7 can both use a SQL server, while Microsoft Access can also use Sharepoint as a back-end. For availability Python scores better, while Microsoft Access scores better for simplicity and clarity. For the DST project the availability of Microsoft Access can be scored excellent as a Microsoft 365 license is available to students and staff. Since Microsoft Access has a shorter development time, the decision was made to go with Microsoft Access. The program was combined with Sharepoint as this was freely available through the TU Delft and it was integrated with Microsoft Teams.

4.2. Implementation and Set-up

In this section the implementation and set-up of the SE Toolset is discussed. A detailed guide/manual for the SE Toolset can be found in Appendix B. A couple of screenshots are shown throughout this section to provide a general idea of the program layout. More figures of the screen layout can be found in Appendix B. In Figure 4.1, the overall program structures is shown. A detailed program structure, with all interactions and logic is provided in Section 4.2.1. With the implementation and set-up of the SE Toolset, some requirements were already met. In Chapter 6, the requirements compliance table can be found in Table 6.1.

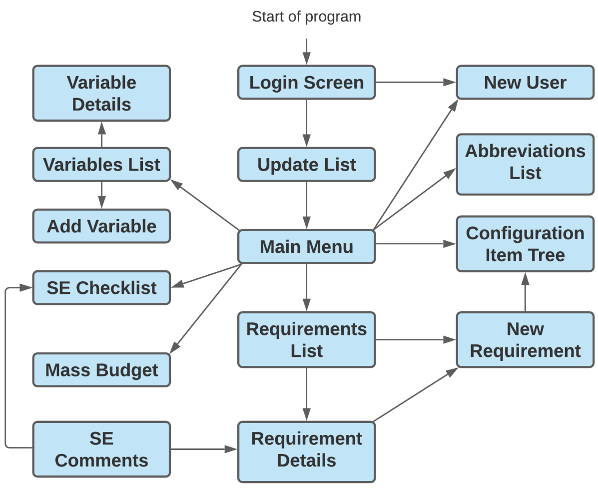


Figure 4.1: Overview diagram of the structure of the SE Toolset.

The SE Toolset is implemented into the DST project via Microsoft Teams and Sharepoint. For the DST team in Microsoft Teams, a Sharepoint storage is available. This Sharepoint storage is used as back-end. The tables created in Microsoft Access can be exported to Sharepoint and a link can be created. When a user changes an entry in Microsoft Access, it is automatically updated in Sharepoint. The change is then also directly updated and visible for other users in their SE Toolset. By storing the front-end Access file on Microsoft Teams, requirement SE-INT-01 is fulfilled. Since the SE Toolset is linked to the Sharepoint storage, the user has to be added to the DST Team to obtain access. Otherwise, an error message will appear that the linked tables cannot be updated. This gives the SE Toolset a good protection layer. Another protection layer is added by setting up user accounts in the SE Toolset. The users can use their NetID and choose a password to create an account. The password is AES 256 encrypted and stored in a linked table. This is the second protection layer. The SE Toolset checks automatically if a user still has access to Sharepoint, otherwise, the user cannot create an account.

When the user logs in, his name is retrieved from Sharepoint and the user abbreviation is stored. This abbreviation is used for all automatically filled fields to show who made a comment or adjustment. The last time the user logged in is also stored. This is used to show the user updates on requirements since his last login, shown in Figure 4.2. Updates are shown for requirements the user watches. Requirements can be watched or unwatched through a watch list. When the user is the owner of the requirement or made a comment on a requirement, the requirement is automatically added to his watch list. A requirement can also be set to "disputed" by a user. The systems engineer will get a notification of disputed requirements, which can then be discussed in the next team meeting. With this functionality requirement SE-COM-05 was fulfilled. With the commenting system communication on requirements can be done through the SE Toolset. Automatic generated comments are added when a linked variable is changed, when a requirement is edited and when a requirement is deleted. These comments provide information on what was changed, by who and what the old value or requirement was. Therefore, requirement SE-COM-04 is also fulfilled.

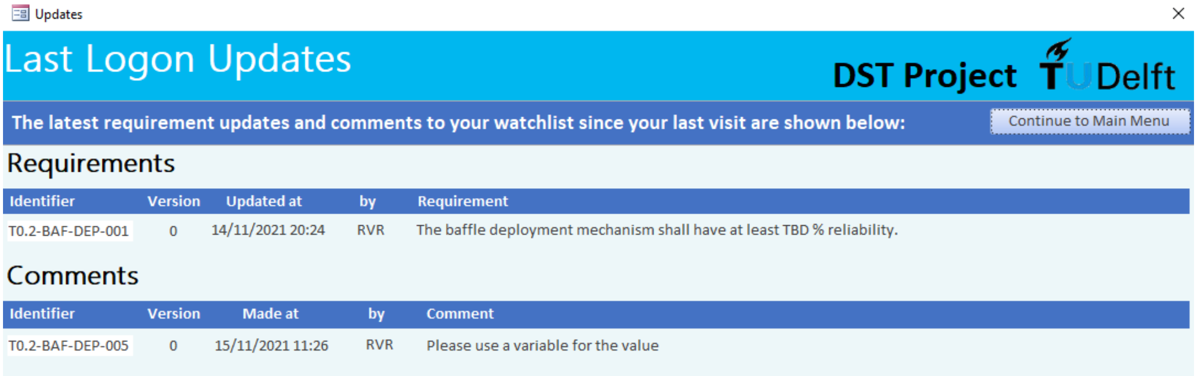


Figure 4.2: Screenshot of the login updates form.

The requirements are structured in a list according to subsystem, shown in Figure 4.3. The list can be filtered by subsystem or search keywords. By double clicking on a requirement the details, previous versions and comments are shown. It is also shown if the requirement is checked by a systems engineer and when. All new requirements and new versions show up in the SE checklist. This list is only accessible to systems engineers. The systems engineer can check if the requirement is correctly formulated and if additional information should be provided. The systems engineer can place a comment through the SE checklist when changes need to be made. The owner of the requirement will get notified through the update system. When the requirement is approved the systems engineer marks the requirement as checked, which can also be seen by the team members. This functionality fulfills requirements SE-INT-02, SE-REQ-02 and SE-REQ-02-01.

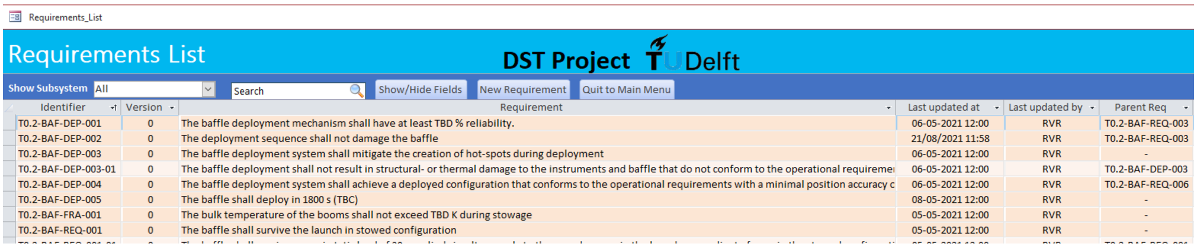


Figure 4.3: Screenshot of the requirements list.

New requirements can be added through a command button in the requirements list. This will open the new requirement form, as shown in Figure 4.4. The following fields are automatically filled in and cannot be changed: owner, creation date and time, last updated by, last update date and time and the requirement comment. Through the configuration item tree the user can select to which sub-system the requirement belongs. The identifier is then automatically generated. The parent requirement can also be selected through the configuration item tree, which in that case is expanded with a requirements list per sub-system. In the requirements detail view new versions of requirements can be created. All fields are automatically filled in and the version number is increased by one. Sub-requirements can also be created here for which the identifier is automatically set to the correct sub identifier. Requirements can also be deleted in the detail view. A reason for the deletion has to be provided after which the requirement comment is set to deleted and the requirement will be greyed out in the requirements list. The requirement will still exist, but cannot be changed anymore, except by a systems engineer. With this functionality, requirements SE-REQ-01-01, SE-REQ-01-02, SE-REQ-01-03 and SE-REQ-01-04 are fulfilled.

A mass budget is present in the SE Toolset, shown in Figure 4.5. This budget is automatically generated from a Sharepoint list containing the data. New component masses can be added and are linked to a variable.

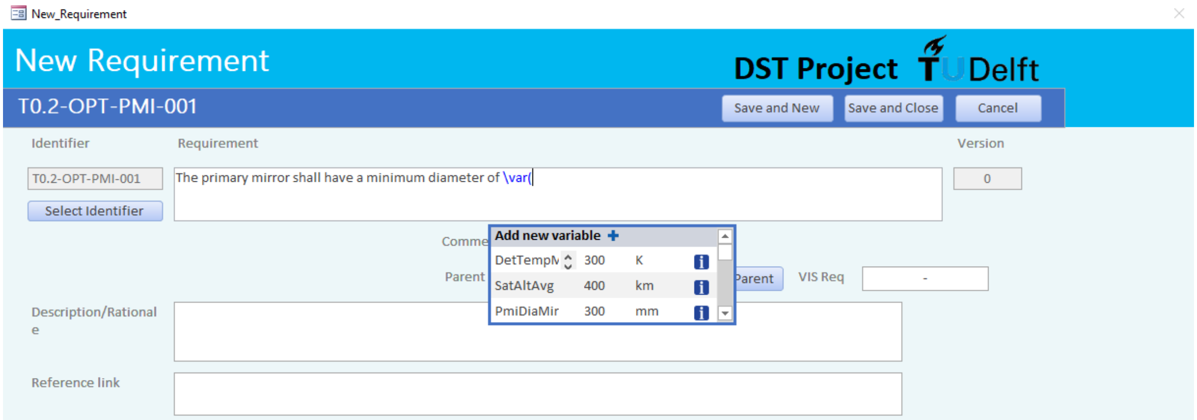


Figure 4.4: Screenshot of the new requirement screen.

This variable can also be linked to a requirement. In case the value of the variable is changed a new version of the linked requirements is created and the mass budget is updated. To select the sub-system to which the component belongs, the configuration item tree is used again. The configuration item tree provides a good overview of the (sub)-systems and is automatically generated from a linked table. More budgets, such as alignment budget, can be added if needed. With the described functionality requirements SE-INT-03, SE-INT-03-01, SE-INT-03-02 and SE-INT-04 are fulfilled. As described in above sections, the SE Toolset automates everything which does not require user input. Therefore, requirement SE-LEA-03 was also fulfilled.

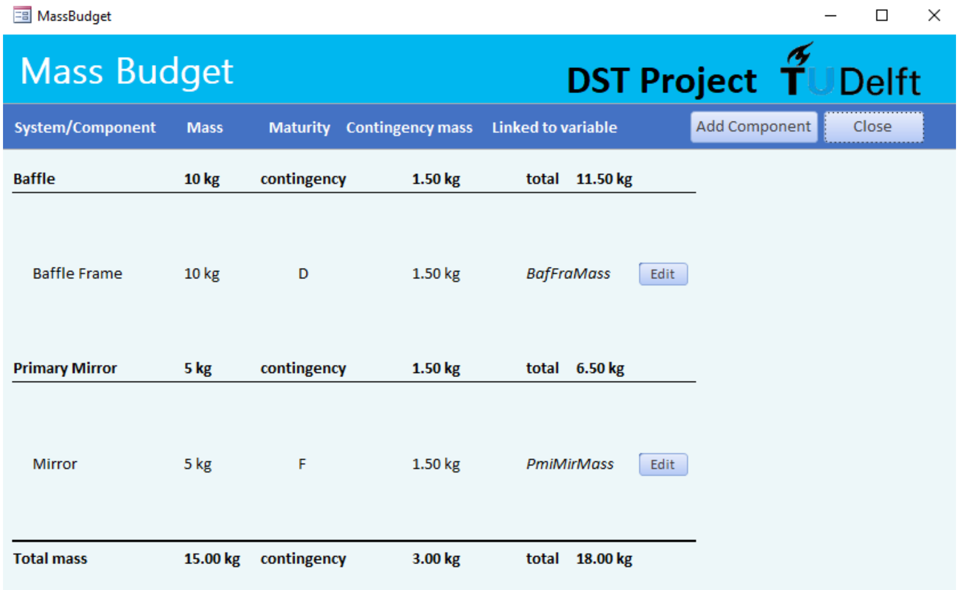


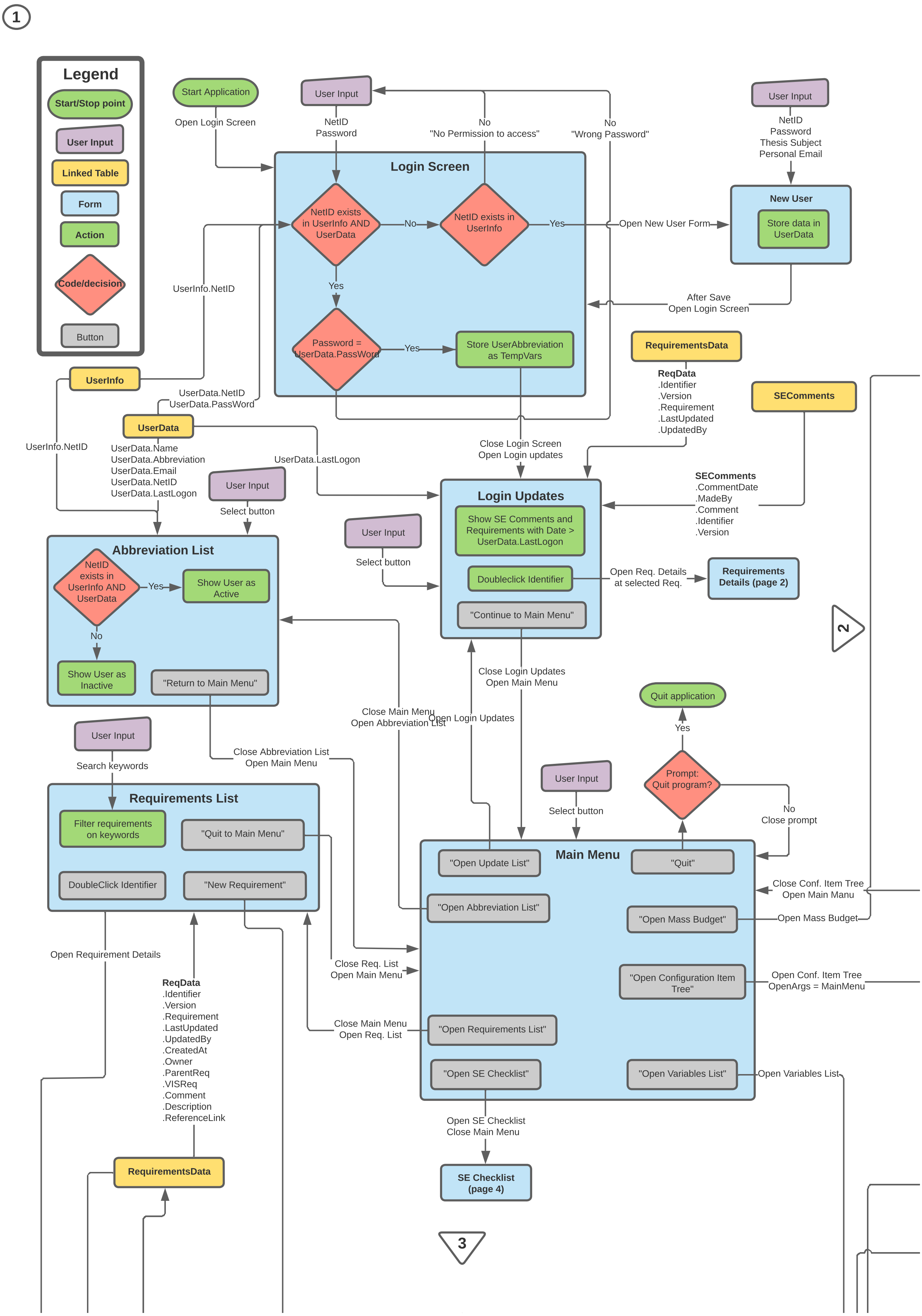
Figure 4.5: Screenshot of the mass budget screen.

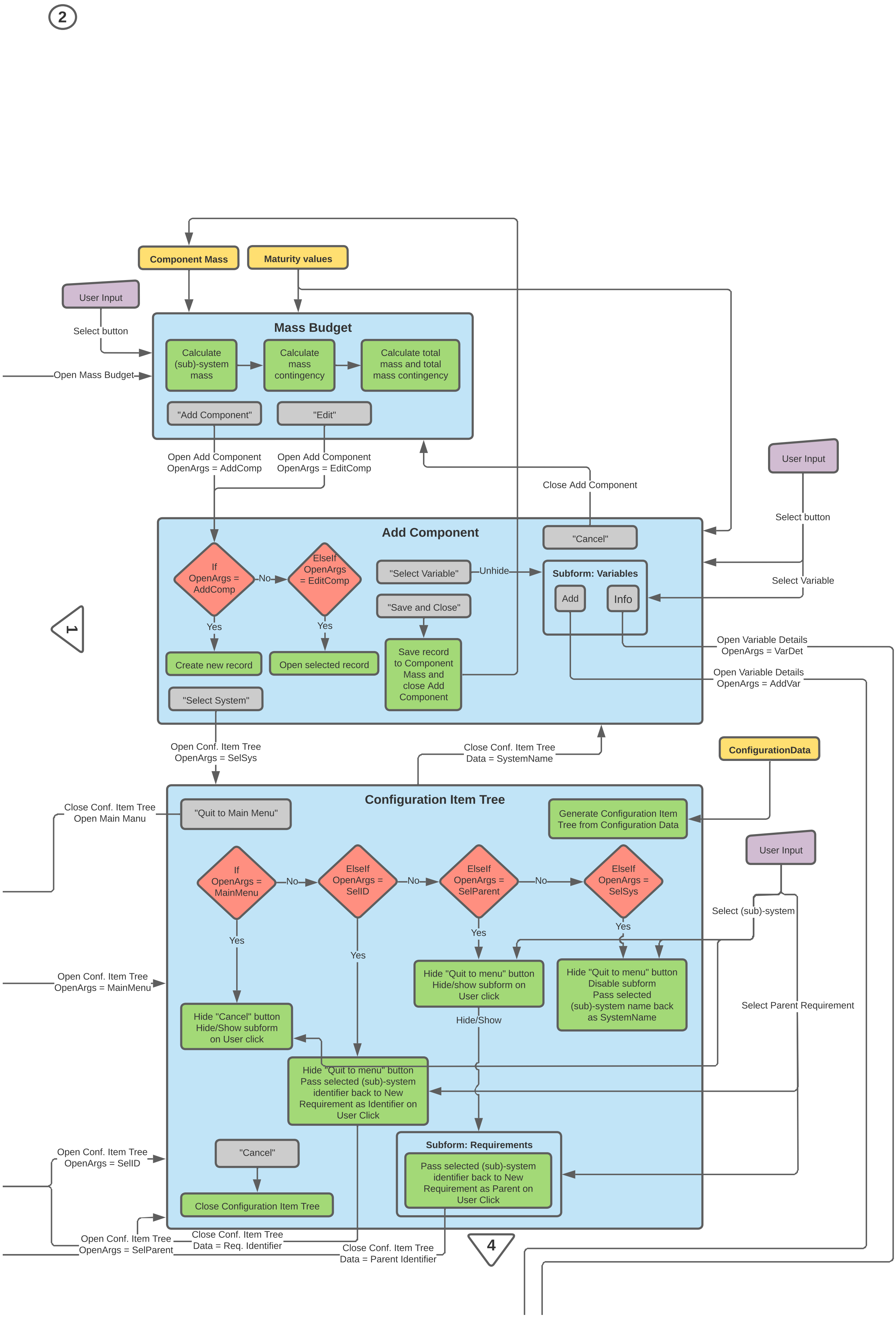
All documentation was stored on a shared drive available from the TU Delft. Information which could be put in the document properties is: document name and identifier, name of the author, date created, date modified, modified by who and version. At first, it was the intention to use this. However, the shared drive from the TU Delft does not allow these fields to be changed. Therefore, this was not an option and it was opted to put this information on the first page of each document. In this way the documentation could already be improved and references should become more traceable. However, since this should still be done manually it is deemed that requirements SE-DOC-01, SE-DOC-02, SE-DOC-03, SE-DOC-04 and SE-DOC-05 are partly fulfilled. SE-DOC-06 is fulfilled with the shared drive.

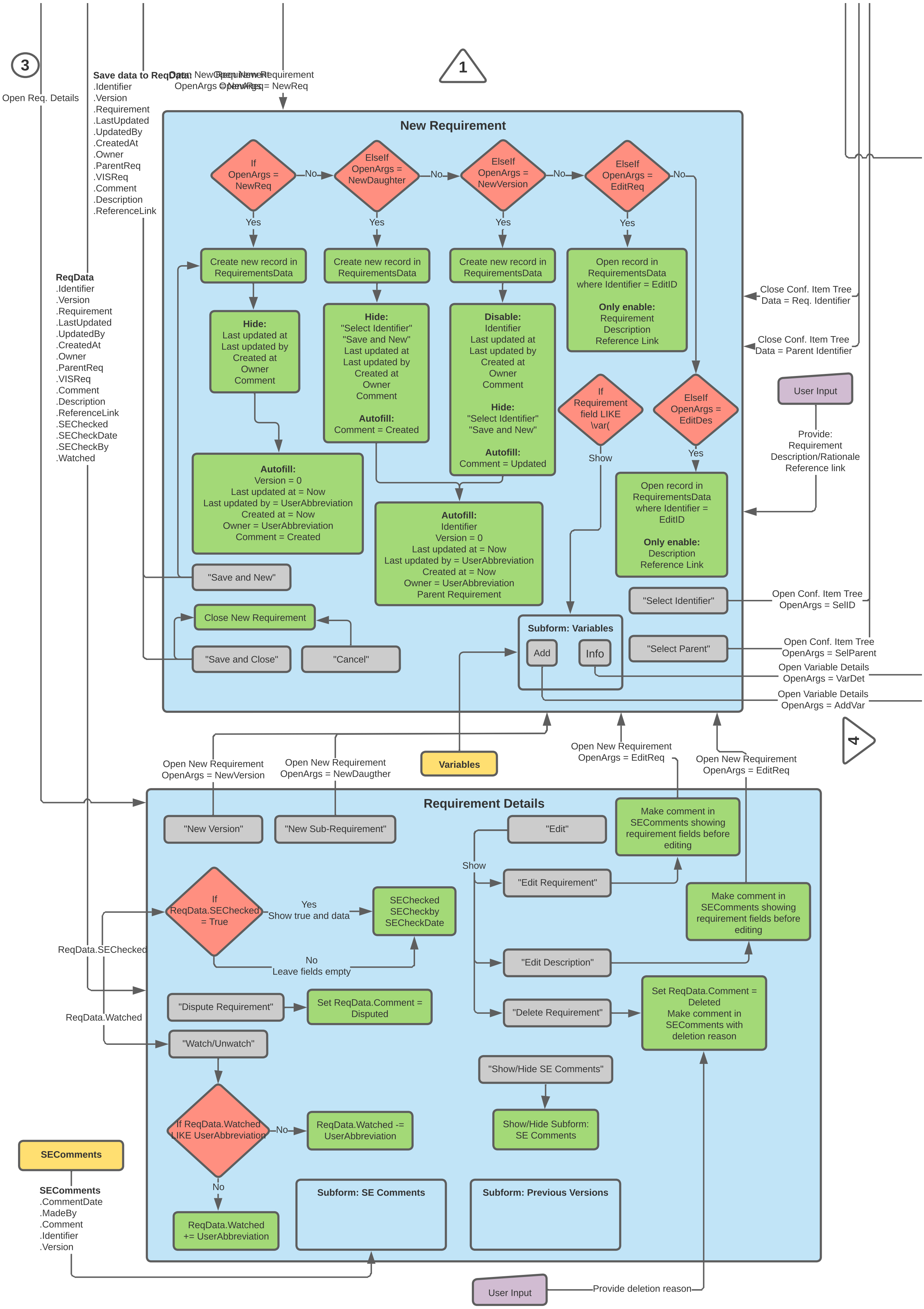
Once the SE Toolset was set up it was important to start using it in the project. However, for the TIR design there were no requirements present yet. Some team members were stuck while waiting for requirements while other team members were using scaled requirements from the visual design. To unify this, it was important to create one set of requirements which was available to all team members. With the SE Toolset this set of requirements could be created and maintained. However, before this could be done, a high level systems engineering analysis was needed to find the best concept for the TIR demonstrator design. This is described in Chapter 5.

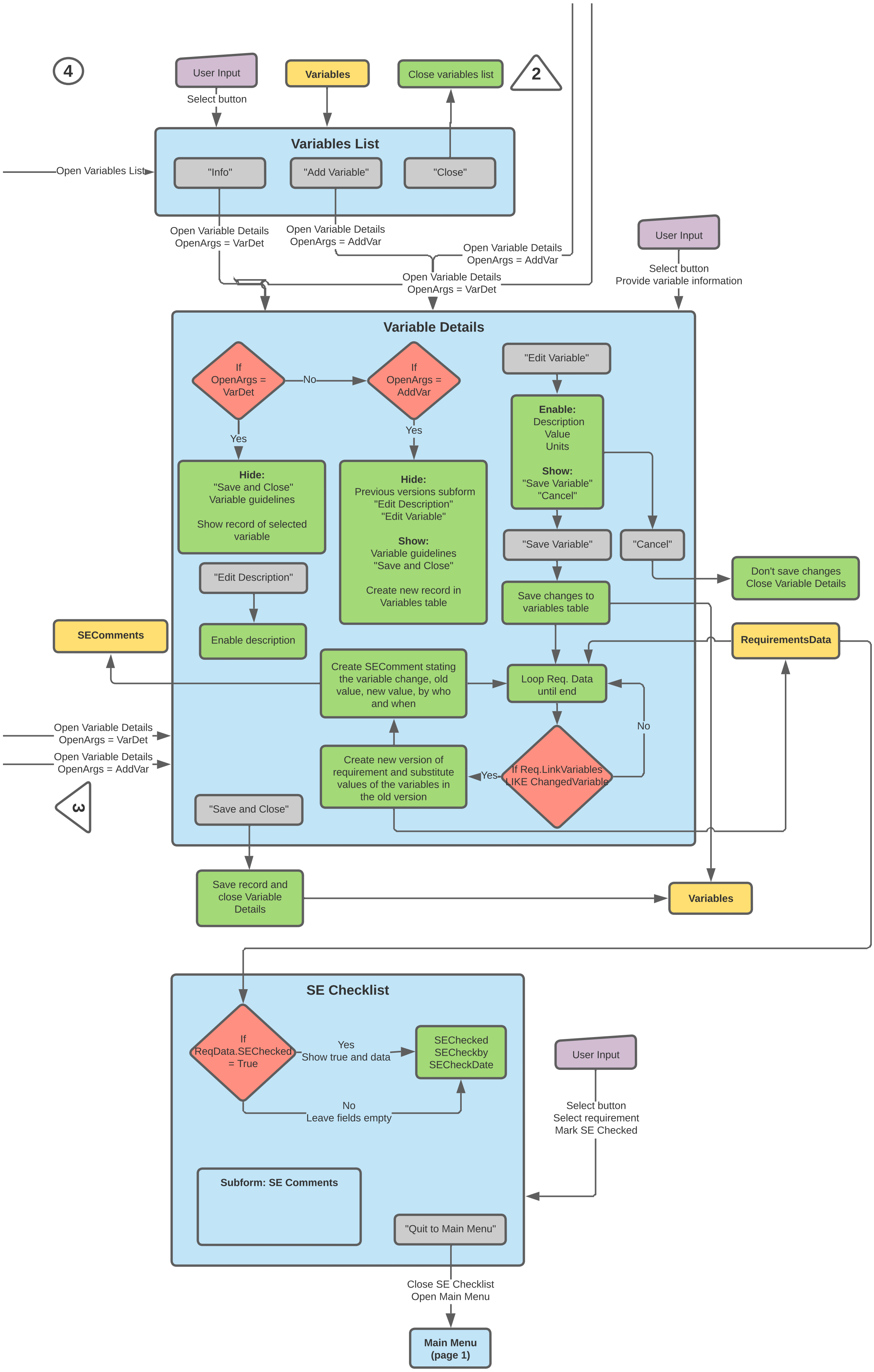
4.2.1. Detailed Program Structure

On the following pages the SE Toolset structure is provided in a diagram. The diagram spans multiple pages. In the top left corner the encircled page number can be found. At the edge of a page a triangle with a number can be found. This indicates which diagram page is located to that side. The diagram consists of four A4 papers, situated as 2x2. The thesis should be printed double sided. The diagram then properly aligns with page 1 and 2 and page 3 and 4 next to each other. The legend can be seen on the first diagram page in the top left corner.


1
Legend
Start/Stop point
User Input
Linked Table
Form
Action
Code/decision
Button
Start Application
Open Login Screen
User Input
NetID
Password
No
"No Permission to access"
No
"Wrong Password"
Login Screen
NetID exists in UserInfo AND UserData
No
NetID exists in UserInfo
Yes
Open New User Form
Yes
Password = UserData.PassWord
Yes
Store UserAbbreviation as TempVars
User Input
NetID
Password
Thesis Subject
Personal Email
New User
Store data in UserData
After Save
Open Login Screen
UserInfo.NetID
UserInfo
UserData.NetID
UserData.PassWord
UserData
UserInfo.NetID
UserData.Name
UserData.Abbreviation
UserData.Email
UserData.NetID
UserData.LastLogon
UserData.LastLogon
RequirementsData
ReqData
.Identifier
.Version
.Requirement
.LastUpdated
.UpdatedBy
SEComments
Close Login Screen
Open Login updates
SEComments
.CommentDate
.MadeBy
.Comment
.Identifier
.Version
User Input
Select button
Login Updates
Show SE Comments and Requirements with Date > UserData.LastLogon
User Input
Select button
Doubleclick Identifier
Open Req. Details at selected Req.
Requirements Details (page 2)
"Continue to Main Menu"
Abbreviation List
NetID exists in UserInfo AND UserData
Yes
Show User as Active
No
Show User as Inactive
"Return to Main Menu"
2
Close Login Updates
Open Main Menu
Quit application
Yes
Close Main Menu
Open Abbreviation List
Open Login Updates
User Input
Search keywords
Close Abbreviation List
Open Main Menu
Prompt: Quit program?
User Input
Select button
No
Close prompt
Requirements List
Filter requirements on keywords
"Quit to Main Menu"
DoubleClick Identifier
"New Requirement"
Main Menu
"Open Update List"
"Quit"
"Open Abbreviation List"
"Open Mass Budget"
Close Conf. Item Tree
Open Main Manu
Open Mass Budget
Open Requirement Details
Close Req. List
Open Main Menu
"Open Configuration Item Tree"
Open Conf. Item Tree
OpenArgs = MainMenu
ReqData
.Identifier
.Version
.Requirement
.LastUpdated
.UpdatedBy
.CreatedAt
.Owner
.ParentReq
.VISReq
.Comment
.Description
.ReferenceLink
Close Main Menu
Open Req. List
"Open Requirements List"
"Open SE Checklist"
"Open Variables List"
Open Variables List
Open SE Checklist
Close Main Menu
SE Checklist (page 4)
RequirementsData
3


2
Component Mass
Maturity values
User Input
Select button
Open Mass Budget
Mass Budget
Calculate (sub)-system mass
Calculate mass contingency
Calculate total mass and total mass contingency
"Add Component"
"Edit"
Open Add Component OpenArgs = AddComp
Open Add Component OpenArgs = EditComp
Close Add Component
User Input
Select button
Select Variable
Add Component
"Cancel"
If OpenArgs = AddComp
No
ElseIf OpenArgs = EditComp
"Select Variable"
Unhide
Subform: Variables
Add
Info
"Save and Close"
Yes
Yes
Create new record
Open selected record
Save record to Component Mass and close Add Component
"Select System"
Open Variable Details OpenArgs = VarDet
Open Variable Details OpenArgs = AddVar
1
Open Conf. Item Tree OpenArgs = SelSys
Close Conf. Item Tree Data = SystemName
ConfigurationData
Close Conf. Item Tree Open Main Manu
"Quit to Main Menu"
Configuration Item Tree
Generate Configuration Item Tree from Configuration Data
User Input
If OpenArgs = MainMenu
ElseIf OpenArgs = SelID
ElseIf OpenArgs = SelParent
ElseIf OpenArgs = SelSys
Select (sub)-system
Yes
Yes
Yes
Yes
Open Conf. Item Tree OpenArgs = MainMenu
Hide "Quit to menu" button Hide/show subform on User click
Hide "Quit to menu" button Disable subform Pass selected (sub)-system name back as SystemName
Select Parent Requirement
Hide/Show
Hide "Cancel" button Hide/Show subform on User click
Hide "Quit to menu" button Pass selected (sub)-system identifier back to New Requirement as Identifier on User Click
Open Conf. Item Tree OpenArgs = SelID
"Cancel"
Subform: Requirements
Pass selected (sub)-system identifier back to New Requirement as Parent on User Click
Close Configuration Item Tree
Open Conf. Item Tree OpenArgs = SelParent
Close Conf. Item Tree Data = Req. Identifier
Close Conf. Item Tree Data = Parent Identifier
4


3
Open Req. Details
Save data to ReqData
.Identifier
.Version
.Requirement
.LastUpdated
.UpdatedBy
.CreatedAt
.Owner
.ParentReq
.VISReq
.Comment
.Description
.ReferenceLink
Open New Requirement
OpenArgs = NewReq
1
New Requirement
If OpenArgs = NewReq
ElseIf OpenArgs = NewDaughter
ElseIf OpenArgs = NewVersion
ElseIf OpenArgs = EditReq
No
Yes
Create new record in RequirementsData
Open record in RequirementsData where Identifier = EditID
Only enable: Requirement Description Reference Link
Hide: Last updated at Last updated by Created at Owner Comment
Hide: "Select Identifier" "Save and New" Last updated at Last updated by Created at Owner Comment
Autofill: Comment = Created
Disable: Identifier Last updated at Last updated by Created at Owner Comment
Hide: "Select Identifier" "Save and New"
Autofill: Comment = Updated
If Requirement field LIKE \var(
Show
ElseIf OpenArgs = EditDes
Open record in RequirementsData where Identifier = EditID
Only enable: Description Reference Link
Autofill: Version = 0 Last updated at = Now Last updated by = UserAbbreviation Created at = Now Owner = UserAbbreviation Comment = Created
Autofill: Identifier Version = 0 Last updated at = Now Last updated by = UserAbbreviation Created at = Now Owner = UserAbbreviation Parent Requirement
"Save and New"
Close New Requirement
"Save and Close"
"Cancel"
Subform: Variables
Add
Info
"Select Identifier"
"Select Parent"
Close Conf. Item Tree Data = Req. Identifier
Close Conf. Item Tree Data = Parent Identifier
User Input
Provide: Requirement Description/Rationale Reference link
Open Conf. Item Tree OpenArgs = SelID
Open Conf. Item Tree OpenArgs = SelParent
Open Variable Details OpenArgs = VarDet
Open Variable Details OpenArgs = AddVar
ReqData
.Identifier
.Version
.Requirement
.LastUpdated
.UpdatedBy
.CreatedAt
.Owner
.ParentReq
.VISReq
.Comment
.Description
.ReferenceLink
.SEChecked
.SECheckDate
.SECheckBy
.Watched
Open New Requirement OpenArgs = NewVersion
Open New Requirement OpenArgs = NewDaugther
Variables
Open New Requirement OpenArgs = EditReq
Open New Requirement OpenArgs = EditReq
4
Requirement Details
"New Version"
"New Sub-Requirement"
"Edit"
Make comment in SEComments showing requirement fields before editing
Show
"Edit Requirement"
Make comment in SEComments showing requirement fields before editing
If ReqData.SEChecked = True
Yes Show true and data
SEChecked SECheckby SECheckDate
No Leave fields empty
ReqData.SEChecked
"Edit Description"
"Dispute Requirement"
Set ReqData.Comment = Disputed
"Delete Requirement"
Set ReqData.Comment = Deleted Make comment in SEComments with deletion reason
ReqData.Watched
"Watch/Unwatch"
"Show/Hide SE Comments"
Show/Hide Subform: SE Comments
If ReqData.Watched LIKE UserAbbreviation
ReqData.Watched -= UserAbbreviation
ReqData.Watched += UserAbbreviation
SEComments
SEComments
.CommentDate
.MadeBy
.Comment
.Identifier
.Version
Subform: SE Comments
Subform: Previous Versions
User Input
Provide deletion reason


4
User Input
Select button
Variables
Close variables list
2
Variables List
Open Variables List
"Info"
"Add Variable"
"Close"
Open Variable Details
OpenArgs = VarDet
Open Variable Details
OpenArgs = AddVar
Open Variable Details
OpenArgs = AddVar
Open Variable Details
OpenArgs = VarDet
User Input
Select button
Provide variable information
Variable Details
If OpenArgs = VarDet
No
If OpenArgs = AddVar
Yes
Yes
"Edit Variable"
Enable:
Description
Value
Units
Show:
"Save Variable"
"Cancel"
Hide:
"Save and Close"
Variable guidelines
Show record of selected variable
Hide:
Previous versions subform
"Edit Description"
"Edit Variable"
Show:
Variable guidelines
"Save and Close"
Create new record in Variables table
"Save Variable"
"Cancel"
Don't save changes
Close Variable Details
"Edit Description"
Save changes to variables table
Enable description
SEComments
RequirementsData
Create SEComment stating the variable change, old value, new value, by who and when
Loop Req. Data until end
No
Open Variable Details
OpenArgs = VarDet
Open Variable Details
OpenArgs = AddVar
Create new version of requirement and substitute values of the variables in the old version
Yes
If Req.LinkVariables LIKE ChangedVariable
3
"Save and Close"
Save record and close Variable Details
Variables
SE Checklist
If ReqData.SEChecked = True
Yes
Show true and data
SEChecked
SECheckby
SECheckDate
No
Leave fields empty
User Input
Select button
Select requirement
Mark SE Checked
Subform: SE Comments
"Quit to Main Menu"
Close SE Checklist
Open Main Menu
Main Menu
(page 1)

5

Top-Level Systems Engineering

After the SE Toolset was implemented, the other two needs of the DST project on top-level systems engineering (SE) were investigated. Therefore, a top-down concept trade-of, based on the goals and development of the DST project and for the DST TIR demonstrator specifically, is described in this chapter. Since the last top-level systems engineering (SE) was performed in 2018, and a switch to TIR was ongoing, it was important that the high level SE was redefined for the new design. At the start of the TIR Deployable Space Telescope (DST) design, it was assumed that the TIR DST design would have a similar layout and deployment concept as the visual (VIS) design, as shown in Figure 5.1. With the switch from a 1.5m aperture to a 30cm aperture for the demonstrator mission, the question arose if the VIS layout was still the most optimal for the demonstrator or that a different configuration was better. This was investigated with an analysis of the different deployment options for the primary mirror (M1), secondary mirror (M2) and baffle and combining these options into concepts. The reference satellite comparison is also discussed in this chapter.

First, the need and mission statement for the DST TIR demonstrator are defined in Section 5.1. Next, the impact of the change from 1.5m aperture to 30cm aperture is discussed and its relevance to the different deployment techniques and stowed volumes. Then, in Section 5.3, the concept generation approach is discussed and the concepts are introduced. In Section 5.4, the relevant parameters for the concepts are defined. In Section 5.5, the concepts are described in detail and the graphical trade-off score per concept is determined. In Section 5.6, the resulting parameters for the concepts are shown together in a table and the concepts are compared on volume. Next, in Section 5.7, the DST concepts are shown in perspective with reference satellites. Once the feasible concepts are identified, the trade-off method, criteria and scoring are defined in Section 5.8. Lastly, in Section 5.9, the graphical and AHP trade-off results are shown and discussed and a conclusion is drawn.

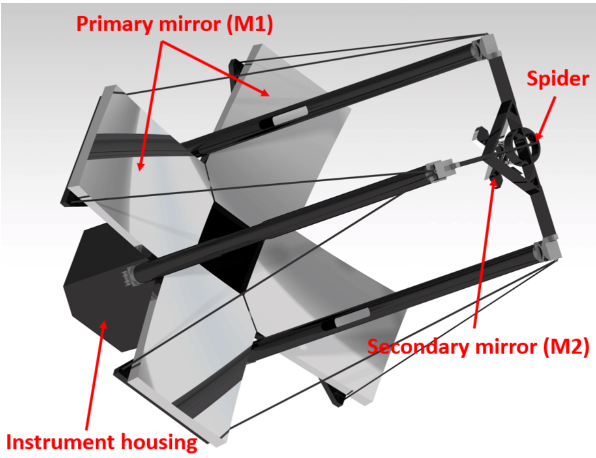


Figure 5.1: Render of the visual (VIS) DST design. The baffle is not shown in the image.

5.1. Need and Mission Statement

With the change from a visual (VIS) design towards a thermal infrared (TIR) design, the need and mission statement had to be reformulated. The need statement provides the necessity why this new design needs to be made. The mission statement defines what the mission is about and how it can satisfy the need. The mission statement was both defined for the DST project on the long term and for the demonstrator mission.

Need Statement
There is a need in the defence and safety & security sector within the Netherlands and Europe to have high spatial resolution telescopes at low cost.

Mission Statement DST Project
Develop high spatial resolution thermal infrared telescopes with deployable optics to have low manufacturing and launch cost.

Mission Statement Demonstrator
Demonstrate that deployable optics can have high spatial resolution relative to their stowed volume.

The focus of the DST project is to develop high spatial resolution TIR imagery at low cost, in terms of launch and serial production. With the demonstrator mission, the concept of deployable optics can be proven and interest and funding can be acquired for the development of a larger version, e.g. 1.5m aperture. A potential that deployable optics have, is to overcome the restrictions of a payload fairing. However, this will only be applicable when apertures larger than 5 to 6 meters are needed, which is not expected to happen on a short term, except for the James Webb Telescope. Therefore, this is not the focus of the DST project, but it is valuable to mention as the DST project could inspire the development towards deployable telescopes.

5.2. Impact of Smaller Aperture

With the change from the visual (VIS) spectrum towards the thermal infrared (TIR) spectrum, a change in aperture was also made. At first, the TIR DST design would have an aperture of 1.5m. However, this would become quite expensive and a smaller demonstrator was needed to prove the concept and obtain funding for a 1.5m design. Another reason for a smaller aperture was that this could provide a stepping stone in the development of a 1.5m design.

In discussions with stakeholders, a 30cm aperture was first considered. However, the ground resolution was expected to be too low after which a 40cm aperture design was proposed. In a later discussion this was deemed to expensive by the stakeholders and a 20cm aperture design was proposed. A wavelength change to 2.5μm - 5μm was also proposed. Since the stakeholder requirements were changing frequently, a decision had to be made for the thesis students to be able to continue their work. With all these changes, they were doing unnecessary work or were waiting for input from the stakeholders. Therefore, the team decided to set a baseline to work with. This baseline was set to a compromise in aperture size of 30cm. It was also decided to stick with the wavelength range of 8μm - 12μm, as the initial optical design was designed for this wavelength.

The smaller aperture also influences the feasibility of some deployment concepts. In the VIS design both the primary mirror (M1), secondary mirror (M2) and baffle were deployed. For a 30cm aperture, it could be more effective to have a fixed M1 or fixed M2. The use of a 90 degree fold mirror in the optical design became also possible according to DST's optical engineer D. Dolkens. For the baffle the change to 30cm aperture meant that linear, telescopic and folding flaps became more feasible. The VIS design was always designed to minimize stowed volume in all three directions. However, a "flat pancake" or long slender rod could perhaps be more efficient in terms of stowed volume. Fixing M1 or M2 would also mean a lower complexity. For the demonstrator this was particularly interesting as it should be reliable while still having a good deployment ratio. To map the possible options a design option tree was made and possible concepts were identified. This is described in more detail in the following sections.

5.3. Concept Generation

To identify all options for the mirror deployment and the baffle deployment, a design option tree (DOT) was created and is shown in Figure 5.2. First, in Section 5.3.1, the options in the DOT are described. In Section 5.3.2, some options are eliminated because they were deemed infeasible. Lastly, the possible combinations of options are described in Section 5.3.3.

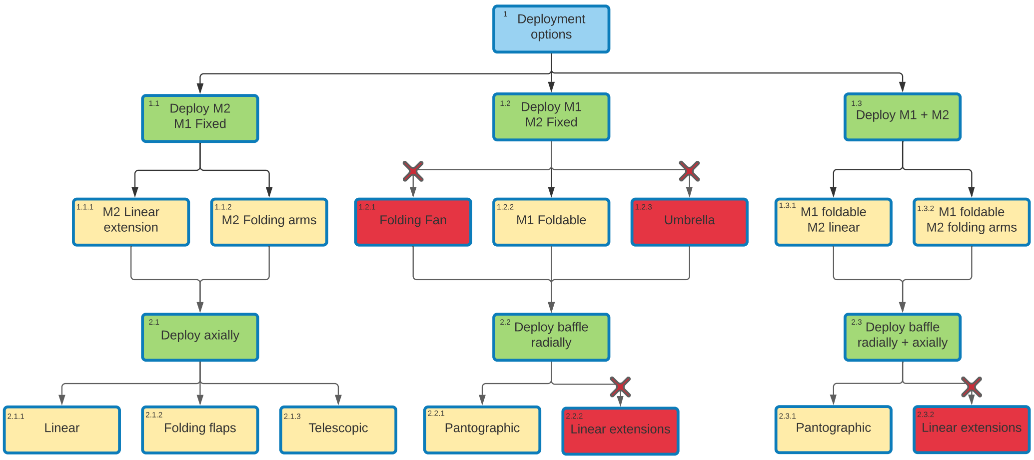


Figure 5.2: Design option tree for primary mirror (M1), secondary mirror (M2) and baffle deployment. Eliminated options are crossed out and marked red.

5.3.1. Design Option Tree

In Figure 5.2, the design option tree for the primary mirror (M1), secondary mirror (M2) and baffle deployment is shown. For the mirror deployment there are three options. The first option is to deploy M2 and have a fixed M1. The second option is to deploy M1 and have a fixed M2. The last option is to have a combination of M1 and M2 deployment.

For M1 deployment there are three options. First of all, M1 can be folded to the top or bottom, as shown in Figure 5.3. Other options are a folding fan or to unfold M1 like an umbrella, as shown in Figure 5.4 and 5.5, respectively.

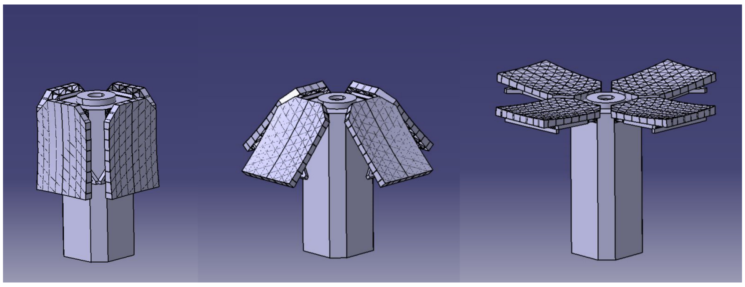


Figure 5.3: Primary mirror (M1) folding deployment. Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.

Figure 5.4: Deployment example of a folding fan lamp. Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.[1]

[1]https://www.red-dot.org/project/ryun-circular-folding-fan-lamp-26820

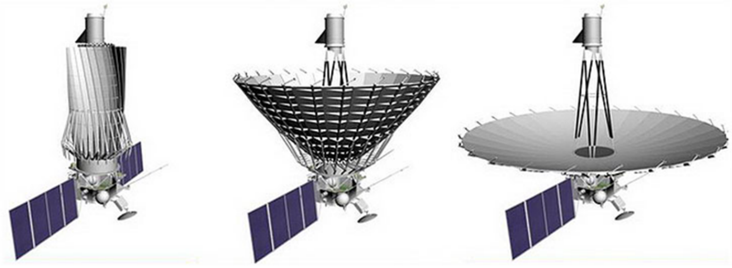


Figure 5.5: Radially unfold M1 mirror like an umbrella, example of Spektr-R.[2]

For the M2 deployment there are two options, either linear extension or folding arms, shown in Figure 5.6 and 5.7, respectively. Linear extension can be done with for instance linear actuators. Folding arms also extend M2 in a linear motion, but do this by a folding mechanism. For the deployment of both M1 and M2, combinations of the aforementioned M1 deployment with either linear extension or folding arms for M2 deployment are possible.

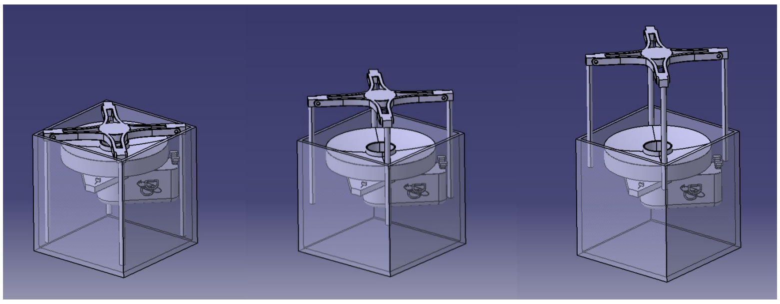


Figure 5.6: Linear deployment of secondary mirror (M2). Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.

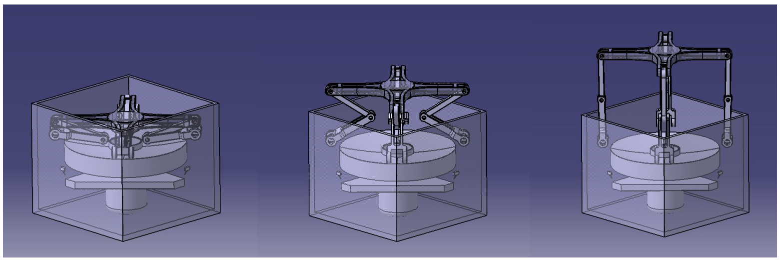


Figure 5.7: Folding deployment of secondary mirror (M2). Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.

The baffle deployment is dependant on the deployment configuration of the mirrors. In case M2 is deployed and M1 is fixed, the baffle only needs to deploy in the axial direction (vertical). For this case there are two options. The baffle can be extended linearly, by a telescopic design or one linear extending baffle section as shown in Figure 5.8 and 5.9, respectively. The second option is to fold the baffle from the sides of the spacecraft up into a vertical position as shown in Figure 5.10.

[2]`http://www.russianspaceweb.com/spektr_r_mission.html`, accessed 5 September 2021

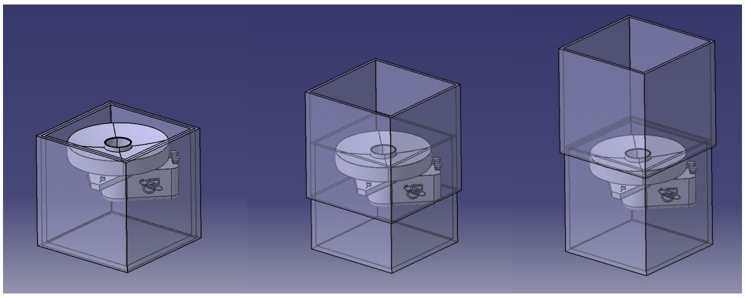


Figure 5.8: Linear deployment of the baffle. Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.

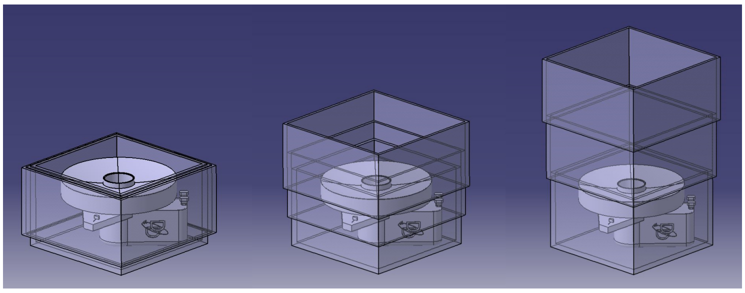


Figure 5.9: Telescopic deployment of the baffle. Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.

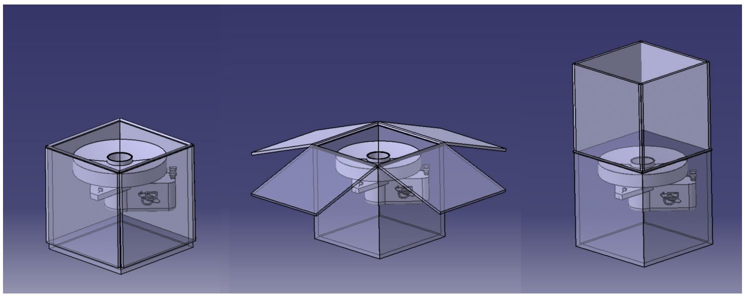


Figure 5.10: Flap deployment of the baffle. Left: Stowed configuration. Middle: During deployment. Right: Deployed configuration.

When M1 is deployed and M2 is fixed, the baffle only needs to be deployed in radial direction. The baffle can be deployed by a pantographic structure or by linear extension. In case both M1 and M2 are deployed, the baffle needs to be deployed in both radial and axial direction. This makes the baffle more complicated, but can again be done with a pantographic structure or by linear extension as was shown by Nagy [18] and Arink [4]. Their designs are shown in Figure 5.11 on the left and right.

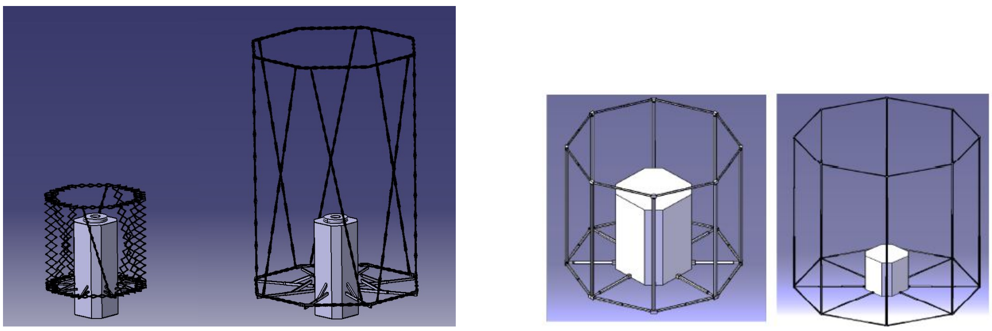


Figure 5.11: Left: Pantographic deployment of the baffle as designed by V. Nagy. [18] Right: Linear extending baffle as designed by J.W. Arink. [4]

5.3.2. Deployment Options Elimination

In order to reduce the total number of concepts, the non-feasible deployment options were eliminated. The following deployment options were eliminated:

- Folding Fan
- Twist M1
- Deploy baffle radially with linear extension

Folding Fan (1.2.1) - The deployable fan is a concept in which the primary mirror (M1) is deployed like a folding fan. An example of a folding fan lamp is shown in Figure 5.4.[3] With this configuration M1 can be made quite small. However, for this concept M1 needs to be made out of a foil like material. Currently, the mirrors are made from solid materials, such as beryllium, aluminum, silicon carbide or fused silica, and require a certain thickness for stability and to reach high accuracy. [21]

One option would be to use a photon sieve membrane, as described by Dearborn [9] and Geoff [3]. This is a thin membrane which can be deployed like a fan. However, this would mean that there is no need for mirrors anymore and the optical design would drastically change. This requires a high order of redesign work, which is not feasible in the resource limited environment of the DST project. Furthermore, research should be done into the applicability of a photon sieve membrane for TIR applications and if it's technology readiness level (TRL) is sufficient. For these reasons, the folding fan deployment is eliminated.

Umbrella (1.2.3) - The umbrella concept is based on the deployment mechanism used in the radio telescope antenna of Spektr-R.[4] A graphic representation of the deployment is shown in Figure 5.5. The image is taken from the Spektr-R satellite which was a astrophysical space observatory with a 10m diameter antenna reflector. For this concept a foil like material is also required for the primary mirror, for which a photon sieve membrane can also be used. However, for the same reasons mentioned before, this would lead to a full redesign. For these reasons this option is discarded.

Linear Extensions (2.2.2 and 2.3.2) - As shown by V. Nagy [18] the pantographic structure is better than telescopic or linear extensions for radial deployment and radial and axial deployment combined. One of the reasons is that the actuation of the linear extensions requires an extra actuator and the segments can get stuck if they get skewed. This is especially troublesome in the DST design of J.W. Arink [4] as there are many sliding segments in multiple directions. For these reasons, linear extensions as a baffle deployment are discarded.

5.3.3. Feasible concepts

From the design option tree (DOT), shown in Figure 5.2, the following combinations are identified:

Deploy M2 with M1 fixed and deploy the baffle axially:

1.1.1 + 2.1.1: M2 linear extension and linear baffle deployment

1.1.1 + 2.1.2: M2 linear extension and folding baffle flaps

1.1.1 + 2.1.3: M2 linear extension and telescopic baffle

1.1.2 + 2.1.1: M2 folding arms and linear baffle deployment

1.1.2 + 2.1.2: M2 folding arms and folding baffle flaps

1.1.2 + 2.1.3: M2 folding arms and telescopic baffle

Deploy M1 with M2 fixed and deploy the baffle radially:

1.2.2 + 2.2.1: M1 foldable and pantographic baffle

Deploy M1 + M2 and deploy the baffle axially and radially:

1.3.1 + 2.3.1: M1 foldable, M2 linear extension and pantographic baffle

1.3.2 + 2.3.1: M1 foldable, M2 folding arms and pantographic baffle

[3] https://www.red-dot.org/project/ryun-circular-folding-fan-lamp-26820, accessed 16 September 2021
[4] http://www.russianspaceweb.com/spektr_r_mission.html, accessed 5 September 2021

5.4. Parameters for Concepts

Before the concepts are shown in detail, some parameters which are used to compare the concepts are defined first. For each concept the following parameters are defined:

Optimized stowed volume is the volume of the concept when the optimized shape is used. In the first iteration the concepts were optimized for minimum volume and best shape to use minimal space when launched in bulk. These concepts can be seen in Appendix D. However, since it is far in the future that constellations will be launched, the concepts where optimized for minimum volume in box shape in the second iteration. The volume of the second iteration is defined as the *boxed stowed volume*. The reason to move towards box shapes is that the available space in a launcher is generally a box shape due to the integration to the launcher. If in the future constellations of the DST are launched, an optimized shape can be beneficial to effectively use all the available space in the launchers payload fairing. For this reason the optimized gain is calculated as the percentage of volume saved by using an optimized shape design.

Boxed stowed volume is the stowed volume of the concept when a box shape is used.

The spacecraft (s/c) integrated bus volume is the volume below the primary mirror (M1), which is available to house other subsystems than the instrument housing. The instrument payload is housed in the box shown below M1. Additional electronics for image processing are not part of this volume and take up part of the s/c bus volume.

The empty volume is defined as volume which cannot be used for the spacecraft bus and which does not house more than one component. An example is concept 3 "M2 folding arms and linear baffle deployment", described in Section 5.5, in which the baffle extends above the spider because the baffle has to have a certain length. This volume is only necessary for the baffle and cannot be allocated to the s/c bus. Therefore this is deemed empty space.

In order to compare the concepts with each other deployment ratios are defined with respect to a fixed reference telescope. To define the deployment ratios, a fixed reference telescope was needed and an estimation of the TIR DST satellite mass and volume was needed. Therefore, available modular s/c busses were investigated. A modular s/c bus means that the DST is added as a box with a predefined electro-mechanical interface on one side, onto a satellite platform. The LEOS-50 platform is shown in Figure 5.12 for reference. In Appendix C, in Table C.1, the investigated s/c busses are shown.

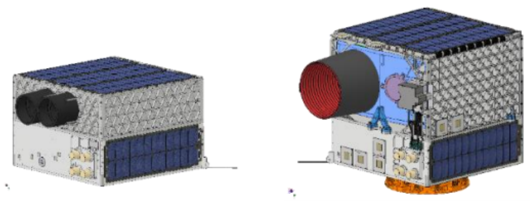


Figure 5.12: Image of the LEOS-50 platform, by Berlin Space Technologies, with different payloads.[5]

The TIR DST instrument was estimated to have a mass of 25kg. This was found by scaling the mass of the visual (VIS) DST instrument linearly by a factor 5, as the aperture is a factor 5 smaller. The VIS DST instrument mass was found to be approximately 125kg from excel sheet "20181121 - SE document - Adapted", found on the DST shared drive. It was found that Altair 27u, MP42, LEOS-50HR and Defiant have the right payload mass range for the TIR DST instrument. Although MP42 has an indicated mass range of 40-50kg it is assumed this can also be lower. The data for Altair 27u, MP42, LEOS-50HR and Defiant is shown in Table 5.1. Taking the average of the s/c busses in Table 5.1, a s/c bus mass of 28.8kg was found and a s/c bus volume of 37 liters. Combining this with the estimate for the TIR DST instrument mass, a total spacecraft mass of 53.8kg was found.

[5] https://www.berlin-space-tech.com/wp-content/uploads/2020/07/PFR-PR28_LEOS-50__V1.00_.pdf, accessed 20 December 2021

Table 5.1: Table containing data on Altair 27U, MP42, LEOS-50HR and Defiant spacecraft busses.

S/c bus	Sys dim [m]	Sys vol [l]	S/c bus vol [l]	Payload vol [l]	S/c bus mass [kg]	Payload mass [kg]	S/c bus ratio
Altair 27U	0.36*0.36*0.37	48	17.1	30.9	20	up to 50	0.36
MP42	0.48*0.48*0.47	108	35	73.7	30	40-50	0.32
LEOS-50HR	0.57*0.57*0.6	195	65	130	45	15-30	0.33
Defiant	0.36*0.36*0.45	57	31	26	20	up to 30	0.54

To find a suitable fixed reference telescope, existing and currently developed telescopes where investigated. The complete table with reference satellites is shown in Appendix C, in Table C.2. From this analysis, Satellogic-NuSat was found to be the closest competitor.[6] Satellogic-NuSat has an operating band between 8 μm and 14 μm, a ground resolution of 90 meter, a swath width of 92km, a mass of 37kg and a volume of 145 liters. The s/c bus volume of Satellogic-NuSat was estimated from figures and the internal architecture, shown in Figure 5.13. It was estimated that from the box shaped volume only 80% consists out of components and that 1/3 of this 80% is allocated to the s/c bus. This leads to a s/c bus volume of 38.7 liters. This is close to the s/c bus volume found for the TIR DST. The mass of Satellogic-NuSat is lower than the estimation for the TIR DST, approximately 31% lighter. Since there are not much satellites found that operate between 8 μm and 12 μm, Satellogic-NuSat was found to be the most suitable reference satellite. Most likely there are more satellites operating in this wavelength range, but they are expected to be military satellites for which data is not publicly available.

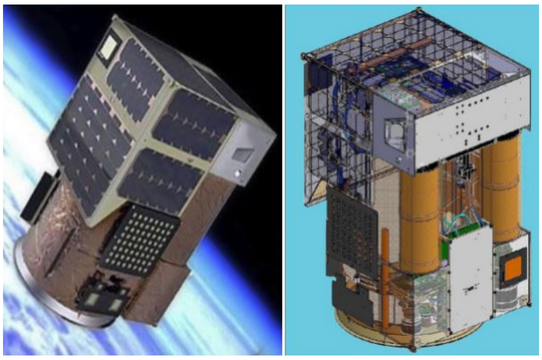


Figure 5.13: Left: Satellogic-NuSat render. Right: Internal architecture of Satellogic-NuSat. [7]

With the values for a fixed reference telescope defined the following deployment ratios can be calculated:

The instrument deployment ratio is the volume of a fixed telescope minus it's integrated s/c bus volume divided by the stowed volume of the concept minus it's integrated s/c bus volume. The latter is also known as the instrument volume. This ratio shows how well the instrument of the concept is stowed and how much volume can be saved with deployable optics. This ratio is used to compare the concepts on instrument level.

Custom s/c bus deployment ratio is defined as the volume of Satellogic-NuSat divided by the concept's instrument volume plus 37 liters to account for custom s/c bus volume. This ratio is used to compare the concepts on spacecraft level.

Modular s/c bus deployment ratio is defined as the volume of Satellogic-NuSat divided by the concept's boxed stowed volume plus 37 liters to account for modular s/c bus volume. This ratio is used to compare the concepts on spacecraft level.

For the custom and modular s/c bus deployment ratio the following grading is used in the graphical trade-off:

- <1.5 - Unacceptable
- 1.5-2.0 Correctable

[6] https://satellogic.com/technology/satellites/ accessed 3 October 2021
[7] https://directory.eoportal.org/web/eoportal/satellite-missions/s/satellogic accessed 3 October 2021

- 2.0-2.5 Acceptable
- 2.5-3.0 Good
- >3.0 Excellent

The minimum deployment ratio which is deemed acceptable is set at 2.0. Between 1.5 and 2.0 is graded correctable. These values are by no means good. However, when the concept is very good in other criteria the concept can still be chosen with the deficit in deployment ratio. If the deployment ratio is below 1.5 the concept is deemed unacceptable. In that case there is not enough gain in stowed volume compared to a fixed satellite to justify the added complexity of deployable optics. Above 2.5 is considered good and above 3.0 is considered excellent.

5.5. Concept Details

The concepts described in Section 5.3.3 are shown and described in more detail in the subsequent sections. To prevent repeating information and to have the data and images close, it was decided to put the reasoning for the graphical trade-off scoring together with the concept details. The graphical trade-off method is described in detail in Section 5.8, on page 52. The concepts are scored for the following criteria:

- **Volume** - Volume is the main driver for a deployable telescope. In particular the deployment ratio with respect to a fixed telescope is important. In case the deployment ratio approaches 1 there is no benefit in the deployment anymore. The volume criteria is separated in two sub-criteria:
    - Custom s/c bus volume: If a custom spacecraft bus is used then the integrated s/c bus volume can be allocated to house the s/c bus.
    - Modular s/c bus volume: In case the TIR DST design will fly on a modular spacecraft bus, then the integrated s/c bus volume of the concept cannot be used to house the s/c bus. In this case it can only be used to house electronics for the optical design. For designs with high integrated s/c bus volume this means that they score bad on modular volume and good on custom s/c bus volume. An example of a custom and modular s/c bus is shown in Figure 5.14.

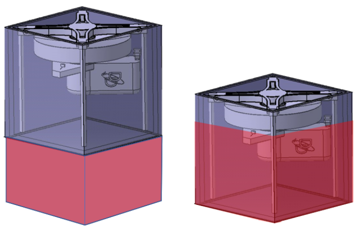


Figure 5.14: Graphical example of a DST concept with on the left a modular s/c bus configuration and on the right a custom s/c bus configuration. The red marked area is the s/c bus.

- **Complexity** - For a demonstrator the complexity should be reduced as much as possible without losing performance. For complexity sub-criteria are the number of deployment systems a concept needs and what the Technology Readiness Level (TRL) is of components which are needed for the concept.
- **Scalability** - Scalability is defined as how well a concept is scalable to a smaller or larger design. Since the focus is now on a small demonstrator, scalability to a larger DST design is considered most important. The potential of a deployable telescope is to overcome the constraints of a payload fairing. For the DST project, with an aperture of 1.5m in the large design, this is not yet relevant. However, proving that the deployable optics are feasible with good ground resolution, is beneficial for the development of deployable optics. In order to limit the waste of research done on the demonstrator when a larger thermal infrared (TIR) DST design is made, scalability is important. Scalability is assessed on the volume increase when the concept is scaled up and if the complexity increases for a larger design.
- **Reliability** - Reliability of the concept is important for a demonstrator as it needs to demonstrate the deployable optics and show the capabilities. The demonstrator also has the purpose to gain more funding for a larger TIR DST design. Therefore, it is important that the demonstrator is simple and reliable.

Reliablity will be quantified by the number of failure points and if there are possibilities for redundancy or not.

The scoring for the criteria is defined as:

- **Unacceptable** - The concept scores unacceptably low on the criterion. There is no room to correct for the bad performance.
- **Correctable** - The concept is not very good on the criteria but with some adjustments or small accepted deficits the concept could work.
- **Acceptable** - The concept meets the minimum required to be accepted for the criteria.
- **Good** - The concept scores above average and has some good aspects. There is still room for improvement and small drawbacks are permitted.
- **Excellent** - The concept scores very good on the criteria. There are no drawbacks or deficits.

The resulting graphical trade-off tables are shown in Table 5.4 and 5.5 on page 54. A summary of the parameters for all concepts is shown in Table 5.2. The following aspects where taken into account in the design of the concepts:

- Aperture size of 300mm
- M1-M2 distance of 262.5mm in deployed configuration
- Baffle to extend 80mm above the spider, as was found by F. Hu

The following assumptions are made:

- Baffle thickness of 10mm
- M1-M2 distance scales linearly with aperture size
- Scale factor of 5 for an increase in aperture from 30cm to 1.5m

Concept 1: M2 linear extension and linear baffle deployment (1.1.1 + 2.1.1)

The first concept is to deploy the secondary mirror (M2) and baffle both by linear extension. From Figure 5.15, it can be seen that in this case the linear extension of both M2 and the baffle are governing for the height. While the width and length can be small with this concept, the height is larger than necessary to house the optical elements. The unused volume below the primary mirror (M1) can be allocated to an integrated s/c bus. Therefore, the concept has no empty volume.

Parameter	Stowed	Deployed	Units
Height	0.350	0.680	m
Width	0.345	0.345	m
Length	0.345	0.345	m
Box shape volume	41.7	80.9	l
S/c bus volume	18.9	18.9	l
Empty volume	-	-	l
Custom depl. ratio	-	2.43	-
Modular depl. ratio	-	1.84	-
Custom s/c bus gain	-	31.6	%
Opt gain	-	22.5	%

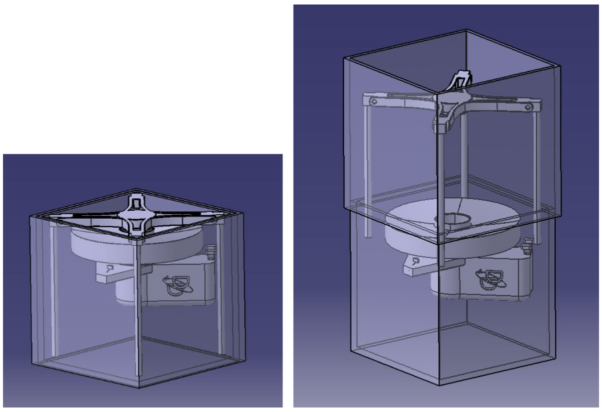


Figure 5.15: Left: Data on concept 1. Right: Model of concept 1 in stowed and deployed configuration.

The concept has a relatively low modular deployment ratio while it has a good custom deployment ratio. With a modular s/c bus, 45% of the stowed volume would be empty volume. The baffle deployment is quite simple and can be actuated by springs and a hold & release mechanism. The M2 deployment is also relatively simple with linear extensions through for instance a lead screw. These components are all easy to come by and have a high Technology Readiness Level (TRL). Therefore, the concept scores excellent on complexity. For scalability the design scores correctable. The concept scales with a factor 5 in all three directions when the aperture is increased to 1.5m, leading to a total increase in volume of factor 125. This leads to approximately 5213 liters. Part of this volume can be used by a custom s/c bus. However, the s/c bus volume does not scale linearly with aperture. It is expected to scale less, leading to a high fraction of empty volume. For reliability the concept scores good. The linear extensions of M2 and linear baffle are relatively simple and there are possibilities for redundancy.

Concept 2: M2 Linear extension and folding baffle flaps (1.1.1 + 2.1.2)

In this concept the baffle is folded from the sides to a vertical position, as shown in Figure 5.16 on the right. The linear deployment of the secondary mirror (M2) is governing for the height. The optimization gain for this concept is rather low due to the fact that for the volume optimized concept already a boxed design was used. For a hexagon or octagon shape, more than four baffle flaps are needed which increases the complexity of the design.

Parameter	Stowed	Deployed	Units
Height	0.350	0.680	m
Width	0.345	0.325	m
Length	0.345	0.325	m
Box shape volume	41.7	71.8	l
S/c bus volume	18.9	18.9	l
Empty volume	-	-	l
Custom depl. ratio	-	2.43	-
Modular depl. ratio	-	1.84	-
Custom s/c bus gain	-	31.6	%
Opt gain	-	9.0	%

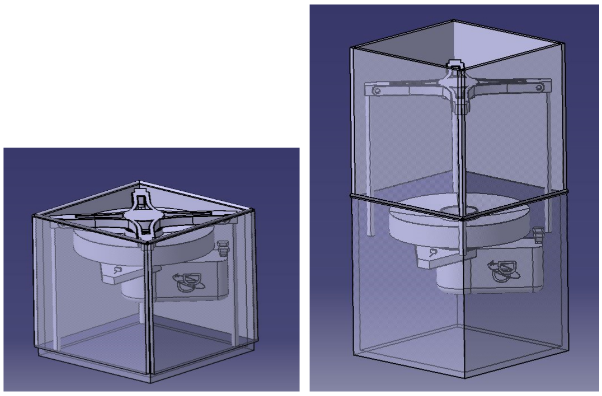


Figure 5.16: Left: Data on concept 2. Right: Model of concept 2 in stowed and deployed configuration.

For complexity the design scores correctable since there are four folding flaps. The folding flaps need to lock into each other and leave no gaps for light to enter the telescope from the sides, which can create image distortion. This increases the complexity over a linear deployed baffle. Another downside of this concept is that it has a large envelope when deploying the baffle flaps. For scalability the design scores correctable. This concept also scales with a factor 5 in all three directions, leading to a total scaling factor of 125. This leads to a volume of approximately 5213 liters. For reliability the design scores unacceptable. Due to the baffle flaps, there are four failure points in one subsystem with little possibility for redundancy. For the mission it would be catastrophic if one of the flaps fails to (fully) deploy as the performance of the telescope would be drastically reduced or even be unusable. The impact of a single failure (or partly failure) in the baffle flaps would be fatal, which is unacceptable for the demonstrator.

Concept eliminated: M2 linear extension and telescopic baffle (1.1.1 + 2.1.3)

This concept is similar to the above described concepts. Again, the linear M2 deployment is governing for the height. The telescopic baffle has therefore no positive effect and even increases the width and length due to the thickness of the telescopic segments. Because of the added volume and complexity the concept is discarded.

Concept 3: M2 folding arms and linear baffle deployment (1.1.2 + 2.1.1)

In this concept folding arms are used to deploy the secondary mirror (M2), as shown in Figure 5.17 on the right. The baffle length is governing. Therefore, the folding arms can be placed above the primary mirror (M1) in stowed configuration. The concept is comparable to the previous concepts regarding deployment ratio. The difference between custom deployment ratio and modular deployment ratio is however smaller due to the lower custom s/c bus volume. The folding arms are a bit more complex than the linear extensions. A separate actuation system for M2 is needed inside the spider, which is is used for correcting the mirror alignment. Both the components for folding arms and the baffle deployment are readily available with a high technology readiness level (TRL). Therefore the complexity is scored good. For scalability this concept scores correctable. Again, the concept scales in all three directions by a factor 5, leading to a total volume of 4163 liters. For reliability the concept scores good. The folding arms have some more failure points than the M2 linear extensions. However, they are still relatively simple and there are possibilities for redundancy. In combination with the linear baffle the concept scores good for reliability.

Parameter	Stowed	Deployed	Units
Height	0.280	0.545	m
Width	0.345	0.345	m
Length	0.345	0.345	m
Box shape volume	33.3	64.9	l
S/c bus volume	7.3	7.3	l
Empty volume	3.0	-	l
Custom depl. ratio	-	2.30	-
Modular depl. ratio	-	2.06	-
Custom s/c bus gain	-	11.5	%
Opt gain	-	8.0	%

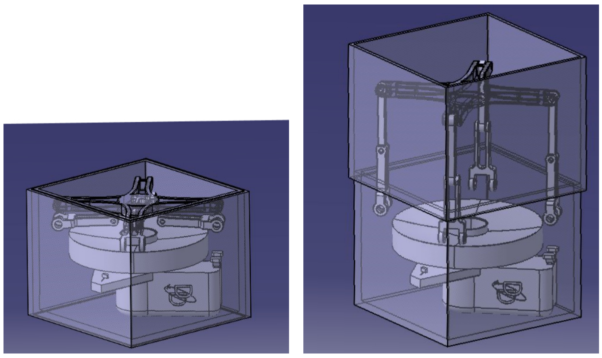


Figure 5.17: Left: Data on concept 3. Right: Model of concept 3 in stowed and deployed configuration.

Concept 4: M2 folding arms and folding baffle flaps (1.1.2 + 2.1.2)

This concept is similar to concept 3 and is shown in Figure 5.18 on the right. However, this concept uses folding baffle flaps for the deployment of the baffle. In order to reach the required baffle length, double folding flaps are used. This reduces the required height which makes the secondary mirror (M2) deployment system governing for the height. Therefore, the folding arms are stowed in between the segmented mirror to reduce the height as much as possible. Due to the low stowed volume and no empty space this concept has good deployment ratios. The design has a very complex deployment system for the baffle and the folding arms need to be stowed in between the primary mirror (M1) segments. Therefore, the concept scores unacceptable for complexity. It is assumed that the height of the internal optics only scales by a factor 2.5. The width and length will scale by a factor 5, which leads to a total scaling factor of 62.5 and a volume of 1713 liters. This is only 41% of concept 3. However, since the concept needs multiple baffle flap segments after scaling, it is expected that the complexity will increase drastically. Therefore, the scalability is scored correctable. For reliability the concept is scored unacceptable due to the many failure points in the baffle flaps.

Parameter	Stowed	Deployed	Units
Height	0.215	0.545	m
Width	0.357	0.325	m
Length	0.357	0.325	m
Box shape volume	27.4	57.6	l
S/c bus volume	7.3	7.3	l
Empty volume	-	-	l
Custom depl. ratio	-	2.54	-
Modular depl. ratio	-	2.25	-
Custom s/c bus gain	-	12.7	%
Opt gain	-	14.9	%

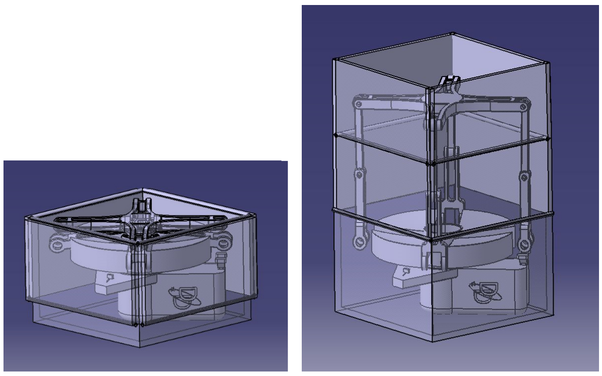


Figure 5.18: Left: Data on concept 4. Right: Model of concept 4 in stowed and deployed configuration.

Concept 5: M2 folding arms and telescopic baffle (1.1.2 + 2.1.3)

This concept is similar to concept 4, except that it deploys the baffle using telescopic segments, as shown in Figure 5.19. The baffle deployment is not very complex as the baffle segments are just sliding along each other like a linear baffle deployment. However, since there are multiple segments the complexity is higher than a linear deployed baffle. Therefore, this concept scores acceptable for complexity. For scalability, the concept is scored acceptable. Similar to concept 3, the total scaling factor is 62.5 which leads to a volume of 1713 liters, which is quite good. The reliability of the concept is scored acceptable. With the additional segments in the baffle, more failure points are present. However, redundancy can be build in which makes the reliability acceptable.

Parameter	Stowed	Deployed	Units
Height	0.215	0.545	m
Width	0.357	0.357	m
Length	0.357	0.357	m
Box shape volume	27.4	69.5	l
S/c bus volume	7.3	7.3	l
Empty volume	-	-	l
Custom depl. ratio	-	2.54	-
Modular depl. ratio	-	2.25	-
Custom s/c bus gain	-	12.7	%
Opt gain	-	14.9	%

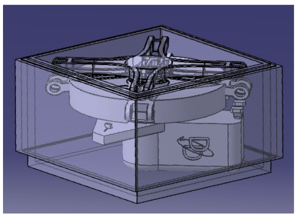
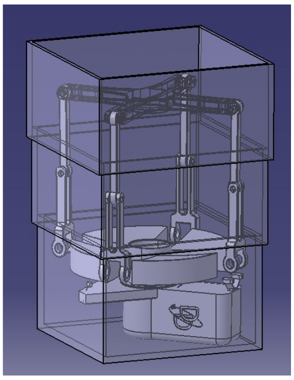



Figure 5.19: Left: Data on concept 5. Right: Model of concept 5 in stowed and deployed configuration.

Concept 6: M1 foldable and pantographic baffle (1.2.2 + 2.2.1)

In this concept the baffle needs to be 80mm above the spider in stowed configuration as it only expands radially, as can be seen in Figure 5.20 on the right. Therefore, the concept is tall, has a high stowed volume and has quite some empty volume above the spider. An improvement for this concept would be to also deploy the baffle axially at the cost of a higher complexity. A reduction in stowed volume of approximately 14% is then possible. In case the secondary mirror (M2) is also axially deployed, as in concept 7, a volume reduction of approximately 36% is possible. Since M2 always needs to be actuated in some way, it is a small step to move towards M2 deployment combined with M1 deployment. Furthermore, as shown by V. Nagy [18], it is possible to combine axial and radial deployment for a pantographic baffle.

Parameter	Stowed	Deployed	Units
Height	0.606	0.606	m
Width	0.225	0.350	m
Length	0.225	0.350	m
Box shape volume	30.7	51.9	l
S/c bus volume	7.0	7.0	l
Empty volume	4.2	-	l
Custom depl. ratio	-	2.39	-
Modular depl. ratio	-	2.14	-
Custom s/c bus gain	-	11.6	%
Opt gain	-	18.5	%

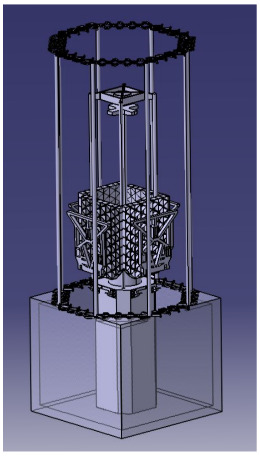
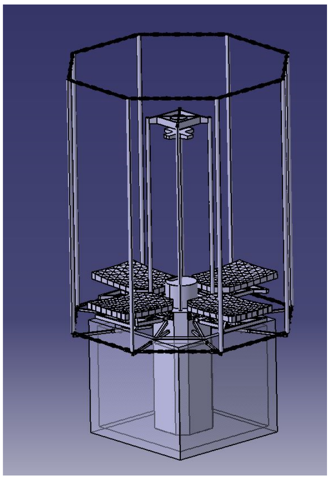



Figure 5.20: Left: Data on concept 6. Right: Model of concept 6 in stowed and deployed configuration.

This is the first and simplest concept for primary mirror (M1) deployment. Deploying M1 has a higher complexity than only deploying M2 and a pantographic baffle has a low technology readiness level (TRL). Furthermore, an actuation system to correct M2 is still required. For these reasons this concept scores correctable for complexity. The concept is not very efficient when scaled. For a 1.5m design the height would become approximately 3 meters. Furthermore, most of the volume will be empty volume as it is only occupied by M2 and the baffle. Therefore, the concept scores unacceptable on scalability. The reliability of this concept is scored acceptable. Four mirror segments have to be deployed and aligned, which is quite complex. However, this can be done with acceptable reliability and redundancy can be build in. The pantographic baffle has a low TRL and prototyping and testing should be performed to determine the reliability. Since redundancy can be build in, the reliability is expected to be acceptable.

Concept 7: M1 foldable down, M2 linear extension and pantographic baffle (1.3.1 + 2.3.1)

This concept is similar to the previous concept, but also deploys M2 and the baffle axially. The concept is shown in Figure 5.21 on the right. The additional deployment adds some complexity, but reduces the volume by approximately 36%. The deployment ratios are very good compared to the other concepts. With the low integrated s/c bus volume it is very suitable for a modular s/c bus. One downside is the linear extension of the secondary mirror (M2), which increases the necessary stowed height. For complexity the concept scores unacceptable. Deploying both M1 and M2 and deploying the baffle both radially and axially is to much complexity for a demonstrator mission. The scalability of this concept is good. Assuming a scaling factor of 125 a total volume of 2038 liters is found. Since the concept has an efficient shape and low s/c bus volume, the scaled concept is expected to leave little empty volume after scaling. For reliability the concept scores correctable. Since in this concept M2 is also deployed, there are additional failure points.

Parameter	Stowed	Deployed	Units
Height	0.321	0.655	m
Width	0.225	0.350	m
Length	0.225	0.350	m
Box shape volume	16.3	60.4	l
S/c bus volume	3.4	3.4	l
Empty volume	-	-	l
Custom depl. ratio	-	2.91	-
Modular depl. ratio	-	2.72	-
Custom s/c bus gain	-	6.8	%
Opt gain	-	29.4	%

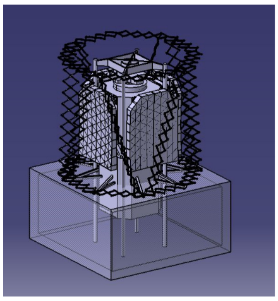
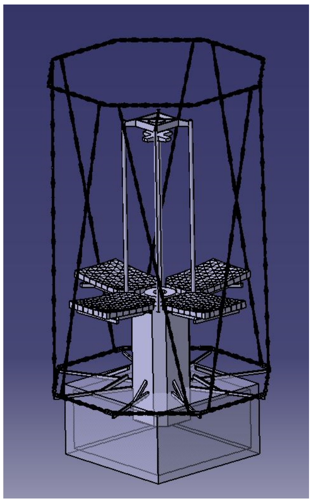



Figure 5.21: Left: Data on concept 7. Right: Model of concept 7 in stowed and deployed configuration.

Concept Eliminated: M1 foldable up, M2 linear extension and pantographic baffle (1.3.1 + 2.3.1)

To investigate if an improvement was possible to the previous concept, the primary mirrors (M1) are folded up in this concept, as shown in Figure 5.22 on the right. The linear extensions of M2 are not governing anymore. However, the volume increased by approximately 20%. Since the M1 mirrors are now governing for the height the stowed height increased by 7cm. Since this concept has similar or lower deployment ratios as concept 7 and a larger stowed volume, the concept is eliminated from the trade-off.

Parameter	Stowed	Deployed	Units
Height	0.390	0.606	m
Width	0.225	0.350	m
Length	0.225	0.350	m
Box shape volume	19.7	51.9	l
S/c bus volume	7.0	7.0	l
Empty volume	-	-	l
Custom depl. ratio	-	2.92	-
Modular depl. ratio	-	2.56	-
Custom s/c bus gain	-	14.6	%
Opt gain	-	29.4	%

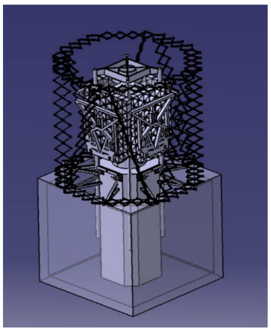
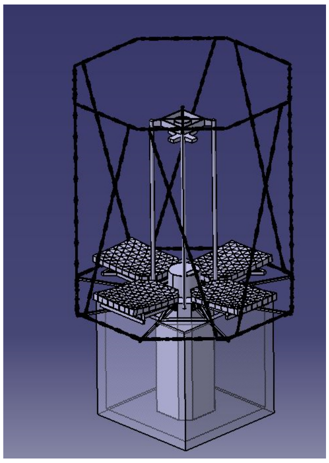



Figure 5.22: Left: Data on concept M1 foldable up, M2 linear extension and pantographic baffle. Right: Model in stowed and deployed configuration.

Concept 8: M1 foldable, M2 folding arms and pantographic baffle (1.3.2 + 2.3.1)

This concept is similar to concept 7. However, in this concept folding arms are used for the M2 deployment as shown in Figure 5.23. This reduces the height by 18mm and the volume by approximately 6%. To improve the concept it would be beneficial to lower the M2 mirror in stowed condition further down. However, this is not possible with only two folding arms. If the lowest folding arm is longer, then the arm will puncture the baffle when M2 is deployed. The deployment ratios of this concept are similar to concept 7 and the concept is also scored unacceptable on complexity. For scalability and reliability the same reasoning as concept 7 holds, which leads to a good and correctable score, respectively. Due to the folding arms, which have more failure points than linear extensions, the reliability of this concept is deemed lower than concept 7. However, it is not considered unacceptable.

Parameter	Stowed	Deployed	Units
Height	0.303	0.605	m
Width	0.225	0.350	m
Length	0.225	0.350	m
Box shape volume	15.3	57.9	l
S/c bus volume	1.8	1.8	l
Empty volume	-	-	l
Custom depl. ratio	-	2.87	-
Modular depl. ratio	-	2.77	-
Custom s/c bus gain	-	3.5	%
Opt gain	-	29.4	%

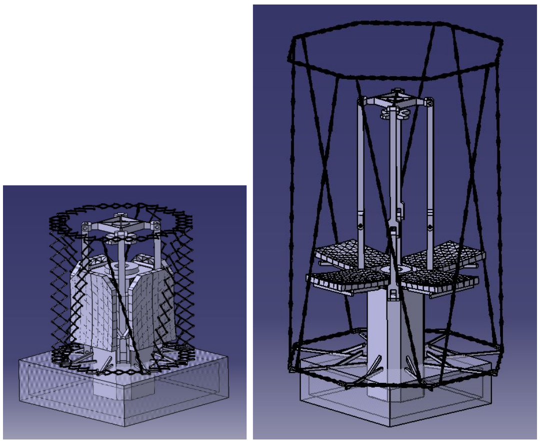


Figure 5.23: Left: Data on concept 8. Right: Model of concept 8 in stowed and deployed configuration.

Concept Eliminated: M1 foldable, multiple M2 folding arms and pantographic baffle (1.3.2 + 2.3.1)

The last concept is an improved version of concept 8. In this concept a more complex M2 deployment system with four folding arm segments or a combination of folding arms and linear extension is used. The concept is shown in Figure 5.24. With the adaptation the height is decreased by 42mm which leads to stowed volume decrease of approximately 14%. This concept, on the other hand, has the highest complexity of all concepts. The gain in deployment ratio is relatively small compared to the increased complexity. Therefore, this concept is eliminated from the trade-off. If concept 8 comes out of the trade-off as winner, then it will be investigated if the complex folding arms could be a beneficial improvement, taking into account their added complexity.

Parameter	Stowed	Deployed	Units
Height	0.261	0.605	m
Width	0.225	0.350	m
Length	0.225	0.350	m
Box shape volume	13.2	57.9	l
S/c bus volume	1.8	1.8	l
Empty volume	-	-	l
Custom depl. ratio	-	2.99	-
Modular depl. ratio	-	2.89	-
Custom s/c bus gain	-	3.7	%
Opt gain	-	29.4	%

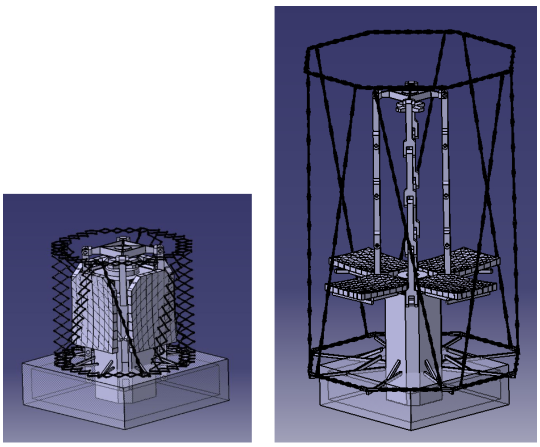


Figure 5.24: Left: Data on concept M1 foldable, multiple M2 folding arms and pantographic baffle. Right: Model in stowed and deployed configuration.

5.6. Concept volume comparison

In this section the volume of the different concepts is compared. In Table 5.2 the results for each concept are shown.

Table 5.2: Table containing the volume values of the different concepts.

Parameter / Concept	$V_{st.opt}$ [l]	$V_{st.box}$ [l]	V_{depl} [l]	$V_{s/cb}$ [l]	V_{inst} [l]	$R_{s/cb}$ -	R_{inst} -	R_{cust} -	R_{mod} -	G_{cust} [%]	G_{opt} [%]
1: M2 linear, Baffle linear	34.0	41.7	80.9	18.9	22.8	0.55	4.67	2.43	1.84	31.6	22.5
2: M2 linear, Baffle flaps	38.2	41.7	71.8	18.9	22.8	0.55	4.67	2.43	1.84	31.6	9.0
3: M2 folding, Baffle linear	30.9	33.3	64.9	7.3	26.1	0.78	4.08	2.30	2.06	11.5	8.0
4: M2 folding, Baffle flaps	23.9	27.4	57.6	7.3	20.1	0.74	5.28	2.54	2.25	12.7	14.9
5: M2 folding, Baffle telescopic	23.9	27.4	69.5	7.3	20.1	0.74	5.28	2.54	2.25	12.7	14.9
6: M1 folding, Baffle pantograph.	25.9	30.7	51.9	7.0	23.7	0.77	4.49	2.39	2.14	11.6	18.5
7: M1 folding down, M2 linear, Baffle pantographic	12.6	16.3	60.4	3.4	12.9	0.79	8.27	2.91	2.72	6.8	29.4
8: M1 folding down, M2 folding, Baffle pantographic	11.9	15.3	57.9	1.8	13.6	0.88	7.83	2.87	2.77	3.5	29.4

***$V_{st.opt}$* - Optimized stowed volume**
Minimum volume possible with an optimized shape, e.g. square, hexagon, octagon, etc.
***$V_{st.box}$* - Boxed stowed volume**
Volume which the concept occupies when outside dimensions are taken as a box.
***V_{depl}* - Deployed volume**
Volume the concept has in its deployed configuration.
***$V_{s/cb}$* - Spacecraft bus volume**
Defined as the volume below the primary mirror which is available to house spacecraft bus components.
***V_{inst}* - Instrument volume**
Volume of the instrumentation only. This is defined as the box stowed volume minus the spacecraft bus volume.
***$R_{s/cb}$* - Spacecraft bus volume ratio**
Ratio between the instrument volume and box stowed volume. Indicates the fraction of volume that is dedicated to the instrument.
***R_{inst}* - Instrument deployment ratio**
Ratio between the volume of Satellogic-NuSat minus it's s/c bus volume and the concept's instrument volume.
***R_{cust}* - Custom s/c bus deployment ratio**
Ratio between the volume of Satellogic-NuSat and the concept's instrument volume plus 37 liters to account for s/c bus volume.
***R_{mod}* - Modular s/c bus deployment ratio**
Ratio between the volume of Satellogic-NuSat and the concept's boxed stowed volume plus 37 liters to account for s/c bus volume.
***G_{cust}* - Custom s/c bus gain**
The percentage gain in deployment ratio when a custom s/c bus is used instead of a modular s/c bus.
***G_{opt}* - Optimization gain**
Percentage of volume decrease which can be gained by using an optimized shape instead of a cubic shape.

From Table 5.2, it can be noted that when the primary mirror (M1) is also deployed that the instrument deployment ratio increases drastically and in some cases even doubles. It can therefore be concluded, that deploying M1 has a significant positive effect on the stowed volume. It can furthermore be seen, that the large differences in instrument deployment ratio are drastically reduced when the deployment ratios are calculated on spacecraft (s/c) level, R_{cust} and R_{mod}. The reason for this is that the spacecraft bus volume does not differ per concept. The effect of the lower stowed volume with more complex deployment options is therefore reduced. It can be noted that when the s/c bus is larger, that the spacecraft deployment ratios get closer together. Therefore, it would be most beneficial to make the s/c bus as small as possible. When looking at the custom s/c gain it can be noted that for all concepts which have a fixed M1, that a custom s/c bus would be most ideal. For the M2 linear concepts with fixed M1 the gain is even 31.6%. For this reason it is concluded that for the M2 linear with fixed M1 concepts, a custom s/c bus is highly favored over a modular s/c bus. While the other concepts can gain up to 12.7% in deployment ratio, it is deemed that these concepts can still operate as a modular payload, as this reduces the complexity of the s/c bus design.

5.7. Reference Satellites Comparison

In this section the DST concepts are compared to reference satellites by using graphs. The focus in finding reference satellites was mainly on satellites that operate solely in the TIR wavelength, 8 to 12 μm, or have a spectral band in this wavelength range. 18 reference satellites were found. Since, the amount of satellites in this range was limited the wavelength range was extended to visual (VIS) (0.38μm - 0.75μm), near infrared (NIR) (0.75μm - 1.0μm), short wave infrared (SWIR) (1.0μm - 2.5μm) and mid wave infrared (MWIR) (3μm - 5μm) as well. Especially for the VIS and NIR spectra many satellites were found. Since now all wavelengths were considered, the comparison could be extended to include the 1.5m VIS DST and an estimation of the 30cm VIS DST design as well. It is furthermore interesting to see the differences in TIR satellites compared to VIS satellites. In the current comparison 49 satellites are used. Although the search for reference satellites was very extensive, more suitable reference satellites can always be found. However, it was observed that additional satellites were mostly situated around the already existing data points. This makes the correlation stronger, but does not change it. Therefore, it was deemed that the current data set is a good representation of the complete data set of satellites. An exception are military satellites for which hardly any information can be found. Therefore, they cannot be taken into account in the data set.

The reference satellite comparison is used to find relationships and a design space for the DST concepts. The blue and purple line in the graphs are approximated by the author, to indicate what appears to be a correlation in the data points. This is only based on the visual location of the data points and can change depending on new reference satellites. The blue line is used to mark the blue design space which is better than most of the current available VIS satellites. The purple line indicates the correlation for satellites with an operating wavelength or band of 8 to 12 μm. The purple indicated area together with the blue area is the design space for TIR satellites. Purple encircled data points indicate satellites that operate or have a band in the TIR spectrum.

The differences between the DST TIR 30cm concepts, regarding stowed volume at spacecraft level, was quite small and are not clearly visible in the graphs. Therefore, it was decided to take the average stowed volume of the concepts and only plot that point. It is preferred for the DST TIR 30cm to fall within the TIR design space. The DST TIR 30cm is indicated in the plots together with the closest reference satellite, Satellogic-NuSat, which is discussed in Section 5.4. The DST TIR 1.5m, DST VIS 30cm and DST VIS 1.5m designs are also indicated in the graphs. The data used for the DST concepts can be found in Table 5.3. The DST VIS 30cm optical performance is an estimation from DST's optical engineer, D. Dolkens. The volume and mass are assumed to be the same as the DST TIR 30cm, because there is no design for a DST VIS 30cm yet. Data for the DST TIR 1.5m is taken from V. Nagy [18] and an optical estimation from D. Dolkens. Data for the DST VIS 1.5m is taken from previous systems engineering documentation from the DST drive.

Table 5.3: Data on the DST designs.

DST design	Operating wavelength [μm]	GSD [m]	Swath width [m]	Mass [kg]	Volume [l]
DST TIR 30cm	8-12	14.2	7.2	54.0	67.9
DST TIR 1.5m	8-12	2.25	4.4	313.6	2400
DST VIS 30cm	0.45-0.65	1.23	24.6	54.0	67.9
DST VIS 1.5m	0.45-0.65	0.25	5.0	313.6	2300

5.7.1. Wavelength vs GSD

The first graph is wavelength vs Ground Sampling Distance (GSD), as shown in Figure 5.25. From the graph it can be seen that most of the reference satellites operate in the visual spectrum. Some satellites operate both in the visual spectrum (VIS) and Short Wave Infrared (SWIR) (up to approximately 2.5μm), while still providing a good GSD of up to 3.7meters, like Worldview-3. There are also satellites that operate in VIS, SWIR and Long Wave Infrared (LWIR), also known as TIR, like Terra, GOES-17, LANDSAT, Jilin-1GP01/02, MTI, Satellogic-NuSat and NOAA-15 AVHRR/3. It can be noticed that once the wavelength increases, the GSD goes up and varies in LWIR between 20m and 2000m. This relationship is also indicated with the blue line. Going below the blue line would be an improvement of the current technology.

One satellite which is interesting is Kompsat-3A. It has a band in Mid Wave Infrared (MWIR) between 3.3 to 5.2 μm with a GSD of 5.5 meters. This is compared to the other satellites operating at this wavelength quite good. Another interesting satellite is MTI which has a good resolution in almost all wavelength ranges. In VIS and NIR it has a GSD of 5 meters and in SWIR, MWIR and LWIR it has a GSD of 20 meters. From the graph it can be seen that the DST concepts are within the marked area. Compared to the reference satellites in the wavelength 8-12μm, the DST TIR 1.5m scores very well. It should however be noted that the optical design of this concept is not fully worked out up to detailed design. Therefore, it is likely to change in the future. It shows however, that if the optical design is feasible that it highly outperforms current competitors in terms of GSD. The DST TIR 30cm also outperforms the current TIR satellites. The only close competitor regarding GSD is MTI. The DST VIS 1.5m satellite outperforms Worldview-3, Worldview-4 and Pleiades NEO which are considered state-of-the-art telescopes. The DST VIS 30cm design performs relatively well for the visual spectrum, but cannot be considered state of the art.

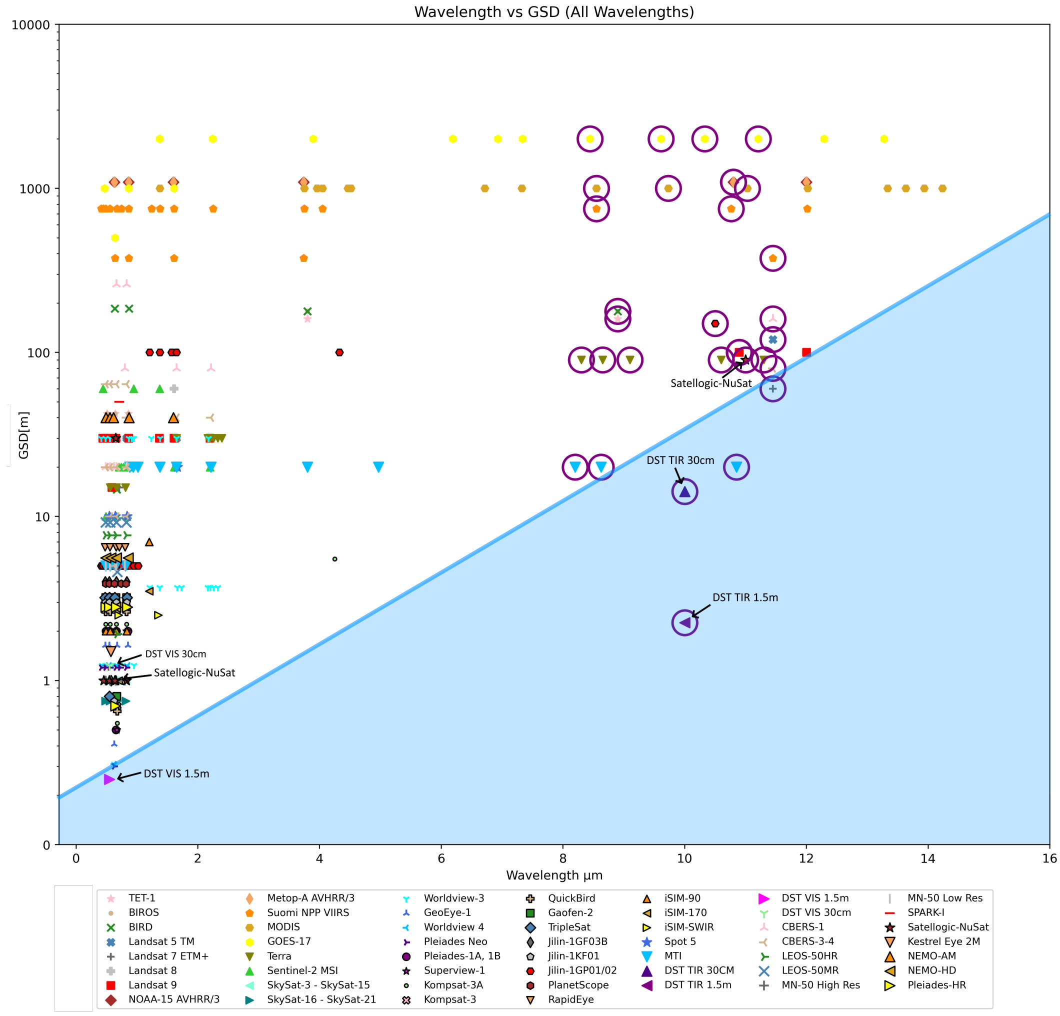


Figure 5.25: Graph of wavelength vs ground sampling distance.

5.7.2. Ground Resolution vs Mass

The second graph is the ground resolution plotted versus the mass of the total spacecraft, as shown in Figure 5.26. From the figure it can be seen that with decreasing mass the GSD increases, which is also indicated with the blue and purple line. Moving towards the right side of this line would be an improvement. This can be done by decreasing the mass and keeping the GSD constant, by decreasing the GSD and keeping the mass constant or a combination. The satellites that are situated to the right side of the blue line are all recent satellites which are state of the art. Since Satellogic-NuSat operates at multiple wavelenghts, it also has multiple ground sampling distances (GSD). The DST VIS 1.5m is located in a good position. If a decrease in mass, without losing performance, is possible for this satellite, then its position can be improved even further. The DST VIS 30cm design performs reasonably well. However, it does not outperform the current state-of-the-art satellites. Strong improvements in GSD or lower mass are needed to strengthen it's position and become better than current satellites.

It can be seen that the state-of-the-art line shifts towards the left for TIR satellites. The inclination seems to stay more or less the same. Satellogic-NuSat, MTI, DST TIR 30cm and DST TIR 1.5m are all located in the purple area. Both DST TIR concepts prove to have a good GSD compared to their mass and outperform the current TIR satellites.

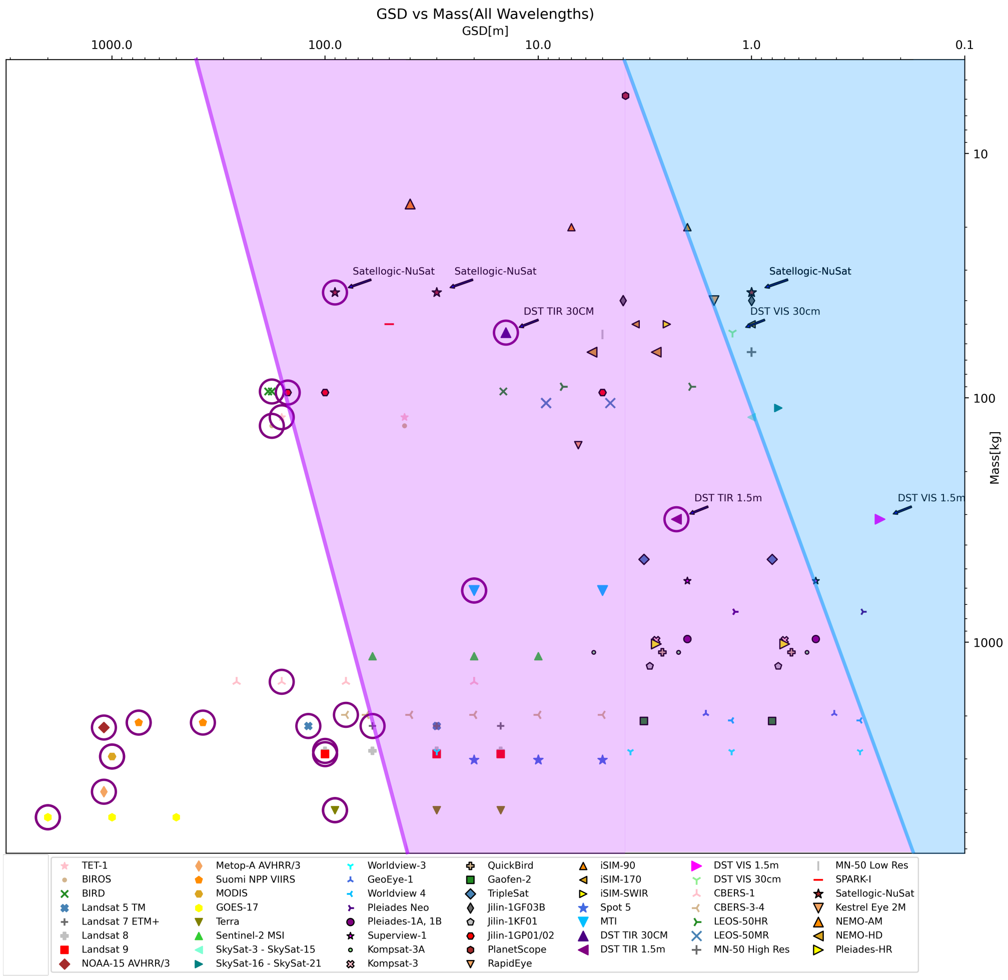


Figure 5.26: Graph of ground sampling distance vs mass.

5.7.3. GSD vs Swath Width

The next graph is the GSD versus swath width, as shown in Figure 5.27. The swath width is defined as the width on the Earth surface that the telescope images. With a larger swath width, less orbits are needed to cover the Earth surface or, when in a constellation, less satellites are needed. Therefore, a high swath width is preferred.

In Figure 5.27, it can be seen that if the GSD becomes lower, lower GSD is better, the swath width generally becomes lower. If the amount of pixels in the detector are kept the same, and the GSD is improved, than the swath width decreases. Moving above the blue or purple line would be an improvement of the current existing technology for the VIS and TIR spectrum, respectively. This could be done by having a larger detector to increase the swath width. One such satellite is Jilin-1KF01. This satellite was launched on 15 January 2020, weighting 1250kg and orbits at an altitude of 481km.[8] According to charmingglobe.com, it has a panchromatic resolution of 0.75m, a multispectral resolution of 3m and a swath width of more than 136km.[9] With a swath width of 136km, it is approximately 3 times better compared to a swath width of 45km for Gaofen-2 at approximately the same wavelength. While having the same GSD of 3m, Jilin-1KF01 weights only a factor 0.6 of Gaofen-2. Therefore, it appears Jilin-1KF01 is a state-of-the-art satellite which lies ahead of the competition regarding swath width.

The DST VIS 1.5m design has compared to Worldview-3, Worldview-4 and Pleiades NEO a three times smaller swath width at a slightly better GSD. The DST VIS 30cm design performs reasonably well, with a swath width of 24.6km. For the TIR wavelength, it can be seen that the DST TIR 1.5m performs quite well with a swath

[8] http://www.charmingglobe.com/EWeb/chanpin_view.aspx?id=682, accessed 14 July 2021
[9] http://www.charmingglobe.com/EWeb/chanpin_view.aspx?id=682, accessed 14 July 2021

width of only 4.4km. The DST TIR 30cm has a better swath width, of 7.2km, but with a higher GSD it is just below the purple line. From the graph it can be seen that with decreasing GSD the swath width appears to reduce faster for TIR satellites than for VIS satellites. However, since there are not much data points with a GSD below 80m this is sensitive to changes.

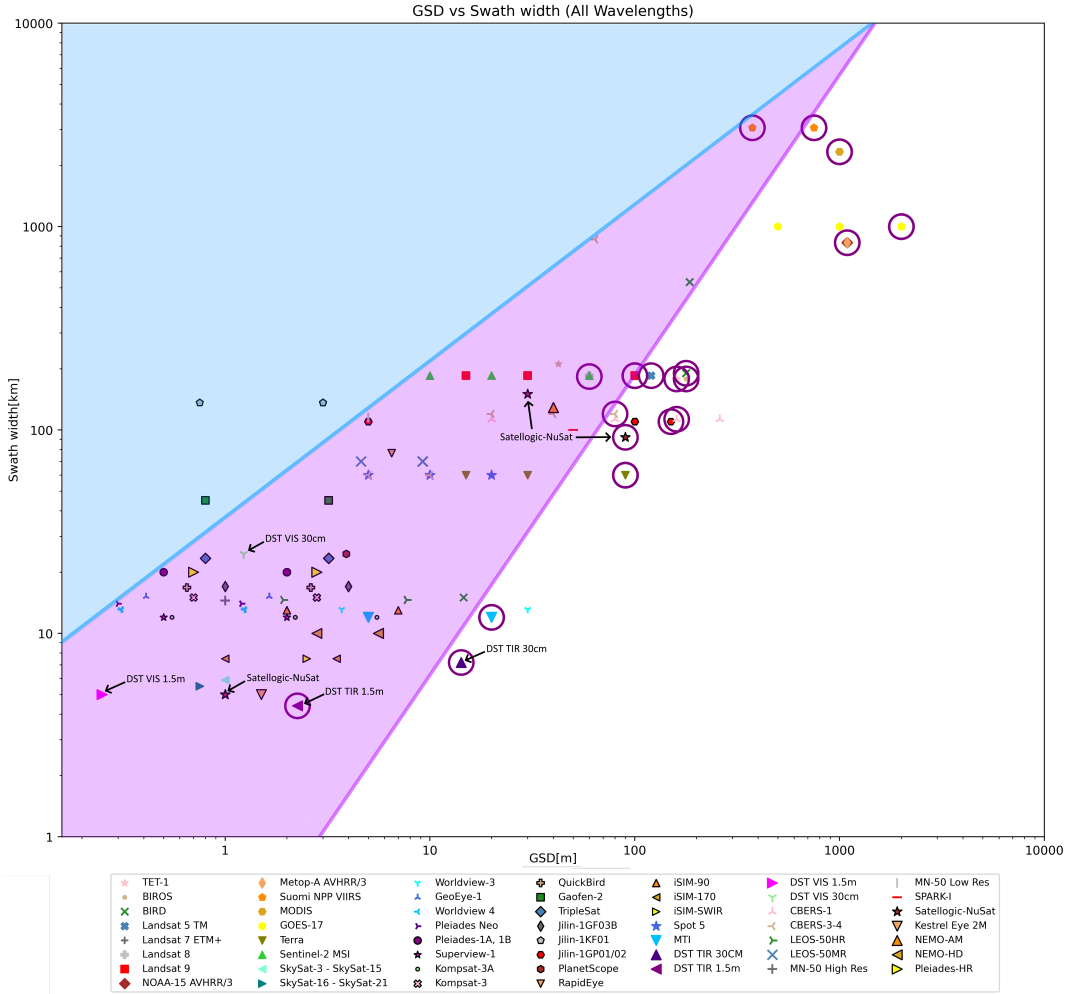


Figure 5.27: Graph of ground sampling distance vs swath width.

5.7.4. GSD vs Volume

The last graph is the GSD vs Volume, as shown in Figure 5.28. A similar relationship is found as GSD vs Mass, which is to be expected since mass and volume are closely related. For the DST VIS 1.5m and DST TIR 1.5m concepts, a s/c bus volume was not available. Therefore, a s/c bus fraction of 50% of their stowed volume was taken as estimation.

DST VIS 1.5m performs for it's volume very well. There are no close competitors. The DST VIS 30cm performs better regarding volume than mass, as was shown in Figure 5.26. The design can be compared with Satellogic-NuSat and iSIM-170, which are both state-of-the-art designs. The DST TIR 1.5m meter design gets relatively close to the visual state-of-the-art design space. The DST TIR 30cm design is also situated in a good position and situated quite far to the right of Satellogic-NuSat's TIR data point.

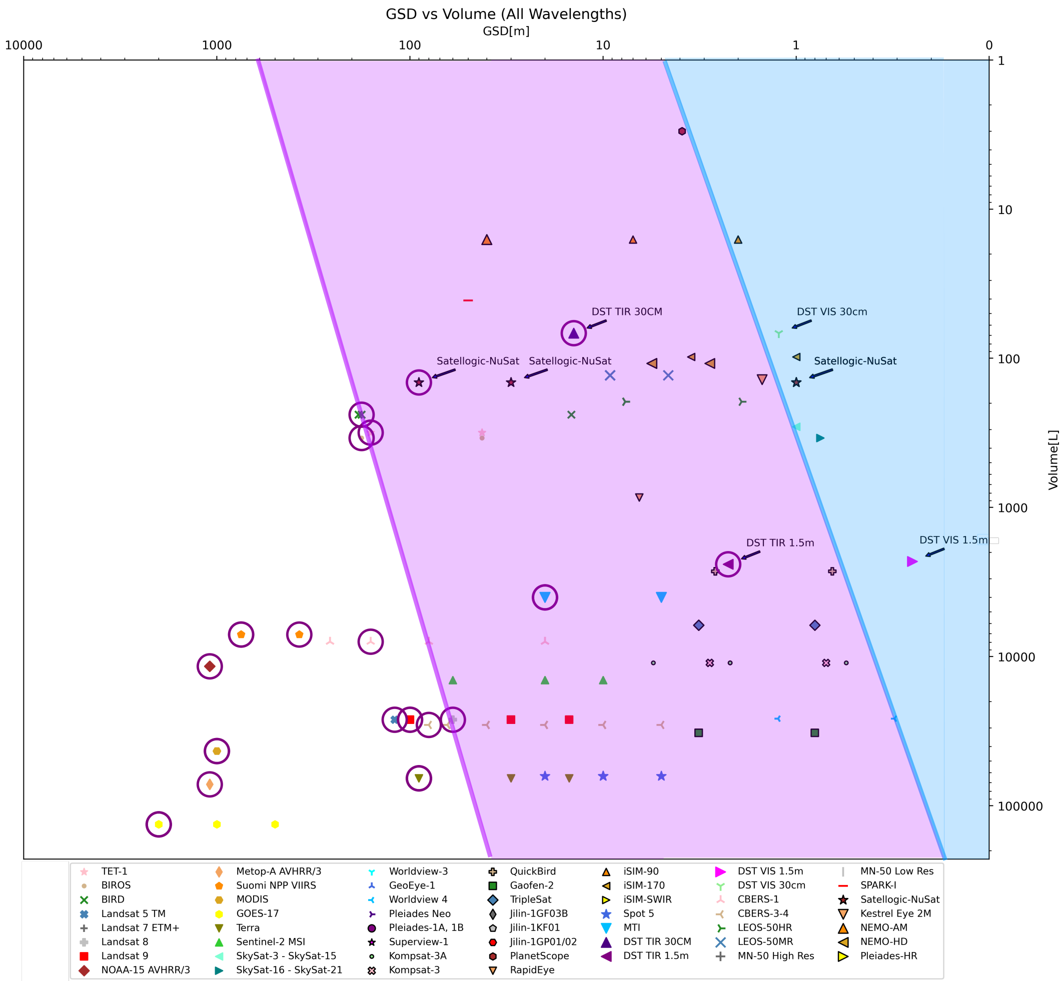


Figure 5.28: Graph of ground sampling distance vs volume.

5.7.5. Conclusion

The design space, with respect to reference satellites, is set out in the previous sections. While it gives a clear overview there are some side notes:

- In the thermal infrared range there are not many reference satellites (MWIR and LWIR)
- Most satellites focus on the visual spectrum and extend this sometimes to Short Wave Infrared (SWIR)
- Otherwise multi- or hyperspectral imaging is used which covers multiple bands ranging from 450nm to 12-14 μm
- Most reference satellites carry multiple payloads besides the thermal infrared imager, which increases the mass of the satellite. This makes them less suitable as a direct reference on spacecraft level. Data on instrument volume and mass is not easily found, which makes comparing instrument size and mass difficult
- Most of the reference satellites have a higher mass than 200kg
- Most likely there are more satellites in the thermal infrared spectrum of which we don't know, e.g military satellites.

When comparing the DST TIR 1.5m and DST TIR 30cm for mass and volume, in Figures 5.26 and 5.28, it is expected that they would lie approximately on a line parallel to the purple line. The DST TIR 30cm performs worse than could be expected based on the reference satellites. This could have several reasons. The first reason is the higher maturity in optical design for the DST TIR 30cm. According to D. Dolkens, the DST

TIR 1.5m optical design could be too optimistic. The second reason is that not all components scale along with a smaller design, think of s/c bus components. Another reason could be that the mass of the DST TIR 30cm, estimated in this thesis using reference s/c busses, is to high or that the estimation that was done in the past for the 1.5m variant, is to low. Since the maturity of the designs is not the same, it is difficult to draw conclusions. It is therefore deemed that the graphs provide insight in how the DST concepts compare to reference satellites, but that there is a uncertainty to their exact position. However, even with this uncertainty, it can be concluded that the DST concepts, except for the DST VIS 30cm, perform quite well compared to the reference satellites. In most graphs they are in a competitive position with a large margin to the closest competitor. The reference satellite comparison should be updated regularly with new satellites to check if the DST concepts are still state of the art.

In the current comparison the DST VIS 30cm design performs similar to Satellogic-NuSat and iSIM-170, except for swath width for which it performs better. These satellites are both fixed state-of-the-art satellites with lower complexity. From the comparison it appears that there is no benefit for a deployable 30cm visual telescope. The DST VIS 1.5m on the other hand, can be deemed state of the art with no close competitors. From this comparison it therefore appears that a larger aperture telescope benefits more from deployable optics than a small aperture telescope.

If we assume that the current position, in Figures 5.26 and 5.28, is correct and that a linear line between the 30cm and 1.5m concepts can be drawn, it can be concluded that a larger design has a higher benefit than a smaller design. This could mean that deployable optics are more valuable for larger aperture designs. However, this is dependant on the maturity of the current DST designs. To investigate this further, a more detailed optical analysis should be performed for the DST TIR 1.5m and DST VIS 30cm to bring their maturity to the same level as the other DST concepts. It would also be interesting to perform a preliminary optical design and sizing of a 3m DST TIR and VIS design and plot it in the graphs. This could provide more insight into the effect of deployable optics in larger aperture designs.

5.8. Trade-Off Method, Criteria and Scoring

After it was determined that the DST TIR 30cm concepts can be considered state of the art and perform well with respect to competitors, the trade-off can be performed. To determine which concept is the most optimal for the demonstrator, two trade-offs are performed. The first is an Analytic Hierarchy Process (AHP) trade-off in which the input of the team members is used. The team members had to fill in different AHP sheets with pair wise comparisons. This method is described in more detail in Section 5.9.2. The second method is a graphical trade-off in which the column width is determined by the weights given to the criteria in the AHP trade-off. This is described in Section 5.8.2. The graphical trade-off is used to qualitatively grade the concepts and determine their feasibility. The AHP trade-off is used to involve the team members and to compare their input with the results from the graphical trade-off. Both trade-offs combined, should give clear insight in feasible concepts and provide the team with a concept to continue on.

The criteria for the trade-offs has been described in Section 5.5, on page 37. The following criteria have been defined: volume, complexity, scalability and reliability. For the volume criterion there are two subcriteria: custom s/c bus volume and modular s/c bus volume.

5.8.1. AHP trade-off method

The Analytic Hierarchy Process (AHP) is a trade-off method which uses math and psychology to organize and analyse complex decisions.[10] For the execution of the AHP trade-off method, the AHP Calculation Tool from http://bpmsg.com, developed by K. Goepel, was used. [12] Since it would be to complicated and time consuming for the team members to fill in a full AHP trade-off for all concepts, a hybrid form of AHP was used. This hybrid form consists of less pair-wise comparisons. The result of the hybrid form was used by the author to complete the full AHP trade-off. In this way the workload for the team was reduced while still involving the team's opinion.

[10] https://www.passagetechnology.com/what-is-the-analytic-hierarchy-process, accessed 20 December 2021

The team members were asked to fill in the following AHP sheets/questions for which a consistency ratio of 10% was maintained:

- **Criteria** - Score the criteria for the trade-off to their importance:
  - Volume
  - Complexity
  - Scalability
  - Reliability
- **Complexity** - For each deployable component, which one do you consider the least complex and with what intensity?
  - M1 fixed
  - M1 deployment
  - M2 fixed
  - M2 linear extensions
  - M2 folding arms
  - Baffle linear
  - Baffle telescopic
  - Baffle flaps
  - Baffle pantographic
- **Custom vs modular** - A modular s/c bus means that the DST is added as a box with a predefined electro-mechanical interface on a third party s/c bus. Therefore the DST is relatively independent on the satellite platform. A custom s/c bus will however mean an integral development of a satellite platform which increases overall cost and complexity. Advantages are a potential smaller overall satellite volume and full control over the platform specification. Considering these arguments, not yet looking into the DST options, which of the two options (custom vs modular) do you consider more important for the DST project and by which intensity?
  - Custom s/c bus
  - Modular s/c bus
- **Deployment ratio** - The deployment ratio indicates the ratio between the volume of a spacecraft with a fixed telescope over a spacecraft with a deployable telescope in stowed condition. Higher deployment ratio is thus per definition better. To obtain the AHP weighting per deployment ratio a few reference points are required. For the below given deployment ratios at spacecraft level, what is the importance of the provided high deployment ratio with respect to the lower deployment ratio?
  - Deployment ratio 1.84
  - Deployment ratio 2.92

The resulting weights from the "Criteria" sheet are used for both the AHP and graphical trade-off. The resulting weights of the "Complexity" sheet are used to calculate the complexity per concept. In for instance the first concept, "M2 linear extension and linear baffle deployment", the weight of M1 fixed, M2 linear extensions and baffle linear are added together, to find the overall complexity for this concept. A higher percentage means a lower complexity of the concept. The complexity percentages are normalized after they are calculated for all concepts. The results of the "Custom vs modular" sheet are used to obtain the weights for the volume sub-criteria, custom and modular s/c bus volume. To grade these sub-criteria, the "Deployment ratio" sheet is used. The team members are asked to give an intensity to the difference in deployment ratio of 1.84 versus 2.92. These are the highest and lowest deployment ratios seen in the concepts. The resulting weight percentage can be plotted in a graph and a linear line between the two points can be fitted. With the formula of this line the weight percentage of each deployment ratio for the concepts can be determined. The results are then normalized to get a distribution with a sum of 100%. The higher the resulting percentage, the better the concept's deployment ratio is. This is done for both the custom and modular deployment ratios. The sub-criteria custom and modular s/c bus volume are then multiplied with the weights from the "Custom vs modular" sheet to obtain the final weights for the volume criteria.

The grading for the criteria scalability and reliability are performed by the author, as for these criteria detailed and high level system engineering knowledge of the concepts is required. The results of the AHP trade-off are presented in Section 5.9.2.

5.8.2. Graphical trade-off method

The criteria weighting found in the AHP sheet "Criteria" is used for the graphical trade-off. This weighting determines the width of the columns in the graphical trade-off table which shows the criterion's importance in a graphical way. For the sub-criteria custom s/c bus volume and modular s/c bus volume a separate trade off will be done first. The results of this trade-off are used in the main graphical trade-off. The scoring for the criteria is described in Section 5.5, on page 37, and consisted of unacceptable, correctable, acceptable, good and excellent.

5.9. Concepts Trade-Off

In this section the concepts trade-off is done. First the results of the graphical trade-off are shown in Table 5.4 and 5.5. The reasoning for the scores in the graphical trade-off is given in Section 5.5, starting on page 37.

5.9.1. Graphical trade-off

For the sub-criteria custom and modular s/c bus volume, it can be seen that for all concepts the custom s/c bus volume scores equally or better than the modular s/c bus volume. The resulting weight of the AHP pair wise comparison between a modular and custom spacecraft bus (s/c), was found to be 74.8% and 25.2%, respectively. The consensus between participants was 94.4% which is very good. The scores of the modular deployment ratio are taken as scores for the volume criteria in Table 5.5, since the weight of the modular deployment ratio is considerably higher.

Table 5.4: Deployment concept trade-off table for the sub-criteria of volume.

Concept \ Criterion	Cust. s/c bus vol. 25.2%	Modular s/c bus volume 74.8%
1: M2 linear, Baffle linear	Acceptable	Correctable
2: M2 linear, Baffle flaps	Acceptable	Correctable
3: M2 folding, Baffle linear	Acceptable	Acceptable
4: M2 folding, Baffle flaps	Good	Acceptable
5: M2 folding, Baffle telescopic	Good	Acceptable
6: M1 folding, Baffle pantographic	Acceptable	Acceptable
7: M1 folding down, M2 linear, Baffle pantographic	Good	Good
8: M1 folding down, M2 folding, Baffle pantographic	Good	Good

The resulting weights from the AHP pair wise comparison were 32.9%, 26.3%, 16.5% and 24.2% for volume, complexity, scalability and reliability, respectively. The consensus for the criteria weighting is rather low at 58.1%. This is further discussed in Section 5.9.2.

Table 5.5: Deployment concept trade-off table.

Concept \ Criterion	Volume 32.9%	Complexity 26.3%	Scalability 16.5%	Reliability 24.2%
1: M2 linear, Baffle linear	Correctable	Excellent	Correct.	Good
2: M2 linear, Baffle flaps	Correctable	Below avg.	Correct.	Unacceptable
3: M2 folding, Baffle linear	Acceptable	Good	Correct.	Good
4: M2 folding, Baffle flaps	Acceptable	Unacceptable	Correct.	Unacceptable
5: M2 folding, Baffle telescopic	Acceptable	Acceptable	Accept.	Acceptable
6: M1 folding, Baffle pantographic	Acceptable	Correctable	Unaccept.	Acceptable
7: M1 folding down, M2 linear, Baffle pantographic	Good	Unacceptable	Good	Correctable
8: M1 folding down, M2 folding, Baffle pantographic	Good	Unacceptable	Good	Correctable

From the graphical trade-off table, it can be seen that the primary mirror (M1) deployment options are unfeasible due to their unacceptable complexity. The same holds for concept 4 "M2 folding, Baffle flaps". This concept also scores unacceptable on reliability, as is concept 2, which is due to the baffle flaps. The last concept which is unfeasible is concept 6 "M1 folding, Baffle pantographic". This concept scores unacceptable for scalability. The remaining concepts are concepts 1, 3 and 5 which are close together and all have their strengths and weaknesses. Concept 1 has a low deployment ratio and therefore high volume. On the other hand, it scores excellent for complexity and good for reliability. Concept 3 has a better deployment ratio at the cost of a higher complexity. For reliability and scalability both concepts score the same. Concept 5 has an even higher deployment ratio but for complexity and reliability the concept scores lower than concept 1 and 3, but still acceptable. For scalability, it scores slightly better due to the telescopic baffle.

Since the criteria weighting is quite close, there is no clear winner at first sight. To put the deployment ratios more in perspective, a trade-off table of the remaining concepts is made including custom s/c bus concepts. Although the team has a strong preference for a modular s/c bus, it is particularly interesting to see the effect on concept 1. This graphical trade-off table is shown in Table 5.6. For the custom s/c bus designs the complexity is deemed one grade lower. The gain in deployment ratio is 32% for concept 1, while the only drawback is a slightly worse complexity. Compared to concept 3, concept 1.c scores similar while the deployment ratio is at the other end of the acceptable deployment range of 2 - 2.5. For concepts 3 and 5 the benefit in deployment ratio of a custom s/c bus is deemed to low to justify the increase in complexity. The gain is only around 12%.

From the graphical trade-off, there is no clear winner. Taking a step back and including custom s/c bus designs complicates it even further, as concept 1.c scores quite well. It could be argued that with a preference weight of 74.8% for a modular s/c bus, a custom s/c bus should not be considered anymore. However, this preference is purely based on a pair wise comparison between a modular and custom s/c bus. If it is put in perspective, as described above, it is a bit more complicated than simply comparing the two. For this reason there is no conclusion yet for the graphical trade-off. In the next section the AHP trade-off results are described. The results of both trade-offs will be used to come to a conclusion in Section 5.9.3.

Table 5.6: Deployment concept trade-off table for the remaining concepts.

Concept \ Criterion	Depl. ratio	S/c bus	Volume 32.9%	Complexity 26.3%	Scalability 16.5%	Reliability 24.2%
1: M2 linear, Baffle linear	1.84	Mod.	Correctable	Excellent	Bel. avg	Good
1.c: M2 linear, Baffle linear	2.43	Cust.	Acceptable	Good	Bel. avg	Good
3: M2 folding, Baffle linear	2.06	Mod.	Acceptable	Good	Bel. avg	Good
3.c: M2 folding, Baffle linear	2.30	Cust.	Acceptable	Acceptable	Bel. avg	Good
5: M2 folding, Baffle telescopic	2.25	Mod.	Acceptable	Acceptable	Accept.	Acceptable
5.c: M2 folding, Baffle telescopic	2.54	Cust.	Good	Correctable	Accept.	Acceptable

5.9.2. AHP trade-off

The setup of the AHP-tradeoff is discussed in Section 5.8.1. The resulting AHP trade-off table for the sub-criteria custom and modular s/c bus volume, is shown in Table 5.7, on page 56. The resulting AHP trade-off table for the criteria volume, complexity, scalability and reliability is shown in Table 5.10, on page 59. To show the spreading of the individual grading in the AHP trade-off, box plots have been created and are shown in Figures 5.29, 5.31, 5.32 and 5.33. The approach is based on the work of J. Bouwmeester. [7] In the box plot, the small horizontal lines at the end of the vertical lines indicate the minimum and maximum. The top and the bottom of the box indicate the first and third quartile and the orange line in the middle indicates the median. The mean is indicated by the blue cross. Lastly, the fraction of participants for which the particular option received the highest priority is indicated with the red dot. Outliers are indicated when they are higher than the third quartile plus 1.5 times the inter quartile range (IQR) or lower than the first quartile minus 1.5 times the IQR. The outliers are indicated to show that there are some different opinions for some criteria. It was furthermore tested if the outliers had a significant effect on the results. This was done through Winsorization. [20] This is a process in which the values of the outliers are edited to the highest or lowest value that is not considered an outlier. With this method it was found that the outcome of the AHP trade-off differed by approximately 0.1% point. Therefore, it was concluded that the outliers did not strongly influence the outcome.

Sub-criteria volume

The participants of the AHP trade-off gave on average an intensity of 5.1, on a scale of 9, to a deployment ratio of 2.92, which is a strong importance. This resulted in a weight of 17.9% and 82.1% for deployment ratio 1.84 and 2.92, respectively. The consensus between participants was 98.4%, which is very good. In Figure 5.29 on the left, this can be seen back in the nearly equal spreading around the mean and median.

To obtain the weights per deployment ratio, the weights were first plotted in Figure 5.30. A linear fit to the two data points was found with the corresponding formula as shown. With this formula the weight of each deployment ratio was determined and then normalized. The resulting weights for the custom and modular s/c bus volume per concept, are shown in Table 5.7. The consolidated column is calculated as a summation of the concept weight multiplied with the column weights. The consolidated weight is input for the volume weights in the AHP trade-off, shown in Table 5.10, on page 59.

The participants gave on average an intensity of 3.75 to a modular s/c bus, which is a moderate importance. The resulting weight percentages are 74.8% and 25.2% for a modular and custom s/c bus, respectively. The consensus between participants was 94.4% which is also very good. The individual participant scores are shown in a box plot in Figure 5.29 on the right. It can be seen that the spreading is quite equal except for two outliers. One of the participants scored custom versus modular equal, resulting in a 50/50 weighting. This is also the reason that the red dots together do not sum up to 1.0, as for that participant there is no winner.

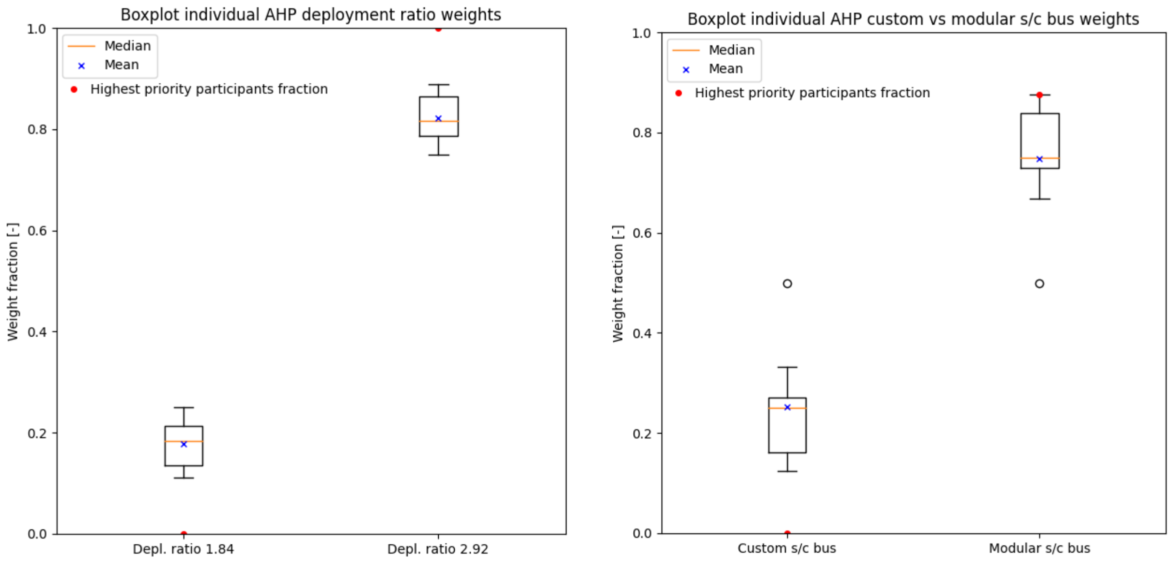


Figure 5.29: Left: Boxplot containing the individual scores for the AHP deployment ratio trade-off. Right: Box plot containing the individual scores for the AHP custom versus modular s/c bus trade-off.

Table 5.7: Table containing the results of the AHP trade-off for sub-criteria custom and modular s/c bus volume.

Concept	Custom s/c bus vol. 74.8%	Modular s/c bus vol. 25.2%	Overall depl. ratio
1: M2 linear, Baffle linear	5.4%	11.0%	6.8%
2: M2 linear, Baffle flaps	5.4%	11.0%	6.8%
3: M2 folding, Baffle linear	9.4%	9.4%	9.4%
4: M2 folding, Baffle flaps	12.8%	12.4%	12.7%
5: M2 folding, Baffle telescopic	12.8%	12.4%	12.7%
6: M1 folding, Baffle pantographic	10.8%	10.5%	10.7%
7: M1 folding down, M2 linear, Baffle pantographic	21.2%	16.9	20.2%
8: M1 folding down, M2 folding, Baffle pantographic	22.1%	16.4%	20.7%

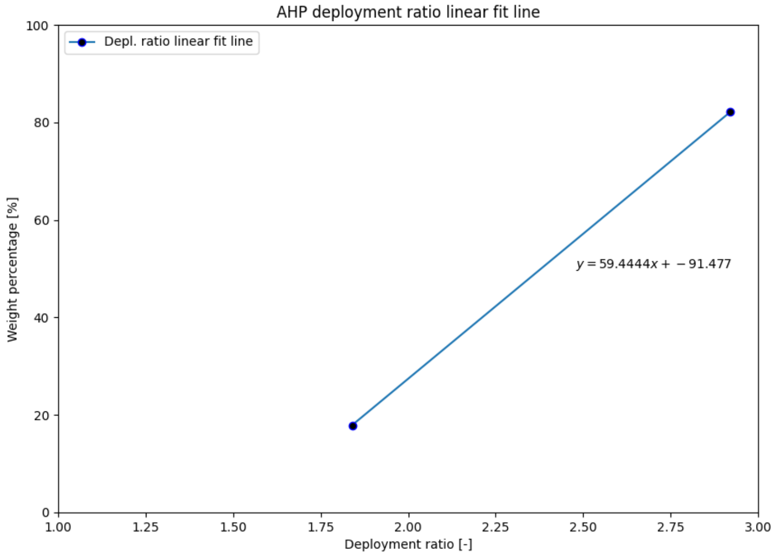


Figure 5.30: Graph showing the resulting AHP weights and a linear fit line.

Sub-Criteria Complexity

The consolidated weight per deployment option is shown in Table 5.8. A high percentage means a low complexity. The consensus between participants for the complexity weights is acceptable with a percentage of 77.8%. The individual participant scores are shown in a box plot in Figure 5.31. It can be seen that most of the spreading is in Baffle linear and M1 fixed. M2 fixed and M2 folding arms have the furthest outliers. The red dot indicates that for most participants, either M1 fixed or M2 fixed came out as winner of the trade-off. According to the position of the box, this is also expected. For one participant the linear baffle came out as winner. The complexity weight for each concept is calculated by summing the weights from Table 5.8 up according to the deployment options present in the concept. For example:

M1 fixed + M2 linear extensions + Linear baffle = 22.9% + 5.0% + 16.1% = 44.0%

After this is calculated the concept's complexity weights are normalized. The resulting weights for the main AHP trade-off are shown in Table 5.10.

Table 5.8: Table containing the results of the AHP trade-off for complexity.

Deployment option	Consolidated weight
M1 fixed	22.9%
M1 deployment	2.1%
M2 fixed	23.1%
M2 linear extensions	5.0%
M2 folding arms	6.8%
Baffle linear	16.1%
Baffle telescopic	9.9%
Baffle flaps	8.6%
Baffle pantographic	5.5%

Table 5.9: Table containing the results of the AHP trade-off for the criteria.

Criteria	Consolidated weight
Volume	32.9%
Complexity	26.3%
Scalability	16.5%
Reliability	24.2%

AHP criteria

The resulting weights for the AHP criteria are shown in Table 5.9. The consensus for this AHP pair wise comparison is rather low at 58.1%. This can also be seen back in the spreading in the box plot shown in Figure 5.32. The criteria weights are used in Table 5.10 to calculate the consolidated weight per concept.

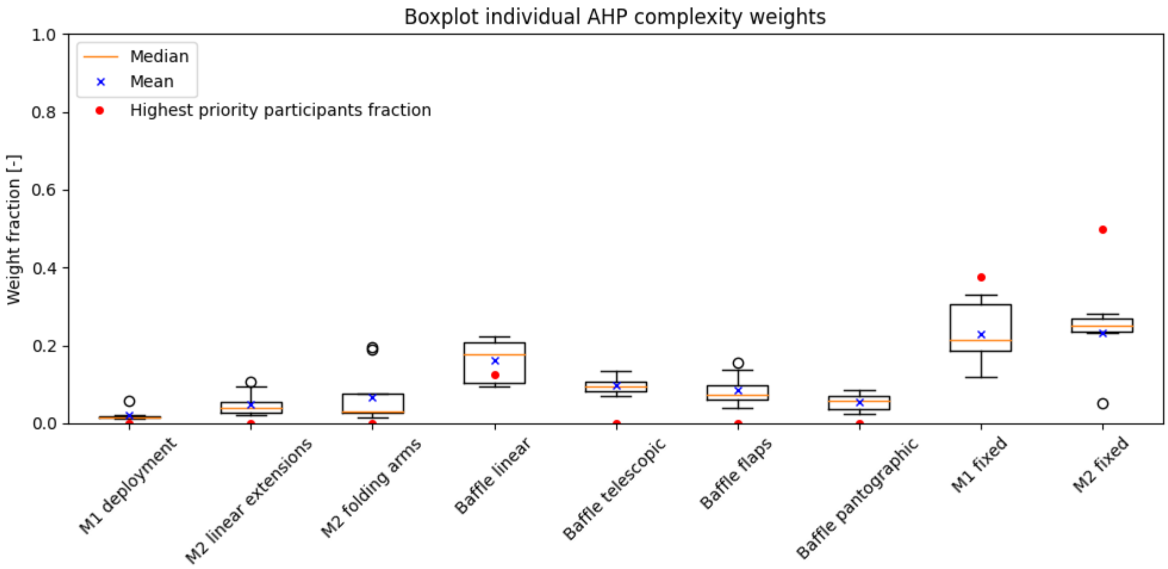


Figure 5.31: Box plot containing the individual scores for the AHP complexity trade-off.

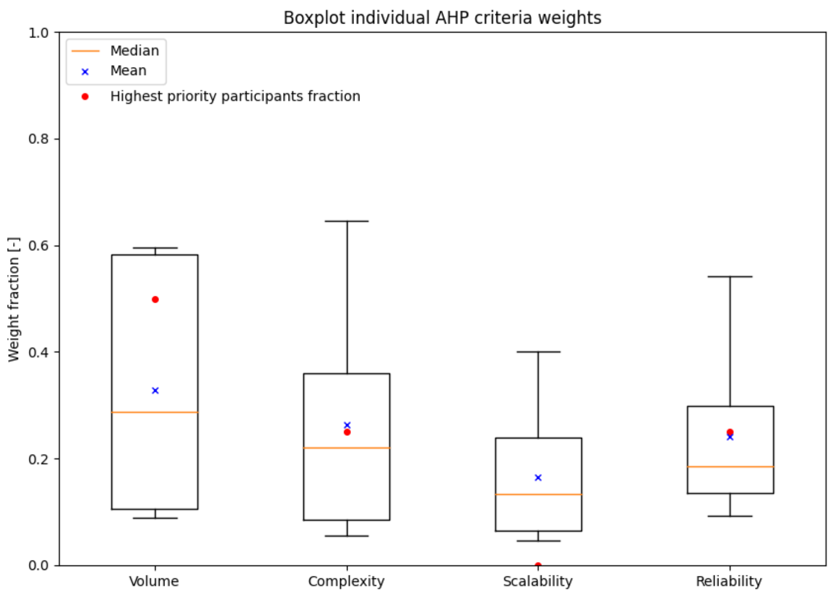


Figure 5.32: Box plot containing the individual scores for the AHP criteria.

AHP trade-off table

The AHP trade-off table is shown in Table 5.10. The winner of the trade-off is concept 1. The second best concept is inconclusive, as concepts 3, 5, 7 and 8 are very close together. Since the consensus on the criteria weighting is rather low, it is interesting to analyse the spread of the participants responses. This spread is shown in the box plot in Figure 5.33. If the mean and median are compared between concepts 3, 5, 7 and 8, it can be seen that they are very close together and that the spread is more or less equal, except for concept 5. The consensus on the score for concept 5 is quite high compared to concepts 3, 7 and 8. To get a better view on the individual scores, the individual ranking is shown in Table 5.11. A score of 1 is best and score of 8 is worst. On the right the average ranking per concept is shown. It can be seen that there is quite a clear consensus that concept 2 is the worst. Concept 1 scores overall and for most participants the best. The individual ranking for the other concepts is a bit more spread out. It is interesting to note that participants P5, P6 and P7 score concept 1 considerably lower than the other participants, while they score concept 8 considerably higher.

Their scoring for the other concepts is also quite similar to each other. These participants value volume over complexity and reliability, while for the other participants it is mostly the other way around. It is interesting to note that participants P5, P6 and P7 are TU Delft staff members and apparently have different priorities than the Thesis and PhD students.

Table 5.10: Table containing the results of the AHP trade-off.

Concept	Volume 32.9%	Complexity 26.3%	Scalability 16.5%	Reliability 24.2%	Consolidated
1: M2 linear, Baffle linear	6.8%	16.8%	3.7%	37.8%	16.4%
2: M2 linear, Baffle flaps	6.8%	13.9%	2.8%	5.4%	7.7%
3: M2 folding, Baffle linear	9.4%	17.5%	4.2%	23.3%	14.0%
4: M2 folding, Baffle flaps	12.7%	14.6%	7.2%	2.6%	9.8%
5: M2 folding, Baffle telescopic	12.7%	15.1%	12.8%	15.9%	14.1%
6: M1 folding, Baffle pantographic	10.7%	11.7%	4.5%	7.9%	9.3%
7: M1 folding down, M2 linear, Baffle pantographic	20.2%	4.8%	32.4%	4.1%	14.3%
8: M1 folding down, M2 folding, Baffle pantographic	20.7%	5.5%	32.4%	2.9%	14.3%

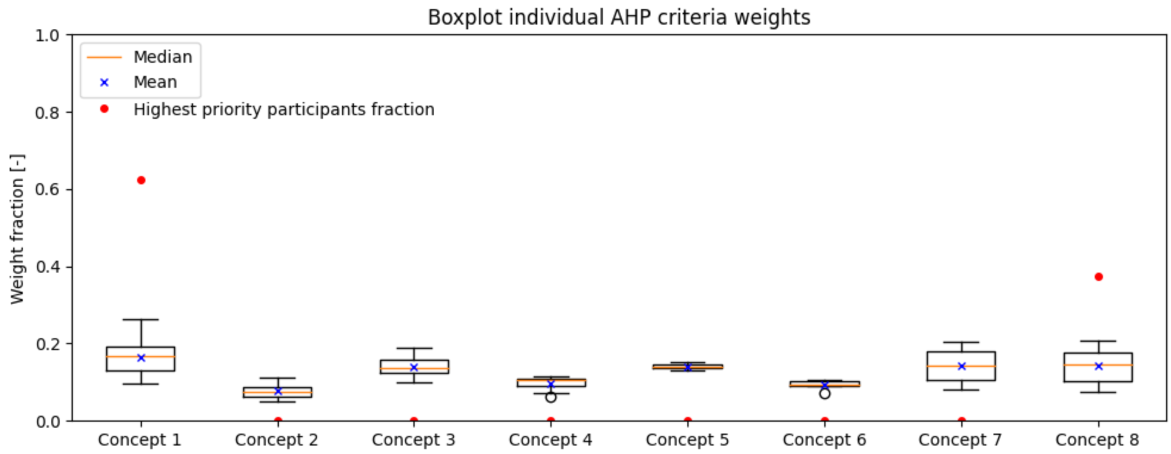


Figure 5.33: Box plot containing the individual scores for the final AHP trade-off.

Table 5.11: Table containing the individual rankings and calculated overall rankings. (P1 = Participant 1, etc.)

Concept	P1	P2	P3	P4	P5	P6	P7	P8	Average
1: M2 linear, Baffle linear	1	1	1	1	6	6	3	1	2.5
2: M2 linear, Baffle flaps	8	8	8	5	8	8	8	6	7.4
3: M2 folding, Baffle linear	4	2	5	2	5	5	5	2	3.8
4: M2 folding, Baffle flaps	6	6	7	4	4	4	6	8	5.6
5: M2 folding, Baffle telescopic	2	3	4	3	3	3	4	3	3.1
6: M1 folding, Baffle pantographic	7	7	6	6	7	7	7	4	6.4
7: M1 folding down, M2 linear, Baffle pantographic	5	4	2	7	2	2	2	5	3.6
8: M1 folding down, M2 folding, Baffle pantographic	3	5	3	8	1	1	1	7	3.6

5.9.3. Trade-off conclusion

While in the graphical trade-off there was no decisive winner, the AHP trade-off showed that concept 1 scores slightly better than the other concepts. However, the differences were relatively small and concepts 3, 5, 7 and 8 scored also pretty good. The deployment ratio of concept 3 is slightly higher while the complexity scores slightly lower. Concept 5 scores acceptable on all criteria. In the AHP trade-off it scores similar to concept 3 while in the graphical trade-off the complexity and reliability are scored lower. This can also be seen in Table 5.10 where it makes up for the lower complexity and reliability by scoring better for scalability and volume. When a custom spacecraft (s/c) bus is put under discussion again, indicated as 1.c in Table 5.12, then concept 1.c scores best out of the concepts in the graphical trade-off. However, the preference of the team members for a modular s/c bus is 76.2%.

Table 5.12: Deployment concept trade-off table for the remaining concepts.

Concept \ Criterion	Depl. ratio	S/c bus	Volume 32.9%	Complexity 26.3%	Scalability 16.5%	Reliability 24.2%
1: M2 linear, Baffle linear	1.84	Mod.	Correctable	Excellent	Correct.	Good
1.c: M2 linear, Baffle linear	2.43	Cust.	Acceptable	Good	Correct.	Good
3: M2 folding, Baffle linear	2.06	Mod.	Acceptable	Good	Correct.	Good
5: M2 folding, Baffle telescopic	2.25	Mod.	Acceptable	Acceptable	Accept.	Acceptable

Concepts 7 and 8 were deemed unacceptable in the graphical trade-off for their complexity. This unacceptable complexity score can be seen back in the AHP trade-off table, Table 5.10, with scores of around 5% versus around 14% for the other concepts. However, since concepts 7 and 8 score very high on volume and scalability their consolidated score is still good. This is something the AHP method does not take into account. Concepts that score average on all criteria have a resulting weight that is the same as concepts that score very poor on some criteria and excellent on other criteria. The result is that these concepts are perhaps infeasible, yet they score quite good. Therefore, another AHP calculation was done with only the three concepts that were deemed feasible from the graphical trade-off. The results of this AHP trade-off are shown in Table 5.13. It can be seen that the result is now reversed. Suddenly concept 5 scores highest, followed by concept 1 and concept 3. This phenomena is known as rank reversal. [24] There are ways to calculate the AHP scores differently, which preserves the correct ranking. However, since the resulting scores are very close together in both AHP trade-offs, it is concluded that there is no clear winner. The rank reversal also proves that the trade-off is very sensitive.

Table 5.13: AHP trade-off table containing the three remaining concepts from the graphical trade-off.

Concept	Volume 32.9%	Complexity 26.3%	Scalability 16.5%	Reliability 24.2%	Consolidated
1: M2 linear, Baffle linear	22.6%	35.0%	11.3%	61.4%	33.4%
3: M2 folding, Baffle linear	32.7%	34.9%	17.9%	26.8%	29.4%
5: M2 folding, Baffle telescopic	44.6%	30.1%	70.9%	11.7%	37.4%

From the trade-off process it can be concluded that the results are very sensitive and that a more detailed analysis should be performed into the concepts. On the system level a decision cannot be made. Since the M2 and baffle deployment are uncorrelated, a detailed analysis and trade-offs at subsystem level can be performed. The top-level trade-off has helped the subsystem design in limiting the options for the subsystem trade-offs. In case the subsystem trade-offs do not provide a conclusion, the decision could be made to prototype the concepts with breadboarding. This should provide more insight and a good basis to make a final decision for the deployment concept.

Furthermore, assumptions are now made for the concepts in the high level SE trade-off. These assumptions can be incorrect and greatly impact the results of the trade-off. If the trade-off showed a clear winner, then the sensitivity of the trade-off for these assumptions could be discarded by the clear results. However, since the results of both the graphical trade-off and the AHP trade-off are very close together, this sensitivity cannot be neglected.

In the trade-off scalability was taken into account. It was found that the designs with both primary mirror (M1) and secondary mirror (M2) deployment score best for this criterion. The main reason for this is that they have a good deployment ratio and have low integrated spacecraft (s/c) bus volume. This makes them excellent for a scaled version. Another option for a scaled version could be M2 folding with a telescopic baffle. The folding arms can scale along with a larger aperture without increasing the height too much. The same holds for the telescopic baffle segments. It can therefore be concluded, that for a 1.5m TIR DST a deployable M1, M2 and Pantographic baffle is the most favourable followed by M2 folding and baffle telescopic.

Team Decision

To move towards the detailed design phase, a decision had to be made regarding the concept. Therefore a team design review was planned to discuss the trade-off results. In this team meeting the following subjects were discussed:

- Definitive decision for a modular s/c bus or can a custom s/c bus still be considered?
- Choose for M2 folding arms or M2 linear extensions?

After the explanation of the trade-off results, the team made the decision to not consider a custom s/c bus and to focus on a modular design. Therefore, concept 1C was dropped. From the trade-off performed by C. Hobijn, folding arms are the winner for the secondary mirror (M2) deployment. The results are 18.0% versus 15.3% for articulated booms (folding arms) and ball & lead screw (linear extension), respectively. These results are close together and since the trade-off was only performed by C. Hobijn, it was deemed quite sensitive. However, to have a clear concept to continue with, the team made the deliberate choice to drop M2 linear extensions, and therefore concept 1, and continue with concepts 3 and 5. From the trade-off performed by F. Hu, the linear baffle deployment appeared to be slightly better than the telescopic baffle deployment. He concluded that both the linear and telescopic baffle could use the same actuation system, with the most simple system being springs. The telescopic baffle could be seen as a linear baffle with multiple segments. The development path of both deployment systems is closely related. Since this trade-off was also sensitive, the team decided not to make a choice between concept 3 and 5 yet. A more detailed analysis is required to make the final decision between a single segment or multiple segment linear baffle.

6

SE Toolset Evaluation

After the the top-level systems engineering (SE) was completed and the team worked with the SE Toolset, the SE Toolset could be evaluated. This is done in this chapter. In Section 6.1, the user experience is discussed. Next, in Section 6.2, a critical analysis into the systems engineering (SE) performance of the SE Toolset is performed. In Section 6.3, the limitations of the SE Toolset are discussed. In Section 6.4, the compliance table. In Section 6.5, the SE Toolset is evaluated for further development and implementation in other projects. In Section 6.6, the comparison is made with IBM Doors, which supports classical SE, and with Capella, which supports MBSE. In Section 6.7, a SWOT analysis is carried out.

6.1. User Experience

To validate the user experience, the team members were asked to fill out an evaluation form. The blank form and results can be found in Appendix E. From the evaluation it was concluded that the ease of use and layout of the program can be improved slightly. With an average of 7 and 7.7, respectively, the score is not bad. However, there are some comments to improve the navigation and to make it a bit more clear where everything can be found. On other aspects the team members indicated that the program works faster than an excel sheet, gives a good overview of the overall system, makes dependencies clearer and improves the SE in the project. The team members indicated that the SE Toolset can especially save time in setting up and checking requirements. It was also found that the team members mostly use the requirements functionality. One of the team members, D. Dolkens, worked with Polarion for his work.[1] He said that Polarion is quite an extensive and sometimes complicated program. He furthermore stated that Polarion has somewhat more functionality,but that the SE Toolset was a lot simpler to work with. For the DST project, and small projects in general, the SE Toolset is perfect according to D. Dolkens. With the positive results of the evaluation it was deemed that requirements SE-LEA-01, SE-LEA-02 and SE-REQ-01 were fulfilled.

The team members responded that the program could save them multiple days when setting up requirements. Furthermore, they responded that the program is also very effective in checking requirements and updating them. This supports the goal of the program to reduce the SE work load for the team members. Keeping track of the identifiers, version control, ownership and update status is all taken out of their hands. It was furthermore indicated by the team members that it is now very easy to view requirements from other subsystems. Before the SE Toolset these requirements were all stored locally. Now, they can just scroll through the requirements list and find what they need, without having to ask team members. Therefore, it was deemed that requirement SE-LEA-05 was also fulfilled.

An improvement point mentioned by the team members is a sorting function for requirements to sort on subject rather than sub-system. For instance, label the requirements that are related to thermal, structural, launch-specific and deployable. Another improvement point is to add labels to requirements to improve the search function. This could be combined with the sorting function. Other improvement points are to add verification and validation and to add a personal list of requirements the user views most. These improvement points will improve the user experience and are advised to implement in the next version of the SE Toolset.

[1] `https://polarion.plm.automation.siemens.com/` accessed 2 December 2021

Some other improvement points, such as scrolling issues, were a bug in the program which have been solved right away.

6.2. SE Performance

When team members update their requirements or budgets, the top level SE is automatically updated as well. The total budget is regenerated and the requirements are visible in the overall SE. In the SE Toolset, the team members perform their SE work while the program handles the overall SE. This could be further improved by adding support for formulas and equations. In this way low level requirements can be linked through variables to top level requirements or to sub-system requirements. With this functionality the program gets a step closer towards Model Based Systems Engineering (MBSE). Therefore, it is deemed that requirement SE-AGI-01 is partly fulfilled.

Regarding the workload on the systems engineer it was deemed that it was sufficiently reduced to meet requirement SE-LEA-04. The systems engineer only needs to check the SE checklist to make sure all requirements are formulated correctly and enough information is provided. Furthermore, the systems engineer needs to check the mass budget and if the component masses are added correctly. This can be done on a weekly basis or even less often, depending on the team's activity. Using the email notification function of Sharepoint, the systems engineer can be notified when changes are made. The changes can already be seen in the email. Hence, for quick verification the program does not have to be opened. In the current format of the SE Toolset, the role of the systems engineer is to check user input and to maintain the SE Toolset. In contrast to classical systems engineering, in which requirements and budgets are maintained manually by the systems engineer. Therefore, it is deemed that the work load of the systems engineer is minimized as much as possible.

Document handling should preferably be done inside the SE Toolset. Document identifiers and naming could then be automated. However, due to time limitations and limited functionality to support this in Microsoft Access, this was not implemented. Requirement SE-REQ-03 is partly fulfilled by having the users fill in the description field of the requirements in which references to documentation can be made. If this could have been done (partly) automatically than the requirement would have been fully fulfilled. In Section 6.5 it is described how this could be implemented.

Two requirements that were not met are SE-AGI-02 and SE-AGI-03. Due to time limitations this functionality could not be implemented anymore. In Section 6.5 it is described how this could be implemented.

6.3. SE Toolset Limitations

Although the SE Toolset proved to be effective in the DST project, it also has some limitations. The first limitation is that there is no customer support. The program was created by the author and after this thesis he is no member of the DST project anymore. In case bugs or errors occur later in the program, then there is no person who can quickly fix it without sorting out all the coding. Commercial programs generally have good customer support which eliminates this problem.

The second limitation is that the SE Toolset became slower as the functionality grew. Some parts were not as efficient as they could be, for example the variables. In Microsoft Access quite complex and slow queries were needed to replace the variables with it's value to show it correctly in the requirements list. With other programs, such as Python, this could have been done more efficiently. This is also one of the reasons why the document handling, system versioning and iterations have not been implemented. It was found that this functionality would be difficult to implement efficiently. Therefore, it was deemed that a better solution would be to transfer the program, to Python for instance, and expand the functionality there. Due to time limitations this could not be done in this thesis and is recommended for further development.

6.4. Compliance Table

The requirements compliance table is shown in Table 6.1. In the table, the requirements are listed and if the SE approach and SE Toolset comply. It is also listed in which section the requirement's compliance is discussed.

Table 6.1: Table containing the requirements, if the SE approach and SE Toolset are compliant and in which section the requirement is discussed.

Identifier	Requirement	Compliant	Section
SE-COM-01	A central communication platform shall be used	Yes	Sec. 3.3.3
SE-COM-02	Team meetings shall be held on a regular basis	Yes	Sec. 3.3.3
SE-COM-03	Emails relevant for the team shall be stored in a central location	Yes	Sec. 3.3.3
SE-COM-04	The SE Toolset shall support communication features	Yes	Sec. 4.2
SE-COM-05	The SE Toolset shall support update notifications	Yes	Sec. 4.2
SE-INT-01	The SE Toolset shall always be available to team members	Yes	Sec. 4.2
SE-INT-02	The SE Toolset shall support requirements	Yes	Sec. 4.2
SE-INT-03	The SE Toolset shall support budgets	Yes	Sec. 4.2
SE-INT-03-01	The budgets shall be linked to requirements	Yes	Sec. 4.2
SE-INT-03-02	The budgets shall update automatically when requirements change	Yes	Sec. 4.2
SE-INT-04	The SE Toolset shall have a clear overview of the sub-systems structure	Yes	Sec. 4.2
SE-DOC-01	Documents shall have a unique identifier	Partly	Sec. 4.2
SE-DOC-02	Documents shall contain the name of the author	Partly	Sec. 4.2
SE-DOC-03	Documents shall contain the creation date	Partly	Sec. 4.2
SE-DOC-04	Documents shall contain the revision history	Partly	Sec. 4.2
SE-DOC-05	The references made in the documents shall be traceable	Partly	Sec. 4.2
SE-DOC-06	Documents shall be stored in a central folder accessible to all team members	Yes	Sec. 4.2
SE-DOC-07	A library shall be created in a central location	Yes	Sec. 3.4.2
SE-DOC-07-01	The library shall contain a search function	Yes	Sec. 3.4.2
SE-DOC-07-02	The library shall contain the previous theses of the DST project	Yes	Sec. 3.4.2
SE-REQ-01	The SE Toolset shall document the requirements in a clear way	Yes	Sec. 6.1
SE-REQ-01-01	The SE Toolset shall support version control	Yes	Sec. 4.2
SE-REQ-01-02	The SE Toolset shall automatically generate a unique identifier	Yes	Sec. 4.2
SE-REQ-01-03	Requirement identifiers shall not be reused after a requirement is deleted	Yes	Sec. 4.2
SE-REQ-01-04	All possible requirement fields shall be automatically filled in	Yes	Sec. 4.2
SE-REQ-02	Changes to the requirements shall be peer reviewed by a systems engineer	Yes	Sec. 4.2
SE-REQ-02-01	The systems engineer shall be notified when a change to a requirement is made	Yes	Sec. 4.2
SE-REQ-03	Requirements shall be traceable to documentation where they are based on	Partly	Sec. 6.2
SE-DEC-01	Minutes of meeting shall be made for every team meeting	Yes	Sec. 3.3.3
SE-DEC-01-01	Minutes of meeting shall be easily accessible in a central location	Yes	Sec. 3.3.3
SE-DEC-02	Design decisions shall be documented in central documentation accessible to all team members	Yes	Sec. 3.3.3
SE-DEC-02-01	The team members involved in the design decision shall be stated in the documentation	Yes	Sec. 3.3.3
SE-DEC-03	Design decisions shall be traceable	Yes	Sec. 3.3.3
SE-DEC-04	Design decisions shall be communicated to all team members	Yes	Sec. 3.3.3
SE-LEA-01	The SE Toolset shall have an intuitive interface	Yes	Sec. 6.1
SE-LEA-02	The SE Toolset shall be easy to use	Yes	Sec. 6.1
SE-LEA-03	The SE Toolset shall automate everything that does not require user input	Yes	Sec. 4.2
SE-LEA-04	The SE Toolset shall reduce the systems engineer's work load to a minimum	Yes	Sec. 6.2
SE-LEA-05	The SE Toolset shall minimize the SE work load for team members	Yes	Sec. 6.1
SE-AGI-01	Adjustments to the low level SE shall be easily adopted by the top level SE	Partly	Sec. 6.2
SE-AGI-02	The SE Toolset shall support the creation of different versions of the design	No	Sec. 6.2
SE-AGI-03	The SE Toolset shall support multiple iterations of the design	No	Sec. 6.2

6.5. Future Use

The SE Toolset proved to be suitable for the DST project and therefore for small and dynamic projects in general. Functionality which is mostly used, is the requirements generation and maintenance. One of the goals of the SE Toolset was to limit the workload for team members as much as possible. This goal was definitely fulfilled as the team members pointed out that the SE Toolset can save them several days of work when setting up requirements. Later on in the project, when they have to check and maintain requirements, they also expect to save time and effort.

Additional functionality, which is important to add, is the compliance of the requirements to extend the program towards verification and validation. Other parts of the program, such as the budgets, were not used much by the team members. The same is expected for risk identification and risk mitigation, which are not yet part of the program. Therefore, it could be argued if these parts add value to the program. The systems engineer could also perform that work in his own documentation. However, if it is integrated with the SE Toolset it can partly be automated. The risk identification should best be integrated with the requirements creation. The systems engineer then only has to check if the requirements need risk mitigation or not and if he agrees with the risk identification. The well known heat maps for the risk identification could be created automatically and criteria could be provided to check which risks should be mitigated to automate it even further. The goal would be to automate as much as possible in order to reduce the systems engineer's work load, while still keeping the systems engineer in the loop to check relevant input/output.

For document handling, the intention was to use a Python script to retrieve the text from previous theses. The retrieved text, preferably per section, could then be stored in Microsoft Access together with a table of keywords per thesis. A search function to search through all theses could then be implemented in the SE Toolset. Other functionality was to link the theses to the configuration item tree. By clicking on a sub-system, the theses with the respective percentage of correspondence could then be showed. If expanded with Natural Language Processing (NLP), the SE Toolset could even provide the user with the respective sections which are most relevant. The SE Toolset could even show the relevant text inside the program. Other possibilities are that an automated overview is created which shows the relations between the theses. This could also be expanded to documentation. If on the first page the author, date, document identifier, version and input documents are stated, then this can be automatically imported to the SE Toolset. The SE Toolset can then create an overview of the documents and their dependencies. For requirements documentation, reports could be generated from the SE Toolset. Although all this functionality could be very powerful, it proved quite difficult and time consuming to implement.

To implement the version handling and iterations the following could be done. An extra layer can be added on top of the current SE Toolset. In this additional layer the user could select the iteration or version he wants to view. This opens the specific data set. Functionality could be added, such that cross links between different versions or iterations can be made. Although this functionality can be implemented, it takes quite some work to set this up. The requirements, variables and comments have to get an extra identifier as to which version/iteration they belong. In this way they can be stored in the same data tables. The most work is the user interface which makes it possible to create new system versions or iterations. Preferably it should be possible to do this not only on system level but also on sub-system level. The implementation of this functionality is left for further development.

For every SE approach, communication is always of key importance. It is therefore very important to continue having team meetings and keep each other updated on progress. A feature that could provide additional support is Power Automate. It is available in Sharepoint and is a feature that can automatically post messages in Microsoft Teams or send emails. It can be fully customized to trigger on a change in a Sharepoint list and act according to a pre-programmed workflow. This functionality was briefly tested with good results. Implementation is left for further development, due to time limitations. A simple alternative is to turn notifications on in a Sharepoint list. An email notification will be send to the user in case something changes in the list. The update frequency can be set to daily, weekly or right away.

It could be argued that by adding all above mentioned functionality, the program becomes too complex and big to still maintain properly. The chance of bugs increases and more maintenance is required, which requires a dedicated systems engineer. Therefore, it could be opted that a commercial available program is more suitable. Although the customer support will be better, the customization possibilities will be less. The SE Toolset can on the other hand be implemented in different formats, depending on the projects needs.

In the most basic form, the SE Toolset could consist of only the configuration item tree, requirements list, requirement details form and the new requirement form. In this basic setting, it can already be very effective for small projects to maintain the requirements. In the current format for the DST project, additional functionality such as budgets, variables, commenting system and SE checklist is available. In this format the SE Toolset is very suitable for small projects with a dynamic setting and limited resources, e.g. lack of dedicated systems engineer. The SE Toolset can be further expanded with verification, validation, risk identification and risk mitigation. The program can then also be applied in later stages of a project, e.g. testing, integration and production. The program can be expanded even further with document handling and automated extracting of data from 3D models. With all these futures it resembles a commercial Model Based Systems Engineering (MBSE) program. However, the core value of the SE Toolset will still be that the input of users and a systems engineer will always be necessary, as the human interaction/intelligence cannot be replaced by pre-programmed automation. The data retrieved from 3D models could for instance be used to automatically check if requirements and budgets are met. However, a systems engineer should always check these results and act accordingly.

6.6. Commercial Available Programs

In this section the SE Toolset is compared to a requirements management tool and a Model Based Systems Engineering (MBSE) program. This is done to provide the framework in which the SE Toolset fills the gap between classical systems Engineering (SE) and MBSE. The program is compared to IBM Doors and Capella, in Section 6.6.1 and 6.6.2, respectively. In Table 6.3, on page 69, the comparison table between the SE Toolset, IBM Doors and Capella is shown.

In the SE Toolset trade-off, in Chapter 4, Capella was considered as a program to construct the SE Toolset. In this section Capella is considered as a stand alone MBSE tool. Therefore, the SE Toolset and Capella can be compared in this section at the same level, independent of the trade-off results.

6.6.1. Requirements Management Tool: IBM Doors

The first big difference with IBM Doors, is that IBM Doors is a paid tool while the SE Toolset is free, as long as you have Microsoft Office. At a cost of 164 dollar per month per user IBM Doors is quite expensive.[2] In the current DST project setting, this would amount to almost 16,000 dollar per year. It was found in Section 6.1, that the team members used the requirement functions most. Especially when they are formulating the requirements, they use the program for several days. However, once the initial requirements are set, the team members only use the program when they need to check requirements or make updates. For this type of usage, the cost of IBM Doors is unreasonably high for the DST project. IBM Doors has similar functionality to the SE Toolset with some additional user settings to customize viewing of requirements, comments and changes. [17] The SE Toolset on the other hand has a dedicated SE Checklist for the systems engineer to work in. The systems engineer only has to check this on a regular basis to stay updated and to make sure the requirements are all correct. The SE Toolset also automatically warns the systems engineer when an ownership transfer is required. The SE Toolset furthermore supports budgets. This is not supported in IBM Doors.

In functionality, IBM Doors and the SE Toolset do not differ that much, except for the budgets. IBM Doors has a higher maturity and customer support while the SE Toolset is simpler, easier to use and free. For small projects with limited resources the SE Toolset is deemed better, mainly due to the pricing. For larger projects the SE Toolset could be limiting due to a low maturity.

6.6.2. MBSE Tool: Capella

Both Capella and the SE Toolset are free. However, Capella has much more functionality because it is a dedicated (MBSE) tool with a higher maturity. Additional functionality includes: operational architecture diagrams, capabilities diagrams, dataflow diagrams, sequence diagrams, modes and states diagrams, classes and interfaces, model validation and more.[3] With this functionality, a computer based model of the design can be made in which changes are automatically adopted. This can be very useful in projects with one or more dedicated systems engineers. The dedicated systems engineers are required to maintain the model and to implement input from team members into the model. For the DST project, with no dedicated systems engineer, it would mean that the team members themselves need to maintain the model. With the dynamic resources. this is very inefficient and is expected to give errors and discrepancies.

In that aspect, the SE Toolset is more suitable as it limits the need for a dedicated systems engineer. A systems engineer is still required, but only needs to maintain the SE Toolset and check the input of team members. A downside to the SE Toolset is that it is custom made with no customer support. If bugs occur, than a team member needs to go through all the coding to solve the problem. With Capella, there is customer support that will help with bugs in the program itself. Furthermore, there is community support. This can be helpful when problems arise in the model. However, detailed knowledge of MBSE and the modelling language of Capella,

[2] https://www.g2.com/products/ibm-engineering-requirements-management-doors-next/pricing?__cf_chl_captcha_tk__=sctsY4mkc2ay83h_Q1neXtZkv6D0u95GJhNBqePCcCY-1638270842-0-gaNycGzNCn0, accessed 30 November 2021

[3] https://www.eclipse.org/capella/features.html accessed 2 December 2021

which is based on SysML and UML, is required.[4] Detailed knowledge of the interactions and how the Capella model was set up is also required. Capella is the tool to create the Capella model, just like Microsoft Access is the tool to create the SE Toolset. The Capella model needs to be adjusted and maintained during the project, while in the SE Toolset only the requirements and budgets need to be maintained. This is a fundamental difference, which makes the SE Toolset much simpler than a Capella model.

For state-of-the-art designs, it can furthermore be difficult to model all interactions and especially the interactions with the space environment. For these cases, a MBSE approach is less suitable. A classical SE approach on the other hand, can be difficult to maintain without errors and requires constant bookkeeping from a dedicated systems engineer. This makes it less suitable for state-of-the-art and fast developing projects. To fill this gap, the SE Toolset can prove useful by automating the classical SE but not becoming as complicated as MBSE.

From Section 6.1, it was found that team members mostly use the requirements functionality and update the mass budget once in a while. It can therefore be concluded that for a project such as the DST project, a MBSE approach is too extensive, complicated and time consuming. The dynamic setting requires a simple program that can handle misuse, without the need for constant monitoring of a systems engineer. The SE Toolset is therefore a good alternative as it saves time by automating as much as possible, while still involving the systems engineer closely to check important input and information. Furthermore, it can provide a better platform for agile projects with multiple different versions and iterations. Although further development is needed, the SE Toolset proved to have potential to fill the gap between classical systems engineering and MBSE, without getting as complicated and time consuming as MBSE.

6.7. SWOT Analysis

To show the potential of the SE Toolset and possible commercialisation, a SWOT analysis was carried out. In Table 6.2 the SWOT analysis is presented.

Table 6.2: SWOT Diagram for the SE Toolset.

Strengths • Good and easy sharing capabilities • Simple to distribute to other projects • Good integration with Microsoft (Sharepoint and Teams) • Easy to use • Transparent • Minimizes workload on user and systems engineer	**Weaknesses** • No customer support • Largely untested • Not certain if it is made in the right programming language • Fully dependent on Microsoft
Opportunities • Fill gap between classical SE and MBSE • Potential commercialization if it proves to fill a gap in the market • Become a stepping stone to transform classical SE based projects towards MBSE	**Threats** • Commercial available programs that have the same functionality • These kind of tools are mostly used in conservative industries, can be applicable to other industries but there is likely a lower need

[4] https://www.eclipse.org/capella/arcadia_capella_sysml_tool.html, accessed 16 December 2021

Table 6.3: Comparison table between SE Toolset, IBM Doors and Capella.

Criteria	**SE Toolset**	**IBM Doors** [17] [5]	**Capella** [6] [7] [8] [9]
Customer support	No customer support	Good customer support	Community support or paid third party support
Customization	Everything can be customized, program workflow, functionality and layout, Access knowledge is required	Program cannot be customized, customization of layout is limited	Model can be customized, program layout cannot be customized, more or less fixed workflow
Learning curve	Easy to start with program, intuitive, to customize/change the program lot of examples on the internet, programming knowledge required to customize program	Easy to start with the program, layout and functionality is clear	Steep learning curve, not easy to start with the program without training, knowledge of workflow/MBSE/SysML/UML/Capella is required
Price	Free of charge as long as Microsoft Office is available	Expensive at 165 dollar per user per month	Free of charge, add-ons can cost money
Sharing	Easily shared through Sharepoint or SQL server	Easily shared when multiple subscriptions are available	Sharing requires paid add-on
Requirements handling	Good functionality, clear overview, version control, traceability could be improved	Good functionality, clear overview, version control, good traceability of changes	Good functionality, also supports model created requirements besides textual requirements, Requirements Viewpoint add-on is advised
Configuration item tree	Can be generated from a table in a diagram, quite static, pre-programmed layout, but is automatically updated according to table, requirements are linked to configuration item tree	Can be constructed as a diagram, layout can be changed through the program, requirements can be linked to configuration item tree (Product Breakdown Structure (PBS))	Can be constructed as a diagram, layout can be changed through the program, requirements can be linked to configuration item tree
Document handling	Not available, further development needed	Not designed to handle documents, import and export functions to word are available	No document handling, only document exportation via M2Doc add-on
Engineering budgets	In current SE Toolset only mass budget available, can be expanded and customized, budget is automatically generated from linked variables that can also be linked to requirements	No functionality found	No functionality for engineering budgets [8]

[5] https://www.ibm.com/docs/en/SSYQBZ_9.5.2/com.ibm.rational.doors.integrating.doc/topics/doorsrmf_user_guide.pdf accessed 3 December 2021
[6] https://www.eclipse.org/capella/addons.html accessed 3 December 2021
[7] https://www.obeosoft.com/en/team-for-capella accessed 3 December 2021
[8] https://www.eclipse.org/community/eclipse_newsletter/2017/december/article5.php accessed 3 December 2021
[9] https://www.eclipse.org/capella/services.html accessed 20 December 2021

7
Conclusion

In this thesis the needs on the top-level systems engineering (SE) for the DST project have been investigated. It was found that there was a need for a top-down trade-off for the DST TIR demonstrator and a comparison with reference satellites. It was furthermore found that a more automated version of the classical SE approach was needed for the DST project. The DST project was used as a running case study to create an automated program, the SE Toolset, to support the new SE approach. In the introduction the research questions have been defined which will be answered here. The first research questions is:

1. *What are the most important needs for top-level systems engineering for the TIR DST project?*

The first important need was that a different SE approach was required which is agile, simple to use, supports research and reduces the work load of both team members and the systems engineer. The requirements on this new SE approach were found by analysing the short comings on the SE previously performed in the project. From this analysis, it was also found that a top-down trade-off based on the goals and development of the DST project, and for the DST TIR demonstrator mission specifically, was needed. The third important need, was to compare the DST concepts with reference satellites to verify if they were state of the art and could compete with fixed telescopes. This answered the first research question.

In the top-level trade-off, it was found that with the change to a TIR demonstrator with 30cm aperture, other deployment options than the options used in the visual (VIS) design, became more favourable. For a demonstrator mission, the complexity of primary mirror (M1) deployment was deemed too complex by the team. Therefore, a design with M1 fixed was most optimal. Three options were very close together. The first option was to use linear extensions for the secondary mirror (M2) and a linear baffle. The other two options were to use M2 folding arms combined with either linear single segment baffle deployment or multiple segment deployment. The results from the top-level trade-off were however very close together and inconclusive. The results were very sensitive as there was no good consensus on the criteria weighting and the concepts were based on estimations. Even rank reversal was encountered in the AHP trade-off when the concepts that were deemed unacceptable in the graphical trade-off, were discarded for the AHP trade-off. Therefore, no winner was found and the sub-system level trade-offs were considered to base a final decision on.

According to the trade-off performed by C. Hobijn, M2 folding arms perform best. However, in this trade-off the results were also close together and the trade-off was deemed sensitive as it was only performed by C. Hobijn. To still be able to continue with a design into the detailed design phase, the team made the deliberate choice to go for M2 folding arms, despite the sensitivity. To resolve the sensitivity, breadboarding is essential. However, given the limited resources, only one concept can be worked out at a time. According to the trade-off performed by F. Hu, both single and multiple segment linearly deployed baffles are feasible options. The results of this trade-off were also close together and sensitive. Since the single and multiple segment linearly deployed baffle are on a similar development path, the team decided not to make a decision yet. A more detailed analysis should be performed before this decision can be made. Therefore, the focus for the TIR demonstrator design was set to a modular s/c bus with M2 folding arms combined with a single or multiple segment linearly deployed baffle. A modular s/c bus means that the DST is added as a box, with a predefined electro-mechanical interface on one side, onto a satellite platform.

In the reference satellite comparison it was found that the DST concepts perform well with respect to reference satellites, except for the DST VIS 30cm. The DST TIR 30cm, DST TIR 1.5m and DST VIS 1.5m all proved to be state of the art and have good performance. The DST VIS 30cm performed similar to Satellogic-NuSat and iSIM-170, which are state-of-the-art fixed telescopes with lower complexity. Therefore, it appeares that there is no benefit in a deployable VIS 30cm aperture telescope. Since the maturity of the DST VIS 30cm is low, further detailed analysis into the concept should be done to confirm this. From the comparison, it furthermore appeared that deployable optics have a higher benefit for larger aperture telescopes, as the 1.5m concepts showed a greater potential than the 30cm concepts. It can be concluded, that the performed comparison provides insight in the position of the DST concepts with respect to competitors and that most of the current DST concepts appear to be promising. Further investigation and research is required to confirm and expand the comparison that was done in this thesis. Combined with a market analysis to investigate possibilities to fill a market gap, this could potentially make the DST road map clearer and more specific. This also leads to clearer short term and long term goals for the DST project. The second research question is:

2. *What is the most suitable systems engineering approach for the TIR DST project?*

It was found that an automated version of classical SE would suit the TIR DST project best. The SE Toolset which was created to support the SE approach, was set up in Microsoft Access. The tool handles the configuration item tree, requirements, budgets and variables. The requirements and budgets are linked to (sub)-systems through the configuration item tree. The requirements and budgets are linked through the variables. The SE Toolset handles the overall SE, while the team members only have to perform their own SE in the program through a simple interface. In this format, the workload for the team members and systems engineer is minimized. The SE Toolset has good traceability and transparency due to version control and a commenting system. This answered the second research question. The third research question is:

3. *How can the suitability of the systems engineering approach be proven?*

The suitability of the SE approach was validated by letting the users work with the program and fill out an evaluation form. The evaluation form focused on the usability and functionality of the program. Their input and suggestions also have been discussed in progress meetings. This provided valuable insight into the users perspective. The suitability of the SE approach was validated from a SE perspective using a qualitative analyses. It was found that for the DST project the SE capabilities of the developed tool are sufficient. At the moment of writing, the team members mostly use the requirements functionality for setting up requirements and administering requirements. For this application the SE Toolset has proven to be suitable. In the future, verification and validation functions should be added to the SE Toolset to extend the applicability to testing, integration and production. It is also advised to add risk identification and mitigation. This is all discussed in Chapter 6 and answers research question 3. The last research question is:

4. *For what kind of projects is the systems engineering approach applicable?*

It was found that the SE Toolset is especially useful in small projects, less than 10 team members, with limited resources. The current SE Toolset is not fully optimized, which can make it slow for large projects. The clarity can furthermore decrease in large projects, due to the amount of requirements, size of the configuration item tree and the variables. If the layout and program are optimized for larger projects, the applicability could possibly be extended. Since the SE Toolset minimizes the need for a systems engineer, it is very suitable for projects with limited resources. By automating the processing of user input it reduces the work load for the users. In this setup the team members do their own SE part and the SE Toolset handles the overall SE. Compared to commercial available requirement management software, which support classical SE, the SE Toolset performs similar. At the other end of the SE spectrum is Model Based Systems Engineering (MBSE) which is supported by programs such as Capella. The learning curve of Capella is very steep and MBSE knowledge is required. The workflow is more or less predetermined with little possibilities for a customized approach. For an MBSE approach, dedicated systems engineers are a must as the model can be incomprehensible. With respect to Capella, the SE Toolset has less functionality. However, it provides an easy workflow which can be customized if necessary. The core value of the SE Toolset is that the input of users and a systems engineer will always be necessary as the human interaction/intelligence cannot be replaced by pre-programmed automation. The SE Toolset provides transparency. Something that in MBSE can be difficult.

The aim of the SE Toolset is to fill the gap between classical systems engineering and MBSE. Both classical SE and MBSE require too much workload to setup and maintain and require a dedicated systems engineer. The

SE Toolset provides the middle ground by being simple, transparent, accessible and minimizing the workload. The current SE Toolset is however not mature enough to support all classical SE, such as N^2 diagrams, risk analysis, requirements compliance, verification, validation and system versioning. By implementing this additional functionality, the program resembles more of a MBSE program. All (classical) SE can then be done in the SE Toolset, which makes is also more suitable for larger projects. For larger projects, the program needs to be optimized to maintain good efficiency and speed to be able to handle many users and requirements. Furthermore, it would be beneficial to transform the SE Toolset into a web-based program. This makes the accessibility more suitable for larger projects. The current SE Toolset is very suitable for small, dynamic and state-of-the-art projects. With further development, the applicability could be expanded without getting as complicated and time consuming as MBSE. MBSE requires a full time systems engineer with extensive knowledge of MBSE and modelling languages such as SysML, UML and Capella. Furthermore, the MBSE model needs to be adjusted and maintained during the project. For the SE Toolset, only knowledge of classical SE is required, which is a fundamental difference with MBSE. This makes the program simpler and more accessible for new team members. If bugs occur or if the SE Toolset needs to be customized, programming knowledge is required.

From this thesis, it can be concluded that the presented SE Approach and SE Toolset provide a good basis for further development of an automated version of classical SE. The downsides of classical SE can be removed with an automated tool, while not making it as complicated as Model Based Systems Engineering (MBSE). The created SE Toolset has been succesfully used in a small project with limited and dynamic resources. Further development is needed to improve the performance and stability and to make it applicable in larger projects. For the top-level SE of the DST project it can be concluded that the DST TIR 30cm, DST TIR 1.5m and DST VIS 1.5m showed good performance compared to reference satellites. The DST VIS 30cm concept appeared to have no benefits over fixed telescopes. From the top-down concept trade-off, no clear winner was found. However, with the input of C. Hobijn and F. Hu, a team decision could be made to focus on a demonstrator design with M2 folding arms and a linear single or multiple segment baffle. Overall, it can be concluded that the DST project is on a good road to develop state-of-the-art deployable telescopes with good performance compared to fixed telescopes.

8 Recommendations

In this chapter, recommendations for further research of the satellite comparison and further development of the SE Toolset are described.

8.1. Reference satellite comparison

For the reference satellite comparison it is recommended to raise the maturity of the DST TIR 1.5m and DST VIS 30cm to a similar level as the DST TIR 30cm and DST VIS 1.5m. Their position in the graphs is then known with a higher certainty, which leads to better insight into their potential. It is also recommended to preliminary size a DST VIS and TIR concept with a larger aperture than 1.5m, to investigate the potential benefit for larger aperture telescopes. It is also recommended to perform a market analysis to investigate the potential market gap for the DST project to fill on the long term. Currently the DST project does not have a clear road map. The aperture size for the demonstrator is a team decision based on changing stakeholder needs. An expansion of the reference satellite comparison with more and higher maturity DST TIR and VIS designs, combined with a market analysis, can improve the road map and make the short and long term goals of the DST project more clear and specific.

8.2. SE Toolset

The most important recommendation for further development of the SE Toolset is to transfer the program to for instance Python or Visual Basic. These programs are more customizable than Microsoft Access. When transferred, the SE Toolset can be further optimized to maintain good speed and efficiency. The current SE Toolset can be used as a basis, as the program proved to have good functionality. Once the program is converted to a different programming language, it can be expanded. Functionality that could be added is discussed in Chapter 6. The functionality which is recommended to add is summarized below.

Notifications through Power Automate

Power Automate, which is available in Sharepoint, can be used to automatically post messages in Microsoft Teams or send emails to team members in case updates or changes occur in a Sharepoint list. This functionality has been briefly tested during this thesis, but was not implemented due to time limitations. Adding this functionality can improve the communication and involvement in the SE Toolset, without the need to open the SE Toolset very often.

(sub)-system version control and iterations

The implementation of (sub)-system version control and iterations can be very beneficial to fast changing projects and research projects. Parts of the design could branch of without influencing the main design. Functionality could even be implemented to merge the main design and branch later back together, similar to a work flow often seen in git based projects.

Expand budget functionality

The budget functionality could be expanded with more budgets, such as alignment and volume budget. Extra values could also be added to the budgets, such as the available budget and current best estimate. This makes it easier for the systems engineer to check if the components and (sub)-systems are still within their limits.

Requirement verification and validation
Requirement verification and validation could be added to expand the functionality towards prototyping, testing, integration and during the life of a product.

Risk identification and mitigation
Risk identification and mitigation could be added, which is beneficial for the systems engineer. The risk identification could be coupled to the creation of requirements to give this task to the team members. The systems engineer can then check the risk identification and determine if risk mitigation is needed. The risk mitigation could be done in consultation with the responsible team member. Predefined conditions could be set to alarm the systems engineer that risk mitigation is required. This can automate the process further.

Documentation handling and documentation overview
Documentation handling could be added to automate documentation identifiers, version control and improve traceability. Documentation overviews could also be created to make dependencies clear. This functionality could be further expanded with Natural Language Processing (NLP) to automatically retrieve relevant data and text. This could also help in automatically structuring documentation and linking relevant sections to requirements, budgets, etc.

Interact with 3D models
Interaction with 3D models could be added to improve the integration of the SE Toolset with 3D modelling software. Data from the models, such as mass, can be retrieved automatically and be shown in the mass budget. The systems engineer can then easily check if the models are aligned with the budget and if actions need to be taken. This can potentially save much time as all individual parts and models don't need to be checked manually. As long as the model naming is done according to a certain format it should work. A simple implementation could be a Microsoft Excel macro which retrieves the data from CATIA. Such an automatic checking of 3D models is already done in the AE1111-II Engineering Drawing course of the TU Delft. Although it could prove very useful, care should be taken that the implementation is done carefully and tested extensively. In case bugs or errors occur the impact could be significant. A solution could be to verify the SE Toolset values and 3D models manually once in a while.

Web-based application
Converting the SE Toolset to a web-based application will improve the accessibility of the tool. This will also make it more suitable for larger project. Therefore, it is recommended to do this while converting the program to another programming language.

Concluding recommendations
With the described functionality, the SE Toolset becomes even more powerful and also becomes applicable to larger projects. The current SE Toolset can also be used as a basic version in which additional functionality can be added according to the specific needs of a project. Even though the SE Toolset can be a good tool to improve working efficiency, the communication stays most important. The SE Toolset will not replace team meetings and regular communication, it will only support it. A certain degree of integration within the team is always required.

Bibliography

[1] O. A. Akanbi, O. Oyedolapo, and G. J. Steven. *Lean Principles in Construction.* Elsevier Inc., 2019. ISBN 9780128117491. doi: 10.1016/b978-0-12-811749-1.00010-9. URL `http://dx.doi.org/10.1016/B978-0-12-811749-1.00010-9`.

[2] I. Akkerhuis. *Deployable Space Telescope, redesign of the Secondary Mirror Support Structure.* Delft University of Technology, 2020.

[3] G. Andersen, M. E. Dearborn, M. G. McHarg, and J. Harvey. Membrane photon sieve telescope. *Optical Design and Engineering IV*, 8167(816714), 2011. ISSN 0277786X. doi: 10.1117/12.898186.

[4] J.W. Arink. Thermal-Mechanical Design of a Baffle for the Deployable Space Telescope. Technical report, Delft University of Technology, 2019.

[5] J. F. G. Baena. Study of the Business Model of three Earth Observation (EO) companies already present in the Very Low Earth Orbit market (VLEO). Technical report, Universitat Politècnica de Catalunya, Barcelona, 2020.

[6] S. J. Baillarin, A. Meygret, C. Dechoz, B. Petrucci, S. Lacherade, T. Tremas, C. Isola, P. Martimort, F. Spoto, and Avenue E. Belin. SENTINEL-2 LEVEL 1 PRODUCTS AND IMAGE PROCESSING PERFORMANCES. *Remote Sensing and Spatial Information Sciences*, XXXIX-B1(September):197–202, 2012.

[7] J. Bouwmeester, S. P. van der Linden, A. Povalac, and E. K. A. Gill. Towards an innovative electrical interface standard for PocketQubes and CubeSats. *Advances in Space Research*, 62(12):3423–3437, 2018. ISSN 18791948. doi: 10.1016/j.asr.2018.03.040. URL `https://doi.org/10.1016/j.asr.2018.03.040`.

[8] E. Caliò, F. Di Giorgio, and M. Pasquinelli. Deploying model-based systems engineering in Thales Alenia space Italia. *CEUR Workshop Proceedings*, 1728:112–118, 2016. ISSN 16130073.

[9] M. E. Dearborn, G. P. Andersen, O. Asmolova, R. L. Balthazor, M. G. Mcharg, H. C. Nelson, T. S. Quiller, G. R. Wilson, T. J. Harvey, and T. W. Murphey. A Deployable Membrane Telescope Payload for CubeSats. *Journal of Small Satellites*, 3(1):253–264, 2014. URL `www.JoSSonline.com`.

[10] D. Dolkens. A Deployable Telescope for Sub-Meter Resolutions from MicroSatellite Platforms. Technical report, Delft Univeristy of Technology, 2015.

[11] J. W. Figoski, N. Zaun, and T. Mooney. Performance results for the Landsat OLI spectral filters. *Earth Observing Systems XIV*, 7452(August 2009):74520T, 2009. ISSN 0277786X. doi: 10.1117/12.823601.

[12] K. D. Goepel. Implementing the Analytic Hierarchy Process as a Standard Method for Multi-Criteria Decision Making in Corporate Enterprises – a New AHP Excel Template with Multiple Inputs. *Proceedings of the International Symposium on the Analytic Hierarchy Process*, page 11, 2013. doi: 10.13033/isahp.y2013.047.

[13] F. Gueguen, A. Bettes, Y. Toulemont, J. Costeraste, and A. Rosak. SPOT series camera improvement for the HRG : very high resolution instrument of SPOT 5. *Design and Engineering of Optical Systems II*, pages 301–312, 2021. doi: 10.1117/12.360020.

[14] R. Guzmán, E. Davis, S.and Ocerin, A. Conde, L. C. Fernández, and M. Massimiani. SSC20-WKIII-10 In-Orbit Demonstration of the iSIM-170 Optical Payload Onboard the ISS. *34th Annual Small Satellite Conference*, page 18, 2020.

[15] R. Guzmán, R. López, E. Ocerin, S. Davis, and J. Tomás. A compact multispectral imager for the MANTIS mission 12U CubeSat A Compact Multispectral Imager for the MANTIS Mission 12U CubeSat. *CubeSats and SmallSats for Remote Sensing IV*, 11505(August):18, 2020. doi: 10.1117/12.2568080.

[16] Z. Hao, Z. Bing, C. Zhengchao, and H. Zhihua. Vicarious Radiometric Calibration of the Hyperspectral Imaging Microsatellites Spark-01 and -02 over Dunhuang, China. *Remote Sensing*, 10(120):36, 2018. doi: 10.3390/rs10010120.

[17] E. Hull, K. Jackson, and J. Dick. *Requirements Engineering*. Springer Science+Business Media, Berlin, 2 edition, 2005. ISBN 1852338792.

[18] V Nagy. Design of a Deployable Baffle for the Deployable Space Telescope. Technical report, Delft University of Technology, 2021.

[19] D. C. Reuter, C. M. Richardson, F. A. Pellerano, J. R. Irons, R. G. Allen, M. Anderson, M. D. Jhabvala, A. W. Lunsford, M. Montanaro, R. L. Smith, Z. Tesfaye, and K. J. Thome. The Thermal Infrared Sensor (TIRS) on Landsat 8: Design Overview and Pre-Launch Characterization. *Remote Sensing*, 7(January):1135–1153, 2015. doi: 10.3390/rs70101135.

[20] N. J Salkind. *Encyclopedia of research design*, volume 1. SAGE Publications, 2010. doi: 10.4135/978141 2961288.

[21] C. R. Sandin, L. N. Allen, E. G. Amatucci, J. W. Arenberg, R. C. Carter, J. A. Corsetti, M. J. DiPirro, M. East, L. G. Fantano, A. Flores, J. S. Knight, D. T. Leisawitz, P. A. Lightsey, G. E. Martins, T. Mooney, A. Nordt, and J. B. Steeves. Materials evaluation for the Origins Space Telescope. *Journal of Astronomical Telescopes, Instruments, and Systems*, 7(01):1–10, 2021. ISSN 2329-4124. doi: 10.1117/1.jatis.7.1.011011.

[22] M. Scardera, M. Baker, R. Reynolds, S. Reddy, K. Kellogg, M. Mahoney, P. Silversmith, N. Rodriguez, and P. Dohm. ALTAIR Millenniums DARPA SeeMe Satellite Solution Technical (R)evolution. *28th Annual AIAA/USU Conference on Small Satellites*, page 13, 2014.

[23] R. J. De Vries. Systems Engineering in Deployable Space Telescope. Technical report, Delft University of Technology, Delft, 2020.

[24] Y. M. Wang and T. M. S. Elhag. An approach to avoiding rank reversal in AHP. *Decision Support Systems*, 42(3):1474–1480, 2006. ISSN 01679236. doi: 10.1016/j.dss.2005.12.002.

[25] S. Yue, H. Wenjiang, Y. Huichun, Chao R., X. Naichen, G. Yun, D. Yingying, and P. Dailiang. Partial Least Square Discriminant Analysis Based on Normalized Two-Stage Vegetation Indices for Mapping Damage from Rice Diseases Using PlanetScope Datasets. *Sensors*, 18(1901):1–16, 2018. doi: 10.3390/s18061901.

A

Information From Team Members

An email containing questions about the project was send at the beginning of the thesis to the team members in order to gain more insight in the project and potential problems which the team members encountered. The questions asked in the email are:

- What are you working on?
- How far/in which phase of the design/thesis are you?
- In which way is systems engineering used by you or your predecessor?
- If yes, was this adequate/suitable enough?
- Is the systems engineering for your part kept up to date/maintained? (changing requirements etc.)
- What does the change from visible to thermal infrared and trace gasses mean for your part?
- What is the impact? (big changes/complete redesign or small adaptations for example)

Response Ilja Akkerhuis

Hoi Roelof,

Goed dat je dit oppakt, dit is inderdaad iets waar ikzelf ook last van heb gehad.

-Ik werk aan de stabilisatie van de secondary mirror tov de primary mirror.
-Ik start nu met de detailed design fase.
-Ik heb natuurlijk de requirements voor mijn eigen design moeten opstellen, maar dit was erg lastig doordat de top level requirements niet meer klopten/ outdated waren. Bijv. mass requirements die nergens op gebaseerd lijken te -zijn (zelfs initiële documentatie verteld niet waarom het de waarde heeft die het heeft). Ik heb deze dan ook naast me neergelegd en een educated guess gedaan naar wat ongeveer voor mijn systeem zou moeten kloppen.
-Dit was niet adequaat.
-Ja dit staat in mijn verslag
-Ik heb hiermee rekening gehouden in mijn ontwerp (bijv. geen IR sources introduceren).
-Niet heel drastisch

Ik hoop dat ik je hiermee heb geholpen, zo niet laat het dan vooral weten! Ik heb net mijn LS geüpload naar de drive dus mocht je die willen inzien ga je gang. Ik heb hier (naar mijn mening) een goed overzicht gemaakt van het project dus wel de moeite waard om intro door te lezen voor jezelf.

Succes met je thesis!

Ilja

Response Dennis Dolkens

Hoi Roelof Jan,

Hier is een antwoord op je vragen. Bijgevoegd zijn wat plaatjes van de huidige optische TIR concepten. Het document is nog een early draft.

Groeten,

Dennis

- Waar werk je aan?

Optical system concepts voor de TIR imager

- Hoe ver/in welke fase van het design/thesis ben je?

Ben bezig met het afronden van mijn thesis. Wat betreft het design van de TIR oplossing, ben ik nu een maand of drie bezig. Er zijn twee concepten af, die ik in meer detail aan het uitwerken ben.

- In welke vorm is er door jou of je voorgangers systems engineering gebruikt?

Requirement definition, trade-offs, verification+validation, etc

- Zo ja, was dit adequaat/passend genoeg?

Ja, voor die eerste fase wel. Toen het project groter is geworden en meer mensen eraan werkten, was er meer interface management nodig, hier is een hoop ruimte voor verbetering. Sean Pepper heeft tijdens zijn thesis aardig wat werk verricht om de SE in het project naar een hoger niveau te brengen.

- Is het systems engineering voor jou gedeelte ge update/bijgehouden? (veranderende requirements etc.)

Ja, maar de optische requirements (SNR, Wavefront, spatial sampling, etc.) zijn dan ook niet veranderd. Ten minste, tot de verandering naar TIR.

- Wat betekent de verandering van visible naar thermal infrared en trace gasses voor jou gedeelte?

- Wat is de impact? (grote veranderingen/drastisch ander ontwerp of kleine aanpassingen bijvoorbeeld)

De impact van de verandering naar TIR is erg groot.

F/nummer:
De TIR telescoop moet ontworpen worden voor een erg klein f-nummer, om zelfs met grote pixels (30 micron) niet gigantisch te over-samplen (f_sampling » f_cutoff). Waar de DST is ontworpen voor een f/nummer van f/8, is er voor de TIR oplossing een f/nummer van f/1.4-1.8 nodig. Simpelweg schalen van het DST ontwerp gaat dus niet werken.

Koeling:
Om een SNR te halen die goed genoeg is moet de detector, en (een deel van) de optiek gekoeld worden, door middel van radiatoren en/of een actieve koeler. Aangezien het koelen van de volledige telescoop waarschijnlijk niet gaat werken, onder meer door het grote volume ervan en door de slechte thermische koppeling door de deployment mechanisms, is het waarschijnlijk beter om slechts een deel van het systeem te koelen. In de concepten maak ik daarom gebruik van een intermediate image en een gekoelde re-imager.

Trace gasses:
Wat betreft trace gasses: dat is nieuw voor mij. Om dat te doen is er (voor zover ik weet) spectroscopie nodig, en niet alleen imaging. We hebben in het huidige design geen ruimte om een spectrometer toe te voegen, dus ik zie niet hoe we deze mission case kunnen bedienen.

Response Thijs Gritter

Dear Roelof Jan,

I am working on the top CORE hinges, of which I will perform both a thermal and a mechanical analysis, with a focus on micro-dynamics. Currently, I am working on the thermal analysis for which the model is nearly completed. I expect to graduate in September or October.

The transition to thermal infrared does not influence my thesis, but it has consequences for the part itself. It is very well possible that in the new design the CORE hinge is not needed. In his thesis, André Krikken has performed several trade-offs for the design of the secondary mirror support structure, but the eventual choice and the subsequent design is specifically related to the visible light spectrum and it is expected that the change to the thermal infrared will have a big influence on the design.

In the first few years of the project systems engineering has been applied. For instance, requirements have been defined for the DST as well as mass and power budgets. At that moment it was adequate, but at this moment it is not adequate and not up to date. As far as I know, it has last been updated in 2018.

For the CORE hinge specifically, not much systems engineering has been performed, but it has been a part of the DST systems engineering efforts. Furthermore, a mass budget of the hinge has been made, which has been lastly altered a year ago by Matys Voorn.

My literature study report is on the drive. In the second chapter, I have written a short summary about the work that has been performed on the DST in the past. I think that it is worthwhile to read this chapter. Additionally, several systems engineering related files are present on the drive, and it cannot hurt to read through the master's theses of other students. Concerning the CORE hinges, Matys Voorn's thesis is most interesting.

Due to my lack of knowledge on optical systems design for TIR, I cannot give you tips and tricks for systems engineering related to the TIR design. For the DST project in general, I think it would be good to have monthly meetings with everyone involved, in order to get a better overview of the progress of different parts of the project. This has been made difficult by the Corona virus, but it would be good to start with these meetings, if possible.

Kind regards,

Thijs Gritter

Response Fabbio Hu

Hey Roelof,

Hieronder mijn antwoord op je vragen, mocht het niet duidelijk zijn, vraag het me.

1. Ik werk aan de optimalisering van de shape van de baffle. De shape van de baffle kan verschillende vormen krijgen, inclusief uitsparingen (sugar scoops) en het plaatsen van baffle vanes. De shape en coating van de baffle zullen hoogstwaarschijnlijk de stray light performance gunstig beïnvloeden, welke ook het thermisch gedrag zal verbeteren. Hierdoor zal ik eerst kijken naar stray light requirements (welke nog niet bestaan) en invloed van satelliet appendages zoals de solar arrays omdat ze ook warmte terugstralen naar de baffle.

2. Ik ben net begonnen aan de thesis. Dit betekent dat ik het begin gesprek met Hans heb gehad. Literatuur studie is af!

3. Waarschijnlijk begrijp ik je vraag hierop niet volledig. Heb je een paar voordelen van typische 'Systems Engineering'-processen of benamingen? Mijn voorganger (J.W. Arink en daarvoor weer E. Korhonen) hebben eerst gekeken naar de requirements van S. Pepper, vandaaruit is J.W. Arink gaan berekenen wat het maximale verschil in temperatuur is tussen de booms (onderling) en met de rods. Door te kijken naar verschillende lengte, diameter, materiaal en coating is gekeken door J.W. Arink of het temperatuur verschil omlaag gebracht kon worden naar het beoogde temperatuurverschil

4. Het beoogde doel was niet bereikt, vandaar de verdere optimalisatie. J.W. Arink gaf de aanbeveling in zijn thesis om meer met stray light parameters te spelen voor verdere optimalisatie

5. Voor zover ik weet niet. Qua stray light requirements zijn er geen requirements opgesteld, hier zal ik een begin aan maken. Aan de hand van de opgestelde requirements, is er een indicatie ook mogelijk om de stray light performance van de nieuwe baffle te quantificeren. Echter zijn (voor mij) de thermische veranderingen belangrijker. Viktor Nagy zal kijken naar deployment mogelijkheden en vandaaruit zullen er hoogstwaarschijnlijk limitaties op de complexiteit van de baffle-shape liggen.

6. Ik heb nog niet gekeken naar de TIR (Thermal Infra-red) implicaties, aangezien er toendertijd nog geen vraag naar was, idem voor trace gas. Maar wat ik begrijp vanuit Hans is dat voor de VIS de warmte-belasting

(de thermal gradients) moeten in bedwang worden gehouden zodat er geen aberraties komen. Voor TIR is dit (verschil in thermal-gradients wat leidt tot abberations) ook van toepassing, maar bij TIR is daarnaast ook de temperatuur binnenin de baffle een bron van ruis (noise) wat gereduceerd moet worden. Hierdoor zal er hoogstwaarschijnlijk voor TIR meer active thermal control moeten plaatsvinden.

7. Voor VIS en TIR: specular baffle, met een andere shape en hoogstwaarschijnlijk een sugar-scoop. Mogelijk om een deployable heat-shield toe te passen voor TIR met ATC (Active Thermal Control)

Ik heb je mijn literatuur study bijgevoegd in de email. Ik zag dat er nog een paar (spel)-fouten en referentie fouten in zaten. Ik ben tevens ook benieuwd naar jou antwoord op je eigen vragen! Als het mag

Met vriendelijke groet,

Fabbio Hu

Response Viktor Nagy

Dear Roelof,

My answers are below.

- What are you working on?

I am working on the thermomechanical design of the DST baffle. There have been two iterations before, but none of them was good enough (requirement compliance), but more importantly, none of them included any kind of prototyping or detailed actuation and integration.

- How far/in which phase of the design/thesis are you?

I am about to finish the trade-off for the deployment concepts, after which comes the detailed design.

- In which way is systems engineering used by you or your predecessor?

Well, maybe a more detailed question could be better, but I try to do my best here. So my predecessors and I both worked like this: first, creating a list of requirements, then design trade-off, then detailed design and analysis, then verification and requirement compliance check.

- If yes, was this adequate/suitable enough?

No, we are missing an important role of a central systems engineer. The different parts of the DST are now being designed separately, and there's minimal engineering communication between those parts. For example, for the SMSS my baffle design would be required to have a thermal analysis, and from the SMSS I would require requirements about what are the allowed temperatures. But we cannot give these to each other, but only at the end of our theses. Also, we don't have up-to-date global budgets, and all the global systems engineering part is just missing.

- Is the systems engineering for your part kept up to date/maintained? (changing requirements etc.)

Well, I have a list of requirements that many DST members checked. Sometimes I change a tiny bit, but they are kept up to date (locally).

- What does the change from visible to thermal infrared and trace gasses mean for your part?

The mechanical design (deployment concept, actuators) probably would stay, however, all the thermal part (required temperatures, coatings, number of MLI layers, etc.) would change. Also the baffle would need more mass and voulme from the budget. However, as I have minimal (close to nothing) information about the TIR and its requirements, I am designing for visible light, and maybe consider later what would change. It is difficult to work with no data.

- What is the impact? (big changes/complete redesign or small adaptations for example)

For the thermal part, as mentioned, complete redesign, for the mechanical part probably just some changes in the diameter and length to be able to support the updated thermal part (shroud).

If you have any more questions, let me know. I attach my literature study.

Cheers, Viktor

Response Victor Villalba Villalba

Hi,

If you want we could set up a skype chat to talk more easily about what your topic is and what I have found out so far regarding systems engineering. Today I don't think I will have the time, but I can call you tomorrow or some other day.

Now, about your questions, yes there was a top-down SE approach at the beginning of the project, but back then I couldn't properly manage the SE aspects with the mechanical and thermal engineering aspects of the project, and with the need to publish stuff, so eventually we drifted from the integrated use of SE to a focus on developing certain key technologies instead. The main reason basically boils down to me not being educated enough on SE, and needing to find out other fundamentals first. If you want a quick summary of what I think needs to be understood prior to having a proper SE approach to deployable telescopes you can check out my paper on the Journal of Astronomical Telescopes, Instruments and Systems (search my name and that, it will pop up). That paper is basically a list of "stuff I wish I had known before trying to do SE for this project". I think the SE approach at the beginning of the project was lacking in focus because we did not really understand what the technology challenges were.

The "switch" to TIR is in that sense not so much of a switch as a new case study, but it will definitely require a better SE approach than the regular DST currently has because there are work packages for external parties and there will be need for management. As far as I am concerned at this stage of my project, I use the budgets that were laid out top-down with minimal modifications and I think others are just applying principles of SE to their project, but not to the overall system. I think Viktor Nagy is doing more of an effort in updating requirements and methods of SE for this project. From y point of view, it is more or less up to date because nobody has touched upon some topics in a long time. That also has to do with a discontinuity in the amount of MSc students that were available at different times.

Now, what I am working on is the development of a piezoelectric actuation mechanism for the primary mirror. This is again technology development mainly, so I am using the framework laid out by Sean Pepper in his thesis. I have been doing thermal, structural and mechanical modeling for that mechanism and for the experiment I am proposing to the NLR to assess piezoelectric actuators for space optics. They are interested in the TIR concept specifically, I do not really care about its context. The switch from visible to TIR wavelengths involves an loosening of the mechanical budgets and likely the creation of all new thermal requirements, but so far I have not really seen any top-down budgets from optical modelling. From what I can see, I expect major changes in the whole architecture of the design and perhaps of the mechanisms, but I do not know enough to tell and therefore I keep working as if the TIR concept didn't exist. I expect more range of motion of the mechanism to be needed, and less resolution.

Well, I know all of this doesn't answer some of your questions, but I am not sure how you want them answered. But in any case you know how to find me. Good luck.

Regards,

Victor

Summary Zoom call 15-6-2020 Victor Villalba

In the DST project both a bottom up and a top down approach are needed. Dennis Dolkens wrote the absolute needs for the top down systems engineering, table with available ranges for alignment and such for the primary and secondary mirrors.

Victor Nagy critically examined the requirements and systems engineering for the baffle. Might consider asking him about systems engineering for his part.

The DSE is mostly top-down, and that means the SE methods typically are easy to adapt. At the end of the DSE students do perform some bottom up work, but you will also notice significantly less SE on that stage. The question here I think is what tools are easily adaptable to bottom up methodology.

It is not really clear were all the problems come from, first you need to understand what the telescope does and how this works. Jan Willem his thesis is interesting, he worked on the baffle design. Maybe also interesting are some pages from Sean Pepper, especially the SE approach for the mechanism he designed. Matthew

Corver and Krikken are also interesting. First thing is 2014/2015 paper from Dennis about the optical design.

Visible telescopes are harder to design, they have a very tight budget. TIR Telescopes have a looser budget. The TIR Telescope is less likely to be affected mechanically by the thermal environment, though it might be optically. Currently there are no deployable telescopes in visible and TIR. See the paper of Víctor Villalba.

Probably it is a bad idea to use the current visible design as baseline. No clear idea on what the TIR requirements will be.

Deployment is a lot easier for TIR, Thermal control becomes the problem. Dennis knows all about it. First top down approach is needed.

For TIR you want the temperature not only stable but also cold, unless there is some clever way to remove the self-emission of the telescope optically.

The mass and volume approximation were done by Dennis. 75kg seemed like a good number, if not feasible it would become 100kg. With a margin we are right now at 120kg.

The technology is variying widely, compososite mirrors, 3D printed mirrors, for small sizes. At 3ME they are currently working on this.

B

Appendix: SE Toolset Setup

In this appendix the SE Toolset is shown and discussed in detail. From the SE analysis in Section 3.1 it is clear that a dedicated systems engineer would be a solution to solve part of the problems in the DST project. However, the resources in the DST project do not allow for a full time systems engineer. Therefore, a requirement on the SE Toolset is to minimalize the need for a dedicated systems engineer. This can be done by automating the SE as much as possible. The team members should be able to perform their SE in a simple centralized environment, which automatically manages version control, has a good overview and has good traceability. They only need to keep their requirements up to date and insert the correct values, links to other requirements/budgets and provide rationale for the requirement. In this way they do not have to spend time on the correct numbering and version control. Instead, they can focus on the rationale behind the requirement. As long as the team members keep their requirements up to date, the SE Toolset will handle the rest. There is still the need for a systems engineer to check the requirements for proper formulation and errors. However, the time he needs to spend on this is limited as a dedicated SE environment is present in the SE Toolset in which he can track new requirements. Next to that, a comment function is available in which the systems engineer and team members can post comments on requirements. In this way, discussions about requirements can be held within the SE Toolset, which improves the traceability of changes and updates to requirements and improves the overall communication.

B.1. Login Screen

The first screen the team members will see when opening the SE Toolset, is the login screen as shown in Figure B.1 on the left. In the login screen they have to give in their TUDelft NetID and a password. The program checks whether the provided NetID is in the list with authorized people to access the SE Toolset. This data is retrieved from a synced Sharepoint list, called UserInfo, which is connected to a Team from Microsoft Teams. As long as a team member is in the Team, he or she has access. As soon as the team member is removed from the Team the program will deny him or her access. On its own this is already quite a good safety measure. However, to improve the security of the SE Toolset another layer of protection is added via a password.

The first time a team member logs in to the program, he is asked to provide a password and some personal details, as shown in Figure B.1 on the right. Once these details are provided, the password is encrypted via AES 256 encryption and stored in a Sharepoint list. The password also prevents the misuse of other team members their account. The user abbreviation is automatically generated and stored in the User_Data table. It consists of the first letter of the first name and first two letters of the last name. In case the abbreviation is already present in the User_Data table, the first letter of the first name combined with the first and third letter of the last name is used. In case this abbreviation is also already taken than a message will show up to contact the supervisor to manually set-up the account.

If the NetID is in the list of authorized team members and if the password is correct, the program opens the login updates form. The program furthermore stores the abbreviation of the user and the last time he logged in. The user abbreviation is later on used to automatically save who made an adjustment or comment.

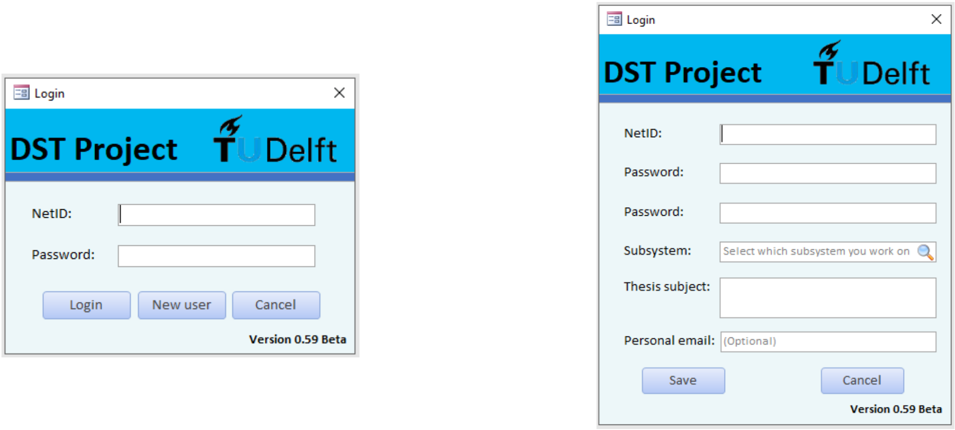


Figure B.1: Left: Screenshot of the login screen. Right: Screenshot of the new user screen.

B.2. Login Updates

The login updates form shows the comments and requirement updates that are made after the last time the user has logged in, as shown in Figure B.2. By double clicking on the identifier, the requirement is opened in detail view. By clicking on "Continue to Main Menu" the form is closed an the main menu is opened. The login updates form can always be viewed from the main menu by clicking the "Open Update List" button.

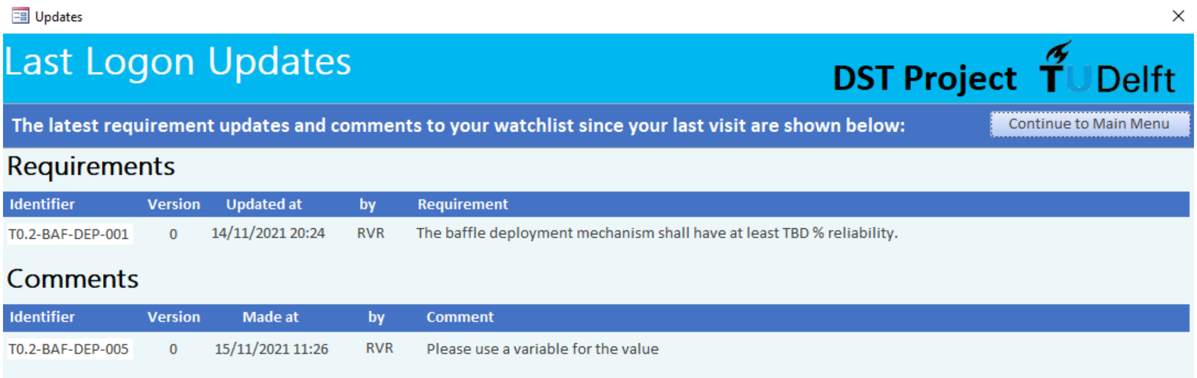


Figure B.2: Screenshot of the login updates form.

B.3. Main Menu

The main menu is shown in Figure B.3. The main menu consists of three parts. On the left there is the systems engineering section, which includes some additional options. This section is only visible and accessible to team members who have the right access permissions, such as the systems engineer and supervisors. This section includes the SE checklist which is discussed in Section B.9 and the option to manage the SE Toolset, as discussed in Section B.11. On the right the section can be seen which contains the normal parts each team member can access. This section contains the requirements list, configuration item tree, variables list and mass budget. These parts are discussed in Section B.4, B.7, B.8 and B.10, respectively.

The last section is the top row, which contains the watch list, update list, abbreviations list, release notes and the current user. The watch list contains a list of the requirements which can be watched or unwatched. When a user selects a requirement to be watched than updates or comments to this requirement are shown in the update list. When a user is the owner of a requirement it will automatically be watched and it cannot be unwatched, unless the user transfers the ownership. The watchlist is shown in Figure B.4. The abbreviation list shows all team members who have worked in the SE Toolset with their full name, abbreviation and email address, as shown in Figure B.5. It is furthermore shown if they are still active and when they last logged in. To know if someone is active or inactive, the program checks if the user is both in the User_Data list and in the UserInfo list. If the user is in both lists, he still has access to the SE Toolset and is active. If the user is in the User_Data list, but not in the UserInfo list, then the user has no access anymore and is inactive. New users are automatically added to the list. A screenshot of the abbreviation list is shown in Figure B.5. Lastly, the name of the current user can be seen at the top op the main menu. If the user clicks on his name than the "New User Form" opens and the user can change his password and personal details.

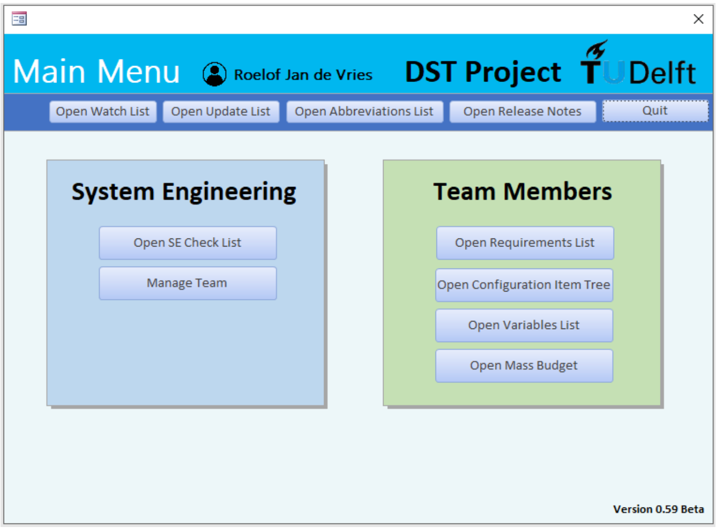


Figure B.3: Screenshot of the main menu.

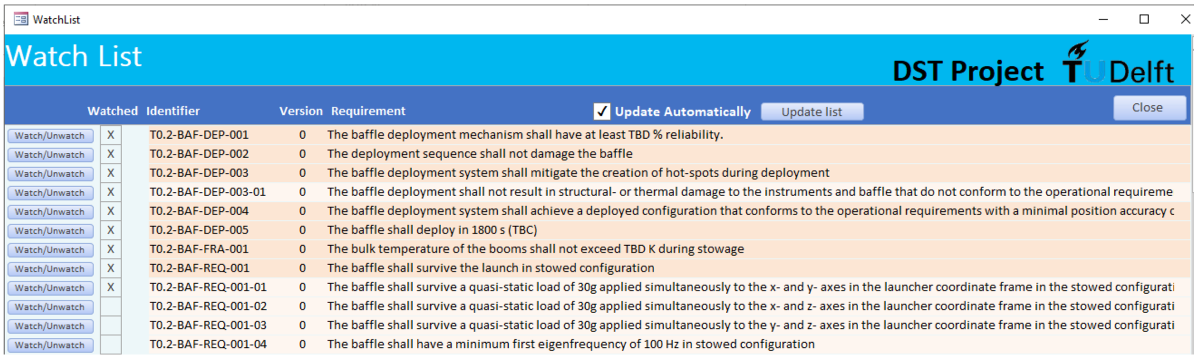


Figure B.4: Screenshot of the watch list.

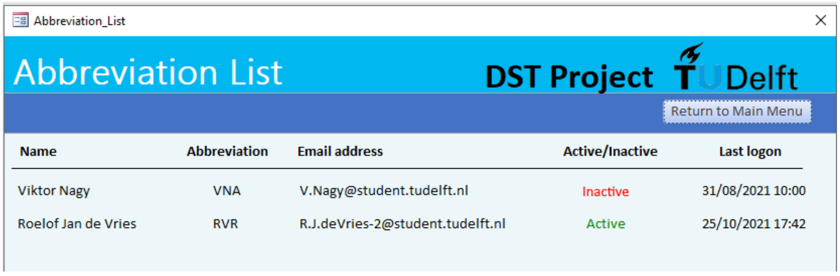


Figure B.5: Screenshot of the abbreviation list.

B.4. Requirements List

The requirements list is shown in Figure B.6. In the requirements list the last version of each requirement is shown. In the top section of the screen there are options to search the list, filter the list per subsystem and to hide columns. There is also a button to add a new requirement which opens the New Requirement form. When you double click on the requirement, the requirement opens in detailed view, which is explained in the next section.

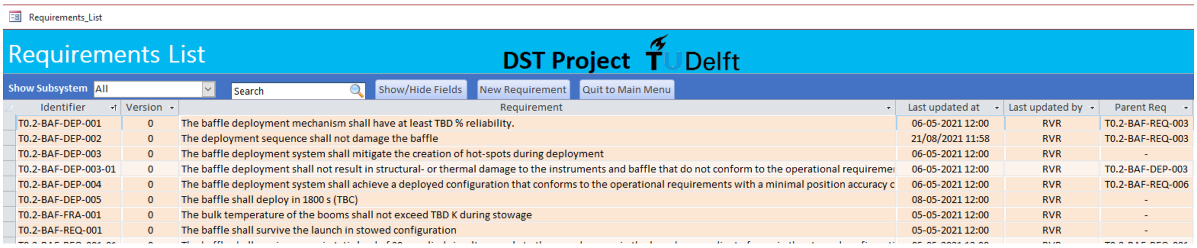


Figure B.6: Screenshot of the requirements list.

B.5. Requirement Details

The requirement details form is shown in Figure B.7 and B.8. At the top, the latest version of the requirement is shown and at the bottom the previous versions are shown. Below the latest version, the parent requirement can be seen if there is one. Below the parent requirement, the daughter requirements can be seen. By double clicking on either the identifier or requirement, the parent or daughter requirement is opened in detail view. In the top section of the form a couple of command buttons can be seen and if the requirement has been checked by a systems engineer. When the requirement is checked, it will show by who it is checked and when. The most left button is used to watch or unwatch the requirement. The next button is to open the comment section for the latest version of the requirement. In this comment section the comments can be seen and new comments can be added. Users their own comments can be edited or deleted. When a comment is edited or deleted it will be shown in small under the comment date. The user can click on this field which will open a dialog screen showing the original comment.

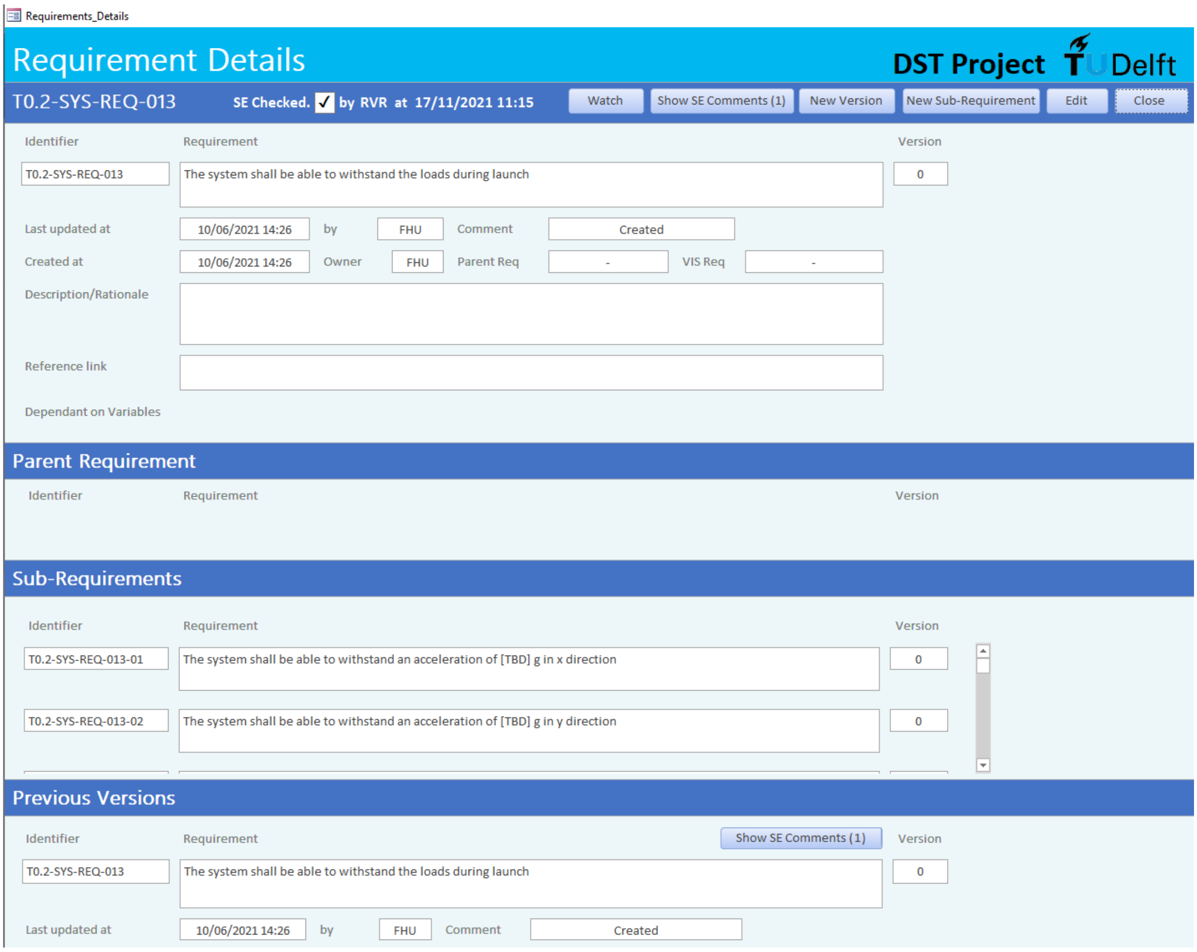


Figure B.7: Screenshot of the requirement details screen.

The third button from the left opens the New Requirement form as a new version form. The data from the requirement, which was viewed in detail view, is copied to the new version form and the version number is increased by one. The requirement comment is also automatically set to updated and the last updated field is automatically set to the current date and time.

The fourth button from the left opens the New Requirement form as a new sub-requirement form. The identifier is automatically set to the detail viewed requirement with correct sub-number. The parent is also automatically set to the detail viewed requirement. Date and time of the creation of the requirement, owner, last updated at, last updated by and comment are automatically set and therefore hidden. After saving the requirement it will show in the requirement detail view.

The fifth button from the left gives the user the option to edit the requirement. When clicking on the edit button three options appear. The first option is to edit the requirement. Two warning messages will appear if this button is clicked as the edit requirement option is only meant for correcting mistakes, like typos. If this option is used for updating requirements than the program will not automatically create a new version. This leads to loss of information and traceability. Therefore, it is only allowed to use this option for editing typos. The second option is to edit the description. It can happen that new information is known for the requirement or additional reference links need to be added. This can be done with the edit description button. In this view only the Description/Rationale and Reference link field can be edited. The last option is to delete the requirement. A warning message will show up to ask the user if he is sure to delete the requirement. If yes is clicked, a form will show up with a warning message to notify the user that the requirement will be set to deleted, but will still show up in the requirements list. The requirement can still be viewed but no changes to the requirement are possible anymore. A deletion can only be undone by a systems engineer. Furthermore, a reason for the deletion needs to be provided. After the reason is saved, the comment field will be set to "Deleted" and the requirement will be greyed out in the requirements list. A comment is also automatically created, which states who deleted the requirement, when, for what reason and which variables were replaced. The program automatically replaces all variables in the requirement with their value to break the link. Otherwise, a new version of the deleted requirement would be created when the linked variables are edited.

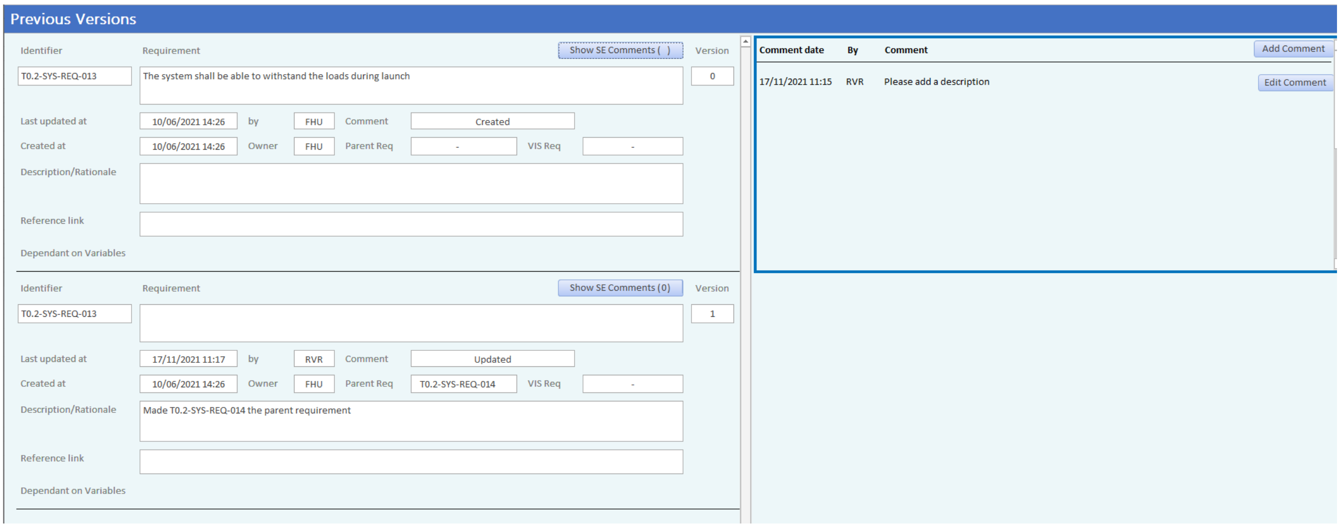


Figure B.8: Screenshot of the requirement details previous versions screen.

B.6. New Requirement

The new requirement form has multiple different functions. While the form remains the same, some controls and fields are hidden depending on where the form is opened from. Below, the different layouts are shown and described when they are used.

New Requirement

When the new requirement button is selected from the requirements list, the form opens in the "New Requirement" layout, as shown in Figure B.9. In this layout part of the fields are hidden, such as last updated, last updated by, created at, owner and comment. These fields are automatically filled in and therefore not adjustable. To select an identifier for the requirement, the user has to click on the "Select Identifier" button. This button opens the configuration item tree in a pop up window. From the configuration item tree, the user can select the sub-system to which the requirement belongs. After selecting the correct sub-system, the window is closed and the identifier field is automatically filled with correct numbering. The "Select Parent" button has similar functionality. Only then a sub-form is shown when a sub-system is clicked, which shows all requirements belonging to this sub-system. The correct requirement should be clicked from this list which closes the window and automatically fills in the parent requirement field.

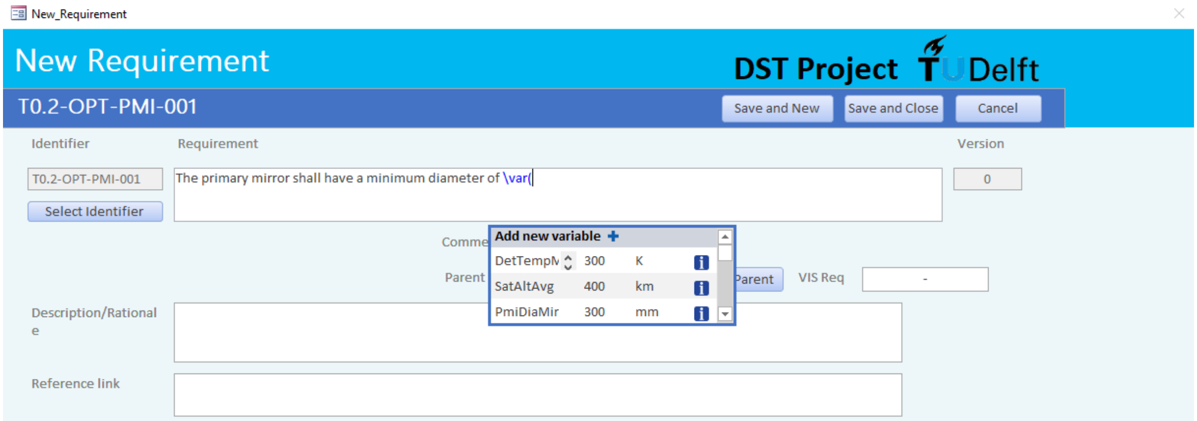


Figure B.9: Screenshot of the new requirement screen with the variable list.

Other functionality are the variables which can be used inside the requirement. When typing "\var(" a sub-form opens which shows all variables, as can be seen in Figure B.9. When the user continues to type the name of a variable, the sub-form is filtered and only shows variables containing the text typed after the opening bracket. In the sub-form the user can also click on the information sign which will open the variable details form. The user can also press the plus sign to add a new variable. This opens a pop up window in which the user can provide the details of the new variable. This form is also discussed in Section B.8. If a variable is selected from the list, then it's name will be automatically placed after the opening bracket. For values in requirements, variables should always be used. This is explained in more detail in Section B.8.

After the requirement and all details are filled in, the user can press the "Save and New" button to create another requirement or click the "Save and Close" button to save the requirement and close the window.

New Version
When a new version of a requirement is made the form is opened in the "New version" layout. This layout shows all information but disables part of it. These fields are automatically filled with the correct data and cannot be changed. The select identifier button is hidden as well as the save and new button. This layout is shown in Figure B.10.

New Sub-Requirement
The "New Sub-Requirement" layout is similar to the "New Requirement" layout. The only difference is that the identifier cannot be selected. The identifier is automatically generated to be a sub-requirement of the requirement which was viewed in detail view.

Edit Description
The "Edit Description" layout shows all fields but disables most of them. Only the description and reference link are editable. This option is present to allow users to add information to the requirement without making a new version. The last updated at and by fields are automatically filled with the current date/time and user. A comment is furthermore created, which states that the description/rationale and reference link have been edited by the user. The comment furthermore contains the previous description/rationale and reference link. Similar functionality is present when the requirement is edited. However, then the previous requirement is also shown. This functionality is present to track changes.

Edit Requirement
The "Edit Requirement" layout is similar to the "Edit Description" layout but also enables the requirement

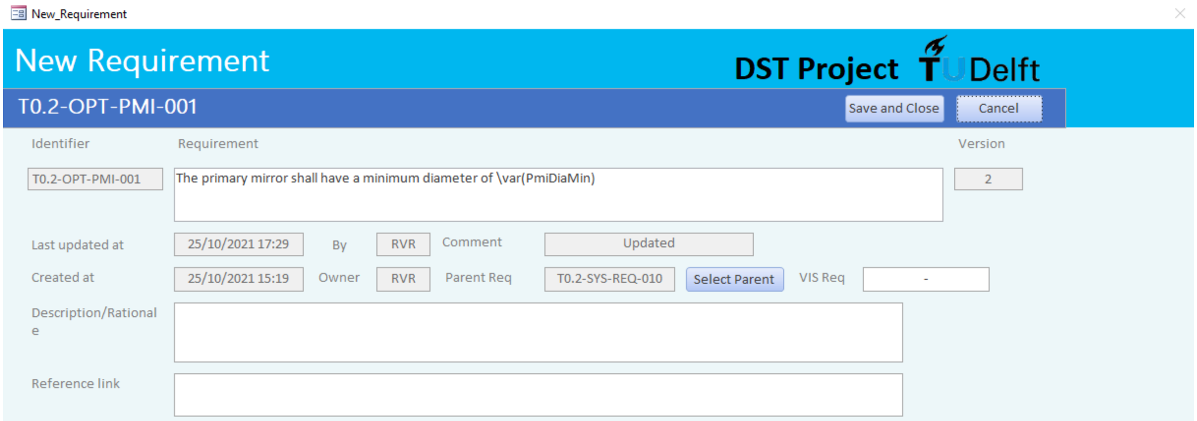


Figure B.10: Screenshot of the new requirement screen when creating a new version.

field. This option is only meant for correcting mistakes in the requirement, such as typos. For changes to the requirement a new version should be created.

B.7. Configuration Item Tree

The configuration item tree (CIT) form is shown in Figure B.11. The form can be opened from different locations and has a slightly different layout, depending on where it was opened from. When the CIT is opened from the main menu, each (sub)-system can be selected to show the requirements which belong to this (sub)-system. When the CIT is opened from the new requirement form, the (sub)-system requirement list will not be shown, but the form will pass the identifier of the selected (sub)-system back to the new requirement form. When opened from the new user form the name of the selected (sub)-system will be passed back to the new user form. The CIT is automatically generated from a linked table called "Conf_Item_Tree_Data". When changes are made to this table the CIT will change along.

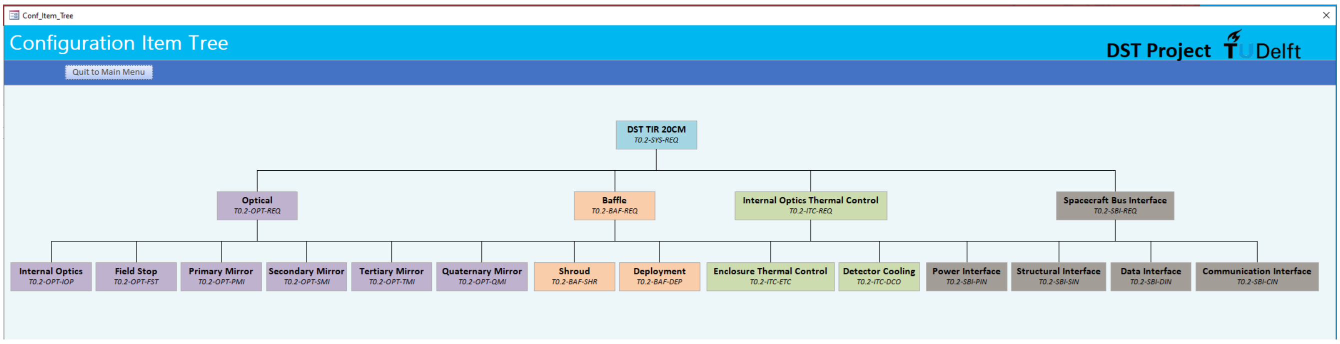


Figure B.11: Screenshot of the configuration item tree screen.

B.8. Variables

The variables list can be opened from the main menu and is shown in Figure B.12. The form contains all the variables with their value, units and a description. By clicking on the information sign the selected variable will open the variable details form. By clicking the "Add variable" button the Add Variable form is opened.

Variable details

The variable details form shows more details about the variable as shown in Figure B.13. Such as version, when it was last updated, by who and which requirements are linked to the variable. It furthermore shows the previous versions at the bottom. At the top there are two buttons to edit the variable. The first button is to only edit the description. The second button can be used to edit the variable. A pop up message will ask the user if he is sure to edit the variable, as this will create a new version of the variable and a new version of all requirements that are linked to the variable. If yes is selected, the value, units and description fields can be edited. The version, updated at and by fields are automatically filled in. The name cannot be changed as

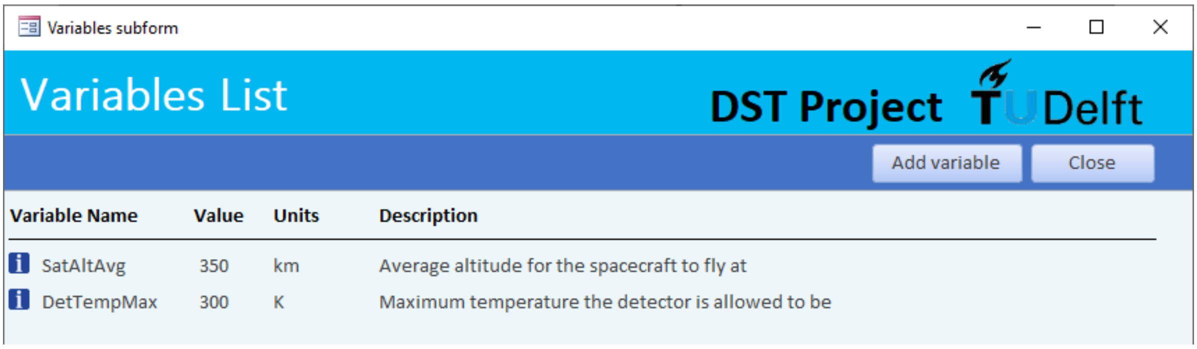


Figure B.12: Screenshot of the variables list.

this would lead to problems with the linked requirements. The field "variable linked to" is not editable and is automatically filled in if the variable is used in a requirement. When the user clicks on a requirement in the field the requirement opens in detail view. After the variable is edited, it needs to be saved with the same button.

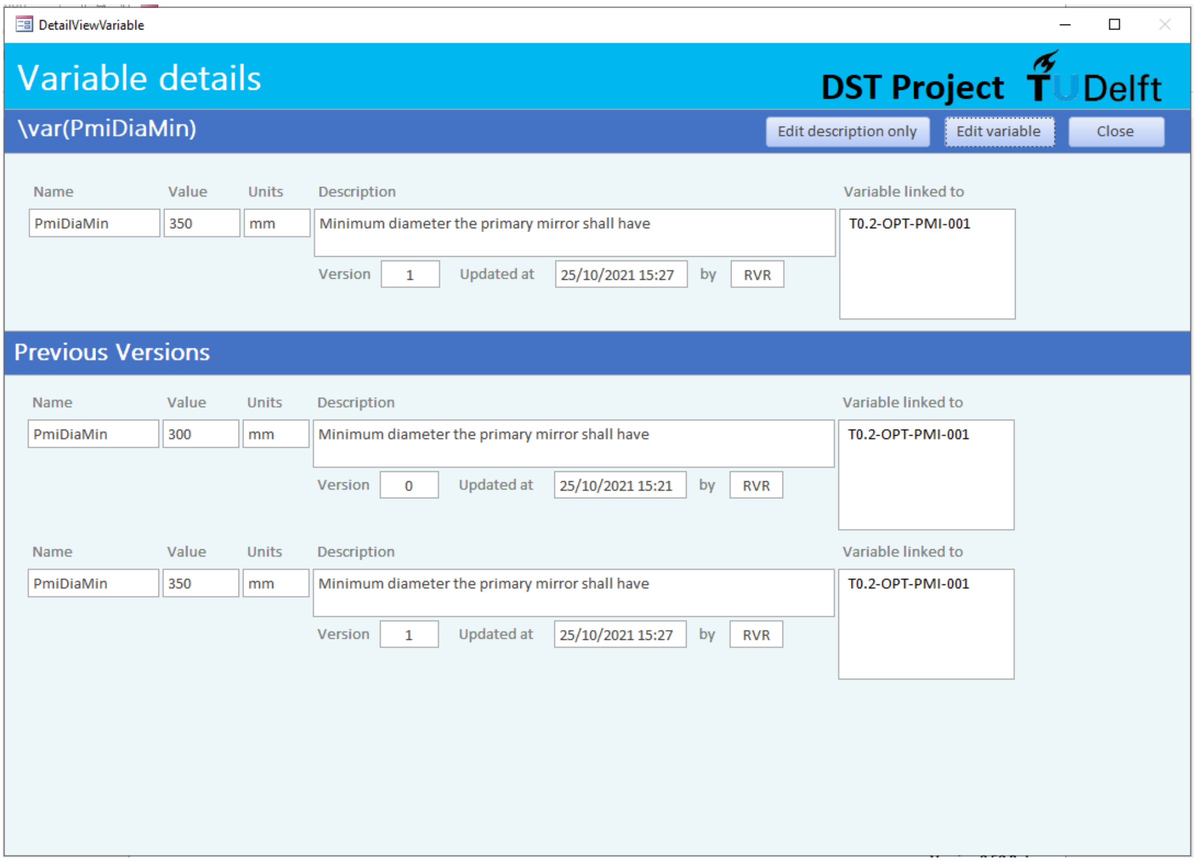


Figure B.13: Screenshot of the variables details screen.

Add variable

The add variable form looks similar to the variable details form but lacks the edit buttons and the previous versions section. A screenshot of the form is shown in Figure B.14. On to bottom left, a comment is displayed which says to maintain the following format for the variable naming:

SatAltAvg - Satellite Altitude Average

DetTempMax - Detector Temperature Maximum

The variable shall consist of three abbreviations put together. The first indicates which (sub)-system or component or subject it is about. The second indicates which aspect and the third indicates a maximum, minimum, average or other kind of description which provides information about the value. The abbreviations should be as much as possible made up out of three letters. However, to still be descriptive the abbreviation can also contain four letters as in above example with temperature. After all fields are filled in, the variable can be saved and the form will close. If not all fields are filled in, a prompt message will appear which states which field is not filled in and the variable cannot be saved.

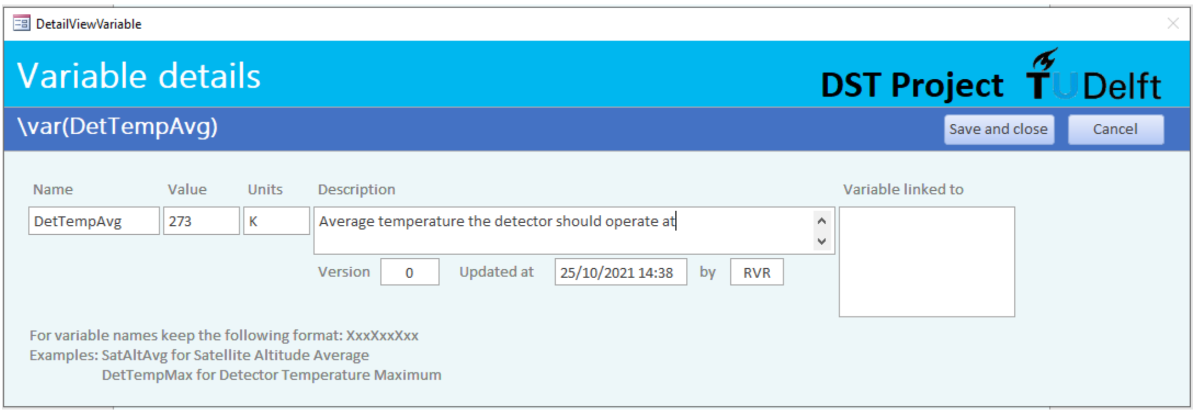


Figure B.14: Screenshot of the add variables screen.

B.9. SE Checklist

The SE Checklist can be seen in Figure B.15. It can be opened from the main menu by users that have systems engineering (SE) permissions. On the left side of the SE Checklist, all requirements can be seen and selected from a list. It is also indicated which requirements have been checked, by who and when. The list can be filtered on already checked requirements, on not checked requirements or on latest version. On the right the details of the selected requirement can be seen. If the requirement is correctly formulated and accepted by the systems engineer, he can click on the SE check mark field. The fields containing the information by who and when the requirement is SE checked are automatically filled in.

In case the requirement is not properly formulated or accepted by the systems engineer, he can place a comment under the requirement. This comment will also be shown in the requirement details form and are therefore also visible to the team members. The program also checks the last time that a user logged in. If there are comments or updates on requirements that are on the users watchlist, and if they are made after the users last login, then the program will show them in the users login updates screen. Besides the comment function in the program the team members can also turn on notifications via Sharepoint. In the Requirements_Data list on Sharepoint they can turn on notifications when requirements are changed. They can change the settings to receive an email once a day or week or get notified right away when a change occurs. It is best to combine the Sharepoint notification with the SE Toolset's comment function, as it is expected that not all DST team members login regularly.

B.10. Mass budget

The mass budget can be seen in Figure B.17. The mass budget is automatically generated from a table containing the component mass. The layout is automatically structured and the (sub)-system masses are calculated from the component masses. A maturity value can be given to the component which determines the contingency factor. The contingency factors can be changed through the manage SE Toolset screen by a supervisor or systems engineer. The mass budget is updated automatically when the contingency factors are changed or when a new component is added. The add component screen can be seen in Figure B.16. The user can fill in the name of the component and select the sub-system it belongs to through the configuration item tree. All components need to be linked to a variable. In this way the requirements are linked to the mass budget through variables and the mass budget is automatically updated with the latest values. If components need to be edited, the edit button can be used.

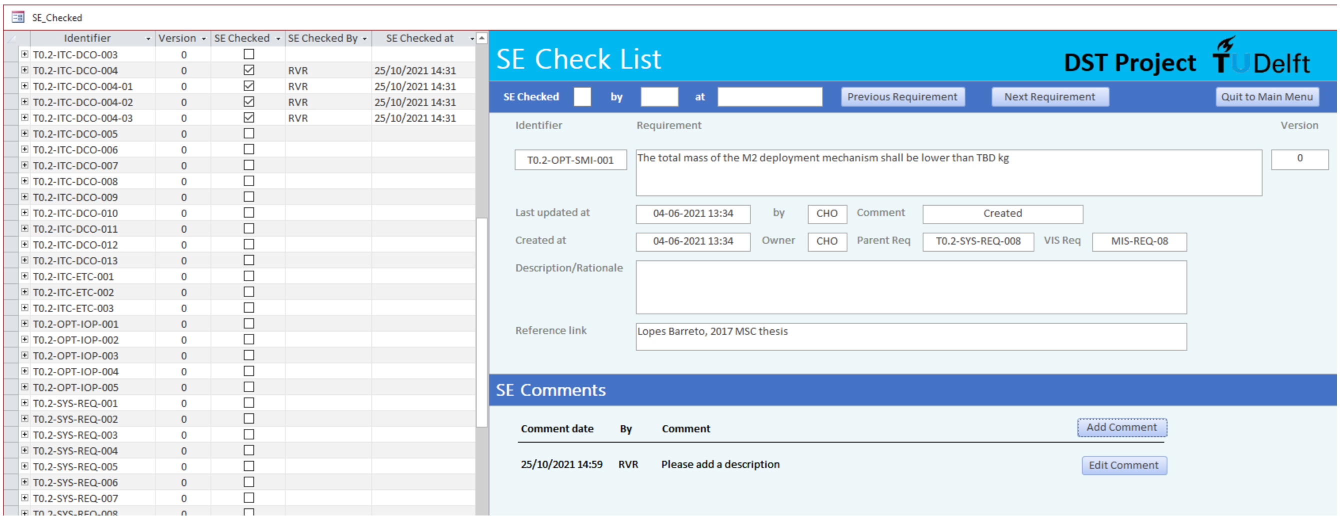


Figure B.15: Screenshot of the SE Checklist.

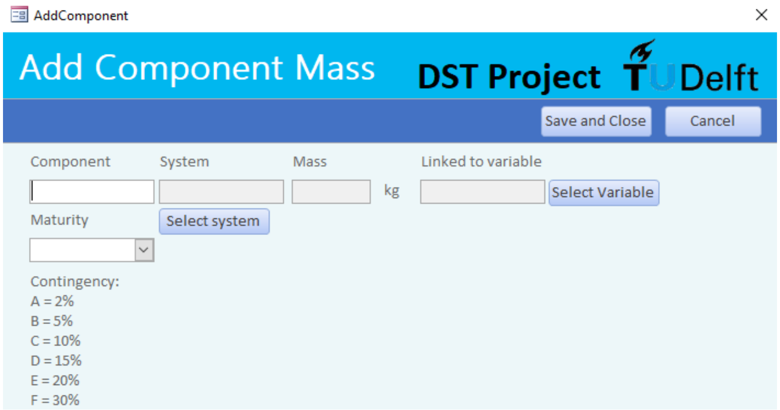


Figure B.16: Screenshot of the SE Checklist.

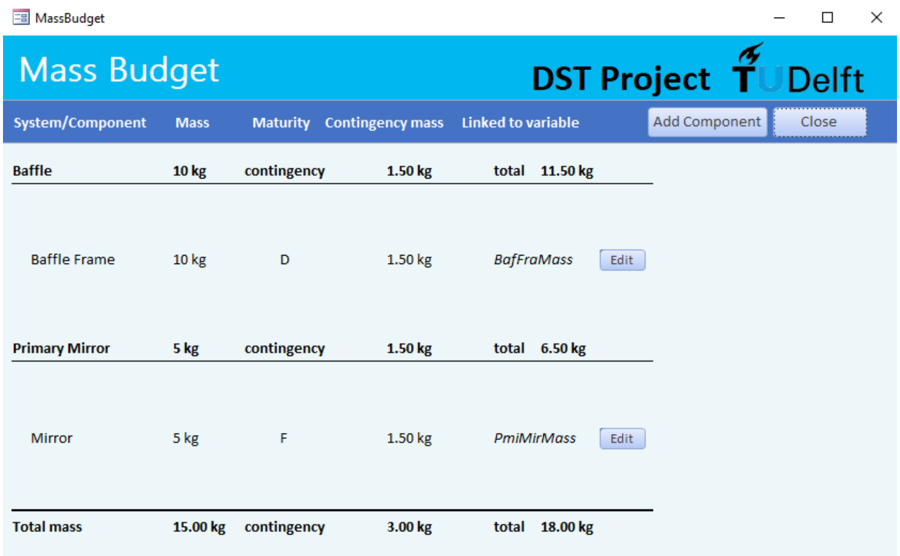


Figure B.17: Screenshot of the mass budget.

B.11. Manage SE Toolset

This screen can be used to manually set team members to inactive, give team members SE permissions and adjust the contingency factors for the mass budget. The supervisors and system engineer are also automatically notified when a user that is inactive, still owns requirements. In the manage SE Toolset they can then transfer the indicated requirements to a new owner. This functionality prevents requirements from having an inactive owner who is not part of the team any more.

C

Reference Satellites and Spacecraft Busses

In this appendix the data on the spacecraft busses is shown in Table C.1. In Table C.2, the reference satellites are shown.

Table C.1: Table containing data on different modular spacecraft (s/c) bus designs.

S/c bus	Sys dim [m]	Sys vol [l]	S/c bus vol [l]	Payload vol [l]	S/c bus mass [kg]	Payload mass [kg]	S/c bus ratio	Reference link
Altair 27U	0.36*0.36*0.37	48	17.1	30.9	20	up to 50	0.36	Ref. [22] https://spaceflight101.com/cygnus-oa7/altair-1/ http://www.millennium-space.com/assets/brochures/altair.pdf
MP42	0.48*0.48*0.47	108	35	73.7	30	40-50	0.32	https://satsearch.co/products/nanoavionics-mp42-microsatellite-bus
LEOS-50HR	0.57*0.57*0.6	195	65	130	45	15-30	0.33	https://www.berlin-space-tech.com/portfolio/leos-50/
LEOS-100HR	0.6*0.6*0.6	216	72	144	70	30-50	0.33	https://www.berlin-space-tech.com/portfolio/leos-100/
Thunder 3U	0.1*0.1*0.34	3.4	1.4	2	3	up to 3	0.41	https://www.utias-sfl.net/?page_id=89
Spartan 6U	0.1*0.2*0.36	7.2	3.2	4	6	up to 6	0.44	https://www.utias-sfl.net/?page_id=89
JAEGER 16U	0.2*0.2*0.45	18	8	10	12	up to 15	0.44	https://www.utias-sfl.net/?page_id=89
NEMO	0.2*0.3*0.44	26.4	14.4	12	15	up to 12	0.55	https://www.utias-sfl.net/?page_id=89
Defiant	0.36*0.36*0.45	57	31	26	20	up to 30	0.54	https://www.utias-sfl.net/?page_id=89
Nautilus	0.6*0.6*0.6	216	56	160	60	up to 90	0.26	https://www.utias-sfl.net/?page_id=89
Dauntless	1*1*1	1000	200	800	200	up to 300	0.20	https://www.utias-sfl.net/?page_id=89
CanX-2	0.1*0.1*0.34	3.4	2.4	1	2.5	1	0.71	https://directory.eoportal.org/web/eoportal/satellite-missions/n/nemo-am
NTS	0.2*0.2*0.2	8	6.3	1.7	4.5	2	0.79	https://directory.eoportal.org/web/eoportal/satellite-missions/n/nemo-am
GNB	0.2*0.2*0.2	8	6.3	1.7	5.5	2	0.79	https://directory.eoportal.org/web/eoportal/satellite-missions/n/nemo-am
NEMO-AM	0.2*0.2*0.4	16	8	8	7.1	9	0.50	https://directory.eoportal.org/web/eoportal/satellite-missions/n/nemo-am https://www.utias-sfl.net/?page_id=227
TET-1	0.65*0.55*0.88	314.6	224	90.6	70	50	0.71	https://directory.eoportal.org/web/eoportal/satellite-missions/t/tet-1

Table C.2: Table containing the relevant data of the reference satellites. Values in the VIS, NIR, SWIR, MWIR and LWIR columns are the respective ground sampling distance for that wavelength range.
VIS: 0.38μm - 0.75μm NIR: 0.75μm - 1.0μm SWIR: 1.0μm - 2.5μm MWIR: 3μm - 5μm LWIR: 8μm - 12μm

Satellite	VIS [m]	NIR [m]	SWIR [m]	MWIR [m]	LWIR [m]	Swath width [km]	Mass [kg]	Volume [l]	Reference links
TET-1	42.4	42.4	-	160	160	211	120	314.6	https://directory.eoportal.org/web/eoportal/satellite-missions/t/tet-1
BIROS	42.4	42.4	-	178	178	211	130	342	https://directory.eoportal.org/web/eoportal/satellite-missions/b/biros
BIRD	14.6	14.6	-	178	178	190	94	238.3	https://directory.eoportal.org/web/eoportal/satellite-missions/b/bird
Landsat 5 TM	30	30	30	-	120	185	2200	26477	https://earth.esa.int/web/eoportal/satellite-missions/l/landsat-4-5 https://www.usgs.gov/core-science-systems/nli/landsat/landsat-5?qt-science_support_page_related_con=0#qt-science_support_page_related_con
Landsat 7 ETM+	30	30	30	-	60	183	2200	26477	https://www.usgs.gov/core-science-systems/nli/landsat/landsat-7?qt-science_support_page_related_con=0#qt-science_support_page_related_con
Landsat 8	30	30	30	-	100	185	2780	26477	https://earth.esa.int/web/eoportal/satellite-missions/l/landsat-8-ldcm https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-7 Ref. [11] Ref. [19]
Landsat 9	30	30	30	-	100	185	2864	26477	https://space.skyrocket.de/doc_sdat/landsat-8.htm https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1350&context=calcon
NOAA-15 AVHRR/3	1090	1090	1090	1090	1090	833	2232	11603	https://directory.eoportal.org/web/eoportal/satellite-missions/n/noaa-poes-series-5th-generation https://directory.eoportal.org/web/eoportal/satellite-missions/n/noaa-poes-series-5th-generation
Metop-A AVHRR/3	1090	1090	1090	1090	1090	833	4085	71672	https://earth.esa.int/web/eoportal/satellite-missions/m/metop https://space.oscar.wmo.int/satellites/view/metop_a https://directory.eoportal.org/web/eoportal/satellite-missions/n/noaa-poes-series-5th-generation
Suomi NPP VIIRS	750	750	750	750	750	3060	2128	7098	https://directory.eoportal.org/web/eoportal/satellite-missions/s/suomi-npp
MODIS - Aqua	1000	1000	1000	1000	1000	2330	2934	42961	https://directory.eoportal.org/web/eoportal/satellite-missions/a/aqua https://modis.gsfc.nasa.gov/about/specifications.php
GOES-17	1000	1000	2000	2000	2000	10000	5192	133224	https://directory.eoportal.org/web/eoportal/satellite-missions/g/goes-r
Terra	15	15	30	-	90	60	4864	65424	https://earth.esa.int/web/eoportal/satellite-missions/t/terra
Sentinel-2 MSI	10	20	20	-	-	185	1140	14382	Ref. [6]
SkySat-3 - SkySat-15	1	1	-	-	-	5.9	120	288	https://earth.esa.int/eogateway/missions/skysat
SkySat-16 - SkySat-21	0.75	0.75	-	-	-	5.5	110	342	https://earth.esa.int/eogateway/missions/skysat
Worldview-3	1.24	1.24	3.7	-	-	13.1	2800		https://earth.esa.int/web/eoportal/satellite-missions/content/-/article/worldview-3 https://directory.eoportal.org/web/eoportal/satellite-missions/v-w-x-y-z/worldview-3
GeoEye-1	1.64	1.64	-	-	-	15.2	1955		https://www.satimagingcorp.com/satellite-sensors/geoeye-1/

									https://earth.esa.int/eogateway/missions/geoeye-1
Worldview 4	1.24	1.24	-	-	-	13.1	2087	26016	https://www.satimagingcorp.com/satellite-sensors/geoeye-2/
Pléiades Neo	1.2	1.2	-	-	-	14	750		https://earth.esa.int/eogateway/documents/20142/37627/1C5+VH-RODA+2021+Pl%C3%A9iades+Neo+Image+quality+indicators+and+products_Airbus-LCoeurdevey_20210422_1C5.pdf/2bab2df3-eeb0-b148-77ad-f1c627b1d3f0
									https://webapps.itc.utwente.nl/sensor/getsen.aspx?name=Pleiades%20Neo
									https://www.satimagingcorp.com/satellite-sensors/pleiades-neo/
Pleiades-1A, 1B	2	2	-	-	-	20	970		https://directory.eoportal.org/web/eoportal/satellite-missions/p/pleiades
									https://earth.esa.int/web/eoportal/satellite-missions/p/pleiades
Superview-1	2	2	-	-	-	12	560		https://www.satimagingcorp.com/satellite-sensors/superview-1/
Kompsat-3A	2.2	2.2	-	5.5	-	12	1100	10996	https://www.satimagingcorp.com/satellite-sensors/kompsat-3a/
Kompsat-3	2.8	2.8	-	-	-	15	980	10996	https://www.satimagingcorp.com/satellite-sensors/kompsat-3/
									https://earth.esa.int/web/eoportal/satellite-missions/k/kompsat-3
QuickBird	2.62	2.62	-	-	-	16.8	1100	2683	https://www.satimagingcorp.com/satellite-sensors/quickbird/
									https://directory.eoportal.org/web/eoportal/satellite-missions/q/quickbird-2
									https://earth.esa.int/web/eoportal/satellite-missions/q/quickbird-2
Gaofen-2	3.2	3.2	-	-	-	45	2100	32412	https://www.satimagingcorp.com/satellite-sensors/gaofen-2/
									https://directory.eoportal.org/web/eoportal/satellite-missions/g/gaofen-2
TripleSat	3.2	3.2	-	-	-	23.4	458	6150	https://www.satimagingcorp.com/satellite-sensors/triplesat-satellite/
Jilin-1GF03B	4	4	-	-	-	17	40		http://www.charmingglobe.com/EWeb/chanpin_view.aspx?id=744
									https://space.skyrocket.de/doc_sdat/jilin-1-highresolution-03a.htm
Jilin-1KF01	3	3	-	-	-	136	1250		https://apollomapping.com/jilin-1-nighttime-satellite-imagery
									http://www.charmingglobe.com/EWeb/chanpin_view.aspx?id=682
									https://en.wikipedia.org/wiki/Jilin-1
Jilin-1GP01/02	5	5	100	100	150	110	95		http://www.charmingglobe.com/EWeb/product_view.aspx?id=676
									https://space.skyrocket.de/doc_sdat/jilin-1-hyperspectral-01.htm
									Ref. [5]
PlanetScope	3.9	3.9	-	-	-	24.6	5.8	3	https://earth.esa.int/eogateway/missions/planetscope
									https://www.planet.com/products/satellite-imagery/files/1610.06_Spec%20Sheet_Combined_Imagery_Product_Letter_ENGv1.pdf
									Ref. [25]
RapidEye	6.5	6.5	-	-	-	77	156	856	https://www.planet.com/products/satellite-imagery/files/Planet_Combined_Imagery_Product_Specs_December2017.pdf
									https://directory.eoportal.org/web/eoportal/satellite-missions/r/rapideye
iSIM-90	2	2	7	-	-	13	20	16	https://satlantis.com/isim-90/
									Ref. [15]
									Ref. [14]
iSIM-170	1	1	3.5	-	-	7.5	50	98	Ref. [14]
iSIM-SWIR	2.5	2.5	2.5	-	-	7.5	50		https://satlantis.com/isim-swir/
Spot 5	10	10	20	-	-	60	3030	63240	https://earth.esa.int/web/eoportal/satellite-missions/s/spot-5
									Ref. [13]
MTI	5	5	20	20	20	12	614	4000	https://earth.esa.int/web/eoportal/satellite-missions/m/mti

CBERS-1	20	20	80	-	160	113	1450	7920	https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cbers-1-2
CBERS-3-4	20	20	40	-	80	120	1980	28650	https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cbers-3-4
LEOS-50HR	7.68	7.68	-	-	-	14.6	90	195	https://satsearch.co/products/berlin-space-tech-leos-50hr https://www.berlin-space-tech.com/portfolio/leos-50hr/ https://www.berlin-space-tech.com/wp-content/uploads/2020/07/PFR-PR28_LEOS-50__V1.00_.pdf
LEOS-50MR	9.2	9.2	-	-	-	70	105	130	https://www.berlin-space-tech.com/portfolio/leos-50mr/ https://www.berlin-space-tech.com/wp-content/uploads/2020/07/PFR-PR28_LEOS-50__V1.00_.pdf
MN-50 High Res	1	1	-	-	-	14.5	65		https://satsearch.co/products/minospace-mn50-platform
MN-50 Low Res	5	5	-	-	-	115	55		https://satsearch.co/products/minospace-mn50-platform
SPARK-I	50	50	-	-	-	100	50	41	https://forum.nasaspaceflight.com/index.php?topic=49898.40 https://www.n2yo.com/satellite/?s=41900 Ref. [16]
Satellogic-NuSat	1	1	-	-	90	92	37	145	https://directory.eoportal.org/web/eoportal/satellite-missions/s/satellogic https://spaceflight101.com/spacecraft/aleph-1/
Kestrel Eye 2M	1.5	1.5	-	-	-	5	40	139	https://space.skyrocket.de/doc_sdat/kestrel-eye-2m.htm https://directory.eoportal.org/web/eoportal/satellite-missions/k/kestrel-eye https://spacenews.com/armys-imaging-satellite-up-and-running-but-its-future-is-tbd/
NEMO-AM	40	40	40	-	-	129	16.1	16	https://directory.eoportal.org/web/eoportal/satellite-missions/n/nemo-am https://www.utias-sfl.net/?page_id=227
NEMO-HD	5.6	5.6	-	-	-	10	65	108	https://directory.eoportal.org/web/eoportal/satellite-missions/content/-/article/nemo–1 https://www.space.si/en/microsatellite/ https://digitalcommons.usu.edu/cgi/viewcontent.cgi?filename=0&article=1017&context=smallsat&type=additional https://geospatialworldforum.org/speaker/SpeakersImages/NEMO-HD-high-definition-video-and-multispectral-earth-imaging-on-a-microsatellite-platform.pdf
Pleiades-HR	2.8	2.8	-	-	-	20	1015		https://eoportal.org/web/eoportal/satellite-missions/p/pleiades

D

Appendix: Minimum Volume Optimized Concepts

M2 Linear extension and linear baffle deployment (1.1.1 + 2.1.1)

Parameter	Stowed	Deployed	Units
Height	0.385	0.648	m
Diameter	0.353	0.353	m
Volume	34.0	57.3	l
S/c bus volume	14.3	14.3	l
Empty volume	-	-	l
Custom depl. ratio	-	2.56	-
Modular depl. ratio	-	2.04	-

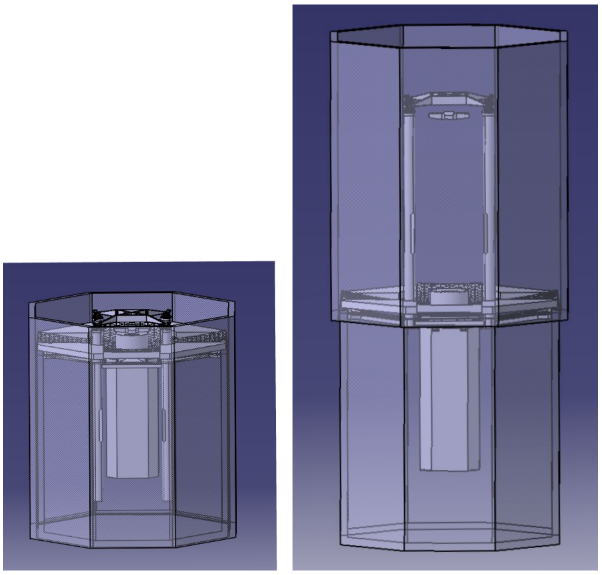


Figure D.1: Left: Data on concept M2 Linear extension and linear baffle deployment (1.1.1 + 2.1.1). Right: 3D model in stowed and deployed configuration.

M2 Linear extension and folding baffle flaps (1.1.1 + 2.1.2)

Parameter	Stowed	Deployed	Units
Height	0.321	0.632	m
Width	0.345	0.345	m
Length	0.345	0.345	m
Volume	38.2	75.2	l
S/c bus volume	21.0	21.0	l
Empty volume	-	-	l
Custom depl. ratio	-	2.67	-
Modular depl. ratio	-	1.93	-

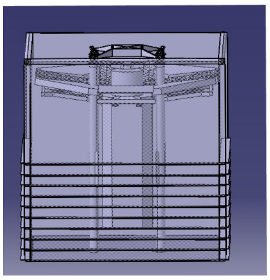
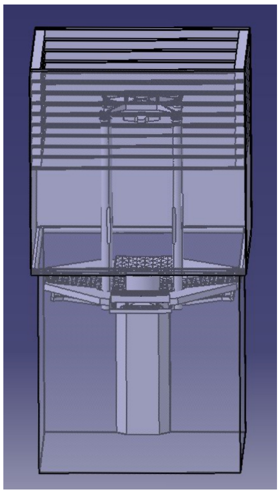



Figure D.2: Left: Data on concept M2 Linear extension and folding baffle flaps (1.1.1 + 2.1.2). Right: 3D model in stowed and deployed configuration.

M2 folding arms and linear baffle deployment (1.1.2 + 2.1.1)

Parameter	Stowed	Deployed	Units
Height	0.291	0.563	m
Diameter	0.412	0.412	m
Volume	30.9	62.0	l
S/c bus volume	7.0	7.0	l
Empty volume	4.0	-	l
Custom depl. ratio	-	2.38	-
Modular depl. ratio	-	2.14	-

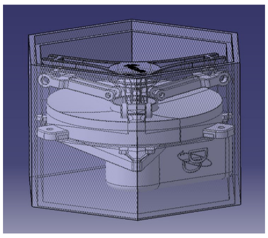
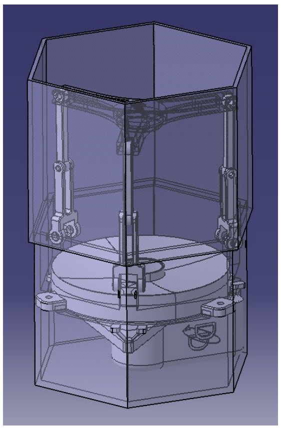



Figure D.3: Left: Data on concept M2 folding arms and linear baffle deployment (1.1.2 + 2.1.1). Right: 3D model in stowed and deployed configuration.

M2 folding arms and telescopic baffle (1.1.2 + 2.1.3)

Parameter	Stowed	Deployed	Units
Height	0.217	0.563	m
Diameter	0.412	0.412	m
Volume	23.9	61.9	l
S/c bus volume	7.0	7.0	l
Empty volume	-	-	l
Custom depl. ratio	-	2.69	-
Modular depl. ratio	-	2.38	-

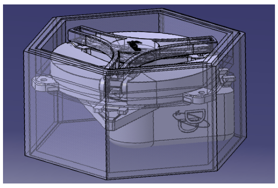
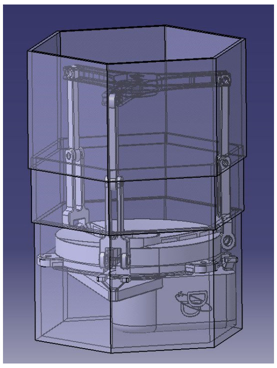



Figure D.4: Left: Data on concept M2 folding arms and telescopic baffle (1.1.2 + 2.1.3). Right: 3D model in stowed and deployed configuration.

M1 foldable and pantographic baffle (1.2.2 + 2.2.1)

Parameter	Stowed	Deployed	Units
Height	0.617	0.617	m
Width	0.220	0.220	m
Length	0.220	0.220	m
Volume	25.9	51.9	l
S/c bus volume	5.7	5.7	l
Empty volume	4.0	-	l
Custom depl. ratio	-	2.54	-
Modular depl. ratio	-	2.31	-

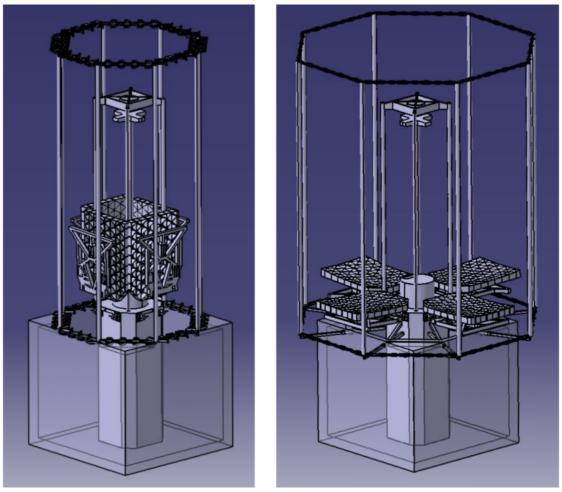


Figure D.5: Left: Data on concept M1 foldable and pantographic baffle (1.2.2 + 2.2.1). Right: 3D model in stowed and deployed configuration.

M1 foldable, M2 linear extension and pantographic baffle (1.3.1 + 2.3.1)

Parameter	Stowed	Deployed	Units
Height	0.390	0.603	m
Width	0.220	0.220	m
Length	0.220	0.220	m
Volume	15.3	51.9	l
S/c bus volume	5.7	5.7	l
Empty volume	-	-	l
Custom depl. ratio	-	3.11	-
Modular depl. ratio	-	2.77	-

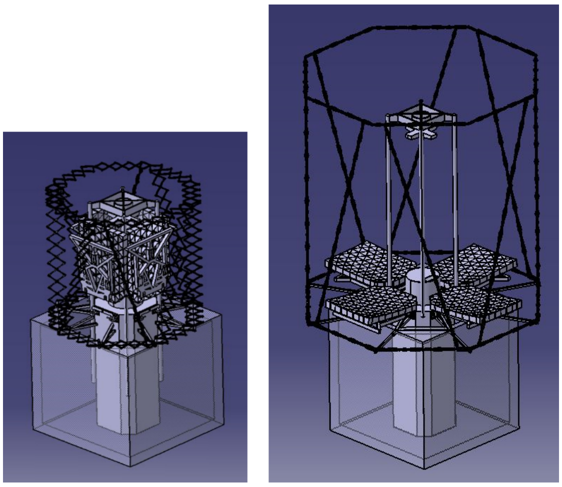


Figure D.6: Left: Data on concept M1 foldable, M2 linear extension and pantographic baffle (1.3.1 + 2.3.1). Right: 3D model in stowed and deployed configuration.

E

Appendix: SE Toolset Evaluation Form

DST Project

SE Toolset V0.65 Beta Evaluation Form

Name:
Working on subsystem:
Thesis Subject:

Give your opinion on the program on a scale of 0 to 10. 0 = Very Bad, 5 = Neutral, 10 = Excellent

	0	1	2	3	4	5	6	7	8	9	10
Layout of the program											
Ease of use											
Is everything easy to find											
Is using the program faster than manually keeping an excel file											
Provides the program a good overview of the overall system											
Does the program make dependencies clearer											
Does the use of this program improve the use of systems engineering in the project											
Is the program suitable for other projects/applications											
Does the commenting system provide a platform to discuss requirements/changes to requirements?											

Explanation on above answers if necessary

....

....

....

What functionality is still missing? And why?

....

Which parts of the program need improvement? And why?

....

How much time per week do you spend on average on SE related tasks? (e.g. requirements, budgets, design dependencies, etc.) And which tasks are this mostly?

………

How much time do you think this program can save you per week/in the long run, in comparison with how the SE was done before? Please explain, also if you think it costs you more time.

………

Do you think you will continue to use this program on a daily/weekly basis? Please explain your answer.

………

Which functionality do you use the most?

………

Which functionality do you value the most?

………

Would you recommend the use of this program in other/future projects? Please explain your answer.

………

Additional comments

……

……

……

DST Project

SE Toolset V0.65 Beta Evaluation Form

Name: Christiaan Hobijn
Working on subsystem: Secondary Mirror Support Structure (SMSS)
Thesis Subject: The design of a Secondary Mirror deployment system for a Thermal infrared deployable space telescope.

Give your opinion on the program on a scale of 0 to 10. 0 = Very Bad, 5 = Neutral, 10 = Excellent

	0	1	2	3	4	5	6	7	8	9	10
Layout of the program								X			
Ease of use							X				
Is everything easy to find						X					
Is using the program faster than manually keeping an excel file											X
Provides the program a good overview of the overall system											X
Does the program make dependencies clearer										X	
Does the use of this program improve the use of systems engineering in the project										X	
Is the program suitable for other projects/applications									X		
Does the commenting system provide a platform to discuss requirements/changes to requirements?								X			

Explanation on above answers if necessary

Overall a very very good program, but by the nature of having a long list of stuff it inevitably gets a bit messy, not sure how you can change that.

What functionality is still missing? And why?

An email notification system to see if a requirement from your department is changed, you don't really have the SE toolbox open 24/7 so you can easily miss if someone in your team has changed something

Which parts of the program need improvement? And why?

Configuration item tree doesn't allow for left to right scrolling when viewing the requirements

How much time per week do you spend on average on SE related tasks? (e.g. requirements, budgets, design dependencies, etc.) And which tasks are this mostly?

It depends, in the beginning a lot, then once the major requirements are calculated, not at all but now in my later design phase Im looking and adjusting requirements as I go along.

How much time do you think this program can save you per week/in the long run, in comparison with how the SE was done before? Please explain, also if you think it costs you more time.

In the beginning? At least a couple of days. It was a collection of like 7 different excel files and each one is semi-complete so that was mess to figure out which requirement is the actual one. Now it would take me… 30minutes to find everything I would need.

Do you think you will continue to use this program on a daily/weekly basis? Please explain your answer.

Maybe not weekly of daily because once I, as a design engineer in this project, have filled in the requirements, there is almost no need for me to look at it again.

Which functionality do you use the most?

Filling in the requirements and checking others in the list.

Which functionality do you value the most?

Having 1 place where all the requirements are and being automatically updated is very nice.

Would you recommend the use of this program in other/future projects? Please explain your answer.

For a small project such as the TIR-DST this is perfect, yes would recommend. When there is a very large project then this might get a bit more cumbersome.

Additional comments

Try to make each window being able to change its size. Having a window on a fixed size can be very annoying if the size is too big when looking at the SE toolbox from a smaller screen.

DST Project

SE Toolset V0.65 Beta Evaluation Form

Name: Fabbio Hu
Working on subsystem: Baffle
Thesis Subject: Optimisation of a deployable baffle for a Deployable Space Telescope

Give your opinion on the program on a scale of 0 to 10. 0 = Very Bad, 5 = Neutral, 10 = Excellent

	0	1	2	3	4	5	6	7	8	9	10
Layout of the program									x		
Ease of use									x		
Is everything easy to find								x			
Is using the program faster than manually keeping an excel file										x	
Provides the program a good overview of the overall system									x		
Does the program make dependencies clearer									x		
Does the use of this program improve the use of systems engineering in the project										x	
Is the program suitable for other projects/applications										x	
Does the commenting system provide a platform to discuss requirements/changes to requirements?						x					

Explanation on above answers if necessary

Commenting/discussion cannot be evaluated yet because no-one comments on it.

What functionality is still missing? And why?

Automatic ordering or option to input a key (or tag) that specifies whether the requirement in a subsystem is part of a structural – or thermal – or deployable – or launch specific – requirement. Makes for easier sorting and shifting through the requirement. Currently the requirements within a subsystem are not ordered, which can lead to missing a requirement due to glancing over the requirements too quickly.

Perhaps an additional ‘personal list could come in handy. The ‘personal list consists of for example a set of requirements which are of great importance when designing. The ‘personal tab’ then gives a quick overview of commonly used requirements. This could be enabled by for example selecting the requirement and adding it to a personal list of requirements.

Which parts of the program need improvement? And why?
Sorting of requirements inside of the subsystem. Visualisation (enable scrolling) in CIT. Overview of date when a requirement is being changed.

How much time per week do you spend on average on SE related tasks? (e.g. requirements, budgets, design dependencies, etc.) And which tasks are this mostly?

This depends on the stage of the project. At first, define the requirement. Later on, the SE toolset would be mostly used to look up requirements in order to design a subsystem.

How much time do you think this program can save you per week/in the long run, in comparison with how the SE was done before? Please explain, also if you think it costs you more time.

The development of the SE tool saves a lot of time of setting up your own requirement list (would probably take someone 2-3 days). The SE toolset also saves a lot of time scrolling through reports/thesis. Everything is structured as well.

Do you think you will continue to use this program on a daily/weekly basis? Please explain your answer.

Most likely not on a daily basis. But I can assume that at the beginning, familiarisation of the requirements is required which would need daily use of the program.

Which functionality do you use the most?

Looking at the requirements

Which functionality do you value the most?

An overview of requirements per subsystem

Would you recommend the use of this program in other/future projects? Please explain your answer.

Yes! Everything is put together in one program

Additional comments

See previous mail

DST Project

TU Delft

SE Toolset V0.65 Beta Evaluation Form

Name: Lars Boer
Working on subsystem: Detector Cooling
Thesis Subject: Detector Cooling

Give your opinion on the program on a scale of 0 to 10. 0 = Very Bad, 5 = Neutral, 10 = Excellent

	0	1	2	3	4	5	6	7	8	9	10
Layout of the program									✓		
Ease of use								✓			
Is everything easy to find										✓	
Is using the program faster than manually keeping an excel file							✓				
Provides the program a good overview of the overall system								✓			
Does the program make dependencies clearer										✓	
Does the use of this program improve the use of systems engineering in the project											✓
Is the program suitable for other projects/applications											✓
Does the commenting system provide a platform to discuss requirements/changes to requirements?									✓		

Explanation on above answers if necessary

Clear program layout, clean user interface, clear work flow, adds some functionalities missing in Excel, but is not necessarily faster. Promotes the tracking of requirements and thus system engineering, but doesn't necessarily give a good overview of choices made during the design. Generic enough to apply to other projects

What functionality is still missing? And why?

Some way to include verification/validation of the reqs and the design, a compliance checklist could be useful. Adding keyword 'tags' to the requirements would improve the search function.

Which parts of the program need improvement? And why?

The navigation can be improved. For example, if I want to check the variable list when writing a requirement, I need to close the requirements and return to the main menu. It would be great if I could have them both open simultaneously as tabs and switch between them without having to close either. I also found some minor UI bugs that I will comment on later

How much time per week do you spend on average on SE related tasks? (e.g. requirements, budgets, design dependencies, etc.) And which tasks are this mostly?

At this moment not much, but that will change after the trade-off

How much time do you think this program can save you per week/in the long run, in comparison with how the SE was done before? Please explain, also if you think it costs you more time.

It saves some time on communications, if I need a value I don't need to wait for an answer, but can look it up myself in the toolset.

Do you think you will continue to use this program on a daily/weekly basis? Please explain your answer.

Probably, a living document with all the requirements is always needed and this program provides nice functionality for that

Which functionality do you use the most?

Checking/adding/updating requirements related to my subsystem

Which functionality do you value the most?

Search function in requirements list. You can't really search that easily in Excel and the like, although it would be even better if the requirements also had tags to search for.

Would you recommend the use of this program in other/future projects? Please explain your answer.

For sure, this is much better than a text document/spreadsheet on google drive, the way you would normally do this.

Additional comments

I have made some screenshots of minor UI bugs that I experienced that I put in an additional document. Do with it what you want haha