Now that more and more universities have joined the CubeSat community and have their own satellite in Earth orbit, it is expected that the planned successors will be of higher complexity. These successors within a university satellite programme will often house more technically advanced subsystems as well as more challenging technology demonstrations for external partners. Also, the number of these third-party experiments is expected to increase throughout the programme.

In order to have successful projects, these developments ask for a robust and well-defined interface control approach. Interface control ensures the proper mutual development of satellite systems and coordination of simultaneously operating design teams. Well-defined and properly implemented interface control procedures prevent engineers from designing non-complying components that are unable to be correctly incorporated into the satellite. Redesigns are thereby less likely.

The characteristics of university satellite projects ask for a different approach to systems engineering techniques than what is common within industry. This is attributable to a scarcity of resources, most notably manpower and budget. Considering these limitations, above all, interface control procedures have to be practically implementable. This paper proposes a set of interface control tools and procedures which are based on common industry practice, but scaled down for university satellite programmes. By elaborating on the proposed tools for interface control one should be able to set up an own set of tools, customized to its own project. Implementation of the interface control tools and procedures is illustrated based on the Delfi-N3xT satellite development of the Delft University of Technology where the procedures are currently in place.

1. INTRODUCTION

With more and more university satellite programmes developing increasingly complex satellites, the need to professionalize certain aspects of the engineering development increases too. One field in which more professionalism would be beneficial is the management of interfaces. A larger technical complexity requires more accurate and traceable design parameters. As often the number of involved third parties increases together with the level of technical complexity, there is a need for good documentation and well-defined processes to optimize the contribution of all involved parties to the project. Interface control methods form a solution for this process. Interface Control (IC) can be generally described as: the management of “specifically defined physical or functional juncture between two or more configuration items” [1].

This paper will show that several university satellite programmes have grown in complexity. The technical capabilities of subsystems as well as the number of involved external parties have increased. However, university projects are limited
by resources, most notably manpower, and therefore have limited capabilities to assign interface management tasks to team members. Moreover, within engineering academics universities place varied emphasis on project management techniques in their curricula. This paper attempts to reduce the effort required for interface management by setting up clear guidelines and tools for interface control.

The guidelines provided in this paper have been specifically tailored to university satellite projects as they are currently in place within the Delfi-n3Xt project (figure 1). They have been drawn up from an extensive survey within both space industry and space agencies’ practices. For example, some particularly extensive guidelines have been published by NASA in an interface control training manual [2].

This paper first lists some considerations regarding systems engineering within university satellite projects (section 2). Section 3 describes interface control throughout a satellite development lifetime. The application of the guidelines in that section is illustrated by the actual approach to interface control within the Delfi-n3Xt project; section 4 describes interface control procedures while section 5 describes the applied interface control tools. The approach to interface control as described in this paper promotes interface control in a consistent manner throughout the whole Delfi programme. Section 6 contains the conclusions and gives some recommendations.

The paper’s main focus is on setting up procedures for external interface control. As compared to internal interface control which handles interfaces between satellite systems within the same responsible organization/university, external interface control focuses on interfaces with external project participants. However, as the essence of interface control does not change, the presented procedures and tools can be applied on both principles of interface control. One difference between internal interface control versus external is that lines of communication are in general shorter, which in general reduces the required efforts for interface control.

2. SYSTEMS ENGINEERING ENVIRONMENT IN UNIVERSITY SATELLITE PROJECTS

Specific characteristics of university satellite projects ask for a different approach to IC as compared to industry. Most notably these characteristics are the scarcity of resources, increasing technical complexity, and the involvement of third parties. As compared to commercial and institutional satellite projects some of these characteristics are quite different and ask for a non-standard approach to IC.

Available resources

According to Elstak [4] the most scarce element within a small satellite student project is the available effective man hours. As most university projects are carried out on a non-commercial base universities have limited financial resources to attract external expertise. The project is thus highly dependent on the number and expertise of students it can manage to attract to their project. This is also reflected in the tasks that are assigned to students. As the human resources are limited, a project must carefully consider on which tasks they will place students. This also holds for the tasks of IC. Some university projects choose to divide IC tasks over practically all team members, while other projects assign the IC responsibility to one team member.

Also, students often are involved in the project for a limited period of time. Within the Delfi programme for example it is common for students to stay up to one year on a project, in the framework of their final thesis work, but some team members are active on a project for just their internship of a few months. These relatively short periods cause the transfer of tasks to new team members to be more intensive as compared to industry where people tend to stay longer on projects and so provide more continuity and build up more expertise. This learning curve effect is less present within university satellite projects. To ensure that information is passed on effectively and to ensure consistency in the work that students deliver, documentation receives much emphasis within the Delfi programme.
Another consequence of the thorough involvement of students in university satellite projects is the need to rely on their expertise. A large part of this expertise comes from the curriculum that a university offers. To effectively conduct interface control tasks knowledge of project management and systems engineering techniques is highly recommended. As Elstak [4] shows in his comparison between university satellite projects it is beneficial to satellite development if students are educated in systems engineering.

The fact that in general the lines of communication are shorter within university satellite projects is beneficial to IC. As university project teams tend to be housed within the same chair, the team members see each other on a continuous base. Also contact with external project partners is relatively easy as compared to industry where external project partners often tend to be situated in other countries or even different time zones. These short lines of communication enable a swift resolution of engineering issues and thereby reduce the need for structured interface meetings.

**Innovative role of universities**

Another characteristic of university satellite projects lies in the fact that universities have the traditional role to be innovative. It can therefore be expected that student satellite projects will have an increasing technical complexity. After the initial acquaintance period in which students and the university’s research departments get familiar with satellite design, one can expect a growing role for more complex designs within the student satellite community. This has been shown by the satellite programme of Aalborg University for example. While their first cubesat, AAU CubeSat, already flew power storage and attitude determination and control capabilities [5], its successor, AAUSAT-II, has a more advanced ADCS [6]. The AAUSAT-II also implemented a more challenging payload: a Gamma radiation sensor. For their third cubesat Aalborg University is taking on another challenge. One of the main objectives for AAUSAT3 is to carry and actually operate an Automatic Identification System [7] (AIS) payload which is used for ship monitoring and is developed by COM DEV. Certain systems require accurate data processing and have additional interfaces with end-users.

Another satellite programme that is gradually increasing in complexity is being carried out at the Tokyo Institute of Technology Matunaga Laboratory for Space System. This institute successfully launched its first small satellite, CUTE-I. This initial satellite was kept relatively simple and flew an attitude determination subsystem and power storage capabilities. Its successor, Cute-1.7 + APD, was improved with an additional attitude control subsystem, a new Avalanche Photo Diode (APD) sensor for technology demonstration, and an electrodynamic tether for de-orbiting purposes [8] This design was further improved into the Cute-1.7 + APD II. Although improvements were minor (improved radio amateur platform and satellite structure [9]) it signals the intentions of the institute to set higher goals for each new development.

A similar process can be recognized within the Delfi-programme. Delfi-C3 is a satellite that is equipped with two payloads from external partners, is passively stabilized and has no power storage. Its successor, Delfi-n3Xt, will fly five payloads from external partners, and will have 3-axis stabilization capabilities and power storage capabilities [10]. The Delfi-n3Xt project has the main objectives to serve as a focal point for education, to demonstrate new technology and to conduct scientific experiments [11]. Development of the Delfi-n3Xt will be more challenging than that of its predecessor when considering the technical complexity of the design. Table 1 provides an overview of the advancements within Delfi-n3Xt over the Delfi-C3 design.

As the technical complexity increases within a university satellite programme the need for IC increases too. For example, it requires more interface management to set-up a well-functioning active ADCS than a passive attitude determination system. As many factors can influence a satellite’s ADCS clear agreements need to be made on for example material usage, mass distribution, payload disturbances etc. Also Delfi-n3Xt’s Multi-Particle Spectrometer needs sufficient interface management. This sensor requires pointing in a specific direction over parts of the orbit and thereby has a clear interface with the ADCS. Also, the instrument radiates a certain amount of heat, requires an opening in the satellite housing and generates scientific data; all these issues are to be dealt with through IC as they influence many subsystems on board the satellite. While Delfi-C3 faced some of these design challenges as well, Delfi-n3Xt has clearly more interfacing issues that need to be managed.

**External partner involvement**

With an increasing technical complexity of satellite programmes the number of involved third parties increases as well. A university satellite
Table 1: Mission characteristics of Delfi-C³ and Delfi-n3Xt

<table>
<thead>
<tr>
<th></th>
<th>Delfi-C³</th>
<th>Delfi-n3Xt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>2.2 kg</td>
<td>3.5 kg</td>
</tr>
<tr>
<td>Dimensions</td>
<td>100 x 100 x 340.4 mm</td>
<td>100 x 100 x 340.4 mm</td>
</tr>
<tr>
<td>Orbit</td>
<td>Sun-synchronous</td>
<td>Sun-synchronous</td>
</tr>
<tr>
<td>Payloads</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>ADCS</td>
<td>Passive damping</td>
<td>3-axis, 3º determination accuracy, 10º control accuracy</td>
</tr>
<tr>
<td>Communication</td>
<td>1.2 kb/s downlink in VHF, Uplink UHF</td>
<td>1.2 kb/s downlink in VHF &gt;9.6 kb/s (variable) downlink in S-band 0.6 kb/s uplink in UHF</td>
</tr>
<tr>
<td>Command &amp; data handling</td>
<td>fail-safe I²C</td>
<td>fail-safe I²C</td>
</tr>
<tr>
<td>Electrical power</td>
<td>Photovoltaic array (4 deployable panels), 3W</td>
<td>Photovoltaic array (4 deployable panels) batteries providing 12W average</td>
</tr>
<tr>
<td>Structural</td>
<td>Modified COTS body</td>
<td>Custom built structure for improved accessibility, detachable side panels</td>
</tr>
</tbody>
</table>

provides an interesting platform for them to for example test their engineering developments on, or even have instruments space qualified. As can be seen with AAUSAT3, serving as a technology demonstrator for a ship monitoring concept, and with Delfi-n3Xt which houses 5 payloads from external partners. In return, these external partners often offer their expertise to the satellite development team and testing facilities are also often made available. For example the Delfi programme benefits greatly from its contacts with industry and has been assisted with, amongst others, external expertise, structural and electrical components and testing facilities. The trend of increasing third party involvement within university projects asks for a more intensive and important IC. This is true even though, as compared to industry, university satellite projects generally have less external project participants to manage. The fact that IC can consume a large part of the project’s resources has been picked-up by industry as well. ESA for example has set-up a new approach to IC within certain types of projects. By shifting the responsibility for IC from the agency to one industry party they have managed to reduce the size of their own engineering team by half. [12]. Sole responsibility for IC thus seems to be beneficial.

In addition, third parties that are involved in a university satellite project on a supporting base through for example testing facilities, also need to be provided with clear and consistent documentation. In order to ensure proper distribution of essential information to involved third parties, some uniform documentation and communication structure should be in place. This way all external parties receive the same necessary information and will deliver the satellite development team with all required information. The communication structure ensures that this information is available in time and with the correct persons.

Contemplating, the need for IC and the level of extensiveness is always a balance between several factors and shall differ from one university project to another. The development team is in the best position to asses IC needs. One has to consider the project’s technical as well as managerial complexity, the available resources, the project schedule etc. This dependence could be very well seen within the Delfi-C³ project where ambitious IC principles were defined in the initial phase of the project but which were neglected once pressure on the project schedule increased. Documentation guidelines shared the same faith within Delfi-C³. As it is hard to define which criteria are most important for considering the need for IC, it is even harder to define a quantifiable threshold to decide whether or not to set-up extensive IC procedures. Summarizing, following factors are heavily influencing the need for IC procedures:
- technical complexity
- number of external partners
- available resources (manpower, time)

3. INTERFACE CONTROL THROUGHOUT A SATELLITE PROJECT

Throughout a satellite development lifetime there is a varied emphasis on IC efforts as the need for
interface management depends on the characteristics of the engineering phase. For example, at the beginning of the satellite development cycle less emphasis is placed on IC as there are no concrete design choices to be made or to be tracked yet. However, in the early development phases engineers should consider the complexity of interfaces that their designs might invoke. Early considerations for modularity have great benefits on interface management later on.

At the start of the development cycle the interface controller is designing IC processes and sets-up initial tools such as Interface Control Documents (ICDs). At a later stage, once design parameters are to be fixed, clear agreements with all involved parties should be made in order to allow all parties to individually develop their contribution to the satellite within certain bounds. These agreements become more and more concrete as the satellite design evolves.

Figure 2 provides an overview of how NASA considers IC to be applied throughout a satellite development cycle. The figure shows system engineering phases that can be defined within a satellite project. Within these phases several tasks are assigned to the interface controller and the level of involvement of the development team and external parties are defined. Although the NASA guidelines are a good starting point they have been adapted to a more reasonable implication for university satellite projects. As can be concluded from the shaded phases, IC efforts are not expected throughout all systems engineering phases. The different IC phases and tasks are summarized below.

![General systems engineering efforts throughout a satellite design life cycle. Phases in which IC efforts are expected are indicated by highlighted cells.](image)

**Concept definition phase**
The main systems engineering purpose of the concept definition phase is to assign functional areas of responsibility. For example responsibilities for development of satellite subsystems are assigned, this can be within a university design team as well as with external parties. In terms of IC following activities take place:

- Define and categorize interfaces: all possible interfaces are to be considered and the path by which each interface can influence others
- Select interfaces that will be designed and managed: as not all possible interfaces are required to be tracked, the interfaces that are to be tracked should be identified. In practice there are few interfaces that deserve less attention over others
- Establish formal IC procedures: define the process in which interfaces shall be defined, registered and managed. Define responsibilities within the IC procedures
- Establish IC tools formats: for all IC tools that are envisioned a format needs to be created. For example, an ICD should be structured according to the interface controller’s vision and must state the type of input that is expected (e.g. drawings, descriptions, tables etc.)

Early establishment of the IC procedures provides a firm foundation for the establishment of design parameters and thereby assists greatly in ensuring individual development of satellite subsystems and eases the documentation review process. Getting project participants involved early in the process of defining IC procedures enables receiving feedback on the proposed procedures. As the collaboration of project participants is crucial for effective IC, involving all participants as early as possible with IC is essential.

**Requirements definition**
In the requirements definition phase mission objectives are translated into requirements on satellite subsystems. These requirements are than ready to be assigned to responsible parties such that all parties can continue their individual contributions. Following IC tasks are defined within this phase:

- Completion of ICDs to the most detailed extent possible: the ICDs that have been established in the previous phase are to be filled in by the responsible project participant
- Register missing interface definition data: any interface data that is missing in an ICD is to
be registered in the ICD and in a tracking tool. Each interface description that is missing shall be assigned a responsible project participant as well as a due date. Whenever possible, missing data shall be assigned bounds to allow further development of the satellite design
- Baselining of ICDs: ICDs are to be signed by designated project authorities. Once this “baselining” has been completed they become official project documentation and any alterations to an ICD shall be reviewed extensively.

Defining requirements and baselining interface documents early in the design process provides timely information to project participants. This transparency should ensure that interface design is done correctly the first time. A proactive attitude of the interface controller to acquire all required data will limit review time and reduce unnecessary paperwork.

**Systems Integration**
In the systems integration phase all attention goes to combining all individual subsystems’ design descriptions into an integrated satellite design. This phase is crucial in the sense that any non-conformance should be addressed as soon as possible in order to prevent large redesigns and possible project delays. In terms of IC following tasks are to be performed:
- Track missing data in ICDs: the inclusion of all interface definition data is to be ensured before the Critical Design Review
- Manage ICD change requests: any request from any project participant to change a design parameter is to be assessed according to predefined procedures. This process should be defined thoroughly to ensure that any change is approved by all other affected interfaces
- Verify design compliance: acquire all individual design data and verify whether ICD requirements are met

Tracking missing interface definition data is important as this information is required by other project participants. If in-time delivery of this data can not be ensured, the project might face delays as project participants rely on certain input data to continue their individual contributions. Also, the verification process of ICD items forms an integral part of IC. All IC efforts are only justified if the ICD requirements are verified with the actual designs being delivered by the project participants. After designs are verified with the ICD agreements IC remains an important discipline. As designs can change also after verification it needs to be assured that any design changes are acceptable. In general IC efforts are needed up to the Qualification Review.

**4. IC PROCEDURES FOR DELFI-N3XT**
IC can be performed on distinct levels. For example, interfaces can be managed according to their engineering discipline (e.g. mechanical interfaces, thermal interfaces, software etc.) or according to payload/subsystem. Within Delfi-n3Xt IC is performed according to the latter option. By focussing on a specific payload or subsystem the interface controller can manage one entire single satellite system. This way great attention can be given to project participants as they in general are responsible for one entire payload or subsystem as well. This approach also eases documentation efforts as each project participant receives one tailored ICD with all relevant agreements.

The main principle of IC for the Delfi-n3Xt project is that IC is carried out by one responsible team member. Based on experience from the Delfi-C project, management of the Delfi-n3Xt has decided to shift all responsibility to a sole team member. This IC manager has the responsibility to design the IC framework and to manage the interface definitions. Together with the project team these interface definitions have been set-up.

The designed IC framework for Delfi-n3Xt is based on procedures and on complementary tools. Both will be described separately in the following sections but should be regarded as inseparable as they both assist one another to implement and perform IC. A more in-depth application of the IC procedures can be found in [3] which describes the management of interface issues of the implementation of the TµPS microthruster into the Delfi-n3Xt satellite.

The IC procedures are defined in such a way that all involved persons (i.e. responsible interface controller, project members, payload partners) have a clear view on what is expected from them during distinct phases of the project’s development.

Also, the procedures should lead to a low demand of effort from the involved project participants as one team member is assigned the main responsibility for IC. In this sense the responsible interface controller manages all interface definitions and ensures that the involved people provide him with the information he needs at that specific phase in the project’s development. Project participants
In the IC framework, the involvement of all parties is important to set the right interface controls. In order to involve everyone, the ICWG was established. In a setting(s) the ICWG was temporarily brought to life to resolve the issue. This process is clearly distinguishable in figures A.1 and A.2. Implementation of a temporary body to discuss specialist interfacing issues is common practice in industry [2], [13].

The ICWG consists of a member from the external project participant side, the relevant subsystem system engineer plus the interface controller. In organized meeting(s) the ICWG resolves any complications regarding interfaces. The interface controller ensures that the proposed solution does not interfere with any other interface definition.

**Interface Control procedures**
Procedures have been set-up taking into account the phases within the IC process. A clear distinction has been made between the initial set-up phase (Interface Definition Phase) and the actual control phase (Interface Control Phase). Additionally, the verification of interfaces can be identified as a distinct task within IC. IC procedures throughout these phases shall be described below. A graphical representation of the IC processes can be found in appendix A.

**Interface Definition Phase**
In the Interface Definition Phase the main objectives are to set-up clear IC procedures, communicate these to all project participants and to create formats of IC documentation and tools. Issues to consider when defining IC procedures are the level of involvement of project participants and their specific role within the IC framework, the sign-off loop of IC documentation, formal

---

**Involved parties in IC**
In order to be able to grasp the designed IC procedures it is essential to describe all involved project participants and their role within IC.

**Interface controller**
The interface controller is the central entity in IC. This team member defines and executes IC procedures, develops IC tools and manages interface definitions. This team member has the task to gather all interface information that is required to perform IC; he does so in close cooperation with both team members as well as the external project participants. The interface controller plays a central role in stimulating effective communication between all project participants.

Management of interface definitions is an ongoing process and needs to be done with constant accuracy. The interface controller has a pro-active behaviour in engaging the project participants to provide all required information in time. In addition, whenever design changes are proposed the interface controller initiates the design change process and communicates this to all involved project participants.

**Subsystem system engineer (project team member)**
The subsystem system engineer is responsible for the development of his subsystem. This development can not be regarded as a separate task within the satellite development; rather the subsystem design is influenced by other subsystem designs and influences others. IC plays an essential role in defining and managing the influence subsystems have on one another. It is thus to the benefit of the system engineer to actively be involved in the IC process by providing all relevant information that is required by the interface controller. Also, if any design changes occur or if any new information comes available, the subsystem engineer provides this information to the interface controller.

**External project participant**
From the external project participant a similar active role is expected as from the subsystem engineer. Although lines of communication are still present, the involvement of subsystem engineers (team members) close cooperation is also expected from external project participants. In a similar fashion they are required to provide all relevant information that is needed by the interface controller and to clearly communicate design changes. Additionally, any external project participant that is involved in for example test campaigns is required to provide and update any interface data as well.

**Interface Control Working Group**
The Interface Control Working Group (ICWG) consists of members that are deeply involved with certain interfaces. Whenever an engineering issue can not be solved via individual communication with the project participants the ICWG is temporarily brought to life to resolve the issue. This process is clearly distinguishable in figures A.1 and A.2. Implementation of a temporary body to discuss specialist interfacing issues is common practice in industry [2], [13].

The ICWG consists of a member from the external project participant side, the relevant subsystem system engineer plus the interface controller. In organized meeting(s) the ICWG resolves any complications regarding interfaces. The interface controller ensures that the proposed solution does not interfere with any other interface definition.

**Interface Control procedures**
Procedures have been set-up taking into account the phases within the IC process. A clear distinction has been made between the initial set-up phase (Interface Definition Phase) and the actual control phase (Interface Control Phase). Additionally, the verification of interfaces can be identified as a distinct task within IC. IC procedures throughout these phases shall be described below. A graphical representation of the IC processes can be found in appendix A.

**Interface Definition Phase**
In the Interface Definition Phase the main objectives are to set-up clear IC procedures, communicate these to all project participants and to create formats of IC documentation and tools. Issues to consider when defining IC procedures are the level of involvement of project participants and their specific role within the IC framework, the sign-off loop of IC documentation, formal
process for resolving conflicting interfacing issues, management of missing ICD data, verification process for interfaces, etc.

More specifically, during the set-up phase meetings are held with all external project participants of the Delfi-n3Xt project. These meetings focus on obtaining a broad overview of essential interfaces with which the satellite design has to deal. By broadly focussing on the following engineering disciplines: mechanical, electrical, command and data handling, environments, little time is consumed while valuable information is provided. During these meetings interfaces are identified and quantified wherever possible.

NASA proposes the use of N² diagrams to obtain a clear overview of all interfaces that are present within a satellite design [2]. An N² diagram furthermore enables digital management of interfaces. This task can proof to be time-consuming however as it requires the creation and maintenance of such a tool. Furthermore, using specialized software to manage interfaces is not in-line with a university’s general aim of developing satellites with very limited resources. Within the Delfi-n3Xt project no use has been made of the N² tool as the need for it has not risen. Within academics the limitations of the N² tool has been recognized as well [14]; an ICD can perform interface management tasks effectively too.

Gradually, as the meetings on interfaces mature and the required data becomes clearer, the documentation of the interface definitions solidifies. This process leads to the creation of ICDs in which all interface descriptions are registered. Figure A.1 shows a flow diagram of how the input of all involved Delfi-n3Xt project participants contributes to the ICD creation. Together with an ICD input manual the external project participants receive the ICD partly filled in with information from the initial meetings and are required to complement the missing interface data.

The aim of the Delfi-n3Xt project is to have the ICD structure defined around the Preliminary Design Review (PDR) and to have all individual ICDs baselined subsequently. The baselined version of the (partly filled) ICD serves as a fundament for the further execution of the IC process as it summarizes interface items that need to be defined. The baselined version therefore does not need to be complete in terms of all interface data but it should state a description of all items that need to be defined. Any missing data in the ICD is clearly indicated by TBDs/TBCs (To Be Determined/To Be Confirmed) and shall include responsibilities, due dates and when possible limiting bounds of design parameters.

**Interface Control Phase**

During the Interface Control Phase the main focus of IC goes out to the actual management of interfaces. With all procedures and tools in place it is now tasks to ensure that the tools and documentation are completed. While ensuring that these items are completed, IC engages in solving upcoming issues.

Throughout the development of both the Delfi-n3Xt satellite and the payloads of external project participants, the design will evolve. With increasing maturity of the designs, the baselined ICDs shall include these new design parameters and TBDs and TBCs are clarified. The interface controller takes an active role in updating missing interface data.

NASA recommends implementing so called Interface-Design-Data-Required (IDDRs) forms to indicate which interface data is still missing and to provide input on these interfaces [2]. In a university satellite project this would lead to additional efforts. In the Delfi-n3Xt project a simple database is created in Excel, the need for extra documentation in the form of IDDERS is therefore not present. Additionally, NASA advises to discuss missing interface data on a monthly base. Since lines of communication within the Delfi-n3Xt team are short, there is less need for structured meetings. The interface controller is frequently in contact with project participants and thereby has great attention for missing interface data.

ICDs are thus “living documents” in the sense that they will be adjusted throughout the project. In order to ensure that adjustments are properly incorporated in the satellite design, clear procedures are defined. The flow diagram in figure A.2 shows how all parties are involved in these procedures. It is of utmost importance that changes in design parameters will not lead to interface non-compliance issues. Therefore, before a change can be agreed upon, the procedures for the IC phase foresee in an elaborate evaluation of any changes. For this goal, the Delfi-n3Xt project team has set-up Change Proposal Forms (CPFs) to ensure that all involved parties take a critical view at changes. The process as defined in figure A.2 identifies conflicting interface issues in time and prevents faulty engineering decisions. Certain extensive change procedures are common for industry as well [15].
**Interfaces Verification**

The interface verification process is an integrated part of Interface Control. Together with all project participants, the Delfi-n3Xt team has to ensure that every defined interface is verified for compliance. Several methods exist to verify interface requirements. In considering these methods several important factors like costs, manhours and time play a role. By determining which methods are applied an acceptable balance between these factors is found. For example, a projects schedule can be reduced by up to 2 years when choosing for a small-scale verification plan instead of a full-out test campaign [16].

Also, by setting up clear requirements early-on in the verification campaign costs can be minimized [17]. Because chances in test parameters are a costly ordeal, it is essential that the verification process is thoroughly considered. According to [14] there are four possible methods for verifying a design:

- Inspection
- Analysis
- Demonstration
- Test

Considerations on these methods should depend on, amongst others, costs, schedule, applicability and available facilities. Some methods of verification are better suitable for a particular interface than others. For example, it is more sensible to use analysis for verifying a payload’s Moment of Inertia (MoI) rather than testing as this requires considerably more effort. An elaborate overview of interface verification consideration within the Delfi-n3Xt project can be found in [3].

Early-on in the definition of IC procedures responsibilities for verification of interfaces has to be assigned so that project participants can take this into account. Assigning responsibilities has to be done together with the project participants in question. As a university in general has limited testing facilities capable of testing satellite systems, it can be strategically beneficial to assign verification responsibilities to external project participants wherever possible. In this case, the interface controller must clearly define what kind of documentation the project requires in order to be able to consider a design as complying with an ICD.

The interface controller should ensure that the verification of interfaces is registered in some type of tool or database. This tool can be used for the tracking of interfaces to be verified as well. Within Delfi-n3Xt a basic Excel file is used to keep track of this. This file allows keeping track of due dates as well.

5. **IC TOOLS**

The IC tools that have been set-up assist in the implementation and performing of the IC procedures. As the procedures form the soul of IC, the tools are the embodiment. The tools enable the interface controller to bring the procedures into action and will lead to the proper registration of interface definitions. Once registered, the tools enable management of the interface definitions.

**Interface Control Document**

The ICD forms the main entity within IC. The ICD registers interface definitions in a standard word processing tool and thereby forms a set of agreements between the Delfi-n3Xt project team and the external project participants. Each external project participants receives a customized ICD which ensures those payloads are given full attention within the Delfi-n3Xt project team.

An ICD should be structured according to a logical division of engineering disciplines. Within the Delfi programme ICDs are structured according to following constituents:

1. Scope: general introduction to the use of the ICD and the payload in question
2. References: summarizing all referred to documents and drawings
3. Mechanical Requirements: mechanical interface definitions between payload and satellite, e.g. mounting characteristics, handling requirements, static envelope
4. Electrical Interface Description: all interfaces on an electrical level, e.g. power demand, connector pin allocation, harnesses
5. Command and Data Handling Description: interfaces on a command and data handling level, e.g. data package size, bit rates, test interface
6. Environments: interfaces with the satellite that occur due to external influences, e.g. payload electromagnetic field, expelled payload heat, payload venting.

**Interface Control Document – input manual**

In order to ensure easy implementation of the ICDs into the Delfi-n3Xt project an ICD input manual is distributed amongst project participants. This manual consists of the ICD structure and instructions on how to fill in the required interface descriptions, i.e. type of data and format (e.g. figure, description, table etc.). Also, the responsi-
able party for delivering data is stated in the manual.

Change Proposal Form
Any changes or additions that are to be made to the ICD after it has been baselined are to be requested with the interface controller by means of a Change Proposal Form (CPF). This form, set up in a standard word processing tool, allows the interface controller to formally communicate proposed changes with involved engineers and eases the review process as shown in figure A.2.
In order for such a form to be effective it must at minimum have provisions for:
- Payload and interface: the initiator has to indicate which payload and which interface the change proposal concerns
- Description of change proposal: the initiator has to describe the design change and provide additional documentation (e.g. updated drawings or tables)
- Effected interfaces: interfaces that are affected by this change have to be stated
- Sign-off table: project participants of which the interface is affected by the change have to agree with the change before the proposal can be authorized

Once everybody has approved the change proposal the change will be processed throughout the whole project. The CPF is then archived and added to the project’s documentation.

Interface Applicability Matrix
The interface applicability matrix is a spreadsheet-based tool that indicates which interface descriptions are applicable to which payload. This overview is to be included in all ICDs. It allows the external project participant to quickly check whether certain interface requirements apply to his payload. Also, as a standalone tool it allows the interface controller to see which interfaces apply to individual payloads and to indicate which interfaces affect other interfaces, both within the individual payload as well as within other payloads. This way the matrix assists the interface controller in filling in the CPFs.

Interface Tracking Matrix
The Interface Tracking Matrix is an internally used tool that allows the interface controller to keep track of interface definitions. In a spreadsheet all interface definitions are grouped in a matrix style according to their payload. Information on responsibility, due dates for delivery of data, and interface verification issues are included in the tool.

6. CONCLUSIONS
We have presented guidelines and tools for effective IC in a university satellite project environment. These procedures and tools are based on industry practices but adapted for the specific characteristics of a university project. The implementation of the proposed procedures has been illustrated by making use of the Delfi-n3Xt project as a real life example.
Effective IC procedures ensure the proper development of individual satellite systems and prevent interfacing issues due to non-compliance. To what level of extensiveness a project should adapt IC procedures is a consideration that differs from one project to another. We have identified key factors that influence the design of IC procedures: a project’s technical complexity, number of external partners and the available resources (i.e. manpower, time).
Out of experience from the Delfi programme we recommended to design the IC processes early on in a project development lifetime. By doing so, project participants are aware of the IC procedures and their expected input in this. This way, project participants recognize the importance of IC early on, to the benefit of their dedication. By having a committed project team and an engaging interface controller, IC shall greatly benefit the development of a university satellite.

REFERENCES


jects, 19th Annual INCOSE International Symposium


Figure A.1: IC process for obtaining baselined ICDs in the Interface Definition Phase. The involved project participant is stated inside the text boxes on the lower right.

Figure A.2: IC process of interface management during the Interface Control Phase which is initiated after ICD baselining. Distinguishable within this scheme are the phases A and B. Phase A is focused on setting up the Change Proposal Form such that every project participant will understand its contents. In phase B more project participants get involved in order to formally accept a change proposal. As these phases require different levels of participation from project participants, the Change Proposal Form’s sign-off loop is indicated with a phase A and a phase B table. This way, one can immediately see where in the change process a change proposal is situated.
Table B.1: ICD structure as in use within the Delfi-n3Xt project

<table>
<thead>
<tr>
<th>Mechanical Interfaces</th>
<th>Electrical Interfaces</th>
<th>Command and Data Handling</th>
<th>Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload envelopes</td>
<td>Harnesses and connectors</td>
<td>Bus standard</td>
<td>Electromagnetic environment</td>
</tr>
<tr>
<td>Dynamic envelope</td>
<td>Connector types and locations</td>
<td>Data handling principles</td>
<td>Payload electromagnetic fields</td>
</tr>
<tr>
<td>Orientation</td>
<td>Grounding</td>
<td>Data definition</td>
<td>Payload susceptibility to electromagnetic fields</td>
</tr>
<tr>
<td>Payload orientation</td>
<td>Grounding requirements</td>
<td>Nominal operations</td>
<td>Payload frequencies</td>
</tr>
<tr>
<td>Payload fields of view</td>
<td>Payload pin allocation</td>
<td>Data type</td>
<td>Non-allowable frequencies</td>
</tr>
<tr>
<td>Apertures/access to space</td>
<td>Harness tiepoints</td>
<td>Data package size</td>
<td>Payload frequencies in use</td>
</tr>
<tr>
<td>Mass properties</td>
<td>Power interface</td>
<td>Data sampling rate</td>
<td>Shielding</td>
</tr>
<tr>
<td>Payload mass</td>
<td>Signal characteristics</td>
<td>Bit rate</td>
<td></td>
</tr>
<tr>
<td>Payload mass variability</td>
<td>Frequency</td>
<td>Other operating modes</td>
<td></td>
</tr>
<tr>
<td>Payload centre of mass</td>
<td>Voltage</td>
<td>Overcurrent protection</td>
<td></td>
</tr>
<tr>
<td>Moments of inertia</td>
<td>Load current</td>
<td>Nominal operations</td>
<td></td>
</tr>
<tr>
<td>Material properties</td>
<td>Power requirements</td>
<td>Data type</td>
<td></td>
</tr>
<tr>
<td>Galvanic and corrosive properties of interfacing materials</td>
<td>Nominal operations</td>
<td>Data package size</td>
<td></td>
</tr>
<tr>
<td>Mechanical characteristics</td>
<td>Other operating modes</td>
<td>Data sampling rate</td>
<td></td>
</tr>
<tr>
<td>Mounting</td>
<td>Overcurrent protection</td>
<td>Bit rate</td>
<td></td>
</tr>
<tr>
<td>Mounting methods</td>
<td>High voltage restrictions</td>
<td>Command structure</td>
<td></td>
</tr>
<tr>
<td>Mounting holes and dimensions</td>
<td>Switching and casualty provisions</td>
<td>Responsibility for commanding payload</td>
<td></td>
</tr>
<tr>
<td>Hole location</td>
<td></td>
<td>Operational payload commands</td>
<td></td>
</tr>
<tr>
<td>Hole dimension</td>
<td></td>
<td>Power switching commands</td>
<td></td>
</tr>
<tr>
<td>Hole patterns</td>
<td></td>
<td>Payload reconfiguration commands</td>
<td></td>
</tr>
<tr>
<td>Mounting hardware</td>
<td></td>
<td>Payload RT (Remote Terminal) address assignment</td>
<td></td>
</tr>
<tr>
<td>Responsibility for delivery</td>
<td></td>
<td>Circuit protective devices</td>
<td></td>
</tr>
<tr>
<td>Mounting sequence</td>
<td></td>
<td>Software interface compliance</td>
<td></td>
</tr>
<tr>
<td>Provisions for Delfi-n3Xt components</td>
<td></td>
<td>Test interface</td>
<td></td>
</tr>
<tr>
<td>Disturbance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Torque profile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location of handling provisions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protection covers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sealing, pressurization attachment and locking provisions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage conditions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load carrying capabilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-flight equipment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calibration equipment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mechanical GSE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical GSE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thermal GSE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equipment removed prior to flight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equipment installed prior to flight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>