APPLICATION OF AUTOMATED DESIGN TOOLS FOR SATELLITE MISSIONS WITH THE DESIGN PLATFORM DCEP

T. Stindl[a], T. Cziep[a], M. Ehresmann[b], M. Fugmann[b], G. Herdrich[b], S. Klinkner[b]

[a] German Aerospace Center (DLR), Institute of Structures and Design, Pfaffenwaldring 38-40, 70569 Stuttgart, Germany
[b] University of Stuttgart, Institute of Space Systems, Pfaffenwaldring 29, 70569 Stuttgart, Germany

Abstract

Within the project IRAS (Integrated Research Platform for Affordable Satellites), digital tools for the fast and semi-automated early design of space missions are being developed. The constellation design tool TOCASTA (Tool for Constellation and Satellite Trade-off Analysis) identifies possible satellite constellation solutions based on coverage requirements using a semi-analytical method. It uses the commercial simulation software ASTOS to refine solutions for optimized constellation design, and performs an automated mission analysis for each solution, aided by the ESA-DRAMA software. The satellite design tool ESDC (Evolutionary System Design Converger) accelerates spacecraft design using heuristic scaling laws and evolutionary algorithms. These laws, in combination with user-defined requirements, generate estimates for subsystems, while parametric models and component-based dimensioning further predict detailed spacecraft designs. Evolutionary algorithms optimize each configuration to minimize the overall system mass. The Digital Concurrent Engineering Platform DCEP offers a web-based service for cooperative model-based systems engineering, and acts as platform for the software-aided design process by providing an intuitive user interface. It contains a parametric representation of the satellite and manages data transfer and integration of other IRAS- and third-party tools. The tools are coupled to the DCEP via an SSH-based method that allows data linking and management of the tools as well as accessing their results via the DCEP user interface, with minimal effort for the tool providers. As a first test of the coupled system of tools and DCEP, an exemplary satellite mission design has been conducted. The tools were utilized successfully via the DCEP to design several satellite constellations with different coverage requirements and altitudes. Mass and power budgets as well as thruster recommendations for the individual satellites were established for the different constellation solutions. The coupling of DCEP and tools allowed a seamless transfer of TOCASTA output data to the ESDC, enabling the rapid computation and evaluation of a large number of designs. Potential improvements in user experience and beneficial additional features were identified.

Keywords: IRAS, MBSE, Concurrent Engineering, Constellation Design, Spacecraft Design, Satellite Constellations

1. INTRODUCTION

The activities presented in this paper are part of the comprehensive project IRAS (Integrated Research Platform for Affordable Satellites, [1] [2]), which strives to develop technologies to reduce spacecraft development costs. The IRAS project investigates novel methods of satellite design and manufacturing by leveraging innovative approaches, including additive, automated manufacturing, novel propulsion systems, system integration, and digital development tools. This paper focuses on the latter aspect of digitalization, illustrating the pivotal role of digital workflows in achieving cost reductions and faster design processes.

A central part of the digitalization effort in IRAS is the conception and implementation of the Digital Concurrent Engineering Platform DCEP [3], developed at the German Aerospace Center (DLR). Its objective is to provide model-based system engineering capabilities that enable a cooperative design process on a shared central data set using shared tools and external data sources. Specialized IRAS- or third-party tools can be connected to contribute additional capabilities to the platform users. Overall, the DCEP with its network of tools and data sources aims to allow for a cohesive and efficient workflow, expediting spacecraft development.

Complementary to the DCEP, two external design tools are developed within IRAS at the Institute of Space Systems at the University of Stuttgart: The Tool for Constellation and Satellite Trade-off Analysis TOCASTA [4] and the Evolutionary System Design Converger ESDC [5] [6].

TOCASTA identifies suitable satellite constellation designs in various altitudes as well as preliminary constellation member design estimates. ESDC generates satellite designs based on a given limited set of requirements and is able to quickly fine-tune individual satellites for each constellation solution or for specific satellite missions.

To demonstrate the current capabilities of these tools and the DCEP, this paper shows the results of a constellation and satellite design case study conducted with them. Section 2 gives an overview of the tools and the DCEP, with the coupling method explained in section 3. Section 4 details the investigated mission, and the results are presented in section 5, followed by concluding remarks in section 6.

2. TOOLS AND PLATFORM

The different software tools used for this paper, TOCASTA, ESDC and the platform DCEP, are complex research software in continuous development. This section presents a short overview of their current capabilities.

2.1. Tool for Constellation and Satellite Trade-Off Analysis (TOCASTA)

Optimal design of satellite constellations is a very tedious task, as it requires many simulation runs until a solution is found that exactly matches the desired coverage characteristics. Small changes in mission requirements or the payload can make these results completely obsolete and require a full re-design of the constellation. Even then, selecting a seemingly optimal constellation without evaluating its impact on the satellites' design can lead to significant challenges in later design phases.

TOCASTA aims to support constellation mission design in three aspects during Phase 0/A studies:

- Automation allows mission design engineers to focus on other tasks
- Simultaneous analysis of many possible solutions allows selection of the best solution overall, not just from a constellation perspective
- Simplicity empowers other stakeholders in the project to evaluate possible changes in requirements or capabilities on their own

2.1.1. Capabilities

Currently, TOCASTA is capable of performing the following tasks:

- Identifying many different constellations that fulfil the given set of coverage requirements (see below), each optimized for its specific case
- Performing an automated mission analysis for all constellation solutions using, amongst other methods, ASTOS [7] and ESA-DRAMA [8]
- Preparing plots and visualizations to compare the different solutions

Coverage requirements are the most important input for constellation design, and TOCASTA supports many different variations of coverage:

- Complete permanent global coverage for Walker-Star and Walker-Delta constellations
- Coverage limited to a latitude belt
- Partial coverage for a specified service level over a specified time period
- Coverage of Earth's surface or altitude levels
- Combination of constellations, in which the lower one covers targets on Earth while the upper one provides communication links to the lower constellation

The mission analysis provides the following output:

- Preliminary mass and power budgets
- Satellite and solar generator sizing
- Cross section estimates
- Propulsion budget, considering station keeping, collision avoidance and end-of-life de-orbit if necessary
- Ground station contact analysis

A more detailed description of the constellation design and mission analysis process and the tools used is given in [1] and [4]. The following section explains the exact workflow for the use-case described below in section 4. With DCEP's tool integration features, highlighted in section 3.2, its users can achieve this without in-depth knowledge of the specific design challenges or TOCASTA itself.

2.1.2. Methodology

TOCASTA's input files, which it receives from the DCEP, contain not only the mission definition itself, such as mission requirements and constrains, but also simulation and calculation settings. This includes, for example, spatial and temporal resolutions for simulations, or constant parameters such as drag and reflectivity coefficients. Once all input and configuration files are parsed correctly, the calculations are started.

First, an initial list of constellations that are potentially able to fulfil the coverage requirements within the mission constraints is set up. While permanent, global coverage problems can be solved completely analytically using Walker-Star or Walker-Delta constellations, a semi-analytical iterative process is used for partial-coverage problems. The process is based on estimating the size of gaps in a constellation's coverage pattern (see Figure 1). One of three possible degrees of freedom (number of orbital planes, altitude, payload field-of-view) is then adjusted until the requirement is satisfied with a reasonable accuracy.

The design case described in section 4 limits the constellations' altitude to a range of 300 to 1200 km, and also assumes a constant payload field-of-view of 25°. In this case, the number of orbital planes is used for iteration. For each number of orbital planes, the necessary altitude is determined by calculating the total gap size and thus coverage achieved, comparing it to the target coverage, and interpolating a new altitude until the method converges on an optimum altitude. Solutions outside of the altitude range are discarded, with some margin to account for likely final altitude changes in the following step.

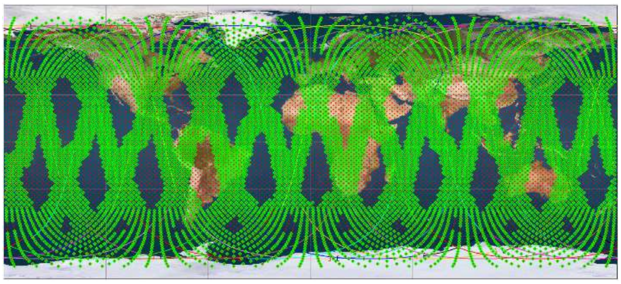


Figure 1. Example simulation of a partial-coverage constellation using ASTOS. Green areas were covered within the given time frame, the size of the gaps in between is calculated.

As the semi-analytical method relies on several assumptions, the results are refined using the commercial simulation software ASTOS. Thus, the coverage can be precisely determined with a HEALPix [9] grid of adjustable resolution. For each solution, an ASTOS scenario is generated, and the solution's altitude is modified until the coverage requirements are met. Typically, this takes only 2 or 3 simulations as the initial estimated solution is already reasonably close.

Afterwards, the mission analysis is started for each constellation case identified. Solutions with very similar altitude and inclination are processed together, since the mission analysis results would be almost identical. First, total size and power estimates are established using literature values, and the necessary solar panel area for the given orbit is calculated. Next, the resulting average cross section is estimated. Usually, TOCASTA makes use of the CROC tool from ESA-DRAMA., However, due to problems with running CROC on a virtual server without graphical output, a simple guess based on satellite and solar panel size is used instead.

Next, the residual end-of-life lifetime is determined using OSCAR from ESA-DRAMA, and active de-orbit manoeuvres are calculated if necessary. ARES from ESA-

DRAMA is used to estimate the delta-v for collision avoidance in the given orbit over the satellite's lifetime. The propulsion budget is finalized by determining the delta-v for compensating atmospheric drag. An average density at the specific altitude is computed using the Jacchia-Bowman 2008 model [10], and the necessary delta-v is integrated over the satellites' lifetime, for which the necessary solar activity and geomagnetic indices are updated automatically from online sources beforehand. The total delta-v is established separately for low-thrust and high-thrust systems, and includes an additional margin. Finally, ASTOS is used to establish ground station contact times and the maximum duration without ground station contact.

Post-processing of the results finally yields the desired output: parameters of each constellation solution, mission analysis results for the satellites of each constellation, plots that show the dependencies and interrelations within the solution space, and finally an ASTOS scenario of the constellation with the lowest satellite count that can be used for further analysis.

2.1.3. Implementation

TOCASTA is implemented in a modular, object-oriented approach using the Python programming language, which allows for platform-independence, and is run on a dedicated tool server located at the IRS. However, since the ASTOS software requires Microsoft Windows, TOCASTA also implements its own server interface to a dedicated ASTOS simulation server hosted at IRS as well. Thus, TOCASTA itself can run on a Linux-based server while still benefiting from ASTOS' functionalities.

TOCASTA also makes use of multiprocessing. As the individual constellation design cases are largely independent of each other, multiple simulations can be run in parallel, increasing the performance significantly. By default, one process per processor core is used, however, it is also possible to limit the number of parallel processes (e.g. if the memory is not sufficient otherwise).

TOCASTA provides all XML files and information for the DCEP interface as explained further in section 3. The tool's internal configuration, e.g. paths to simulation files, installation directories of other software, or TOCASTA's graphical output are defined in the XML format as well.

2.2. Evolutionary System Design Converger (ESDC) for Spacecraft Design

The continuous advancement in various space technologies and the demand for efficient and effective spacecraft systems, configurations and operations as well as the need for cost reduction necessitate an automated capability for rapid and unbiased design iteration. The Evolutionary System Design Converger (ESDC) is a holistic spacecraft design software tool currently in development to address these challenges. The scope of this tool is to support Phase 0/A, hence in-depth system design and analysis is limited and preliminary. Appropriate margins are applied to maintain feasibility. This section outlines the key features of ESDC, its solver stages, and its interface with the Digital Concurrent Engineering Platform (DCEP) for enhanced functionality and collaboration.

ESDC uses three solver stages to subsequently refine the fidelity of the system prediction, as shown in Figure 2.

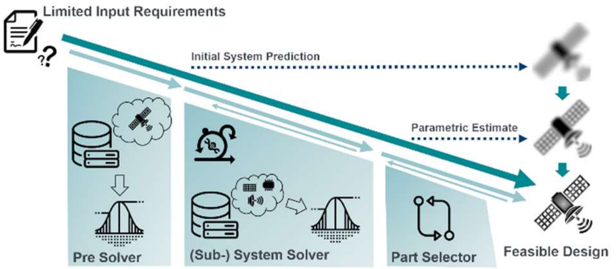


Figure 2. The process flow of the ESDC software tool with limited input system requirements and three subsequent solver stages to obtain refined spacecraft designs.

Stage 1: Preliminary Estimation and Scaling

The first stage of the ESDC utilizes a heuristic model to perform a preliminary estimation and scaling of spacecraft subsystems in terms of mass and power. This stage leverages heuristic scaling laws derived from a comprehensive database of spacecraft with flight heritage, enabling the tool to quickly estimate an initial spacecraft composition from a low number of requirement parameters [11]. The spacecraft is segmented into the following subsystems: payload, structure, propulsion, power, on-board computer, thermal, attitude determination and control and telemetry, tracking and control. Stage 1 is not only used a single time during start-up, but during each design iteration, when major system changes have been made.

Stage 2: Parametric Model and Evolutionary Algorithms

The second solver stage employs a parametric model for budgeting mass, power, heat and data between subsystems. This model then builds upon the heuristic scaling approach used in the preliminary solver stage to facilitate the parametric estimation of system and component parameters while considering available degrees of freedom in the design. To achieve optimal solutions, ESDC utilizes evolutionary algorithms that iteratively evolve and refine the design based on predefined fitness criteria such as minimum mass or maximum design margin mass. The evolutionary algorithm can distinguish between categorical degrees of freedom (e.g. technologies, propellant type, …) and quantitative degrees of freedom (e.g. specific impulse or thrust of the propulsion system) and can automatically mutate both types.

The design space is initially seeded with randomly set degrees of freedom, which is then completed with feasible estimates of the Stage 1 pre-solver. Random incremental change is applied to a current design, parametric models solve for a feasible full system solution and non-random selection permits for successive design improvements over several generations.

The second solver stage uses similar scaling estimates as stage 1 to directly solve otherwise complex correlations that require a highly detailed system design (e.g. mass of a thruster of specific technology and propellant for a certain thrust and specific impulse) by utilizing an implemented hardware data base of which the respective scaling laws are derived.

Stage 3: Component Selection and Recommendation

In the third solver stage, ESDC selects specific components based on the data-driven space hardware component database. This allows the tool to deliver reliable and feasible spacecraft designs using real-world components and hardware that have demonstrated the necessary performance characteristics. Nonetheless, it has to be noted that a significant margin has to be reserved to ensure feasibility of a Phase 0/A design and permit flexibility for detailed design downstream.

2.2.1. Advantages and DCEP support

ESDC's fast and unbiased iteration capabilities significantly reduce manual efforts and design biases, resulting in a multitude of feasible and optimal spacecraft design solutions.

The tool's interface has been updated to be operated by the DCEP. The ESDC itself is currently hosted and made available by an on-premises server at the Institute of Space Systems of the University of Stuttgart (IRS). This empowers the tool owner and developer with full control and update capabilities without requiring privileged access for DCEP users. Moreover, the seamless collaboration within the DCEP facilitates efficient information exchange and enhances overall spacecraft system design efficiency

2.2.2. Open-Source Implementation and Version Support

ESDC is implemented in Octave, an open-source alternative to Matlab®, ensuring accessibility and enabling further community contributions. The current Octave version, 8.2.0, offers most features of Matlab® and contains sufficient functionalities for robust and fast spacecraft design endeavours. The ESDC tool is available on GitHub [12].

2.3. DCEP

The Digital Concurrent Engineering Platform (DCEP) aims to facilitate the cooperative satellite design process with different institutions and corporations by providing a web-based software platform for model-based system engineering in conjunction with the digital tools developed within IRAS. Current focus of the platform's development is to support the IRAS-based digital tools and to provide a prototype for the early design phases 0/A, with a continually expanding feature set, all based on free and open software and frameworks. In this section, a brief overview of the architecture and the currently available features is given, further details can be found in [3].

2.3.1. Architecture

The DCEP uses a common server-client-architecture for its internal processing and visualization tasks, amended by a network of software tools and databases connected via different interfaces, as shown in Figure 3. The DCEP's backend, frontend and database, as well as its communication interface are briefly presented in the following, whereas the blockchain module, an experimental system to aid in protecting intellectual property, and the external databases and tools (other than TOCASTA and ESDC) have not been used for the activities described in this paper.

DCEP Backend

The DCEP backend manages access to the DCEP functionalities and applies the business logic to requested data or services. As core of the DCEP, it also manages communications with client, external databases and software tools and platforms, including handling the tool connection and usage with TOCASTA and ESDC. It is built with Flask [13], a light-weight Python framework for web applications.

DCEP Frontend

Developed with the Flutter UI Toolkit [14], the graphical user interface can be deployed directly as desktop client or as frontend web server to present the DCEP data to its users. By sending requests to the interface of the DCEP backend, all functionalities are easily accessible via the DCEP frontend, although direct communication with the interface is possible as well.

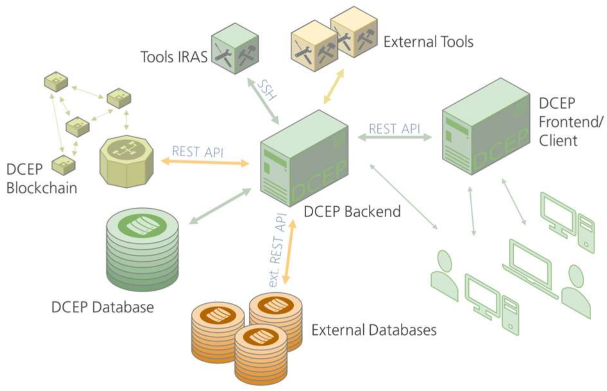


Figure 3. DCEP architecture and interfaces

DCEP Database

Internal DCEP data is stored in a Non-SQL database (MongoDB [15]) in so-called collections and documents, reflecting the DCEP data model and structure.

Client and Server Communication

As frontend and backend are two independent entities, the frontend has to be capable of understanding the response based on an application protocol to visualize the requested data provided by the backend. A representational state transfer interface (REST API) enables requests to read, create, update or delete data or to execute more complex operations. Not only the client or frontend can make use of the REST API, but also software tools or other servers, leading to easy access and data exchange.

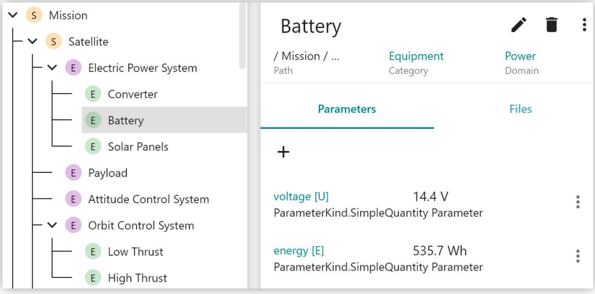
2.3.2. Data Model and Capabilities

Currently, the following features, which are also represented by the internal data model, are implemented in the DCEP. For a more detailed description of the DCEP's data model and basic functionalities, see [3].

Product Modelling

The DCEP's data model stores project information in a hierarchical product tree of elements, each representing a (sub-)system or component described by textual meta information and parameters. A product, satellite system or complete mission can be decomposed into one root element with freely nested child elements which can be created, edited, deleted, copied or moved within the tree. Defined decomposition rules can be applied by categorizing these elements, but are not strictly enforced.

The single elements are further defined by parametric descriptions which are added by instantiating parameter types such as “mass” and assigning a value e.g. “1.0” and an appropriate unit such as “kilogram”. An example from the graphical user interface for a product tree and an element is shown in Figure 4. Quantifiable parameters also may contain optional margin values. In addition to simple quantity parameters, multiple, diverse kinds of parameter types for textual, Boolean or array data can be assigned to an element. Furthermore, individual parameters can be defined to be state-dependent, meaning the parameter’s value changes with defined discrete operational states or state combinations (e.g. available power from solar panels depending on current pointing mode). Parameter values can be linked to external REST API data sources and be updated unidirectionally on request.



/ Mission / ... Path
Equipment Category
Power Domain

voltage [U] 14.4 V
ParameterKind.SimpleQuantity Parameter

energy [E] 535.7 Wh
ParameterKind.SimpleQuantity Parameter

Figure 4. Graphical user interface, showing a simplified product tree and one element with parameters (image detail)

As the parameters define the design and behaviour of a system, their values are the essential data exchanged with other tools. Parameter values are used to create input files for those tools, and the tools’ resulting output file values can be transferred to the product tree’s parameters.

Reference Data Library

A so-called reference data library includes all user- or pre-defined general data that may be referenced or used in the product design, such as units, prefixes and possible parameter types.

Files

Files, such as data sheets or other additional information can be attached to elements and are stored in the DCEP database. Files created or received during the tool usage process are stored as well. Users can upload, download or delete a file and/or edit the file’s meta information.

User Management and Data Provenance

The DCEP contains an implemented authentication and authorization system together with a user management system where users can be administered. Users can be assigned to freely definable roles such as “administrator” or “observer” and can belong to one or more, also freely definable, expert domains. Registered users are initially authenticated by the DCEP via a password and then receive a time-limited access token (JSON Web Token), used for authorization for all further requests. Access to and usage of the different functionalities and individual data objects is additionally restricted depending on the user’s assigned role and expert domains. User operations on data such as creating, updating or deleting are recorded, with information about the user and point in time to trace data changes.

External Tool Usage and Data Sources

With the provided REST API, other tools, platforms or services can send requests directly to the backend, enabling them to access the complete DCEP feature set and data, if authorized to do so. Using this, external tools can exchange data with the DCEP, e.g. a CAD tool extracting model data and transferring it back after changes done by a user. A more closely integrated method to make file-based external tools usable through the DCEP user interface has been developed to reduce the necessary effort to connect the tool to the DCEP and to improve ease of use for tool users. It has been used for the computations presented in this paper and is described in the next section.

3. DCEP AND TOOL COUPLING

While the DCEP and the individual tools can be used independently, integrating them more closely yields numerous benefits, from efficient data exchange to ease of use. This does not only apply to the tools mentioned above, but also to other tools developed within IRAS or potentially provided by a partner in a joint DCEP project. Therefore, a generalized tool coupling concept has been developed and implemented, in addition to the previously available method of connecting tools via REST API.

3.1. Concept

The basic requirement for the tool coupling method is that the tool uses files as input and output. In these files, the data is contained in (optionally hierarchical) data elements/objects in a specific format, with XML currently being the only supported one. The tool is located on a separate tool server, which has to be accessible by the DCEP backend via an SSH connection. Figure 5 shows the general process of coupling and using a tool via DCEP, described in more detail in the following sections.

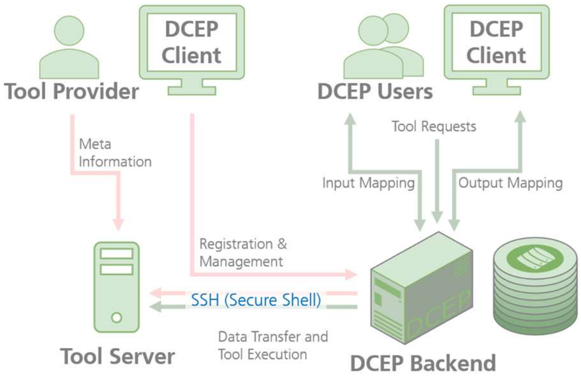


Figure 5. DCEP tool coupling and usage concept

3.2. Tool Coupling and Registration

Before connecting a tool to the DCEP, the tool owner has to provide meta information and prepare a start script on the tool server. The meta information is a simple text file in JSON format and contains information about available tool versions and their respective paths to the start script, input and output files and their templates as well as to a log file. The log file has to contain specified keywords that the

DCEP uses to determine the current status of a running tool. Additionally, the owner has to provide templates for the tool input files, optionally also for the expected output files. The templates are based on the original tool input files and can be amended by special DCEP blocks.

These special blocks provide additional information to the DCEP and its users. Structural information such as the number of possible occurrences of a data element allows DCEP users to adjust the input accordingly. This includes making elements optional or correlating them with elements in different files. Restrictive information, like minimum or maximum values, enumeration options or expected data types, is enforced by the DCEP and helps users avoid errors while using the tools. Additional possible information that can be added includes descriptions, expected units or default values, which are shown to the users to help them use the tool correctly. The special blocks allow references to other elements of the same or different files. They also allow conditional information, enabling the DCEP to generate structurally complex and dynamic input files.

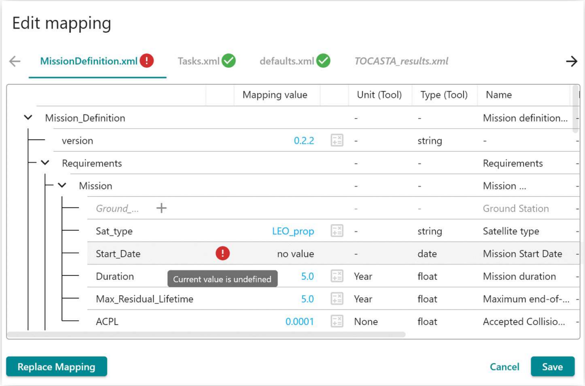
The tool is then registered in the DCEP by adding it via the DCEP GUI or REST API, providing server information and the location of the meta file as well as defining authorized user domains. The DCEP then parses the meta file and the file templates, checks them for errors, and makes the tool available to authorized DCEP users.

This method allows the tools to be used through the DCEP without giving users direct access to the tools or having to distribute them to the users. Basing the templates on readily available input files, with optional, easily provided amendments for additional information, allows tools to be added with minimum effort. This makes it feasible to even add smaller scripts or rarely used tools.

3.3. Tool Usage

After registering the tool in the DCEP, it is made available to its users, who can then create so-called tool jobs to map DCEP data to the parametric elements in the tool input and output files. For this, users are presented with a graphical interface (Figure 6), which shows the data elements and their hierarchy as well as additional information, depending on the special blocks added to the templates. A minimum viable use case, using provided default values, is initially generated by the DCEP. Users are then able to create a so-called tool mapping by changing the data structure and by entering input values or linking DCEP parameter values to the individual data elements. This allows future runs of the tool to be conducted with up-to-date values of those DCEP parameters without having to link the values again, enabling quick iterations with minimal preparation effort. The DCEP dynamically checks the validity of the mapping and informs the user of potential issues.

Running the tool is handled fully by the DCEP. Input files are generated based on the template and the completed mapping, including the current values of linked DCEP parameters. The files are transferred to the tool server and the tool is started via the command script. A unique run identification number is associated with the tool run and is optionally used in the file paths to allow multiple simultaneous tool runs and differentiation of runs on the server. The log file on the server is read by the DCEP at the request of a user to determine the tool run status based on specific keywords. On tool run completion, the results found in the designated output folder, including additional files such as plots, are transferred back to the DCEP. All files are made available for download by the users. The data output files specified in the meta file are parsed and presented to the user to create (or re-use) an output mapping, i.e. linking the received data elements to DCEP parameters. The data then is submitted into the DCEP data model, according to the mapping.



	Mapping value	Unit (Tool)	Type (Tool)	Name
Mission_Definition		-	-	Mission definition...
version	0.2.2	-	string	-
Requirements		-	-	Requirements
Mission		-	-	Mission ...
Ground_...		-	-	Ground Station
Sat_type	LEO_prop	-	string	Satellite type
Start_Date	no value	-	date	Mission Start Date
Duration	5.0	Year	float	Mission duration
Max_Residual_Lifetime	5.0	Year	float	Maximum end-of-...
ACPL	0.0001	None	float	Accepted Collisio...

Figure 6. DCEP tool mapping interface (clipped)

This method enables users to use the coupled tools easily without having to manually edit the input files. The additional information helps users identify errors or better understand input data. Directly linking DCEP data allows quick and easy tool runs with changing input data, while the DCEP handles file generation, data transfer and tool status management. Finally, every tool run is preserved in the DCEP, including all generated and received files, making it possible to consult earlier results or get a quick overview of completed tool runs. By linking outputs of one tool to DCEP parameters and inputs of a different tool to the same parameters, a tool chain can be created to easily iterate designs over multiple tools.

4. DEMONSTRATION USE-CASE

The use-case to demonstrate and verify the capabilities of DCEP, TOCASTA and ESDC when coupled together is based on a preliminary small satellite design study conducted within earlier phases of the IRAS project, called OREUS (Observation of Re-Entry Events Using Space 4.0) [1]. It was meant to serve as a platform to demonstrate and validate multiple new technologies, such as novel propulsion systems, use of 3D-printed structures with function integration, and COTS electronics. Besides technology demonstrations, it also included a scientific payload capable of observing atmospheric (re)-entry events.

OREUS' scientific payload consists of two cameras in the visual spectrum to detect events, one for day-side and one for night-side operations [16]. The cameras cover a wide field-of-view of 25° and detect any entering object, primarily meteoroids. On top of that, when a much brighter, but slower event is detected (indicating an artificial object such as rocket stages or disused satellites), a UV spectroscope with a narrow field-of-view can be used to investigate their break-up [17]. This allows to establish the break-up sequence by identifying individual parts of the object by material via characteristic spectral lines, enabling better demisability models.

Since the satellite is supposed to point the UV spectroscope at its targets only after detection, the field-of-view of 25° is defining for constellation design. The complete payload module (without the star trackers, which are not part of the payload, but including the technology demonstrations) as shown in Figure 7 has a total mass of 13.6 kg and a peak power consumption of 32.19 W.

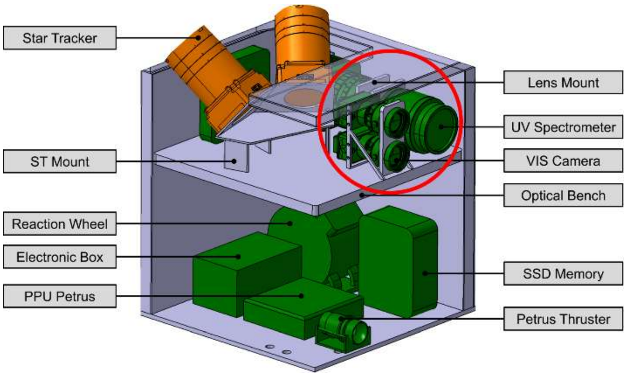


Figure 7. The payload module of the proposed OREUS satellite, with the coverage-defining cameras encircled [17]

(Re)-entry events are typically short, and even slow and large objects are only observable for a few minutes; with meteor entries being even shorter. A partial coverage of 90% within 10 minutes was established as coverage requirement. Although complete permanent coverage would be desirable from a scientific point-of-view, a full-coverage constellation requires significantly more satellites than one with only partial coverage. As full-coverage problems can be solved analytically, this would also not allow to demonstrate TOCASTA's capabilities. The maximum latitude had to be limited to 85° due to problems in the convergence of TOCASTA's estimation method. The target altitude (at which break-up events typically occur) was set to 80 km. All relevant mission requirements and constraints are summarized in Table 1.

Coverage	90% in 10 minutes
Target altitude	80 km
Maximum target latitude	85°
Altitude range	300 – 1200 km
Mission start date	January 1st, 2028
Mission duration	5 years
Payload field-of-view	25°
Payload mass	13.6 kg
Payload power	32.19 W
Allowed end-of-life lifetime	5 years
Acceptable collision probability	10^{-4}
Ground station (Stuttgart)	Latitude: 48.75° Longitude: 9.1° Altitude: 500 m
Min. elevation at ground station	5°

Table 1. Mission requirements summary

5. RESULTS

TOCASTA and ESDC were registered in the DCEP to conduct the design of the use-case from section 4 via the platform, with the mission requirements of Table 1 used as input parameters for TOCASTA to identify suitable constellation configurations (e.g. numbers of satellites and orbital planes) for differing orbit heights. The resulting solutions and their analysis results were then submitted back to DCEP to each serve as input parameters for the ESDC tool runs for finding optimal spacecraft design solutions for the given parameters. The overall experience of working with DCEP and utilizing the expert tools as well as the respective results of both tools are described in this section

5.1. DCEP Usage

In preparation for the tool runs, the mission requirements from section 4 were inserted manually into the DCEP as hierarchical elements with parameters, building a basic product tree. In addition, the necessary input parameters and output parameters of both tools were created as a blueprint, so the subsequent process of mapping tool values to DCEP data was accelerated. The coupling of both tools to the DCEP worked well, and the mapping of input and output parameters was easy to do and intuitive for the tool developers and users.

First, TOCASTA was started through the DCEP with the mission requirements as inputs. The possibility to perform lengthy tool runs during night or weekends, independent of the user's terminal, was especially significant as the TOCASTA run took appr. 9 hours for its calculations and simulations. While the current status of a tool run as provided via the log file keywords is visible to the DCEP user, a closer integration might be beneficial to assess the current status on the server system level. Additionally, tools providing an estimation of projected run time to the DCEP would be useful for users unfamiliar with the tool.

Transferring data between expert tools from different domains in a correct and consistent way is challenging to do manually in all engineering projects. Here, the DCEP's approach with pre-defined input and output information by the tool developers and the reuse of already created mappings led to time savings and avoided errors. This was particularly noticeable when the 30 solutions of TOCASTA were successfully used as input cases for individual ESDC tool runs, which needed just a few easy user actions.

One advantage of the implemented DCEP architecture and the interface definition to external tools is the ability to perform updates and upgrades in the background. During the process described above, it became necessary to add functionality to the ESDC tool, including adaptions to the input/output templates, which were pushed to the tool server. After requesting the DCEP to update the ESDC tool coupling based on the new template, the user mappings were automatically extended by the newly available input parameters which could then be easily linked to DCEP data.

A desirable, but yet missing feature was identified when working with tools that generate multiple, competing outputs: ideally, it should be possible to generate multiple design paths out of the different solutions that can be followed simultaneously, which is not yet implemented in the DCEP.

5.2. TOCASTA Results

A total of 30 possible constellation solutions were identified and analysed, distributed across the given altitude range as shown in Figure 8. All constellations meet the desired coverage of 90% within 10 minutes with a tolerance of 0.1%.

The total satellite count was between 230 for the highest solution at 1101 km and 801 at 323 km altitude, with 23 to 89 orbital planes and 9 or 10 satellites per plane. A notable outlier at 377 km requiring 870 satellites (87 planes, 10 satellites each) was identified. This case was likely initialized at an altitude at which 10 satellites per plane would be necessary, but it only converged when the altitude was low enough for 9 satellites per plane to suffice. In future versions, a revision method for such cases should be implemented.

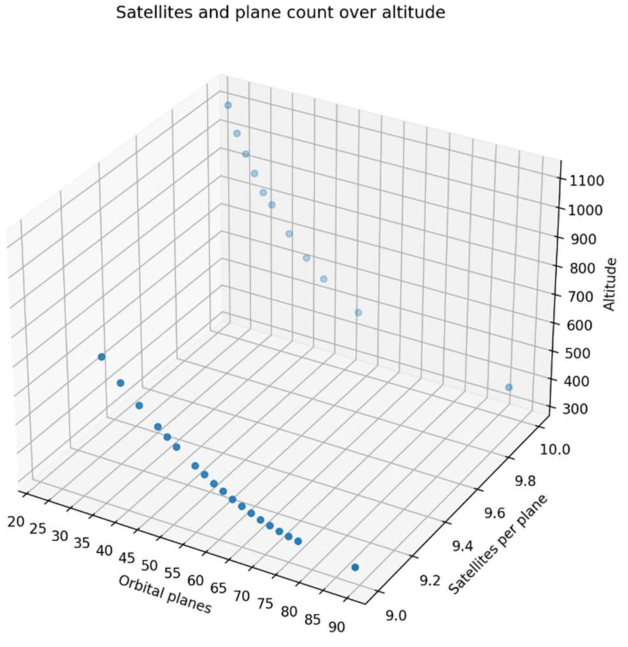


Figure 8. Constellation solutions identified

The propulsion budgets for low thrust systems is shown in Figure 9. As expected, station keeping is dominant at low altitudes, whereas the propulsive deorbit makes up most of the delta-v (Δv) budget above 550 km. The Δv for collision avoidance is too small to be noticeable. The determined annual collision probability, and thus the required Δv, peaks at appr. 800 km, as can be seen in Figure 10. This is to be expected as a result of the FY-1C destruction and the Iridium-33 / Kosmos-2251 collision. More mission analysis results, such as a ground station analysis, are available, and could be used for example by ESDC once it supports more detailed communication subsystem models.

It has to be noted that the size, mass and power estimates of TOCASTA are based on average LEO satellites. The large power demand of electric propulsion systems for drag compensation below 500 km would lead to comparably large solar generators, further increasing the Δv for low-thrust systems. Iterating the entire design after a better propulsion power estimate was produced by ESDC would be advisable.

Overall, TOCASTA provided suitable constellation solutions across the entire altitude range. The automated mission analysis reproduced the expected dependencies between orbital parameters and satellite subsystem requirements, allowing their use in the subsequent satellite design.

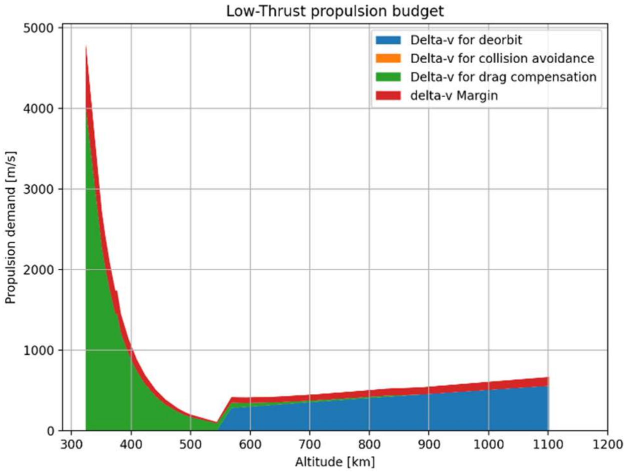


Figure 9. Low-Thrust propulsion budget

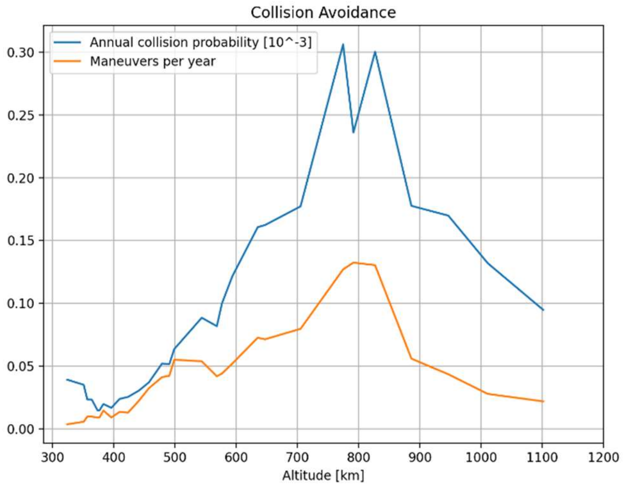


Figure 10. Collision avoidance results

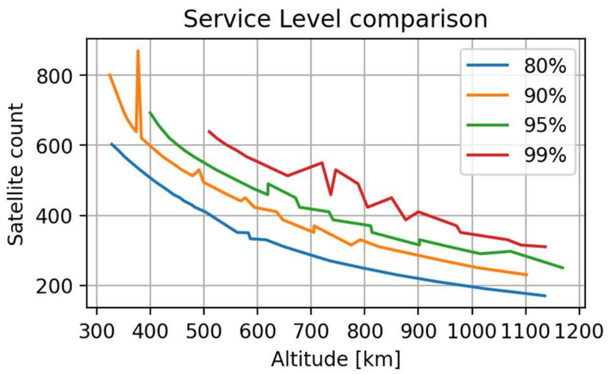


Figure 11. Comparison of different service levels

Finally, additional TOCASTA runs were performed to show the impact of mission requirements on constellation parameters. From the graphs in Figure 11, it can be seen that the methodology works better with lower coverages, as TOCASTA fails to find solutions at low altitudes for high service levels and the graphs seem highly unsteady. The reason and possible solutions for this have to be investigated further. Nonetheless, the expected connection between service level and altitude or satellite count can be clearly seen, and an automated evaluation is possible

5.3. ESDC Results

TOCASTA's results are transformed by the DCEP to be utilized as input for ESDC. This means that all 30 solutions found by TOCASTA are mapped via the DCEP to generate 30 tool runs for spacecraft optimization using ESDC. For this show case, only the parameters total mass, total power, payload power, velocity increment and orbit height are mapped from TOCASTA to ESDC.

As ESDC currently does not support parametric scaling of chemical propulsion systems, only the low-thrust case relevant for electric propulsion systems is considered.

The analysis is initialized by seeding the design space with 50 random designs (i.e. the initial population), which is then iterated for at least 30 generations. Both the number of initial designs and the minimum iteration number are parameters that can be defined by the user. A generation is not necessarily a successful improved design, but rather the attempt of improvement. If a worse design is found – in terms of mass or mass margin – the previous generation is reiterated again. Convergence is assumed after 20 successive unsuccessful mutations. If a successful mutation is found, the counter is reset and at least 20 additional attempts of design mutation will follow.

The ESDC considered input requirements are illustrated in Figure 12. These requirements are made up of the 30 cases of constellations designed by TOCASTA, each containing the mapped parameters of the desired total mass, payload mass and power demand. An additional arbitrary input of 100 W for the power available to the propulsion system was added via the DCEP as initial guess. The range for the considered orbit height spans from 300 to 1200 km.

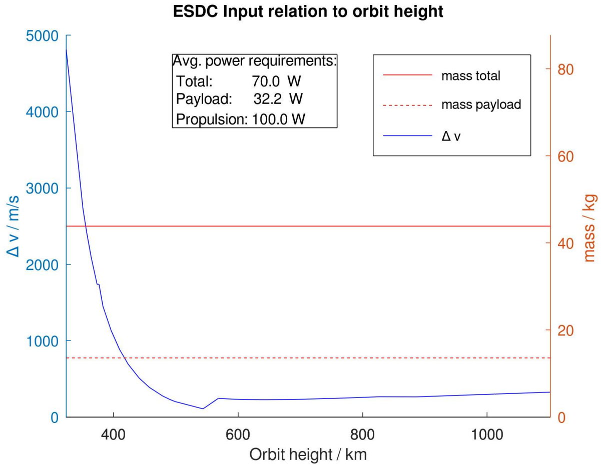


Figure 12. ESDC analysis input parameters, variation of Δv over the relevant orbit height, with given mass and power constants.

It is worth noting that the total required input power in Figure 12, estimated by TOCASTA, is lower than the combined power draw of the payload and propulsion system. This is an input oversight caused by manually adding the 100 W propulsion power while TOCASTA does not specify power demands for this system and therefore does not include those in the total power used as input for ESDC. ESDC has the ability to automatically rectify this situation. The correction is prompted by the complexity of accurately estimating the total power from the outset, owing to intricate interdependencies among individual system power consumption. As a result, the total power draw is adjusted to match the sum of all subsystem powers, and the power system's mass is appropriately scaled and cross-checked against the allowed mass margin.

After 33 generations of evolution, the outcomes for the optimal solution among the initial 50 random designs are presented. These results are provided as absolute system mass compositions in Figure 13. These should be considered as exemplary output, as manual correction was required to correctly accommodate the payload requirements, other systems have been adapted to fit total mass requirements. A cumulative graph is employed to illustrate these results, with coloured areas corresponding to the respective lines above them and the total mass corresponding to the overall mass including margin. The resulting distributions align closely with the inputs shown in Figure 12. This alignment reveals a linearization above ca. 500 km orbit height, while at lower orbits, the influence of velocity increment begins to impact the system composition. The solution for the lowest orbit with the highest velocity increment yields a negative margin mass, i.e. in this case no feasible solution has been found by ESDC.

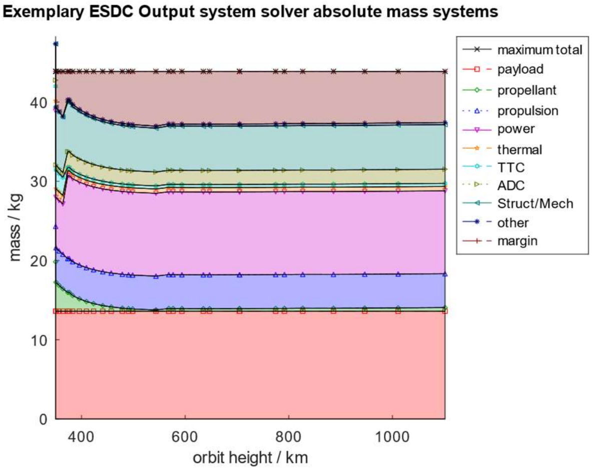


Figure 13. Exemplary ESDC system design output as cumulative depiction of absolute system mass composition corresponding to orbit height

The individual mass fractions for all subsystems are similar across the complete altitude range. An important exception is the propellant fraction, where the increased delta-v requirements for lower altitudes lead to increased propellant mass due to their correlation governed by the Tsiolkovsky rocket equation. Similar observations can be inferred when considering power scaling of individual systems, which are therefore omitted for the sake of brevity of this paper. The input oversight of a total power draw of 70 W is increased to 162 W by ESDC to accommodate operating all systems according to estimations.

Unfortunately, this linearity and self-similarity in the system estimation is carried downstream. Thus, the result of the propulsion system analysis is that for all 30 cases of differing orbit height, the best four solutions of the 50 seed points are all a grid ion thruster (GIT) to be operated with Xenon propellant at an effective exhaust velocity 31.4 km/s at a thrust of 1.7 mN. This corresponds to the jet power of the thruster to be approximately 26.5 W, which correspond to the 100 W maximum propulsion system power draw, including typical losses for this type of thrusters.

To have at least some variation in terms of propulsion system configuration, one has to go as deep as the fifth best found solution of the individual tool run. Usually a result used in the DCEP only considers the best design solution of all 50 evolutionary lineages. When considering less-optimal solutions, more variance can be observed. Then, two different solutions appear: An indium-fed field-emission electric propulsion (FEEP) thruster at an altitude of 364 km, and an ammonia-fed thermal arcjet at 490 km altitude. The parameters of these solutions are shown in Table 2.

	GIT	***FEEP***	***Arcjet***
Orbit height / km	All	364	490
c_e / km/s	31.4	35.3	7.8
F / mN	1.7	0.7	0.90
P_{jet}/ W	26.5	12.5	22
$\eta_{thruster}$ / -	0.29	0.13	0.24
η_{PPU} / .	0.92	1*	0.92
Δv / m/s	full range	2092	230
Propellant	Xe	In	NH3
Model	RIT µX	Nano	Velarc
Reference	[18]	[19]	[20]
*PPU efficiency is assumed here as 100%, as PPU efficiency is already considered within the thruster efficiency via available data sheets			

Table 2. ESDC Output on subsystem solver level, with an optimal (i.e. GIT) and two non-optimal propulsion system (i.e. FEEP and arcjet) solutions shown.

The quantities shown in Table 2 allow to elaborate on system design to a limited degree. First, it is notable that although a FEEP provides the largest effective exhaust velocity, a GIT is always more suitable due to its efficiency advantage. The respective power system has to be scaled accordingly to accommodate the increased power draw, while an increase in effective exhaust velocity has a squared impact on power draw. Thus, a FEEP has already two significant power disadvantages when compared to a high efficiency GIT thruster. The NH3 arcjet offers slightly lower efficiencies than a GIT, at a significantly lower specific impulse, resulting in a considerably higher need for propellant mass.

It has to be noted that the current version of the ESDC database does not include Hall-effect thruster performance information and is therefore unable to consider and scale such systems, which could result in additional alternative designs. Improvements on database extensions here are ongoing. Furthermore, the consideration of chemical propulsion systems is currently not implemented in ESDC. Especially for the low Δv demands given by TOCASTA for high-altitude constellations using high-thrust systems for end-of-life deorbit, chemical systems should be a viable mass-optimal solution.

Furthermore, the skilled designer has to carefully consider the presented solutions and cross-check feasibility of parameters that are not directly mapped via DCEP. In this show case, for example, the thrust level of the GIT is likely to be insufficient when considering the minimum thrust requirement determined by TOCASTA.

6. CONCLUSIONS

In this paper, the current capabilities of the digital tools developed within the Integrated Research Platform for Affordable Satellites (IRAS) project were showcased. While not quantifiable, the cost and time savings enabled by the DCEP with its well-integrated tools ESDC and TOCASTA became evident.

TOCASTA has proven its capabilities to generate and analyse many viable solutions, its interaction with DCEP and ESDC was seamless, and setting up the use-case was quick and easy after the mapping was done. However, improvements to the coverage estimation at near-polar inclinations are necessary, and the behaviour at service levels close to 100% requires more investigation. Additional approaches to solve constellation problems, e.g. using constellations at multiple altitudes and inclinations, are in development.

The outputs of TOCASTA served as inputs for ESDC and enabled the design and optimization of the individual satellites for each constellation solution. ESDC was able to show its capabilities for the given use case. It provides comprehensive estimations and compositions for spacecraft systems, offering valuable insights into subsystem design and yields specific hardware recommendations for a mass optimal spacecraft. Notably, ESDC demonstrates remarkable efficiency, performing an input job with typically creating and analysing more than 1500 designs in just approximately 3 minutes to reach convergence. This significant acceleration in Phase 0/A of the project underscores its crucial contribution to streamlining spacecraft development within the IRAS initiative. Additionally, the tool's ability to reveal clear trends of spacecraft design and composition for varying orbit heights further enhances its value in system optimization.

By coupling the tools and providing a graphical interface, the DCEP enabled seamless interaction between the tools and rapid iterations of different designs. Both tools were successfully integrated into the DCEP, using the newly implemented method for user-friendly, lightweight tool coupling with communication via SSH. The successive, iterative use of the tools through the platform with prepared and quickly (re)generated input mappings as well as the automated design through the expert tools speeds up the space project design process. It eases finding the best design out of a broad selection, resulting in fast decision making and significant cost savings

Looking ahead, the focus of upcoming development is set on enhancing the user experience and expanding the feature set of the DCEP and the expert tools TOCASTA and ESDC. As the IRAS project continues to evolve, these advancements hold the potential to overall improve satellite mission design, enabling greater efficiency, innovation, and affordability.

ACKNOWLEDGEMENTS

This work is funded by the Baden-Württemberg Ministry of Economic Affairs, Labour and Tourism, with the grant numbers 3-4332.62-DLR/49 (IRAS), 3-4332.62-DLR/56 (IRAS II), 3-4332.62-DLR/59 (IRAS III) and WM3-4332-157/68 (IRAS IV).

Contact:

torsten.stindl@dlr.de

REFERENCES

[1] T. Stäbler, K. Chen, T. Cziep, M. Echsel, J. Fischer, L. Friedrich, M. Fugmann, D. Galla, N. Gottschalk, J. Hildebrandt, S. Hümbert, J. Skalden and T. Stindl, “IRAS - new technologies for low cost satellites,” *72nd International Astronautical Congress,* 25-29 10 2021.

[2] T. Cziep, F. Atzler, M. Echsel, M. Ehresmann, S. Fasoulas, M. Fugmann, D. Galla, N. Gottschalk, G. Herdrich, J. Hildebrandt, S. Hümbert, S. Klinkner, I. Sakraker, J. Skalden, T. Stindl, A. Vikas and H. Voggenreiter, “Overview of Activities in the Project Integrated Research Platform for Affordable Satellites,” *Deutscher Luft- und Raumfahrtkongress DLRK,* 19-21 09 2023.

[3] T. Cziep, T. Stindl and T. Stäbler, “The Digital Concurrent Engineering Platform DCEP,” *72nd International Astronautical Congress (IAC),* 25-29 10 2021.

[4] M. Fugmann and S. Klinkner, “An Automated Constellation Design & Mission Analysis Tool for Finding the Cheapest Mission Architecture,” *34th AIAA/USU Annual Small Satellite Conference,* 01-06 08 2020.

[5] M. Ehresmann, G. Herdrich and S. Fasoulas, “An automated system analysis and design tool for spacecrafts,” *CEAS Space Journal 14,* pp. 327-354, 2022.

[6] M. Ehresmann and G. Herdrich, “Generic Spacecraft Design Prediction and Modelling,” *72nd International Astronautical Congress (IAC),* 25-29 10 2021.

[7] Astos Solutions GmbH, “Analysis, Simulation and Trajectory Optimization Software for Space Applications,” Astos Solutions GmbH, 2019. [Online]. Available: https://www.astos.de/products/astos. [Accessed 21 November 2019].

[8] ESA, “DRAMA - Space Debris User Portal,” [Online]. Available: https://sdup.esoc.esa.int/drama/. [Accessed 31 07 2023].

[9] C. Deil, T. Robitaille and D. Lang, “What is HEALPix?,” 21 January 2019. [Online]. Available: https://astropy-healpix.readthedocs.io/en/latest/. [Accessed 21 November 2019].

[10] B. Bowman, W. K. Tobiska, F. Marcos, C. Huang, C. Lin and W. Burke, “A New Empirical Thermospheric Density Model JB2008 Using New Solar and Geomagnetic Indices,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit,* 18-21 08 2008.

[11] J. R. Wertz, D. F. Everett and J. J. Piuschell, Space Mission Engineering : The New SMAD, Hawthorne, CA : Microcosm Press : Sold and distributed worldwide by Microcosm Astronautics Books, 2011.

[12] M. Ehresmann, “github,” 2023. [Online]. Available: https://github.com/aerospaceresearch/ESDC. [Accessed 29 08 2023].

[13] “Flask Documentation,” [Online]. Available: https://flask.palletsprojects.com/en/2.3.x/. [Accessed 01 09 2023].

[14] “Flutter - Build Apps for any Screen,” [Online]. Available: https://flutter.dev/. [Accessed 01 09 2023].

[15] “MongoDB - The Developer Data Platform,” [Online]. Available: https://www.mongodb.com/. [Accessed 01 09 2023].

[16] J. Petri, “Satellite formation and instrument design for autonomous meteor detection,” Institute of Space Systems, Stuttgart, 2022.

[17] S. Lutz, “First Design of the Satellite Mission OREUS - Observation of Re-entry Events Using Space 4.0,” Master's thesis at the Institute of Space Systems, Stuttgart, 2020.

[18] ArianeGroup, “RIT Thruster Family Performance Data,” 2023. [Online]. Available: http://www.space-propulsion.com/spacecraft-propulsion/propulsion-systems/electric-propulsion/images/rit-thruster-family-performance-data.jpg. [Accessed 29 08 2023].

[19] Enpulsion, “Nano,” 2023. [Online]. Available: https://www.enpulsion.com/wp-content/uploads/ENP2018-001.G-ENPULSION-NANO-Product-Overview.pdf. [Accessed 29 08 2023].

[20] J. Skalden, L. Ackermann, P. Upadhyay, G. Herdrich, S. Fasoulas, T. Schönherr, J. Gonzalez del Amo and D. Feili, “Electrostatic Probe Investigation of Very Low Power Arcjet VELARC in IRS and ESA-ESTEC Facilities,” *36th International Electric Propulsion Conference,* 15-20 09 2019.