CEDESK Documentation

Dominik Knoll

November 2019

1 Collaboration Tool

We analyzed tools referenced in the literature about concurrent engineering in conceptual design as practiced by space agencies. The result of this comparative analysis is compiled in the section 3, in Table 1.

The major differences between these tools are in the specialization on certain life cycle phases and in the focus and level of abstraction of parametric models. Models used in the conceptual design of space systems primarily describe the system's behavior; hence, the tools have little to do with geometry. In fact, [Fortin et al., 2017] found out that only around 20% of the parameters in conceptual design describe geometry. Differently, product data or Product Life Cycle Management tools have their strength in supporting design and manufacturing and hence the management of geometric models. Tools for Multidisciplinary Design Optimization (MDO) are agnostic to the nature of the parameters and their use as a means for analysis and optimization. Ultimately, there are SE tools, which are used to make models which describe the system decomposition structure and subsystem interactions.

For the purpose of the conceptual design for feasibility studies, we decided to collocate our tool in the group of concurrent conceptual design tools. This type of tools focus on connecting different engineering disciplines. Examples are: VirSat[1] from DLR, CDP[2] from RHEA, Valispace[3]. Based on the experience with existing data exchange tools, we derived the following key requirements for our own tool: user friendliness, easy synchronization, data compatibility, and integration with third-party engineering tools. Moreover, it serves the double purpose of supporting conceptual design and feasibility studies in our Concurrent Engineering Design Laboratory (CEDL) and enabling research on the concurrent design methodology.

We designed our tool to focus on this primary function: exchange parametric model information between discipline experts. Different from VirSat for example we decided not to include visualization of basic three-dimensional geometry. And different from CDP we decided not to use Excel™ as primary user interface for the interaction with the system model. To allow direct interaction with third-party engineering tools typically installed as desktop applications, we decided not to use a web interface, where as Valispace did.

State-of-the-art PLM tools allow to leverage knowledge accumulated in an organization by managing models in a way to facilitate their reuse. Such knowledge-based engineering approach is particularly advantageous when system designs can be generated based on models available for a product family [Prasad and Rogers, 2005]. The fact that the integration of knowledge management into product development tools remains challenging is confirmed by Chandrasegaran et al. [2013].

1.1 Software Architecture

Like any collaboration software, CEDESK is built according to a client-server architecture, as illustrated in Figure 1. The server part embodies the central model repository and consists of a relational database and the client part consists of a desktop application. The application was implemented in Java™in order to be able to be run on all major desktop platforms, such as Windows®, MacOS™, and Linux™. Multiple clients can connect to a server at the same time and work on the

[1] https://www.dlr.de/sc/en/desktopdefault.aspx/tabid-5135/8645_read-8374/
[2] https://www.rheagroup.com/cdp
[3] https://www.valispace.com/

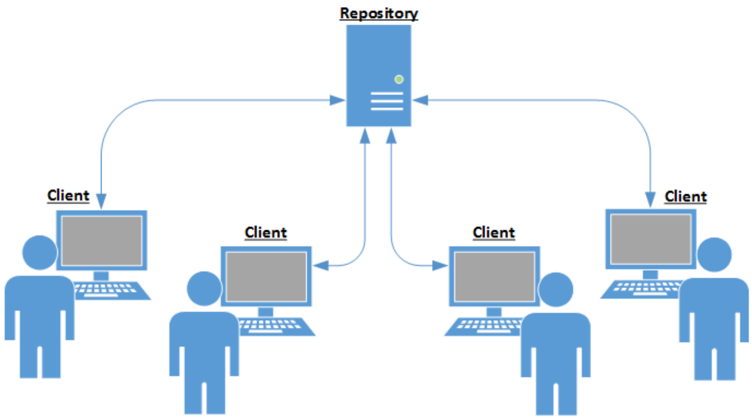


Figure 1: The client-server architecture of CEDESK

same model. Model synchronization is built on top of atomic database transactions. The data storage relies on a MySQL™ from Oracle [2016] database and the client application uses the Hibernate™ Object Relational Mapping from Redhat [2014] framework to store models in the database.

The user interface is built with JavaFX™ technology. Moreover, CEDESK makes use of various open-source libraries for logging, access to workbook files, spreadsheet-like user interface components, and handling of graph data structures.

1.2 CEDESK user interface

The client application is the primary user interface for the user to access the central study repository and to interact with the models. Similar to many collaborative design tools, with CEDESK, users load projects from the repository, then operate on a local working copy, and it can then be saved back to the repository when needed. Figure 2 shows a screenshot of the client application's main window.

At the top, the name of the current study, the logged-in user, and his active roles are displayed. According to the roles, a user is assigned; he can either view or modify a subsystem, its parameters, and external models.

The user interface consists of four major parts enabling the user to work with the system model (see numbers in Figure 2).

(1) Shows the structure tree for the systems hierarchical decomposition. The buttons allows users to add, modify, and delete model nodes. The screenshot above shows the model of the "demoSAT" spacecraft and its subsystems.

(2) This is the list of external models belonging to a model node. External models (files of third-party tools) can be attached, detached, and opened with the respective tool directly from there with the respective buttons. In the screenshot above, an Excel™ workbook "Orbit.xslm" is attached to the subsystem "Orbit" which was selected on the left.

(3) Shows the list of parameters belonging to a model node. The buttons on the bottom allows a user to add a new parameter, remove an existing parameter, and see the version history of a parameter.

(4) This is the area for parameter details, which also allows for immediate editing. In particular, this editor allows a user to create a link to another parameter or to set up the reference to external models.

There is the possibility to clone a study by exporting the full system model to an XML archive and then re-importing it with a new name. This can save a user time when building a new model based on similar conceptual design studies. Basic knowledge management capabilities are implemented in the form of a model library, with the possibility to store, search, and instantiate models from a component library as laid out in [Fortin et al., 2017].

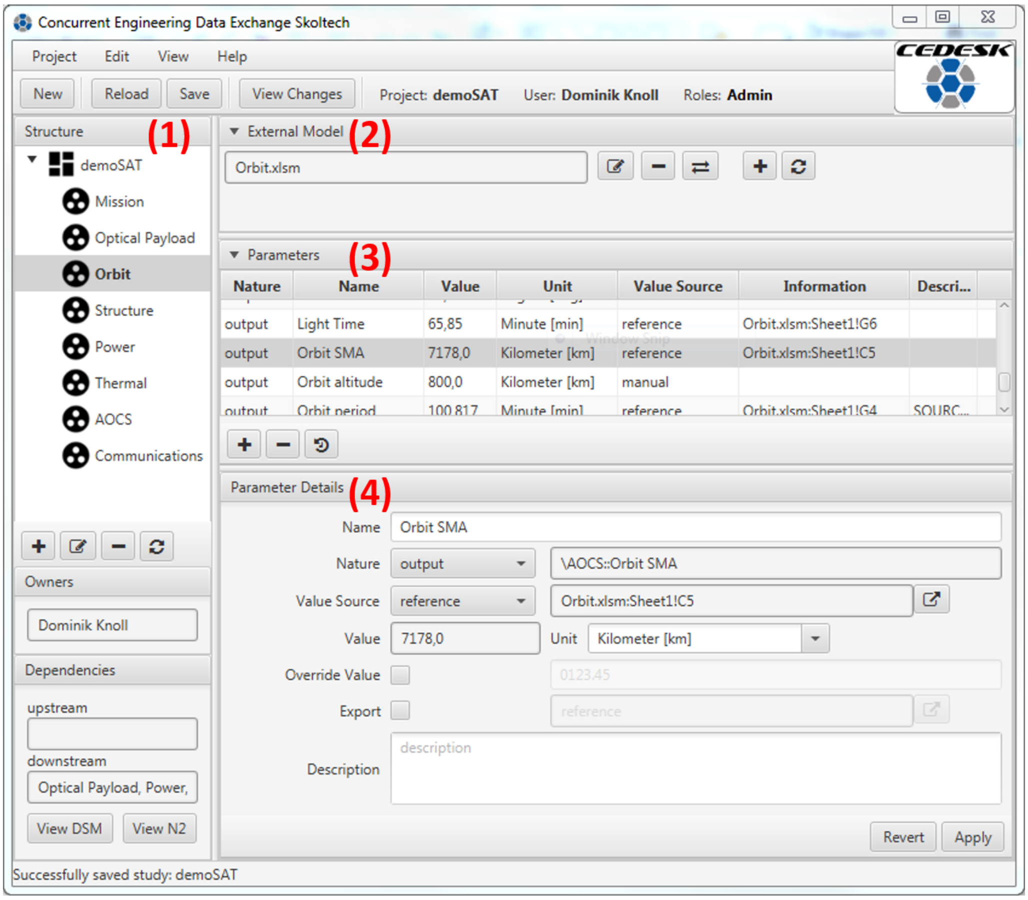


Figure 2: The main screen of CEDESK

A feature invisible to those designers who are using the tool, but of high relevance to researchers, is the detailed logging of user activities. The application logs each action such as loading and saving of models as well as modifications to the models' structure and parameters, along with all of the relevant meta data such as the related entity, time and user information. The log is stored in the same central database as the modeling information. This allows for a deeper analysis of the logs even after a design study is completed.

Upon close inspection of the various screenshots provided as examples it can be noted that the application appears to change its appearance. This is because the screenshots were made when running on different operating systems, which shown the application windows in different styles. This is also meant to demonstrate the cross-platform capability of CEDESK.

1.3 Modeling capabilities

The data model in CEDESK is structured similar to ECCS-E-TM-E-10-25A, as much as it concerns parametric system models. The primary model entities represent the system structure, its parameters, units of measures, users, and roles. Figure 3 describes the data structures, using the graphical notation of a UML class diagram.

A study is composed of a system model, which is a tree structure of model nodes. A node represents the parametric model of an engineering discipline or a system component. Each model node contains a set of parameters and a set of external models. External models encapsulate files made by third-party engineering tools. Parameters are of one of the following natures: input, internal, or output. All parameters have a numerical value and can be associated with a unit of measure. The value is obtained either from manual entry, from a link to another parameter, or a calculation based on other parameters.

Each model node, such as system, subsystem, element or component encapsulate a paramet-

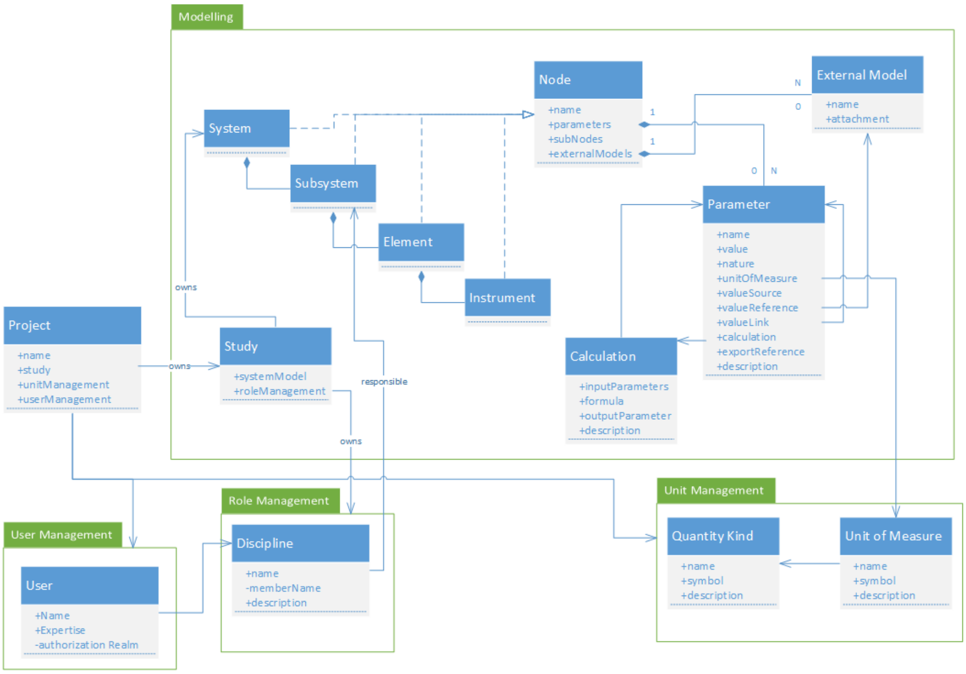


Figure 3: The data structures for integrated system models in CEDESK

ric model, with input and output parameters. Figure 4 shows how parametric models work in CEDESK: values of input parameters are fed into a calculation, a simulation or a human design decision and values for output parameters are produced.

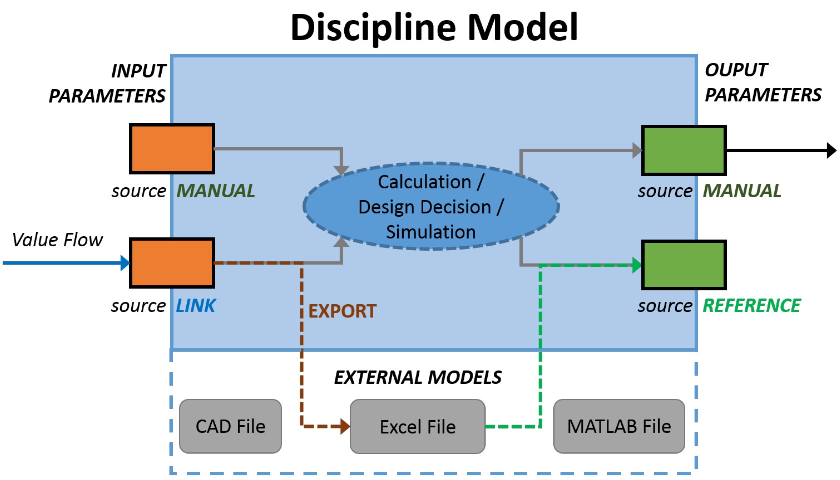


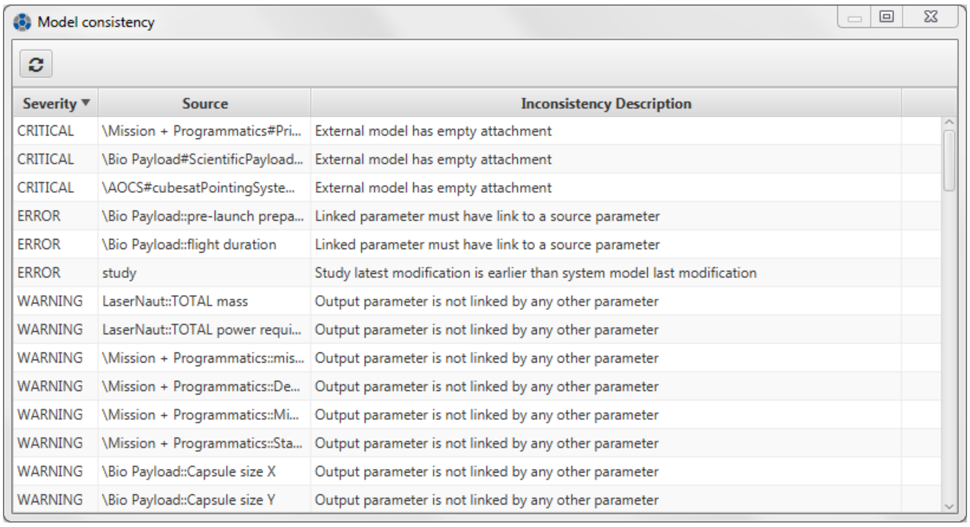
Figure 4: The structure and information flow in a parametric discipline model in CEDESK

An input parameter can obtain its value either from manual setting or from a link to another subsystem's parameter. Actually, only output parameters are visible to other subsystems and can be linked. An output parameter can obtain their value from setting it manually or from a reference to an external model. In the example shown in Figure 2, the parameter obtains its value from a reference to a specific cell from the Excel™ workbook "Orbit.xlsm:Sheet1!C5". Whenever the value of a parameter is obtained from a link or an external model, there is an option to override the value. This is useful at the beginning of the concurrent conceptual design, when a discipline engineer works with assumptions before being provided with calculated values by another discipline. Finally, a parameter can also export its value to an external model (e.g. to a cell of an Excel™ spreadsheet).

1.4 Consistency Check

Making sure that the integrated design model does not contain inconsistencies is a major concern. Some changes to the model that would introduce inconsistencies are blocked by the user interface right away. For example, links can only be established to output parameters, and removing a parameter that is already linked is restricted. The tool also ensures that the units of measures of two linked parameters will always correspond.

For incomplete models or inconsistencies that can occur in the model, the tool offers the ability to run a model check. The issues of the model are categorized by severity: critical, error, warning. An example of the result of such a check is shown in Figure 5.



Severity	Source	Inconsistency Description
CRITICAL	\Mission + Programmatics#Pri...	External model has empty attachment
CRITICAL	\Bio Payload#ScientificPayload...	External model has empty attachment
CRITICAL	\AOCS#cubesatPointingSyste...	External model has empty attachment
ERROR	\Bio Payload::pre-launch prepa...	Linked parameter must have link to a source parameter
ERROR	\Bio Payload::flight duration	Linked parameter must have link to a source parameter
ERROR	study	Study latest modification is earlier than system model last modification
WARNING	LaserNaut::TOTAL mass	Output parameter is not linked by any other parameter
WARNING	LaserNaut::TOTAL power requi...	Output parameter is not linked by any other parameter
WARNING	\Mission + Programmatics::mis...	Output parameter is not linked by any other parameter
WARNING	\Mission + Programmatics::De...	Output parameter is not linked by any other parameter
WARNING	\Mission + Programmatics::Mi...	Output parameter is not linked by any other parameter
WARNING	\Mission + Programmatics::Sta...	Output parameter is not linked by any other parameter
WARNING	\Bio Payload::Capsule size X	Output parameter is not linked by any other parameter
WARNING	\Bio Payload::Capsule size Y	Output parameter is not linked by any other parameter

Figure 5: Inconsistency check in the integrated design model

1.5 Collaboration

To enable multiple users work on a project, CEDESK also furnishes a user management feature which allows for the assignment of users and roles. A quick turn-around in collaboration is facilitated by notifications to the user, whenever changes have been stored to the model repository. Changes made by other team members or the user's own unsaved changes can be reviewed in a dedicated window (see Figure 6).

Whenever a user stores the system model, the tool not only saves the latest state of it, but also keeps track of each modification. All changes to the structure of the model and parameters are recorded and a full version history is kept. This allows to reconstruct the complete state of the model back to any stored version as part of posterior analysis. The tool provides functionality to tag the state of the system model at any point in time. The user can also restore any tagged version from the history.

To distinguish the responsibilities of study participants over parts of the system, the tools implements access control based on roles Figure 7. Roles can be created for all disciplines. Model nodes can be associated with roles, and roles associated with users.

1.6 Process Guide

The aim is to assist the user in following the proposed design process. Therefore we included the process model and description into the tool, such that it is available for consultation during the design process. The process guide in CEDESK, as shown in Figure 8, allows for the navigation through the major design steps. This should help newcomers to easily learn the about the process, and give all team members awareness about the current activities.

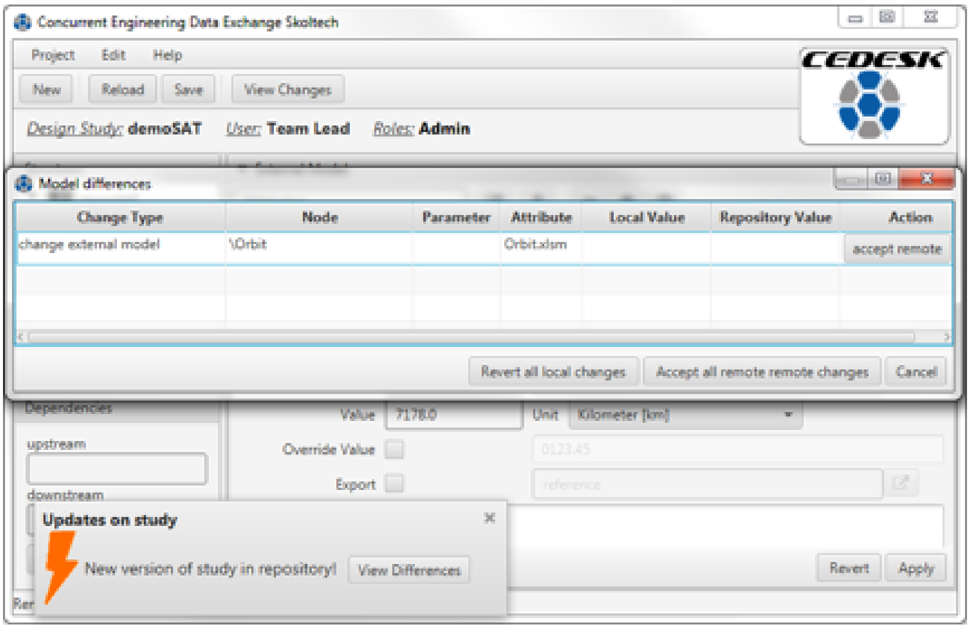


Figure 6: Change notification and possibility to review changes

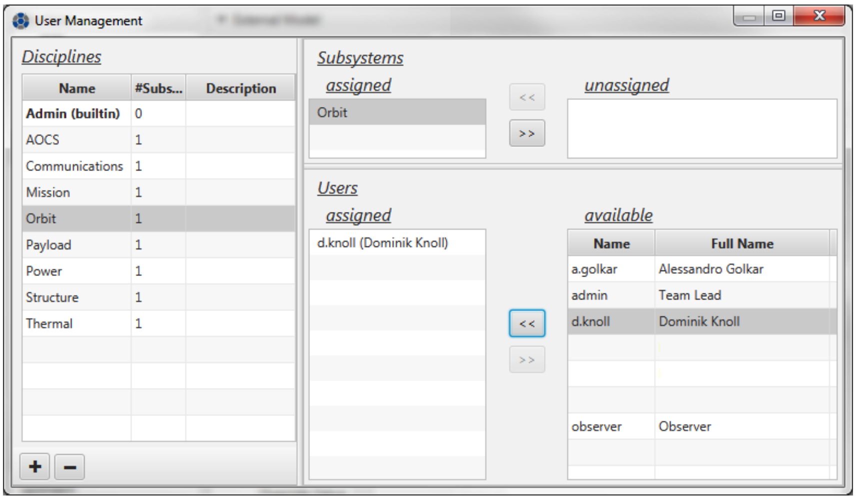


Figure 7: The discipline and user management

1.7 Coordination

An important goal of CEDESK is to support the team leader in coordinating the design effort. A graphical representation of the dependencies is given in the form of an N^2-Diagram (see Figure 9). The boxes in the diagonal represent the subsystems, and the lines indicate the parameter links. The arrows at the end of the lines show the direction of information flow. The width of a line reflects the number of parameter links, and at the bend in the lines the names of the parameters are shown.

The tool allows users to interact with charts by clicking on single arrows and subsystems to highlight them (see red border). Moreover the user can change the sorting of the subsystems via the buttons at the top, and export a snapshot of the chart as a picture. Charts not only give a view of the static parameter links between the subsystems, but also their status. The arrow tip changes is shown in blue in the case that the value of the output parameter is explicitly overwritten at the input parameter. The arrow line turns blue when the output value is not yet propagated to the input parameter.

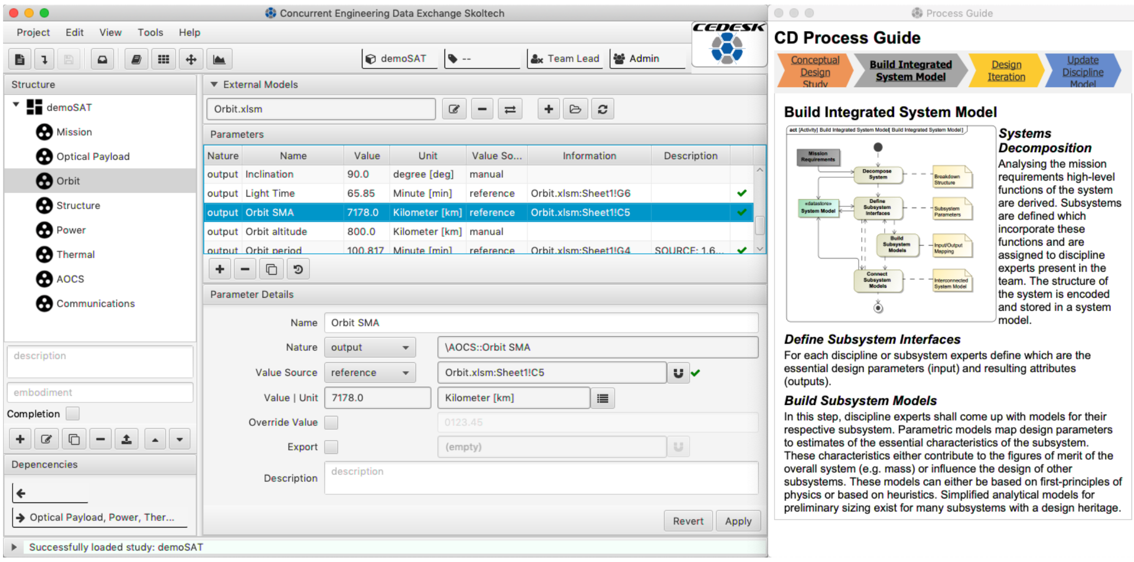


Figure 8: The process guide integrated into CEDESK

The user can choose to have the view updated automatically at any change made to the model in the repository. This allows a team lead or moderator to use this view to observe the design process and the propagation of changes in real time.

Another view in the application allows the user to visualize the dependencies among subsystems in the form of a Design Structure Matrix (DSM). The cells of the matrix in Figure 10 show the number of parameters linked from the subsystem in that row to the subsystem of that column. The user can interact with the chart by using the mouse. Hovering over a cell lists the names of the parameters, and by clicking on a cell it will highlight the two involved subsystems. The interface offers the user the possibility to run a DSM clustering/sequencing algorithm. The results are shown on the same view, just through the re-arranging of the discipline names and marking the clusters with a square.

A moderator or team lead can used this sequence to synchronize the disciplines in order minimizes rework.

1.8 Tradespace Exploration

The conceptual design elaborated during a conceptual design study is commonly evaluated according to a few characteristics, or Figure of Merits. In the case that the system to be designed can be associated to a bigger family of products (e.g. communication satellites), the new design is compared with other planned or existing solutions. A very powerful tool for this comparison is the tradespace chart. The integration of tradespace exploration and parametric modeling is not available in any other tool.

CEDESK has the related functionality of tradespace exploration built-in. The respective screen is shown in Figure 11. This part allows a user to create tradespaces, define the respective figures of merit (left upper part), and visualize tradespace charts (left side). The screenshot shows the data points, that were imported for the comparison of car designs.

Data points represent designs that are characterized by a name, and values for a set of FOMs, as well as the epoch (year) the design was made. During the import of the data points from spreadsheets in CSV or XLS format, the user can choose the meaning given to each column, whether it contains the name, a FOM or the epoch.

The definition of the figures of merit allows for the selection of a figure of merit optimality criteria. In the example, the FOM "horsepower" is set to maximal. A FOM for cost, would be set to minimal. On the top left, the user can choose which FOM to shown on which axis of the chart. A line connects the data points which form the Pareto-front among the known designs.

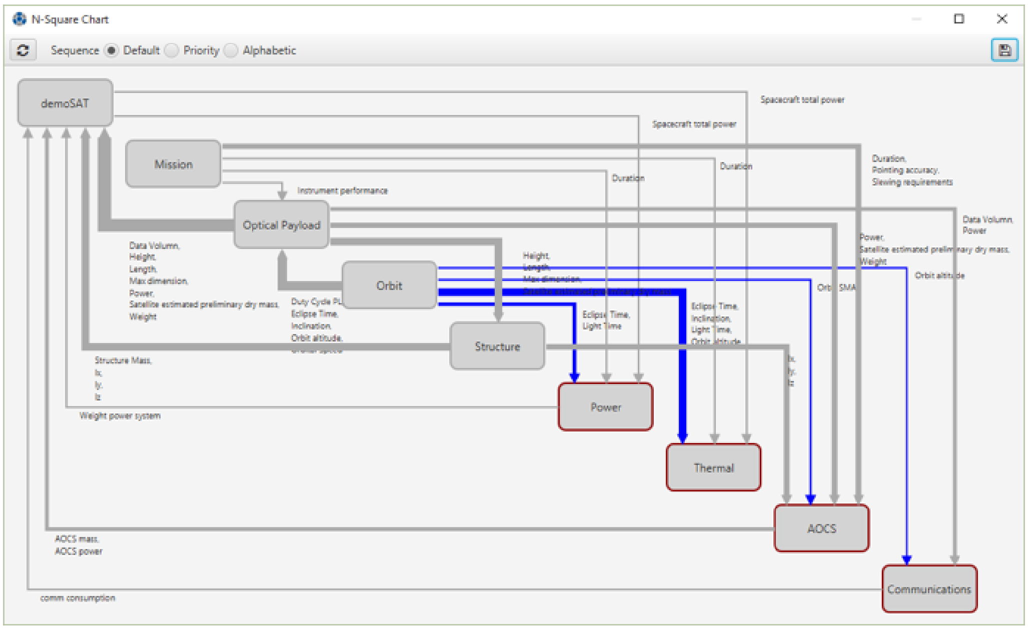


Figure 9: The subsystem dependencies visualized in a N^2-Diagram

FOMs can also be linked to parameters of the design model. For example, the parameter of engine size and horsepower of the design model can be connected to the figures of merit of the cars tradespace. In this way, any state of the current design can be compared to all existing designs in terms of these FOMs. This allows for immediate feedback between the concurrent design and analysis of competitive products.

1.9 Distinctive features of CEDESK

This application was designed and developed to provide the necessary functionality, which supports the user throughout the concurrent design study. Like all other available tools, CEDESK allows teams to store and share the integrated parametric model. Besides these, it comes with the following features which are not available elsewhere:

Consistency check Is an automated analysis which reveals possible mistakes in the parametric models, and helps the team to make any corrections accordingly.

Coordination With the visualization of dependencies as interactive N^2 charts user have the ability to see the changes as they are propagated. The automatic generation of the DSM based on the parameter links allow the user to control the order in which the various disciplines make updates.

Tradespace Explorer This tool can import and visualize competitive solutions in terms of key system characteristics (Figure of Merit). Thereby the users have the ability at any moment of the design study to compare the design with the best-in-class.

These features are actual innovation in the area of tools for concurrent conceptual design.

2 Summary

This chapter presents the comprehensive methodology for model-based co-located conceptual design. The methodology (MoCoDeM) was devised from literature review and expert survey, as well as the author's learning from conducting conceptual design studies with a co-located team and within a Model-Based Systems Engineering environment. It included the description of formalized models of the 5 pillars: facility, team, model, process, tool.

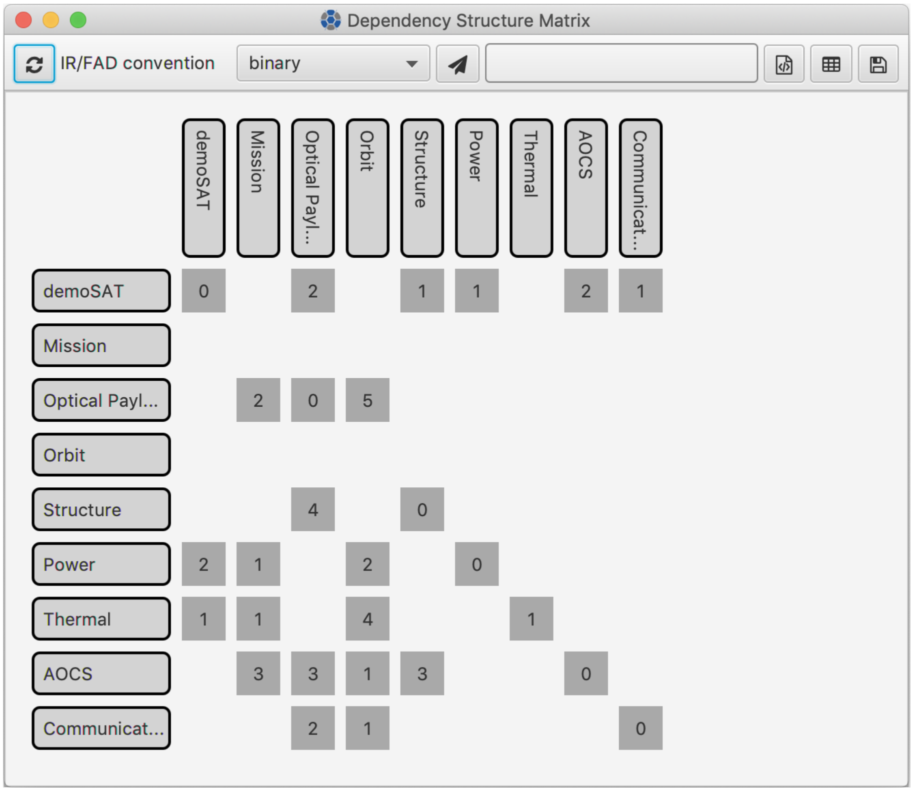


Figure 10: A visualization of the subsystem dependencies in the form of a DSM

In the following chapters we will describe the verification of MoCoDeM interviews with experts from estabblished CDFs as well as perform initial validation of this methodology through case studies.

3 Tool Comparison

We analyzed and compared the tools in use CDFs for conceptual design of space missions. The tools can be split in two categories: those focused on conceptual parametric modeling (see Table 1), and others (see Table 2). This comparison is not meant to be exhaustive, but it covers the tools reported in a survey and that are not proprietary to the facility.

Table 1: Comparison of Tools, I

Group	CONCURRENT CONCEPTUAL DESIGN TOOLS					
Tool	IDM	VirSat 4[4]	OCDT	CDP 4[5]	Valispace[6]	CEDESK[7]
References	Bousquet et al. [2005]	Schaus et al. [2010], DLR [2016]	ESA [2014], Braukhane [2015]	Fijneman and Matthyssen [2010], RHEA-Group [2019]	Valispace [2017]	Knoll and Golkar [2018]
Aspect						
Multi-User Support	Yes	Yes	Yes	Yes	Yes	Yes
Lifecycle Phase Focus	conceptual design	conceptual design	conceptual design	conceptual design	conceptual design	conceptual design
Parametric modeling Focus	behavior	behavior and geometry	behavior	behavior	behavior	behavior
Version Control	Limited	Yes	Yes	Yes	No	Limited
Primary User Interface	Excel™	Own client	Excel™	Own client, Excel™	Own Web	Own client
Integration With 3rd Party Tools	No	No	Yes	Yes	Yes	Limited
Availability	ESA community	Open Source	ESA community	Open Source	Commercial	Open Source

Table 2: Comparison of Tools, II

Group	PLM TOOL	MDO TOOLS		SYSTEMS ENGINEERING TOOLS	
Tool	ENOVIA[8]	Model Center[9] 11.2	OpenMDAO[10] 2.6	Magic Draw[11]	Open MBEE[12]
References	Dassault Systems [2016]	Phoenix Integration [2015]	Gray et al. [2019]	NoMagic [2015]	Kulkarni et al. [2016], NASA JPL [2016]
Aspect					
Multi-User Support	Yes	Yes		Yes	Yes
Lifecycle Phase Focus	design, manufacturing	design	design	conceptual design	design
Parametric modeling Focus	geometry and geometry	analysis and optimization	analysis and optimization	description	description
Version Control	Yes	No	No	Limited	Yes
Primary User Interface	Own client	Own client	Python Code	Own client	MagicDraw
Integration With 3rd Party Tools	Yes	Yes	No	Yes	Yes
Availability	Commercial	Commercial	Open Source	Commercial	Open Source

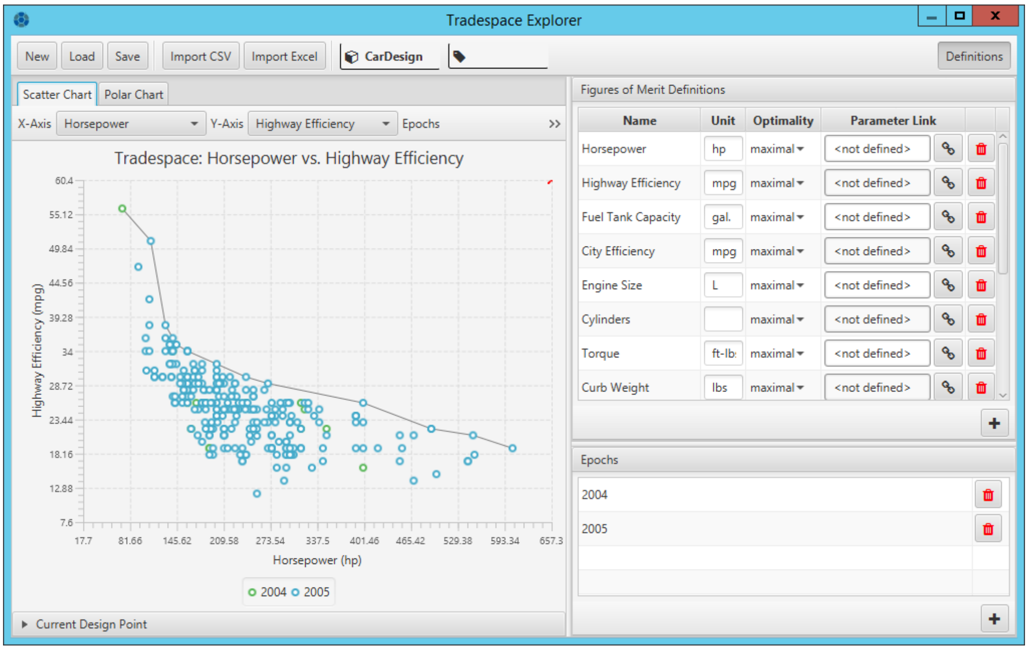


Figure 11: The tradespace explorer in CEDESK

4 Software Resources

4.1 CEDESK

CEDESK stands for Concurrent Engineering Data Exchange Skoltech.

This tools was conceived, designed and implemented by the author and his colleague Nikolay Groshkov.

It was released in July 2017 as open source under the Apache License.

4.1.1 General Concept

CEDESK is a tool to facilitate co-located collaborative model-based conceptual design for complex engineering systems. This type of tool is also known as data exchange for concurrent engineering studies. Multidisciplinary design teams can use CEDESK to facilitate their work together by building shared parametric models of their system of interest.

4.1.2 Use and Contribution

Installation packages of the tool are available for Windows, MacOS and Linux on the website `https://cedesk.github.io`. Also available there is an introductory users guide, as well as a guide for developers. The source code is published on GitHub `https://github.com/cedesk/data-exchange`.

4.1.3 Features

CEDESK does not duplicate the functionality of discipline-specific engineering tools, but rather integrates them with a shared parametric system model for easy exchange among engineering disciplines.

This tool supports conceptual design with the following features:

- Integrated parametric system model and internal links
- Simple and user-friendly graphical interface
- Multiple users working on a same model
- Centralized repository, local working copy
- Users of different disciplines work on distinct subsystems
- Easy resolution of conflicting changes
- Integration with calculation spreadsheets

Furthermore, the tool distinguishes itself from similar tools by the following unique features in support for a structured design process:

- Online help with a process guideline
- Automatic visualization of change propagation as N^2-Diagram
- Calculation of optimal discipline sequence with DSM algorithms

4.1.4 Compatibility

The software is compatible and was last used with:

- Java Development Kit, version 1.8.0_202
- Maven, version 3.3.9
- MySQL Community Server, version 5.7.25 (also tested with version 8.0.16)

4.2 jDSM

jDSM is a Java library for representing and analyzing Design Structure Matrix. It was originally made to analyze any Java software with regards to modularity.

This library was found as open source project on Sourceforge http://jdsm.sourceforge.net/index.html. It was developed and published in 2008 by Roberto Milev, as part of his master's thesis at the Technology Innovation Management program at Carleton University, Ottawa, Canada. Since then no further development happened.

For the inclusion of DSM algorithms into CEDESK, this library was adapted and is available on GitHub https://github.com/cedesk/jdsm/.

This code was last used with Java Development Kit 8.

4.3 Matlab DSM

The MATLAB® macro for analyzing DSMs was found at http://www.dsmweb.org/en/dsm-tools/research-tools/matlab.html.

We adapted the original code for our research and made it available on GitHub https://github.com/djknoll/dsm_matlab/.

This code was last used with MATLAB® version 2018b.

5 Process

Work in progress ...

Glossary

CDF Concurrent Design Facility. 9

CDP Concurrent Design Platform. 1

CEDESK Concurrent Engineering Data Exchange Skoltech. 2–8, 11–13

CEDL Concurrent Engineering Design Laboratory. 1

DLR Deutsches Zentrum für Luft- und Raumfahrt, German Aerospace Center. 1

DSM Design Structure Matrix. 7–9, 12, 13

FOM Figure of Merit. 7, 8

MBSE Model-Based Systems Engineering. 8

MDO Multidisciplinary Design Optimization. 1

MoCoDeM Model-based Co-located Conceptual Design Methodology. 8, 9

PLM Product Life Cycle Management. 1

SE Systems Engineering. 1

UML Unified Modeling Language. 3

VirSat Virtual Satellite. 1

References

P W Bousquet, J Sire, and F Vigeant. Concurrent Engineering at CNES. *56th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, pages 1–11, 2005. doi:10.2514/6.IAC-05-D1.3.06. URL http://arc.aiaa.org/doi/10.2514/6.IAC-05-D1.3.06.

Andy Braukhane. Evolution of the European Concurrent Design Tool for Space-based System-of-Systems studies. In *AIAA SPACE 2015 Conference and Exposition*, pages 1–12, Reston, Virginia, 8 2015. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-334-6. doi:10.2514/6.2015-4438. URL http://arc.aiaa.org/doi/10.2514/6.2015-4438.

Senthil K. Chandrasegaran, Karthik Ramani, Ram D. Sriram, Imré Horváth, Alain Bernard, Ramy F. Harik, and Wei Gao. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design*, 45(2):204–228, 2 2013. ISSN 00104485. doi:10.1016/j.cad.2012.08.006. URL https://linkinghub.elsevier.com/retrieve/pii/S0010448512001741.

Dassault Systems. ENOVIA, 2016. URL https://www.3ds.com/products-services/enovia/.

DLR. Virtual Satellite 3, 2016. URL https://software.dlr.de/p/virsat/home/.

ESA. Open Concurrent Design Tool, 2014. URL https://ocdt.esa.int/.

M Fijneman and A Matthyssen. Application of Concurrent Design in Construction, Maritime, Education and Other Industry Fields. In *Proceedings of the 4th International Workshop on System & Concurrent Engineering for Space Applications (SECESA 2010), Lausanne, Switzerland*, 2010.

Clément Fortin, Grant McSorley, Dominik Knoll, Alessandro Golkar, and Ralina Tsykunova. Study of Data Structures and Tools for the Concurrent Conceptual Design of Complex Space Systems. In *IFIP 14th International Conference on Product Lifecycle Management 9-12 July 2017, Seville, Spain*, pages 601–611. Springer, Cham, Seville, 2017. doi:10.1007/978-3-319-72905-3_53. URL http://link.springer.com/10.1007/978-3-319-72905-3_53.

Justin S. Gray, John T. Hwang, Joaquim R. R. A. Martins, Kenneth T. Moore, and Bret A. Naylor. OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, pages 1–30, 3 2019. ISSN 1615-147X. doi:10.1007/s00158-019-02211-z. URL http://link.springer.com/10.1007/s00158-019-02211-z.

Dominik Knoll and Alessandro Golkar. A coordination method for concurrent design and a collaboration tool for parametric system models. *Concurrent Engineering*, 26(1):5–21, 3 2018. ISSN 1063-293X. doi:10.1177/1063293X17732374. URL http://journals.sagepub.com/doi/10.1177/1063293X17732374.

Tejas Kulkarni, Kevin DeBruin, Adam Nelessen, Kevin A. Reilley, Russell Peak, Stephen J. Edwards, and Dimitri N. Mavris. A Model Based Systems Engineering Approach Towards Developing a Rapid Analysis and Trades Environment. In *AIAA SPACE 2016*, pages 13–16, Reston, Virginia, 9 2016. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-427-5. doi:10.2514/6.2016-5472. URL http://arc.aiaa.org/doi/10.2514/6.2016-5472.

NASA JPL. Open Model Based Engineering Environment, 2016. URL http://www.openmbee.org/.

NoMagic. MagicDraw, 2015. URL https://www.nomagic.com/products/magicdraw.

Oracle. MySQL, 2016. URL https://www.mysql.com/.

Phoenix Integration. ModelCenter, 2015. URL http://www.phoenix-int.com/product/modelcenter-integrate/.

Brian Prasad and Jeff Rogers. A Knowledge-Based System Engineering Process for Obtaining Engineering Design Solutions. In *Volume 3: 25th Computers and Information in Engineering Conference, Parts A and B*, volume 2005, pages 477–488. ASME, 2005. ISBN 0-7918-4740-3. doi:10.1115/DETC2005-85561. URL http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1588184.

Redhat. Hibernate ORM, 2014. URL http://hibernate.org/orm/.

RHEA-Group. CDP™ – Concurrent Design Platform, 2019. URL http://www.rheagroup.com/products/cdp/.

Volker Schaus, P Fischer, D Lüdtke, A Braukhane, O Romberg, and A Gerndt. Concurrent engineering software development at german aerospace center-status and outlook. In *4th International Workshop on System & Concurrent Engineering for Space Applications*, 2010.

Valispace. Valispace, 2017. URL http://www.valispace.com/.