CONCURRENT ENGINEERING SOFTWARE DEVELOPMENT AT GERMAN AEROSPACE CENTER - STATUS AND OUTLOOK -

Schaus, V.[1], Fischer, P.[1], Lüdtke, D.[1], Braukhane, A.[2], Romberg, O.[2], Gerndt, A.[1]

[1] German Aerospace Center (DLR), Simulation and Software Technology
Lilienthalplatz 7, 38108 Braunschweig, Germany
Email: volker.schaus@dlr.de
Email: philipp.fischer@dlr.de
Email: daniel.luedtke@dlr.de
Email: andreas.gerndt@dlr.de

[2] German Aerospace Center (DLR), Institute of Space Systems
Robert-Hooke-Str. 7, 28359 Bremen, Germany
Email: andy.braukhane@dlr.de
Email: oliver.romberg@dlr.de

ABSTRACT

Virtual Satellite is an internal research project within the German Aerospace Center (DLR) and has been running for more than three years. The main objective is to support the newly equipped Concurrent Engineering Facility (CEF) at the DLR Institute of Space Systems in Bremen with a modern software environment. During the project a software solution has been developed to support the design process and the simulation of space systems in an efficient way. It is designed as modular framework to cover early conceptual design and mission requirement studies (Phase 0/A) which are more and more performed in concurrent engineering environments. By using a component model repository, the framework is capable to import and export models based on Simulation Model Portability (SMP2) standard which can be used for simulation in succeeding design phases (Phase B). The underlying consistent data model assures that the system requirements are met at all times.

INTRODUCTION

The Concurrent Engineering Facility (CEF) [1] at DLR's Institute of Space Systems in Bremen became operational in 2008 and has been well received by engineers and users. Since then, 15 internal and external studies have been worked out, mainly conceptual designs, mission requirement and feasibility studies for future DLR missions including satellite configuration, launcher performance optimization, and exploration mission architectures. The related outcomes exceeded the expectations and objectives which have been defined each time prior to the start of the concurrent engineering studies.

Since the beginning in 2008, the DLR Department of Simulation and Software Technology was deeply involved in the set-up process. Besides selecting hardware and equipment for the facility, the main focus was on concurrent engineering practices and how to support them efficiently with software. In a first step, DLR adopted ESA's spreadsheet-based design software to be quickly operational. Over the time it became obvious that Excel as a design tool has several limitations [2]. Based on this realization, a new software was developed as part of the DLR internal research project Virtual Satellite to overcome the drawbacks and to replace Excel as design tool for early designs.

The other main objective of the project Virtual Satellite is to provide a design platform for subsequent design phases, in which nowadays simulations play an increasingly important part [3]. The idea is to use the additional value of simulations as early as possible in the design process and to offer a way to quickly build-up an overall satellite simulation. These two aspects indicate that the developments cover various research fields and this initiative is grouped under the term Virtual Satellite within DLR. The important areas are systems engineering, concurrent engineering, model-driven engineering, and simulation of spacecraft systems and will be discussed to explain the topical context of Virtual Satellite in more detail.

STATE OF THE ART

Systems Engineering

Satellites and spacecrafts are complex to design and operate. They involve many different disciplines from engineering over budget and project management to logistics. Usually, space projects are handled by multiple partners of different nationalities based in different locations. This creates additional needs for team, tool, and work process coordination. The various disciplines involved in spacecraft design have interdependencies. Changing a design parameter in one discipline might have strong effects on one or more other disciplines. These effects are primarily managed on system level and involve discussions between the respective domains.

Model-driven Engineering

Models play an important role in systems engineering. They are used as abstract representation of the reality and help to show and comprehend the complexity of the system. In recent years, the promising approach of model-driven architecture (MDA) in software engineering has been applied to systems engineering as well. In software engineering, MDA brought several improvements. MDA starts with a platform independent, conceptual model. It raises the design to a new abstraction level where designers can focus on logic, behaviour, and functionality of the software product before dealing with implementation issues. This model can be used throughout the whole product life-cycle from design to maintenance and potentially be re-used in future projects. It also provides a consistent design basis at all times. This helps to shorten the time-to-market of products, which is a general demand in all companies [4]. Another interesting aspect of model-driven architecture is that some succeeding tasks in the development process can be done automatically, e.g. the source code generation for a specific programming language from the platform independent model. It is desirable to take the advantages of MDA in software engineering across to systems engineering. Currently the multiple disciplines involved in an engineering design process create their own specific models, e.g. a geometry model, structural model, thermo dynamical model, power model and control model. Detailed examinations are performed with these models. Over the time, many disciplines have developed standard software tools in their domain, or use commercial products such as MATLAB/Simulink for control design or Satellite Tool Kit (STK) for orbit analysis. It becomes very difficult, time-consuming and error-prone to exchange the results between the disciplines and to optimize the design. Usually many iteration cycles are necessary to come to a design which satisfies all the requirements. The idea behind model-driven engineering is to have a system model superior to the discipline specific models in analogy to the abstract, platform independent model in MDA. The models needed by specific engineering disciplines can be derived by model transformations from the system model. This process is comparable to the automatic code generation in the MDA approach.

Design Phases and Simulation

The development process of spacecraft is divided in development phases, Phase 0/A, B, C, D, the operational Phase E and the disposal Phase F [5]. The main focus of Virtual Satellite lies on the development phases. Concurrent engineering is generally used in Phase 0 and A only. At the end of each phase there are reviews. The development process is document based. This means that the end of each design phase is marked with a review meeting and a written report capturing the results of the preceding phase. The next phase basically starts all over again based on the outcome of the last review. This denotes an interruption in the design process. Sometimes such a break is inevitable, e.g. when a project is handed over from a space agency to industry partners. However, in order to shorten the design time and assure consistency over the development process it is desirable to take as much information across from preliminary design phases and re-use the work that was done before.

Spacecraft systems engineering follows among others a classical V-Model. All design requirements are elaborated first, then fabrication, component testing, and their integration takes place. Finally, system tests are performed to qualify the whole satellite. The use of simulations offers a significant short-cut in the development process. The aim is to establish an overall system simulation and verify and validate the design with it. Later, the simulation is re-used again to qualify the actual physical system. The endeavour is to have simulations available as early as possible, preferable even in the concurrent engineering phase at the very beginning [6].

Concurrent Engineering

Concurrent engineering is a relatively new design approach in systems engineering. It has been used in the space domain for over a decade now. Starting from ESA's Concurrent Design Facility (CDF) [7], more and more organisations, institutes and industry partners in Europe have established such collaborative design centers [8][9][10]. Concurrent engineering has brought great improvement to early mission design and concept studies. It brings a team of experts together and allows for quick iterations and exchange of design data to find a strong baseline concept. The collaboration makes everyone aware of design changes and assures that everyone is working with the latest design parameters, thus bringing consistency to the design. This new design philosophy cuts the time needed for a pre-phase A studies from over half a year down to several weeks only which reduces the cost significantly [11].

Concurrent engineering consists of five key elements [7][12]:

- The concurrent engineering process
- The interdisciplinary team of experts
- The integrated design model
- The concurrent engineering facility
- The soft- and hardware infrastructure

The Virtual Satellite project addresses mainly two of these points, the integrated design model and software to support the process. The integrated design model collects and holds all design information on system level. It is used to exchange design parameters between disciplines and ensures consistency. When concurrent engineering was first introduced, ESA had developed a spreadsheet-based design tool of the same name – Integrated Design Model (IDM). Here, engineers work in discipline specific workbooks and bring together individual design parameters to form an integrated system model. Virtual Satellite aims to completely replace the IDM tool with modern software technologies. It is designed to support the concurrent engineering process in an efficient way and to remove drawbacks of spreadsheet-based design tools.

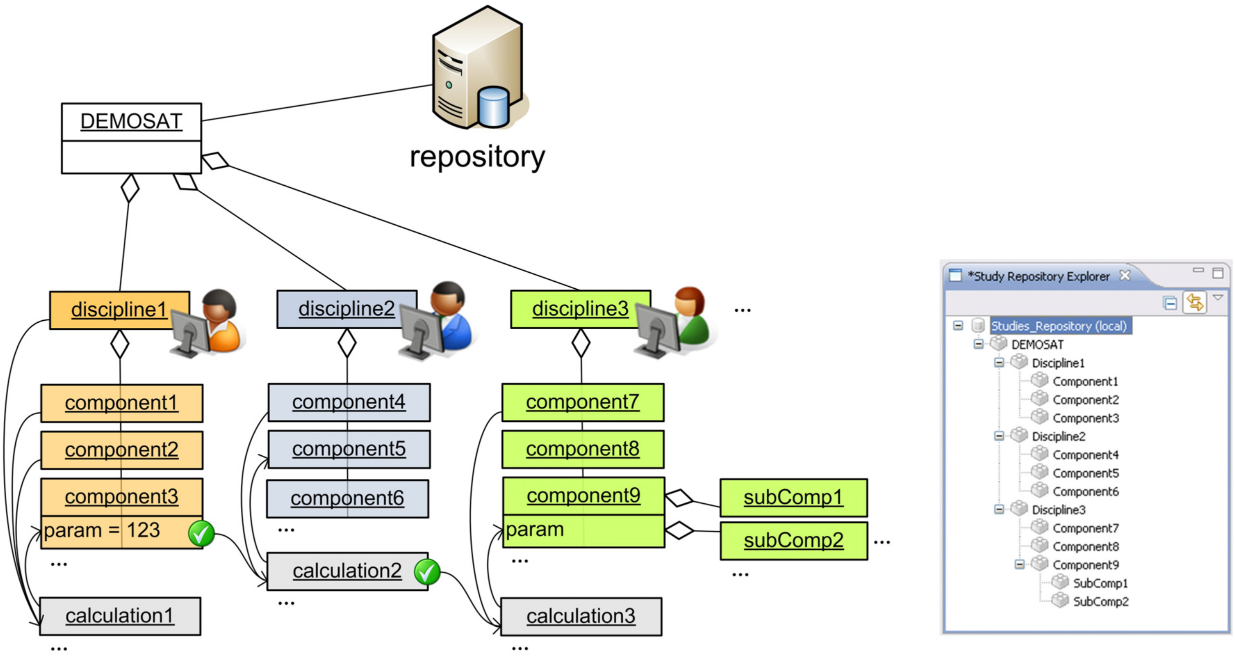
STATUS AND ACHIEVEMENTS

For its main purpose of being a supporting design tool in the CEF, Virtual Satellite aims to replace the well established Integrated Design Model (IDM) from ESA. For compatibility reasons, special care was taken to allow a mapping of IDM to the data model of Virtual Satellite. During the studies at the CEF, the engineers learned to use the Excel-based IDM tool and discovered its features and possibilities. The functionalities were closely examined and the process was analysed. The gathered information was turned into a first set of requirements for a new software to replace Excel as the design environment. Over time, Virtual Satellite grew towards a modular software framework which is easily extendable using plug-in mechanisms. Therefore, it has evolved into an integration platform for related projects.

Features

The core element of the Virtual Satellite software is the underlying data model to represent aspects of satellite design offering the necessary flexibility and extensibility. It is based on the data models developed at the TU Munich [13][14] and has been extended in several ways. The model uses a object-oriented approach. It represents both static and dynamic behaviour in order to be used for simulations and later phases of the design. Abstract elements can be defined in the model to create a semantically correct model first and then filling it with detailed information during the design process. Like the TU Munich's concept, Virtual Satellite's data model uses a hierarchical approach which can be displayed as a top-down tree. However, the data model is also designed to allow for discipline specific views, e.g. it is possible to view all objects related to power management and consumption.

Fig. 1 shows a typical set-up of the Virtual Satellite and the usage of its data model during Phase A in the CEF. The study is hosted on a central repository server. Every expert responsible for one subsystem of the satellite runs one instance of the Virtual Satellite installed on a local workstation and connects to the repository server. The server itself manages the study using a version control system and each client accessing it can contribute and receive data from it. This approach is useful in several ways. Firstly, it enables each expert to control when to contribute data to the team as well as it allows deciding when to receive changes from the central repository. Secondly, it allows members of the study to work remotely or offline, accessing the repository later and even via internet if needed. Additionally the repository



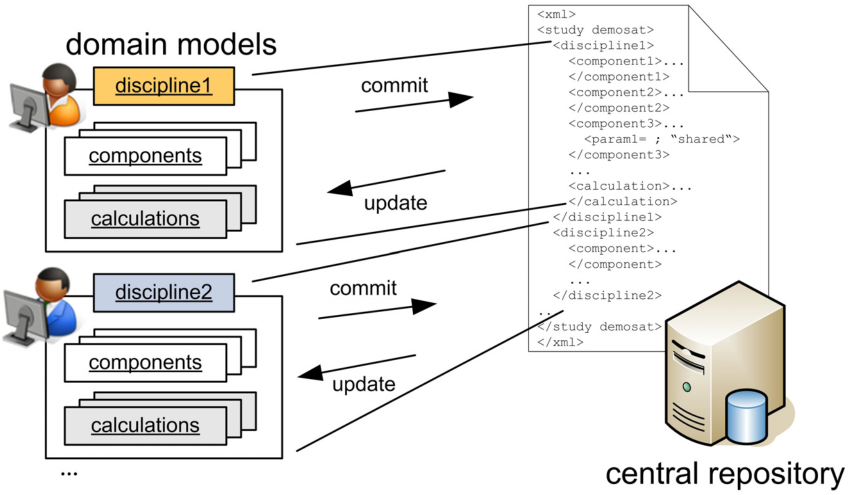
*Study Repository Explorer

- Studies_Repository (local)
  - DEMOSAT
    - Discipline1
      - Component1
      - Component2
      - Component3
    - Discipline2
      - Component4
      - Component5
      - Component6
    - Discipline3
      - Component7
      - Component8
      - Component9
        - SubComp1
        - SubComp2

Fig. 1. Schematic representation of a Phase A data model built-up in a concurrent engineering session; Screenshot of the tree-view of the study in an explorer view

tracks the progress of the study and the published changes making the design process traceable. It offers the possibility to step back to any point of the design, e.g. after realising design flaws. Working on the study, each user can insert new components by starting with an empty component or use a standard system component from a library, the so called component repository, which is a second database (not illustrated) besides the already mentioned repository. The number of units that can be added per discipline is not limited. Each system component can contain subcomponents representing refinements of the parent component. Again, the number of nested elements is not limited as well as the number of nesting levels. It is self-evident that in most cases, system component entries represent actual physical components in the satellite design. So the expert for power systems will create system components like solar array and accumulator in his sub-tree. Each unit has parameters like mass, power consumption and dimensions which can be freely edited and new parameters can be added as they are needed. Incorporating a basic authorization mechanism, each instance of the Virtual Satellite in the CEF is assigned to a unique discipline and restricts it to certain operations that can be performed on the repository. This ensures that e.g. the expert for one discipline can only modify (add/remove components, alter parameters, add calculations, etc.) the part of the study one is assigned for, but has read access to all other disciplines. Accordingly, all participants are still permitted to have a complete overview of the design as it is depicted in the screenshot on the right hand side of Fig. 1. There, the study explorer shows disciplines and components as hierarchical top-down tree, a way that has proven to be very intuitive to navigate in. The data model also offers the possibility to create simple calculations. At the beginning of a satellite design it is very important to keep track of critical system parameters like overall mass and power consumption. Such summaries are called budget equations. Therefore, the values of system components are brought up in the hierarchy and are finally collected on system level. Each discipline can add calculations to their domain model using a wizard dialog. Generally, all parameters within a domain can be used, but valuesfrom other domains can only be referenced in a calculation when the other discipline actively shares the parameter with others. This is indicated by a little green tick in the GUI of the software. This is the easiest and most natural way to link disciplines together. The initial hierarchical character is slowly changing into a complex network creating dependencies directly between the different disciplines.

The save strategy is depicted in Fig. 2. As explained earlier, workstations create their individual domain model and commit an update to a central repository with version control. This repository saves the study in XML Metadata Interchange (XMI) format. Each discipline contributes to just a part of the XMI file. The authorization mechanism guarantees that a discipline can only edit his/her domain which eliminates possible merge conflicts. This work concept is distributed, assures data integrity and enables good team collaboration.



```
<xml>
<study demosat>
  <discipline1>
    <component1>...
    </component1>
    <component2>...
    </component2>
    <component3>...
      <param1= ; "shared">
    </component3>
    ...
    <calculation>...
    </calculation>
  </discipline1>
  <discipline2>
    <component>...
    </component>
    ...
  </discipline2>
...
</study demosat>
</xml>
```

Fig. 2. Save strategy of the study with XML file

As already mentioned, the software also offers a component library which can be filled with standard components, previous studies, and even simulation models. It helps the users to quickly build up a study, by reusing already approved components. If it is not possible to take a component directly across to a new design, the component repository still functions as knowledge base. The experts can easily browse through old studies and quickly derive or develop components that fit the new requirements.

Fig. 3 illustrates the idea of creating a system simulation. It generally consists of three blocks shown in grey, the dynamics, the environment, and the mission information. Verified and validated models exist for these three blocks because they describe the physics around the satellite, like Earth's magnetic and gravitational field, the atmospheric models, the equations of motion and orbit models. Once implemented, these models can be stored in the component repository and re-used in each study. The part of the simulation that changes with each mission is the satellite's behaviour itself. And it is a time-consuming task to create tailored models. That is why Virtual Satellite specifically addresses this issue and provides means to shorten the time needed to create a simulation of the satellite.

One way to do so is to create high quality component models that are well documented and therefore are easy to be understood by others and can potentially be re-used in other satellite designs. In regards to this, two projects contribute to Virtual Satellite: the projects WieMod [15] and SimMoLib. WieMod defines a process and practical guidelines for high quality modelling and re-use of models. SimMoLib stands for Simulation Model Library and addresses the fact that models often exist in different institutes in different formats. They contain a lot of knowledge. Knowledge that should be incorporated in future designs. The situation we find today is that there is no central way to access this knowledge. Therefore SimMoLib claims to establish a simulation model database which can be directly accesses in Virtual Satellite and in the CEF by connecting it to the component repository.

The fact that models are usually designed for a specific platform or even for a specific version of a software does not help in regards to reusability. ESA's Simulation Model Portability (SMP) 2.0 standard [16] raises model development to a new abstraction level in order to make models portable to different platforms and reusable in other studies. In the Virtual Satellite software, it is currently possible to import dynamic models in SMP2 format into the component repository. The dynamic behaviour of each component is needed in order to generate a system simulation of the satellite. Ideally, corresponding models should be available in the repository, but if this is not the case or a new component is designed, there is no way around creating it. To connect the models together, a port mapping is needed to set up a complete simulation. It defines which components exchange values, how they are linked and what interactions and dependencies they have. The port mapping is done with a simple text file that can be saved with the study and is parsed to feed the relevant data to the model. Virtual Satellite offers a special editor with text highlighting and auto completion functionality based on a domain specific language for this task.

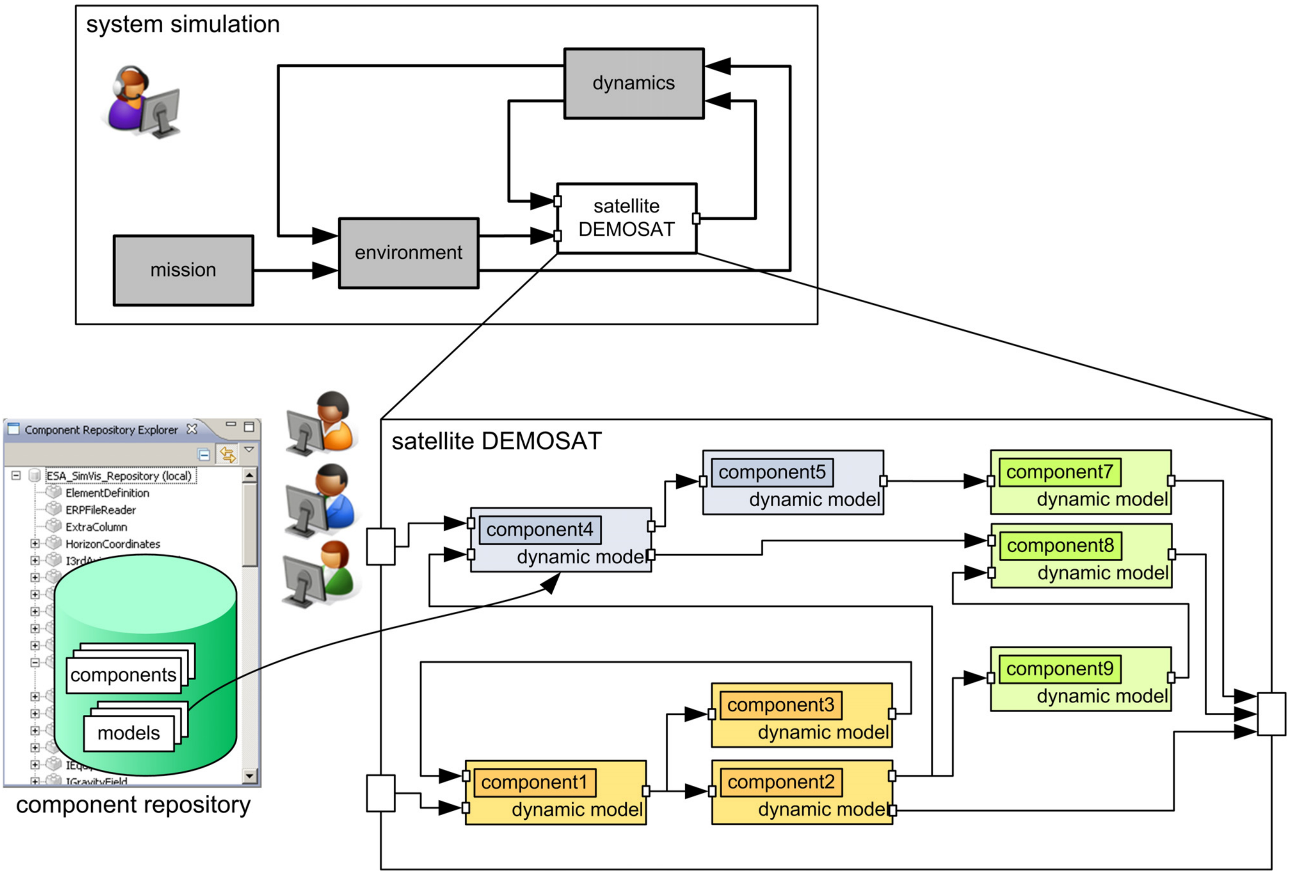


Fig. 3. Concept to develop a satellite simulation on system level using the component repository

The complete study can be exported as SMP2 simulation. Due to the prototype status of this functionality to create SMP assembly and catalogue files, this process is not yet working fully automatically, but the necessary XML frame can be created with one mouse click already reducing the time to initialize the simulation in Phase B.

Technologies

The software is developed using modern project setup and latest software technologies based on the Eclipse IDE. The development strictly follows the model driven architecture (MDA) approach using Eclipse Modeling Framework Project (EMF) [17] and is based on DLR internal or open source projects. The data model is represented as an Ecore metamodel. Data exchange between distributed workstations is implemented using XMI and a Subversion repository. For simulation support, the model import, transformation and export is based on open source technologies like Xtext and Xpand from EMF exploiting the great potential of the underlying Ecore model. Persistence of the model is guaranteed by introducing a separate layer in the software accessing SQL databases underneath.

The project uses a complete software configuration management system. On the one hand, this includes an issue management system called Mantis for bug reports and requirement and release planning. On the other hand, the source code is managed with the version control system Subversion. Last but not least, the software is built and unit tests are run automatically using Maven and the continuous integration system Hudson. This approach allows fast implementation cycles and gives flexibility to incorporate new software requirements quickly, thus enabling the development in close relation with the users at the CEF.

CONCLUSION AND OUTLOOK

The software Virtual Satellite has now reached a prototype stage and was evaluated during a one day test session with a small group of users in the CEF. It was shown that the software has the same basic functionalities as the IDM tool and already extends it. This live evaluation created a lot of feedback from the users; feedback necessary to start discussing new features and requirements to further enhance the software. The short-term goal is to improve the software in regards to user-friendliness, intuitive use and robustness, and to establish it as standard tool for CE studies during early design phases at DLR. The software development team will accompany more design studies at the CEF, deriving new requirements and use cases to further improve the software. Until now, several new functionalities are already under consideration and shall be discussed hereafter.

Improving and adding more functionality to the simulation support will have the highest priority. The data model is currently modified by editors parsing XMI files and a text-based editor is used to define the port mapping of the simulation. One idea is to use graphical editors instead for these tasks. This would mean a great improvement towards user-friendliness since spacecraft engineers are used to work with graphical editors. The Eclipse Graphical Editing Framework (GEF) [18] provides promising concepts to easily do so.

To support and enhance Phase B studies by simulation, an already developed satellite model with environment [15][19] will be integrated as generic template for simulations and used as first shot to gain experience on a minimum risk basis. Developed to be transformable from its MATLAB/Simulink origin into the SMP2 standard, the complete process of importing, storing, and exporting the dynamic models can be tested and evaluated. Analysis of future Phase B studies using this model in the CEF will bring extra value of simulations to early design stages and give indications on which aspects to emphasis work on.

The incorporated version control system gives full transparency of the design parameters. A new revision is created with every upload. This data could be used to display the history of a selected parameter over the design process, e.g., it enables decision makers on system level to monitor the satellites launch mass at all times. Critical design decisions can be clearly identified, questioned, and evaluated. Additionally, having the complete design history available, it is also possible to compare two or more stages of the design and clearly show the evolution.

Visualization is a powerful way to present results and can be used in manifold ways in virtual satellite design. During Phase 0/A sessions, users in the CEF expressed the need for an instant preview of the design in order to quickly have an overview or see the latest changes. A first glance of the satellite could be quickly generated using geometric primitives, like cylinders, spheres, and cones, a method used e.g. in the Space Station Design Workshop [20].

Once the output of simulations is fed back into the integrated data model, it can be imagined that results of different analysis can be visualized together or superimposed in one view in Virtual Satellite. First attempts exist to visualize the orbit and the ground track.

The current calculation support in the prototype is limited to basic mathematical operations like add and multiply. Excel offers great freedom to make complex calculations and use the incorporated math engine. The intention is to create an interface to Excel. An Excel window will then open in an embedded editor. The engineer can freely work in the workbook and select specific cells as input or output value.

REFERENCES

[1] H. Schumann, A. Braukhane, A. Gerndt, J. Grundmann, R. Hempel, B. Kazeminejad, O. Romberg, and M. Sippel, “Overview of the new concurrent engineering facility at DLR”, *3rd International Workshop on System & Concurrent Engineering for Space Applications (SECESA)*, Oktober 2008.

[2] H. Schumann, H. Wendel, A. Braukhane, A. Berres, A. Gerndt, and A. Schreiber, “Concurrent systems engineering in aerospace: From excel-based to model driven design”, *8th Conference on Systems Engineering Research*, 2010.

[3] J. Eickhoff, *Simulating Spacecraft Systems,* Springer, 2009.

[4] P. Hertkorn and A. Reichwein, “On a model driven approach to engineering design”, *International Conference on Engineering Design (ICED)*, August 2007.

[5] *ECSS-M-ST-10C Space project management - Project planning and implementation*, ESA Std., Rev. 1, March 2009.

[6] P. Fritzen, S. Kranz, P. van der Plas, and M. Arcioni, “Rapid simulation development for concurrent engineering”, in *SpaceOps 2006 Conference*, Rome, Italy, June 2006.

[7] M. Bandecchi, B. Melton, B. Gardini, and F. Ongaro, “The ESA/ESTEC Concurrent Design Facility”, *Proceedings of the 2nd Concurrent Engineering Conference (EuSEC)*, Munich, Germany, September 2000.

[8] R. Mager and R. Hartmann, “The satellite design office at Astrium - a success story of an industrial design center application”, Proceedings of the *2nd Concurrent Engineering Conference (EuSEC)*, Munich, Germany, September 2000.

[9] J. Smith, “Concurrent engineering in the jet propulsion laboratory project design center”, *Aerospace Manufacturing Technology Conference & Exposition*, 1998.

[10] G. Filippazzo, “The Italian Space Agency Concurrent Engineering Facility (ASI CEF) Project”, *1st Concurrent Engineering for Space Applications Workshop*, 2004.

[11] D. Di Domizio and P. Gaudenzi, “A model for preliminary design procedures of satellite systems”, *Concurrent Engineering*, vol. 16, pp. 149–159, June 2008.

[12] O. Romberg, A. Braukhane, and H. Schumann, “Status of the concurrent engineering facility at DLR Bremen”, *German Aerospace Congress*, September 2008.

[13] M. M. Schiffner, *An object-oriented modelling method for simultaneous system design*, Ph.D. dissertation, Technical University Munich, 2007.

[14] M. Wilke, *Integrated model-based satellite design*, Ph.D. dissertation, Technical University Munich, 2002.

[15] WieMod Project Homepage: www.wiemod.de, last accessed: 10/01/2010

[16] ESA, *SMP 2.0 Handbook*, Issue 1, Revision 2, 2005, doc Ref: EGOS-SIM-GEN-TN-0099.

[17] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework*, 2nd ed., Boston, MA, Addison-Wesley, 2009.

[18] W. Moore, D. Dean, A. Gerber, G. Wagenknecht, and P. Vanderheyden, *Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework*, IBM Corp., 2004.

[19] A. Berres, M. Berlin, A. Kotz, T. Terzibaschian, and A. Gerndt, “A generic Simulink model template for simulation of small satellites”, 7th *Symposium on Small Satellites for Earth Observation*, pp. 247–250, Mai 2009.

[20] *Final report of the Space Station Design Workshop*, Tech. Report, Institute of Space Systems (IRS), University of Stuttgart, 2005.