Dependency Analysis in Complex System Design using the FireSat example

Johannes Groß, Stephan Rudolph
Institute of Statics and Dynamics for Aerospace Structures, University of Stuttgart
gross@isd.uni-stuttgart.de, rudolph@isd.uni-stuttgart.de

Abstract. Using the FireSat mission from literature (see Larson 1999), the satellite design process is analyzed. Prior to this analysis, the physical and functional properties of the satellite system are encoded in a graph-based design language. During the automated design language compilation process several graph representations are generated. These graph representations are generated from the design constraints. From the mathematical analysis of the graphs three important interpretations can be derived. The first interpretation concerns the derivation of the exchange rates based on the analysis of the functional coupling. The second interpretation yields a feasible design sequence. Third, a generic backtracking method for resolving engineering design conflicts is presented and illustrated with an antenna example. All three interpretations are illustrated through a detailed view into the FireSat communication subsystem. Furthermore, the demonstration of the method closes with a system-level analysis of the satellite design process to show its applicability to all levels of detail in system design.

1 Complex System Design

For the design of complex engineering systems such as a satellite, using model-based systems engineering (MBSE) techniques become more and more a state of the art approach. For the design of complex software systems, the model driven architecture (MDA) approach was developed. Both of the MBSE and the MDA technique have commonalities but have not yet merged into a common wording. The software engineering terms of a platform independent model (PIM), a platform specific model (PSM), or a domain specific language (DSL) have without any doubt their correspondences in the engineering domain. Yet this mapping is not yet established or agreed on and should therefore be the topic of future consideration.

In engineering design, many efforts have been taken to establish and develop data models (e.g. (Eisenmann 2009)) and modeling techniques (e.g. Delp 2008, 2009) in the field of space systems. In addition to these approaches, further interesting potential is brought up by MBSE in the field of the model-based analysis of the designed systems. Such analyses contain not only the structural and mathematical analysis of the systems, but also the establishment of simulations from the systems engineering models. In (Schaus 2011) for example, the benefits of modeling simulation models are pointed out. In (D`Ambrogio 2010) the model-driven architecture (MDA) approach to the development of simulators is shown. On the basis of these undertakings, a clear systematic methodology for the creation of systems and variants thereof has to be established.

The dependency analysis shown in this paper is an excerpt from a larger satellite design language based on the Unified Modeling Language (UML). Initiated by the graph-based satellite design language of (Schaefer 2005), an updated version of this design language was developed by (Gross 2011). The features of a graph-based design language are hereby combined with the semantic elements of a standard modeling language, in this case the UML. An exemplary FireSat mission is laid out in more detailed in (Gross 2012). The work presented here concentrates on the conclusions of the dependency analysis presented in the following sections.

1.1 The Pahl and Beitz Construction Systematic

The German "Pahl and Beitz construction systematic" (Pahl 2003) describes the design task as a mapping sequence in between 4 different spaces (Fig. 1). Firstly, the initial requirements of a system are set up in the requirements space. These requirements (REQs) have then to be mapped to abstract product functions (FUNs) which describe the problem independently from a specific implementation. Then a solution principle (SOLs) has to be selected for the problem solution. Usually different embodiments (EMBs) from different suppliers can be chosen from to satisfy a solution principle.

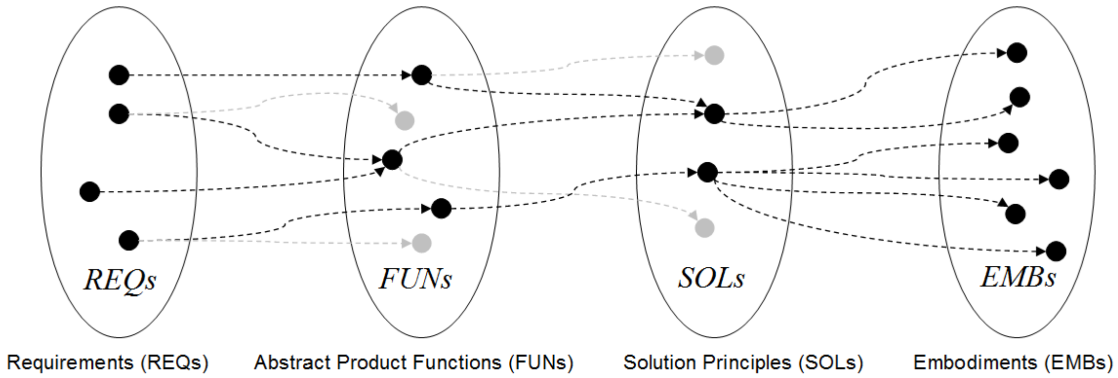

Fig. 1: Construction Systematic according to [PB03]

This construction systematic is appropriate for top-down design (i.e. from left to right in Fig. 1). Such a design systematic is helpful in the creation of variants, because the (very) same requirements are mapped to different variants. Conversely, a bottom-up design procedure can be seen in the systematic in Fig. 1 as a step from a more concrete level to a more abstract level (i.e. from right to left). In the bottom-up design process, an abstract phenomenon which fulfils the required function has to be found. This can also be regarded as a pattern recognition process. Consequently, the verification process of a successful bottom-up design can be understood as a top-down design step.

For the often heritage-based design task in space systems engineering, both top-down and bottom-up methods can be applied because each change in a given design can be understood as a mapping from a more concrete level to a more abstract level and then switch back to the concrete level in another branch as shown in Fig. 1.

1.2 Pahl and Beitz in the Unified Modeling Language (UML)

To enable the usage of the shown design systematic in a Unified Modeling Language (UML) based systems engineering environment, it is necessary to establish clear interfaces between different solution principles or embodiments. These interfaces can be created by the inheritance of the common attributes of different solution principles. In Fig. 2 this practice is shown for the example of three different antennas (e.g. a horn, a parabolic and a helix antenna). While the three antenna types have a set of mainly distinct geometric attributes, they all have in common the attributes of antenna gain, material density, carrier frequency and wavelength. By inheriting these four features, a system using an antenna can integrate this part by simple referencing to these common attributes. Then these parts can be exchanged later without any adjustments required in the embedding system.

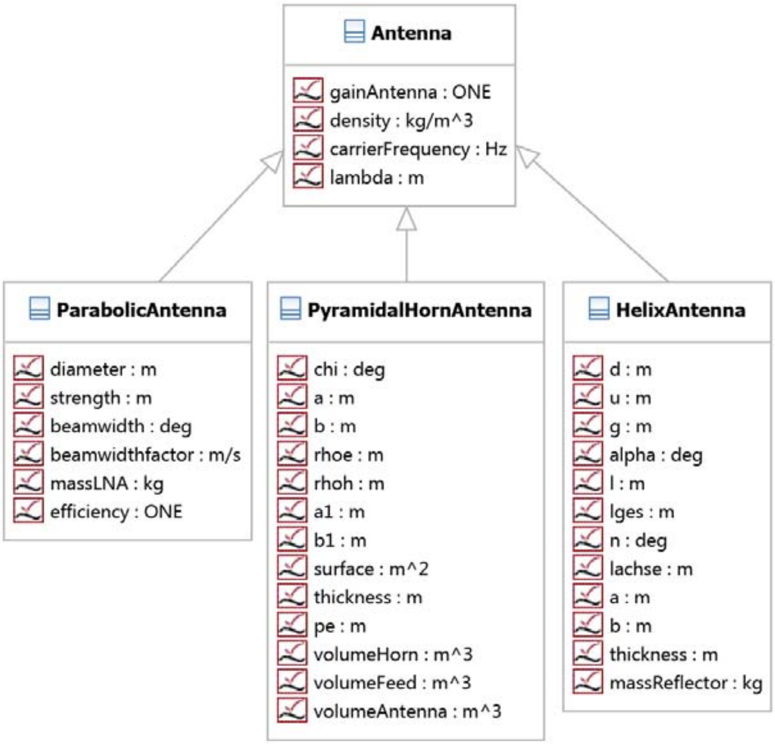

Fig. 2: Classes for the antennas used in the example

2 Dependency Analysis

As background for the examples the FireSat mission described in *Space Mission Analysis and Design*, edited by Wiley J. Larson and James R. Wertz (Larson 1999) is used. It describes a low earth orbit mission for the observation of forest fires. In this section, the communication downlink of the payload data, shown schematically in Fig. 3, is discussed.

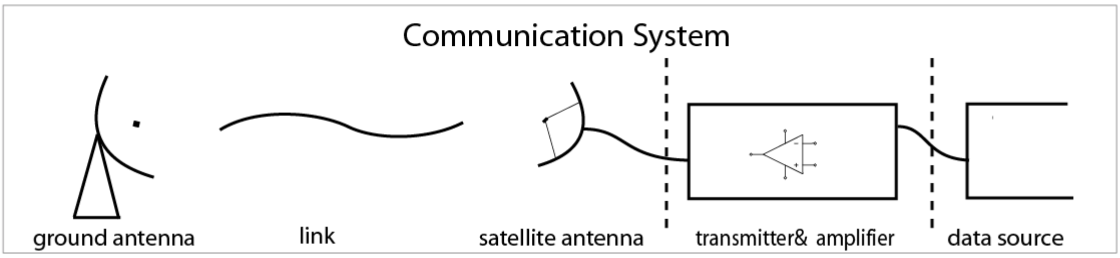

Fig 3: Design Cycle of a Communication System with Parabolic Antenna

The data generated by the payload, marked as data source in Fig. 3, are sent to a ground antenna by the communication subsystem which is built up by a transmitter, an amplifier and the satellites antenna. Since in satellite design the system is highly constraint in terms of size, mass and power, these variables are the main requirements for the layout of each subsystem.

For the purpose of the following trade-off study, a dependency analysis is shown for the aforementioned three different antennas. First, on the example of a parabolic antenna, the analysis method and the corresponding analysis graphs are introduced. Second, the horn antenna with some more variables will be analyzed followed then by the helix antenna with special emphasis on some peculiarities for the design procedure.

In Fig. 4 the systematic from Fig. 1 is shown with the three different antenna solution principles. The scope of the analysis in this section is the step from the abstract product function “align radiation” to (one of) the three different solution principles.

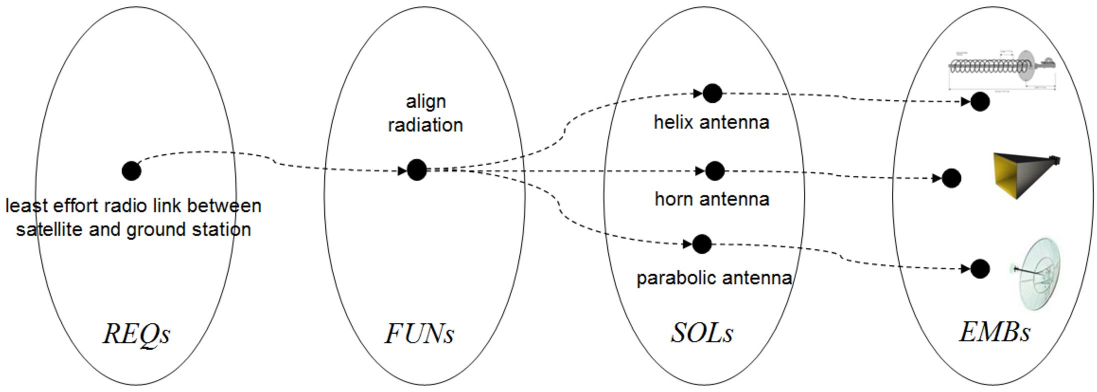

Fig. 4: Construction systematic for the antenna example

The antennas are later evaluated in respect to the two different properties of mass and gain. The mass shall be as small as possible. The gain, expressing the efficiency of the antenna, is a classical evaluation parameter in antenna design and shall be maximized.

2.1 Description of the Parabolic Antenna

The parabolic antenna used is shown in Fig. 5. The main antenna parameter is the diameter. The position of the focal point is disregarded because it has no effect on the antenna gain.

Fig. 5: Drawing of a Parabolic Antenna

The calculation of the gain of the parabolic antenna and the mass are given in equations (1) and (2). These equations are stored in mathematical syntax in the corresponding class of the parabolic antenna shown in Fig. 2. Equation (1) is the calculation of the antenna gain after (Larson 1999) with *lambda* as the wavelength. Equation (2) constitutes an approximation for the calculation of the antenna mass based on a flat antenna, which is sufficient for this context of the dependency analysis, since all relevant variables are contained in the equation.

$$gainAntenna = \pi^2 * diameter^2 * \frac{1}{lambda^2} \qquad (1),$$

$$mass = diameter^2 \cdot \frac{\pi}{4} \cdot strength \cdot density \qquad (2)$$

2.2 Dependencies between Requirements

With the constraint processing technique presented in (Rudolph 2000) the two equations can be automatically solved either to calculate the mass from a desired gain or to calculate the gain from a given diameter. With the mentioned technique this can be done equally with huge equation systems (here: 10 ~100000 equations.). In Fig. 6 the dependencies given by the equations above are drawn as a graph, in which every variable occurs only once. Mathematically, a graph G consists of edges E and vertices V and is used to express the topology of a system as G(E,V). The variables of the equations represent the edges of the graph. Each variable consists of an instance name followed by a "." and the variable name. The instance name is given to the UML instance specification which represents the embodiment in the design language. The links (i.e. the vertices of the graph) represent the propagation of the variables during the calculation. The given variables are marked orange and the arrow heads give the direction in which the variable is propagated.

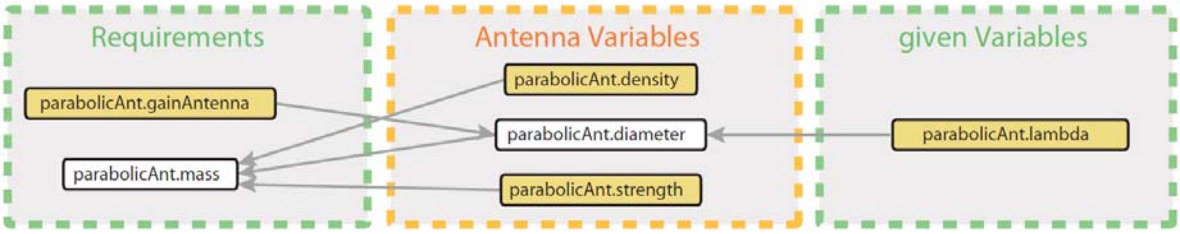

Fig 6: Dependencies between the Requirements and the Antenna Variables

In this graph it can be seen that the two requirements mass and gain are coupled by the variables of the antenna solution principle. Hereby the equations of the antenna (Eq. (1) and (2)) determine the exchange rate between the requirements variables mass and gain. In Fig. 7 the function of the requirements mass and gain for the parabolic antenna is drawn.

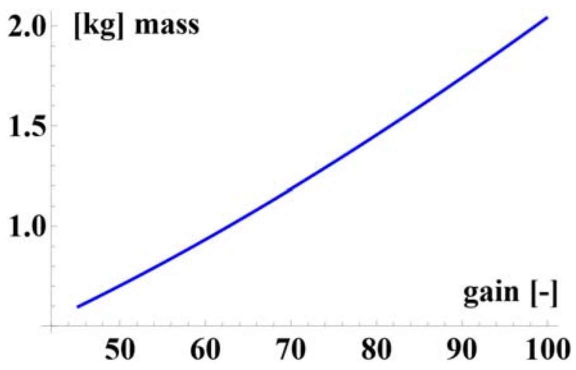

Fig. 7: Function of the Requirements for the Parabolic Antenna

This leads to a crucial point in the analysis of system dependencies: In a complex system engineering example, most of the requirements are interdependent. If they can not be met all at the same time, they have to be adjusted. The functional coupling between the variables is determined by the solution principle used in the design. This could be interpreted as "exchange rates" between the requirements, since a “delta” in one variable is worth a “delta” in another variable. This is due to the coupling of the equation system defined by the solution principle. In this example, the resulting question is: "How much gain do we get per mass?" The answer to this question is dependent on the equations (i.e. the equation system) of the antenna, and the equations are a representation of the chosen antenna solution principle. This can be stated as a rather philosophical statement:

“The choice of the solution principle determines the "exchange rates" between the requirements.”

For the reader unfamiliar with this type of analysis this seems at first to be a useless statement due to its philosophical nature, but to help the engineer understand the different levels of action, in the case of top-down design scenarios it can be quite helpful as will be shown later on.

2.3 Sequence of Design Problems

If the graph in Fig. 6 is rearranged to show the solution sequence one obtains Fig. 8. In this sequence, the given variables are put in the row, one step before they are required to solve an unsolved variable. By this it can be determined at which step of the design a certain variable is required to be known or to be determined.

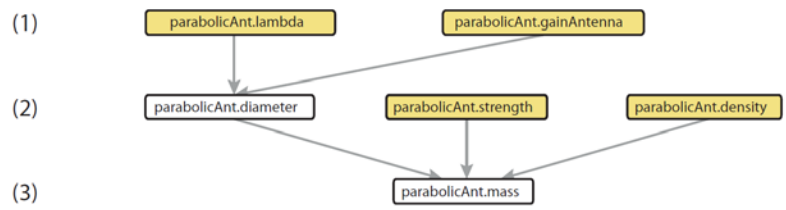

Fig. 8: Solution Path of the Variables in the Design of a Parabolic Dish Antenna

For example, in the first step, the wavelength "lambda" and the requested "gainAntenna" are required. In the second step the "diameter" of the antenna can be calculated. Then the "strength" and "density" of the antenna are a prerequisite to calculate the variable antenna "mass".

In large and complex system design examples (as shown later in section 3), the different engineering domains working together on a product development can acquire by this analysis the information at which step in the product design their results are required. If for instance the determination of the diameter might takes several months in Fig. 8, the material department can wait until then to deliver the required strength and density of the antenna. Thus it can be stated:

"The solution path of the design variables, determines the sequence of design tasks."

2.4 Cycles in Design

Since equation systems may not always necessarily possess a solution, one has to back-track and start from (more or less) scratch again. Design is therefore considered to be an iterative task, thus Fig. 8 is drawn in a cyclic manner as shown in Fig. 9. The starting point is hereby the (set of) initial requirement(s) to the given system. These are drawn in the middle of Fig. 8 on top. From these, the diameter of the antenna can be calculated. Now the size can be checked with the surrounding system environment. If the size is not ok, the initial requirements can be changed to meet this constraint.

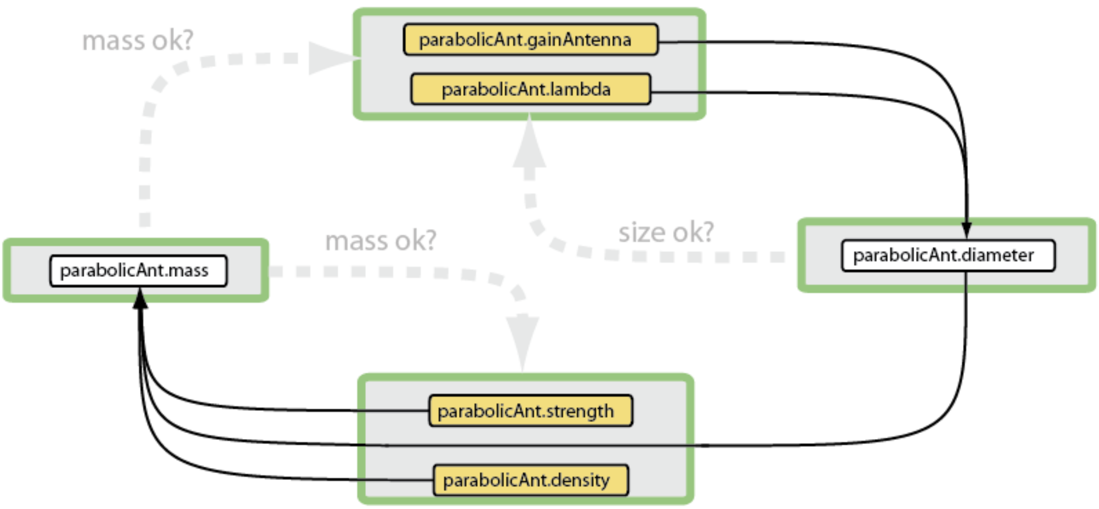

Fig. 9: Design cycle of a Parabolic Antenna

The next and last result of this cycle is the mass of the antenna. Here also the check against the system environment constraints has to be effected. If the mass is not ok, not only the initial requirements can be changed, but also the material variables on the bottom of the graph. Since in this graph, the predecessors of a variable have fewer side effects on the design than the variables upstream in the cycle, it is likely to assume, that the effort is smaller, if the change is done closer to the concerned check parameter. If the strength and density of the antenna are changed, only the mass if influenced. If the gain or the wavelength is modified, also the diameter antenna will change its value. From these observations, a backtracking principle for engineering applications can be drawn like Fig. 10.

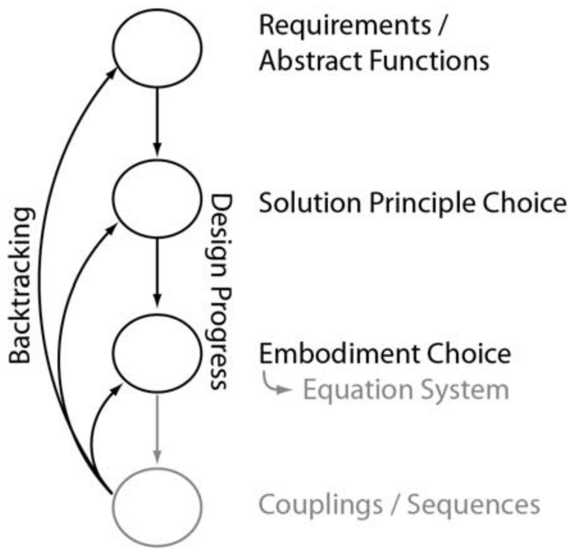

Fig. 10: Backtracking in Systems Engineering

In the schema of Fig. 10, the design has the requirements as a starting point. From the given requirements, a choice of the different solution principles to fulfil the requirements can be selected e.g. horn antenna or parabolic antenna. By choosing the solution principle and subsequently an embodiment, the behaviour of the system is set and thus the set of equations for the calculation of the system behaviour are fixed. Based on the equation system the couplings and the above shown sequence can be generated. The grey parts of the drawing are thus generated automatically from the design language, whereas in the black parts the engineer takes dedicated design decisions.

If now a problem in the design process arises, the nearest starting point for a solution of the problem is to try to change some of the parameters in the equation system of a given embodiment. In this example, the strength of the parabolic antenna could be reduced to meet the mass requirements. This is shown by the shortest bent arrow in Fig. 10 from the sequence to the equation system. If the strength is reduced to a certain degree, the effort to reduce it further may be too high in respect to other possibilities of problem solution. In this case a change of the solution principle e.g. from a parabolic antenna to a horn antenna could be appropriate. If the new solution principle with the changed equation system won't fulfil the requirements, either another solution principle or a change in the requirements might be appropriate.

This procedure for trouble-shooting can be seen as an analogy to the backtracking graph algorithms known in artificial intelligence and thus it can be stated:

"A general problem solving method can be obtained by applying backtracking in engineering."

2.5 Analysis of Horn Antenna Dependencies

If we take now a pyramidal horn antenna into consideration, the engineer has to deal with some more variables. Fig. 11 shows an image of the analyzed antenna and most of its parameters.

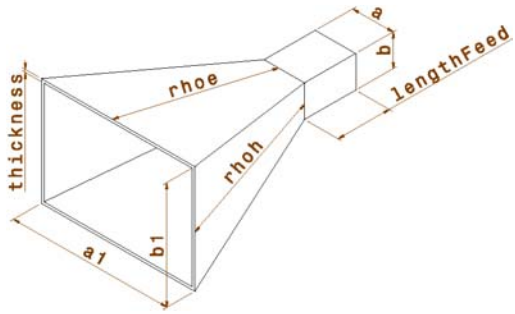

Fig. 11: Drawing of Horn Antenna

The dependencies between the parameters are shown in Fig. 12.

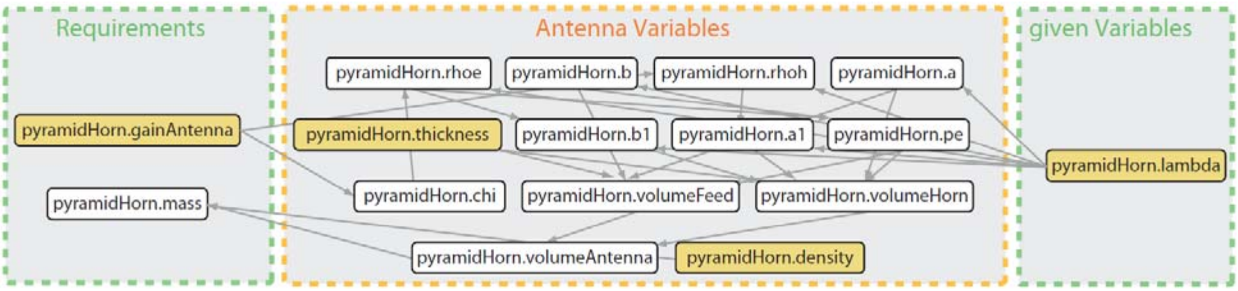

Fig. 12: Dependencies in Horn Antenna Design

It can be seen that the same requirements are now coupled quite differently by the variables of the horn antenna. This changes also the functional relation between the requirements.
The resulting design cycle for the horn antenna is shown in Fig. 13.

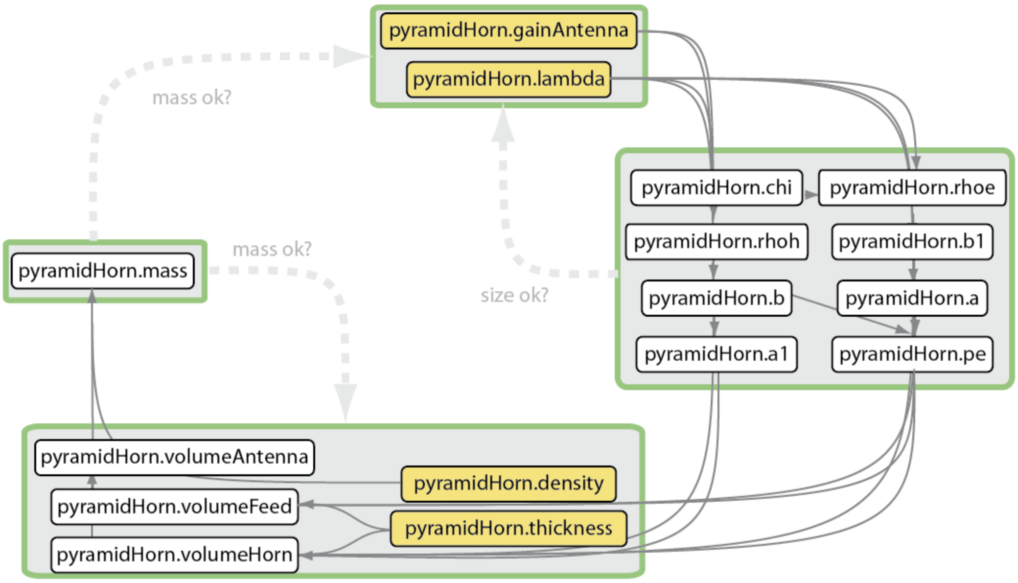

Fig. 13: Design Cycle of a Horn Antenna

2.6 Analysis of Helix Antenna Dependencies

In Fig. 14 a drawing of the analyzed helix antenna is shown.

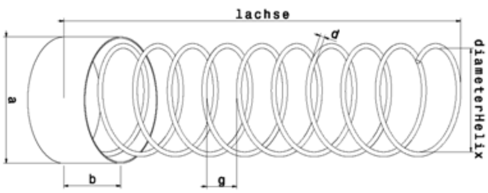

Fig. 14: Drawing of Helix Antenna

In the graph of dependencies of the helix antenna in Fig. 15 it can be seen that the antenna gain is not a given variable. This is due to equation (3) for the calculation of the helix antennas gain according to (Kark 2010). This equation cannot be solved for *n*, so the design of the antenna has to be done iteratively.

$$gainAntenna = 8.3 \cdot \left(\frac{u}{\lambda}\right)^{\sqrt{n+2}-1} \cdot \left(\frac{n \cdot g}{\lambda}\right)^{\frac{4}{5}} \cdot \left(\frac{\tan(0.218)}{\tan(\alpha)}\right)^{\frac{\sqrt{n}}{2}} \qquad (3)$$

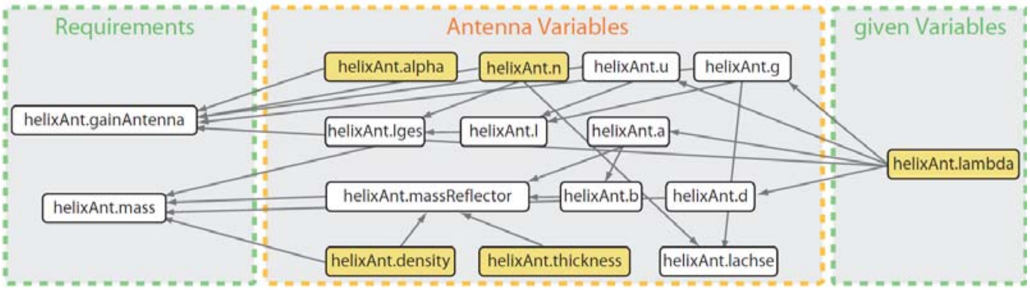

Fig. 15: Dependencies in Helix Antenna Design

The design cycle of the antenna (Fig. 16) starts therefore with the angle of the helix, “alpha”, the number of windings “n” and the wavelength. From this, the geometrical values can be calculated. After the size check, with the material values the mass can be calculated and also from the geometrical values the gain can be calculated.

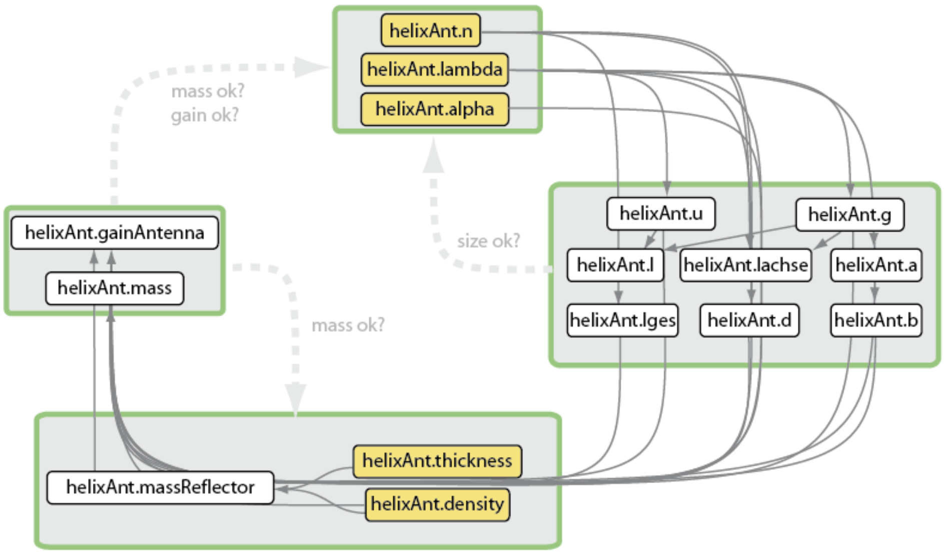

Fig. 16: Design Cycle of Helix Antenna

3 Analysis of Dependencies in Complex System Design

In this section it will be shown that the graphs for the embodiments shown above can also be generated on subsystem level and on system level. This applicability is given automatically through the recursive definition of the satellite as a "system of systems". The graphs for the subsystem level are on a broader scope and show the order in which different embodiments of a subsystem can be laid out. The graph on system level shows the order in which the different design problems in satellite engineering can be tackled. This graph is very similar to processes already implemented in satellite industry companies because it results from and is conform to the natural dependencies in the specific design problem.

3.1 Cycles in Communication Subsystem Design

When changing the scope from the antenna design to the layout of the communication sub-system, the requirements for the system considered do change. The communication subsystem as shown in Fig. 3 consists of a ground antenna, the link to the satellite and (on the satellite) an antenna and a transmitter with an integrated amplifier. Fig. 17 shows the UML instance specifications of the system components. The satellite link instance contains the different variables and equations for the link calculation. The transmitter instance calculates the mass of the transmitter depending on a correlation to the required power. The values calculated for this exemplary layout are given in the slots of the instances. The physical unit information of the values is indicated behind the value, "ONE" stands hereby for a dimensionless variable.

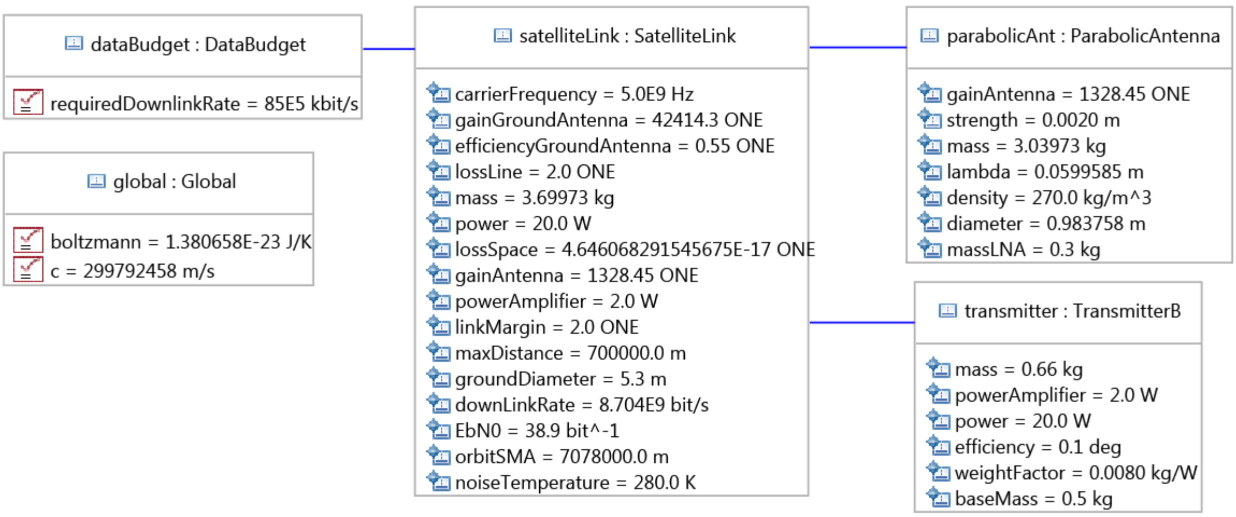

Fig. 17: Instances of the Communication System Classes

To show once the full dependencies of all the design variables, in Fig. 17 the detailed design cycle for the communication system with a parabolic antenna is presented. It can be seen that starting from the requirements the link equations are solved to calculate the required gain of the antenna. Then transmitter and amplifier are laid-out roughly prior to the antenna design.

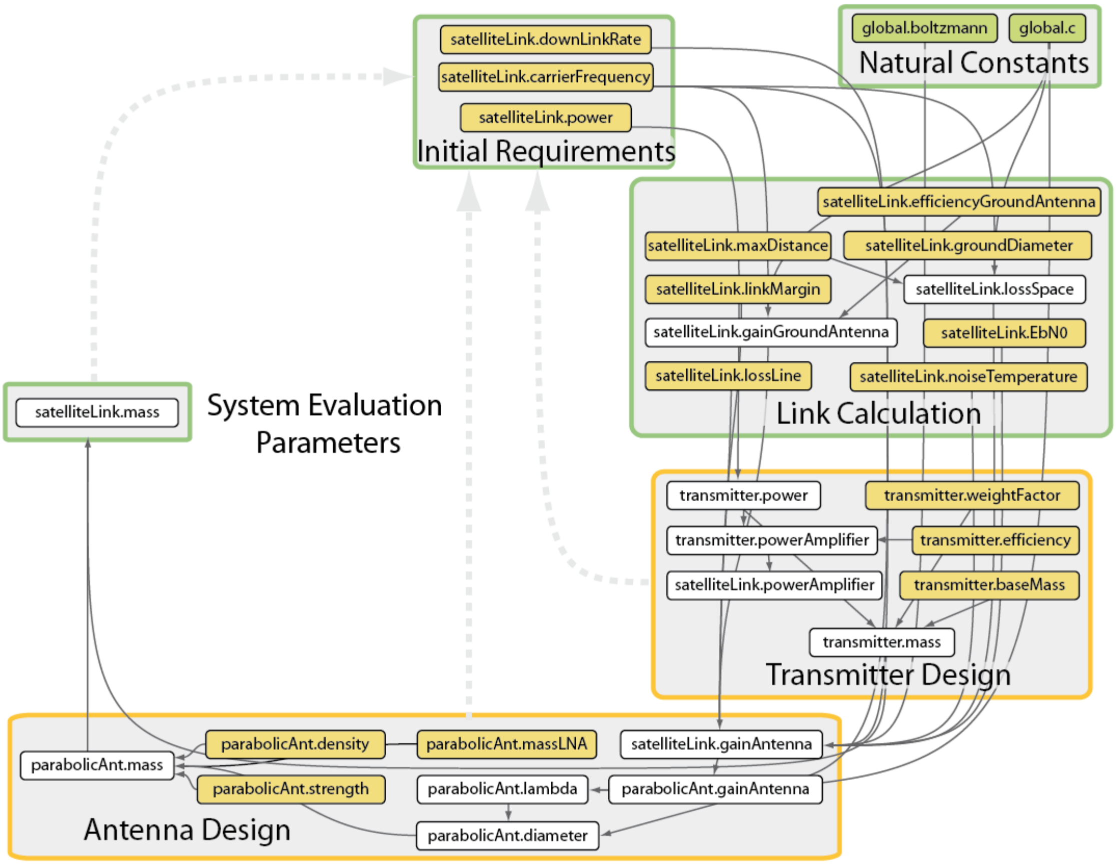

Fig 17 Detailed Design Cycle of a Communication System with Parabolic Antenna

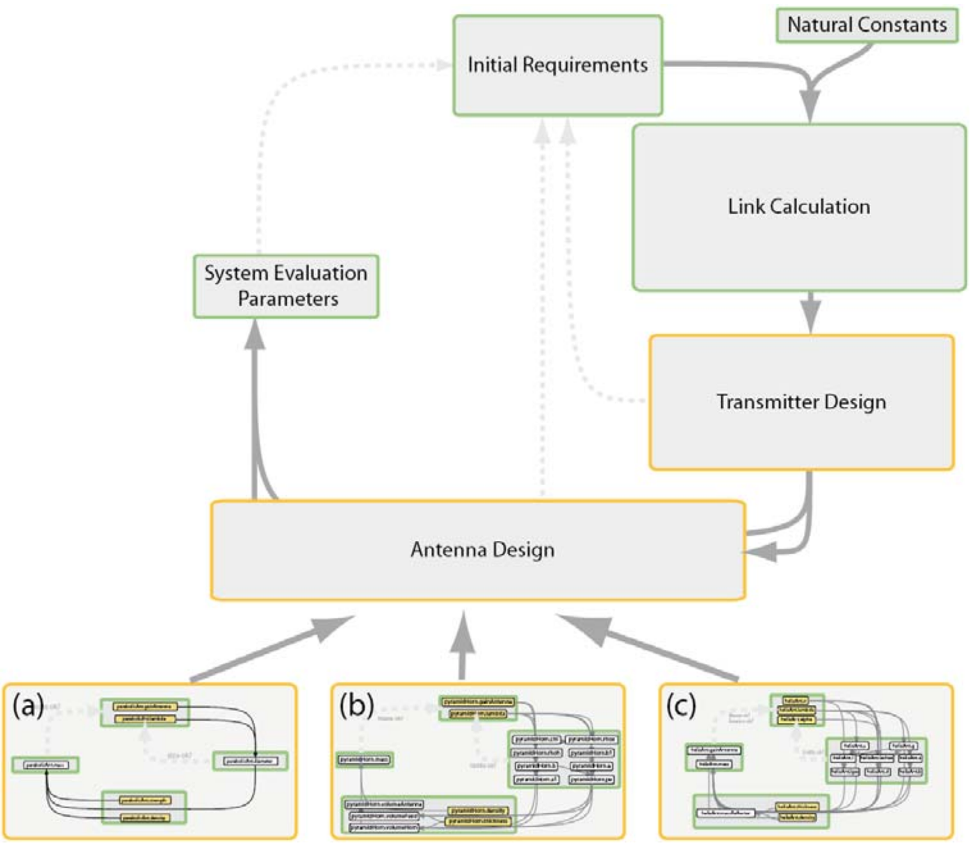

Fig 18: Simplified Design Cycle on Communication System Level

In Fig. 18 a simplified design cycle for the communication subsystem is drawn. In the bottom line the alternative antenna technologies with their corresponding design cycles are shown. The design cycle in this view is derived from the dependency analysis of the chosen solution principles and the respective embodiments. By the temporal ordering of the solution steps, the calculation order of the different variables can be determined (Rudolph 2000). From the calculation order of the single variables, the layout order of the different embodiments can be retrieved. Thus on system level the design cycle gives a temporal order of the layout of the subsystems and of their embodiments in a top-down design process.

3.2 Cycles in Satellite Design

With this analysis method also a design cycle for the complete satellite system can be created. The basic ideas for this proposition were already established in (Schaefer 2005). It is based on a top-down satellite design procedure using graph-based design languages. Since then, the FireSat example was modelled and laid-out in more detail (Gross 2011, 2012).

The satellite design loops shown in Fig. 19 start with a mission description from which a potential payload design is obtained. The preliminary payload design contains estimates for e.g. the mass, power requirement, pointing requirements and the required data downlink rate. After the payload design an orbit is selected to fulfil the mission requirements. Then, the subsystems can be created and laid-out.

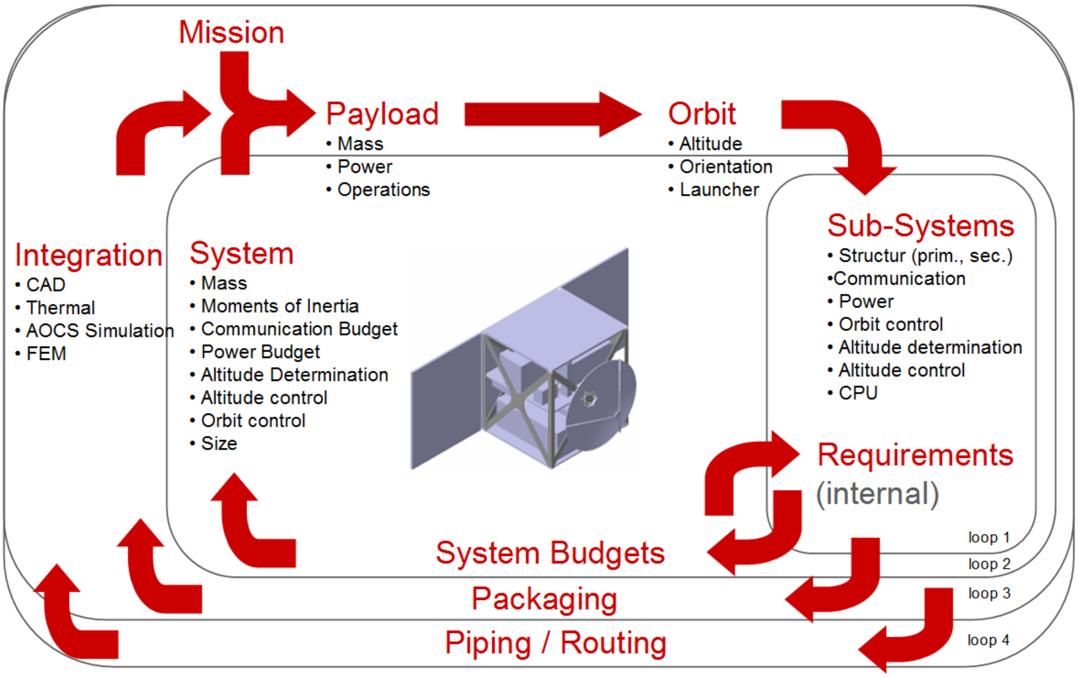

Fig. 19: Extended Satellite Design Cycle after (Schaefer 2005)

First the mechanical subsystem is created with a satellite bus based on an estimation of the size. With the given size and the presumed mechanical bus, the power system can be created and the size of the solar arrays can be calculated preliminarily. Then the communication system is created based on the first values.

These subsystem creations can trigger further embodiments to be created and integrated in the satellite as shown in (Schäfer 2005). Afterwards, the system properties are retrieved by the so called system budgets. This is the sum of all masses, all power requirements and all downlink requirements in this case. Then the whole design cycle is repeated until the system values converge.

The outer loops describe the subsequent packaging of the satellite and if the packaging is fulfilled, the solution of the field problems, such as the thermal model and the finite element model can be investigated. Another subsequent loop comprises the routing of the tubes and cables in the satellite to finish the geometrical model.

4. Reflection and Outlook

The shown graphs and analysis techniques depend on the explicit nature (i.e. that the equations are known explicitly) of the considered domain. Since it is designated to top-down problems, it will loose its strengths when bottom-up procedures dominate the design problem (i.e. when no closed-form symbolic equations are known). Consequently the systems have to be built up of formalized discrete models instead of continuous field models.

However in the shown example of a preliminary satellite design, these preconditions are met and so with the shown methods satellite specific know-how can be extended in a specific project setting. When the automated constraint solving as shown in (Rudolph 2000, Rudolph 2004) are established in the design of large complex systems the knowledge of the design sequence and the order of the problem solution can be retrieved automatically. In combination with the methods of a graph-based design language (Rudolph 2003, Gross 2012) the changes from one conceptual level of Fig. 1 to another can be done by previously specified rules. By this, an automatic variation of design alternatives becomes feasible within a graph-based design language.

The potential of graph-based design languages for later design and development phases is partly shown in (Gross 2012). The dependency analysis shown herein is based on the analytical equations modelled in a graph-based design language. Such dependencies can be analyzed for practically any size of an equation system. In later design phases, the dependencies of the differential equations are mostly encoded in external simulation programs and are thus not modelled explicitly in the design language. If however such dependencies are of interest for the shown analysis, they can still be modelled abstractly in the form of a mathematical dependency such as z=f(x,y) expresses that z depends on x and y. Thus in a rule-based design language the shown analysis can be provided not only for the conceptual design phase.

References

D´Ambrogio, A.,Gianni, D., Risco-Martin J.L., and Pieroni, A. 2010: "A MDA-based Approach for the Development of DEVS/SOA Simulators". 2010 Spring Simulation Multiconference, Orlando, US.

Delp, C. L., et al. 2008. "The challenge of model-based systems engineering for space systems" INSIGHT 11 (5): 14– 18.

———. 2008. The Challenge of Model-based Systems Engineering for Space Systems, Year 2. INSIGHT 12 (4): 36– 39.

Eisenmann, H., Miro, J. and de Koning, H.P. 2009: MBSE for European Space-Systems Development. INSIGHT 12 (4): 47-53.

Gross, J. and Rudolph, S. 2012: "Generating Simulation Models from UML - A FireSat Example". 2012 Spring Simulation Multiconference, Orlando Florida 2012.

———. 2011. "Hierarchie von Entwurfsentscheidungen im modellbasierten Entwurf komplexer Systeme". Tag des Systems Engineering der GfSE, Hamburg 2011.

Kark, K.2010: *Antennen und Strahlungsfelder*, 3. erw. Auflage, Wiesbaden, 401-405: , Vieweg+Teubner.

Larson, W. J., and J. R. Wertz, eds. 1999: *Space mission analysis and design*. 3rd ed. El Segundo, CA: Microcosm.

Pahl, G. and Beitz, W.2003: *Konstruktionslehre*, 6. Auflage, Berlin: Springer.

Riestenpatt genannt Richter, M. 2011: "Eine Entwurfssprache zur Auslegung der Lage- und Bahnregelung von Satelliten". Studienarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart.

Rudolph, S. 2000: "Visualization and Comparison of Solution Paths with Artificial Metrics". In: Proc. of DETC00, 2000 ASME Design Engineering Technical Conferences, Baltimore, Maryland USA, September 2000.

Rudolph, S.2003: "Aufbau und Einsatz von Entwurfssprachen für den wissensbasierten Ingenieurentwurf". 3. Forum Knowledge Based-Engineering, CAT-PRO, Stuttgart.

Rudolph, S. and Boelling, M. 2004. "Constraint-based conceptual design and automated sensitivity analysis for airship concept studies". In: J. of Aerospace Science and Technology 8, 333-345 : Elsevier.

Schaefer, J. and Rudolph, S. 2005: "Satellite Design by Design Grammars". In: J. of Aerospace Science and Technology 9, 81-91: Elsevier.

Schaus, V., Groekatthofer, K., Ludtke, D., and Gerndt, A. 2011: "Collaborative Development of a Space System Simulation Model". Collaborative Modeling and Simulation Workshop, WETICE 2011 Conference

Biography

Johannes Gross works in the Similarity Mechanics group at the Institute of Statics and Dynamics for Aerospace Structures (ISD) at the University of Stuttgart. He is PhD candidate in the group of Dr. Rudolph since 2009 and developed a design language for satellites comprising a geometry representation in the UML, the automatic routing of the cables in a satellite and the generation of a thermal model, a Simulink model and an OpenCascade model.

Stephan Rudolph is the leader of the Similarity Mechanics group at the Institute of Statics and Dynamics for Aerospace Structures (ISD) at the University of Stuttgart. PhD in Aerospace Engineering in 1995, PostDoc in the Systems and Design Group at the Massachusetts Institute of Technology (MIT), German Habilitation thesis on Design Methodology in 2002. Within the research for an universal engineering design approach, the main focus lies on graph-based design languages as well as on dimensionless similarity analysis methods for design evaluation.