"Towards animated executable specifications for satellite systems"

Volker Zerbe, Ulrich Freund, Petri Solanti

Faculty of Computer Science and Automation
Technical University of Ilmenau, Germany

E-Mail: zerbe@theoinf.tu-ilmenau.de

Abstract

In this paper we present how system level simulation can support the design process of satellite systems at the early design stage. Using the 'Meet-in-the-Middle' design approach system level models representing specifications can be executed and animated to give an early feedback whether the overall system can meet its requirements. Animation is an essential means to overcome communication gaps between different design teams. As an example we built a system level model of the data- and power bus of an earth observation satellite.

1 Introduction

A lot of different design tasks have to be done to design a satellite system. Normally, the overall system is divided into subsystems and components. Different subsystems are assigned to dedicated design groups. Influences of one group's design to the other subsystems are discovered at a very late stage of the design. Reason for this might be:

- unawareness of the impact of the own design to other design groups.
- lack of information between different design groups.
- too detailed information delivered by a group leads also to a lack of information

Furthermore, every design group adds a safety margin to their subsystem so that if something serious happens during system integration the major part of the subsystem's design can be kept. However, since not every margin is necessary, this approach leads in the end to a too conservative system.
Every design group hosts a lot expertise to design a subsystem. Documentation of this expertise is easy compared to the overall system design because only

one domain of knowledge, e.g. solar panels, is involved. System expertise is still a certain knack of a distinct aero- and astronautical company. System Level Simulation in conjunction with the 'Meet-in-the-Middle' design approach can formalize the system level expertise and reduce the design time of satellite systems.

2 System level modelling of satellites

2.1 The Meet-in-the-Middle approach

The expertise for designing a satellite can be divided into different domains. This knowledge consists of detailed and high-level components models, anticipated reference models by NASA, ESA, DARPA,..., measurement values and identified models, and, last but not least, own experience.

The 'Meet-in-the-Middle' approach proposed by [2] and [1] starts with the translation of the requirements into an executable model. This model is as abstract as possible but considers all relevant aspects of the system. It consists of well known, more or less textbook equations of the different domains, parameters and look-up-tables. These parts are identified and taken from the domains information base. It is the task of the domain's design group to deliver the 'right' high-level model to system design group.

An initial system level simulation is performed and provides key parameters, tolerances, and system constraints to the subsystem design. The subsystem groups then perform with the constraints a detailed simulation and identifies critical elements for the systems design, look at Fig. 1.

Information between the different domains is only exchanged via the system level model. The direct way between the domains must not be used. Information exchange via the system level ensures that an abstraction process is involved and only necessary information is supplied.

2.2 Adaption of the approach to satellite systems

If we consider a satellite network for mobile communication, we might identify the following subsystems: Network Services, User Equipment, Network Equipment, Satellite Equipment, Satellite Environment, Power System, Attitude Control and On-Board Processing. According to the 'Meet-in-the-Middle' approach we take an abstract model of each subsystem. If these are not available because the company has not build it yet, it is derived from the specifications. The system model is then checked via scenarios against the requirements. The design of a subsystem is done by starting at the system level and refining only the subsystem. While refining the subsystem we check our subsystem together with the abstract models of the overall system.

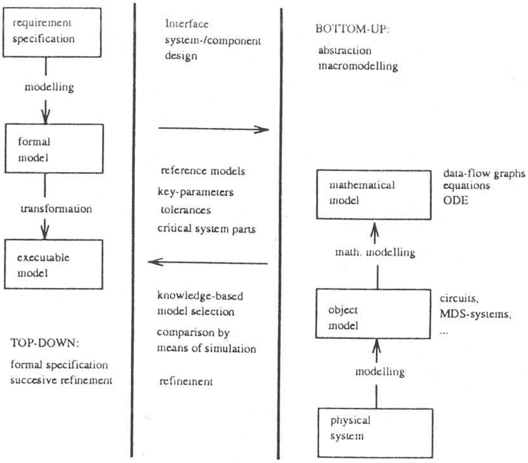

Figure 1: A view of the Meet-in-the Middle strategy

The system level model is an ideal means to communicate between different design domains. The models are easy enough to be understood by the all design groups. Executing this model, i.e., performing system level simulation,

- discovers (functional) dynamic problems not mentioned in the requirements.
- checks the systems performance against the dynamic requirements.
- reveals the dynamic influences of different domains.
- shows via animation how the overall systems works.

After animated system level simulation runs have been performed, almost all involved designers have at least a rough understanding of the overall systems behavior and performance. It is this understanding which constituted the knack of system expertise of a company.

3 System-Level-Simulation Tools

There exists a variety of system level simulation tools which can be used for system level simulation. The range is from spread sheet (no joke), hand coded

C- and Fortran programs of over continuous system simulators to discrete event simulators.

For our system level model of a research satellite we chose BONeS-Designer [4] in conjunction with SatLab. BONeS-Designer uses the concept of discrete event execution together with dynamic data flow models. Furthermore, finite state machines, queuing models and a C-Code interface are available. The execution of data-flow-diagrams and finite-state-machines can be animated during debugging. During animation the block which is executed is highlighted. In external animation mode the executed block is shown in a separate display with additional information about the current simulation time. An interface for exchanging information between SatLab and BONeS-Designer exists.

SatLab is a tool for the design and animation of satellite systems. Since it is based on textbook equations which can be solved analytically, it can be used for the modelling of overall satellite constellations which are common in mobile satellite systems. On system level, it is normally not necessary determine the orbit of a satellite by more detailed models. Since it is possible to show the orbit of a satellite from different points of view, even designers not involved with the orbital aspects get a feeling of the orbital impacts.

4 The Satellite Model

We consider the power and the on-board computer system of a research satellite. Its mission is the observation of the earth. The payload, the earth observation instrument, is turned off when the sub-satellite point is over the water, the instrument is turned on if the sub-satellite point is over land. This switching is done by the on-board computer. While the sub-satellite point is over land, data is sensed and stored onto an on-board hard disk. This data transfer is also controlled by the on-board computer. If an earth station becomes visible, the data stored on the hard-disk is transmitted to the earth. This is done as long as the the earth-station is insight of the satellite.

Power is generated by solar panels. The total solar flux absorbed by the panels determines their temperature. Via a lookup-table of the current bus voltage and the temperature the actual current is calculated. If more current is produced than necessary the supplement energy is stored in a battery. If the opposite is true, the battery serves the power-bus.

Power-Consumer is the on-board computing systems. During sensing, more power is needed than in standby-mode. The same hold for the transmitter to the earth-station.

4.1 SatLab-Model

The orbital modelling is done by means of the SatLab-Tool. SatLab provides actual sub-satellite point, visibility of the earth-station, sun-angle of the satellite and a flag which indicates whether the satellite is over land or water.

4.2 BONeS-Designer Model

The BONeS part of the model can be divided in two sub-models fed by the information from SatLab. The sub-models are the power bus model and the data bus model. The satellite model is depicted in Fig. 2.

The power model receives the solar angle from the SatLab model.The angle of the panels to the sun is calculated. This information is fed to the textbook equations, e.g.[3] describing the panel's temperature. The current temperature depends also on the old temperature, indeed we are performing an integration step.

Certainly this kind of integration constitutes the Euler-Cauchy algorithm, but, on system level this is for this example sufficient. The temperature-voltage characteristics are supplied via a look-up-table for one string. This procedure is done for every string and the available current is determined. This current is compared to the actual required current by the power control block. This block also implements the battery model which contains another integration algorithm.

The on-board computer software is modeled via finite-state-machines. A land-water edge causes an event to the software so that a corresponding event can be generated. This event is transmitted via a data packet to the earth observation instrument which is turned on. The sensing and transmission procedures are modeled in an analogous way.

4.3 Results

In initial simulation runs we got results for the temperature of the solar panel, the bus-voltage, the bus-currents alongside the bus commands and the commands on the data bus shown in Fig. 3: This plot shows that every time there is a land water crossing the associated command is issued by the CPU. Furthermore, the transmission of an image from the EOS-instrument to the hard disk is shown. If an earth-station is visible, some frames are sent to the earth station. A lot of parameter studies can be performed with this model. However, this has still to be done.

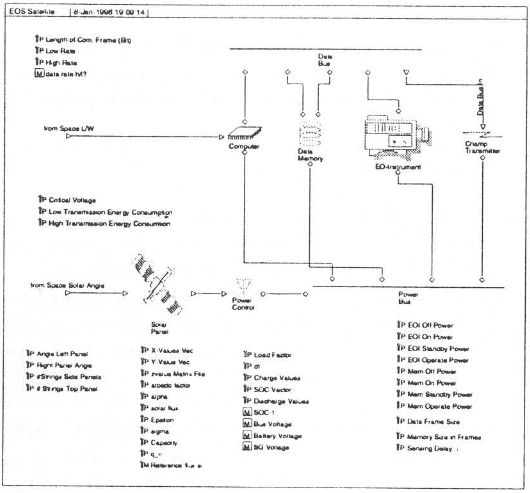

Figure 2: The satellite model

5 Conclusions

In this paper we showed how system level simulation can support the design of a satellite. The system model serves as an executable discussion basis between different design domains. Results can be checked against the requirements and influences between different design domains can be unveiled by animation. Further investigations include the development of an integrated system level simulation language capable of simulating more sophisticated integration algorithms alongside the discrete event features of BONeS-Designer and developing a common theoretical framework for system level description. First results can be found in [5].

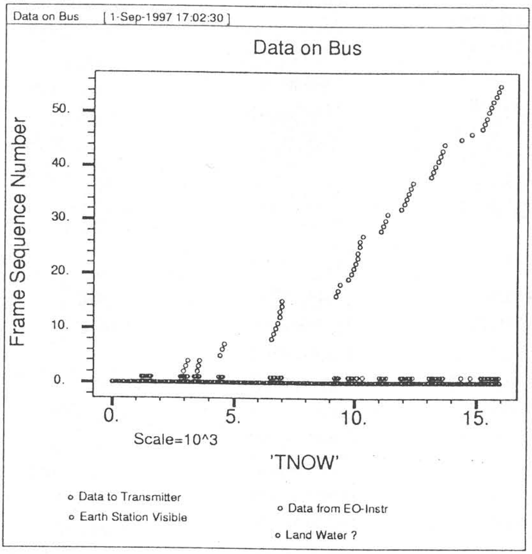

Figure 3: Packets on the data bus

Literatur

[1] K.D. Mueller Glaser, Systems Engineering in Microsystems Design, Proceedings of the IFIP Workshop on Modelling of Microsystems, Stirling, 1997

[2] J. Bortolazzi, *Untersuchungen zur rechnergestuetzten Erfassung, Verwaltung und Pruefung von Anforderungsspezifikationen und Einsatzbedingungen elektronischer Steuerungs- und Regelungssysteme*, Dissertation, Universitaet Erlangen-Nuernberg, April 1994

[3] P. Fortescue, J. Stark *Spacecraft Systems Engineering*, Wiley 1995

[4] Cadence Alta Group *BONeS-Designer Modelling Handbook* Sunnyvale 1995

[5] U. Freund, V. Zerbe *Hierarchical Simulation Techniques for the Design of Heterogeneous, Embedded Systems*, IFIP Workshop on Modelling of Microsystems, Stirling, 1997,

INDEX