Mechatronics simulation on system level: BroSAnT approach

Dr. Petkun
Sergey.Petkun@brose.com
Brose Fahrzeugteile GmbH & Co. Kommanditgesellschaft, Hallstadt
Max-Brose-Straße 1, D-96052 Hallstadt

Abstract

At present, the modelling of mechatronics system is “buzz” word in simulation world. There are a lot of different concepts how to get the problem, dealing with many physical domains: mechanics, electrics, electronics and software, under control. In principal, all approaches to model mechatronics system can be divided in two classes: “all in one” and “combination of domain specific tools”. Analysing the results of mechatronics simulation gives us already first hints to general requirement to such simulation: very important time of modelling for construction and simulation duration. The depth of modelling is the most important part for such simulation. Right relation should be found between precisions of modelling different counterparts of the whole model. Not less important is the criterion how close the model behaviour is to “reality”. The question “how precise is the model ?” is not correct without specified estimative criteria.

The system analysis has character of multi criteria because the model behaviour will be estimated from different point of views: mechanical, electrical and etc. from such point of view, **BroSAnT** (**Bro**se **S**ystem **An**alyse **T**ool) tries to realise the new conceptual solution. The origin of false thinking is the meaning that the system modelling should be made by experts in calculation of specific domains. In our opinion, just joint activities between the application experts from different domains and simulation developer can lead to successful modelling concept. Only from system view, the necessary modelling level of corresponding domain counterpart can be right determined. For successfully using of simulation, the time taken from problem definition until result analysis should be below as product development cycle. It requires a number of prepared models, which leads to choose modelling language, in which the model data base will be based and built.

1 Introduction

Every simulation should give some answers on the certain questions. Not a trivial question is what is waited from simulation of whole system. Under whole system, is meant some mechatronics system: system, consisting of electrical power source, energy transformer (electrical into mechanical), mechanical executive mechanism and intellectual control system (algorithms realised in software and controlling hardware). There are a number of approaches to simulate the components: solid multi bodies in mechanics, Matlab-Simulink in Algorithms/Software, Spice/VHDL-AMS modelling in hardware. If there are some models of different components in system, as first idea that comes is to combine these available models in whole system. It leads to co-simulation concept. There are a lot of techniques for co-simulation but it is not an optimal way for system analysis. The system analysis should be begun from question “is the system appropriate to its purposes?”.

During developing mechatronics systems it is highly recommended to optimize appropriation to constitutive components. Common functionality of whole system is connected with functionality of every component: if the system does not right work – from system point of view, it is not so important which component is defect – the result is clear: the device is defect. It doesn’t matter whether electronics fails or mechanics breaks down.

There is a false conviction that with help of advanced software the constructional errors can be corrected. The poor constructed mechanical structure can be brought to functional state, but compelled solution is not the best solution. The optimization on system level has a number of different and very often contradictive criteria. This contradictory consists in reducing price of component on the one hand and on the other hand in providing the required functionality. It is not seldom case when the device is defect only because of some trifle fails.

2 Requirements on system simulation solutions

Functionality under “all conditions”, described in specification, predestines the choice of parameters for mechatronics simulation: all working condition: climatic, constructive, dynamic and energetic (e.g. voltage fluctuation on the power source). It is clear that the whole range of all parameter variations can not be investigated by different reasons but it is known (and partly it is company know-how) which effects should be taken into account during modelling components.

All these questions to simulation can be yet further extended or reduced, depending on situation. For example, for some industrial project with huge number by mass production, the constructive tolerances take great importance which is not so interesting in the case of some single sample.

The most of mechatronics systems have some “adjustable” part in form of algorithms parameter set (firmware parameter part), which allows optimizing the system for certain “working range”.

To summarize all these requirements on mechatronics simulation, all input variation can be divided in groups:

- Constructive setting
- Climatic environment
- Dynamic environment
- Energy supply
- Algorithmic setting

The purpose of system is usually described in specification on system. The working environment is usually prescribed in the specification. For mechatronics system, it is, as a rule, ambience, dynamic and constructive tolerances. Under all these conditions, the system should be complied with the purpose determined in specification.

From this list, it can be already seen that thre are enough many parameters only at system level without taking component level into account. That is why, some flexible cable, working far from large deformation range, can be treated as only force transfer element with certain elasticity properties. So the complex constructive element will be reduced to some functional property as force transfer with definite stiffness.

Such approach is well known since a long time through modelling the electronic schemas where complex electrical behaviour of transistor will be replaced by behaviour model at working point. From system point of view, mechanics is nothing else as force-velocity transformer in dynamical system. From view behavioural modelling, the whole mechanical effects can be described as combination of elastic, inertial and frictional properties, describing these effects.

Elasticity is responsible for recoverable energy transfer, inertia – for time delay by the movement, frictional for energy loss. Naturally, all these properties strongly depend on conditions, mentioned above, but requirement in specification is also based on certain “working range”. Replacing the complex mechanical geometry through behavioural description is not simple task but brings enormous advantages with it.

To provide industrial projects with CAE support, the simulation should be “real time” able in sense of, not only fast calculation but also system building, overview- and analysis-able. Nobody needs simulation results after successful start of production. At present, the development of mechatronics system is connected with combination of prefabricated hardware components (electrical, mechanical) available on commercial market and that is why the importance of simulation at system level rises continually. The lack of domain compatible models leads to the development of new modelling languages as Modelica [1] or effort to describe the wanted components of system in term of old one as VHDL-AMS [2]. There is already first effort in standardisation of models on component level [3]. The electronics development sets a good example for future technology of development – the product is impossible to sell without accompanying model with it. The risk of implementation of no appropriated component will be reduced through virtual assembling test.

Successful and reasonable simulation of whole system is not possible without participation of domain experts in it, because the simulation engineers can not right estimate if the behaviour of the whole system reflects all necessary effects under investigation. The aim of simulation developers is to give appropriate component models with minimal number of parameters but with all necessary effects in it. It leads to the well-known solution: server-client application.

There are not so many problems connected with embedding control algorithms, because usually using platform specific C-code at end application phase allows to bind this code directly as “black box” in simulation model. Almost all simulators can use the “black box” with little adaptation by embedding it in simulation model. This feature is very important for system simulation because it give us some so necessary criteria for estimation how close the rest of model is to reality.

Very important point of system simulation is the ability of model verification. The focus of system analysis is dynamical behaviour, and that is why, the comparison with real measured behaviour must be possible. What can not be measured is not well appropriate for the estimating the quality of model.

The traceability of simulation runs is obligatory by system simulation. The number of parameters, even in the case of simple simulation, is not small and the number of possible investigation variants is very big.

There are some different meanings about GUI (Graphical User Interface) for end user. In our opinion, the same simulation interface should be adapted to end user for specific domains. By analyzing the same system, the experts from different domains have different points of interests to the same object and various analyzing methods: for mechanics the forces are in the focus of analyses, current or voltage can be in the focus of analysing for electronics.

System simulation is in certain sense also documentation for making some conceptual decision. That means the simulation results should be reproducible in a later time point.

The requirements to system simulation depend also on simulation users. System developers have other demands to simulation as system application team.

3 Requirements on modelling of system components

To divide system into components is not complex – it is determined by a construction of product. The constituent part of product can be used from self manufacture or bought on the commercial market. The most important feature is that this component can be non-destructively replaced in device. The second one is the part should possess some self-completed functionality.

The system analysing tool should be based on some data base with component models. It determines the choice of modelling language. It can be some standard language as VHDL-AMS or another popular one as Modelica or at all tool specific language like MAST for simulator Saber (Synopsys). The modelling of components is very important to have, if it is possible, the similarity to reality by choosing connection points. For example, for drive there are three connection points: two electrical and one mechanical. It gives possibility to use different drive models and exchange them in a light way. It does not matter if the used drive with another parameter set or at all some other models, maybe with another modelling depth. From system point of view, a drive has the main functionality to transform electrical energy into mechanical movement.

This sameness allows the product developer to use different drives by simple replacement of component. Already at modelling level the interface between system components should be clear determined.

There is big risk during determining the depth of component description to go down too deep in the component description. Component level should not be mixed up with system level. Domain experts are trying continually to deep in each own domain. The practical rule is following: main parameters, which are needed for component description on system level, can be found in the specification sheets of this component. It stands to reason that, if the number of parameters is not enough for self-consistent description of component, it means that this specification gives component supplier more freedom and can lead later to problems on the side system responsible. It is well-known case: the component is complied with its specification but system with it does not work right.

It is obvious that every additional requirement in specification is connected with price or opposition from supplier side but the depth of specification on the product shows “gold” practical middle between customer and supplier and also reflects current industrial state in this manufacturing branch.

Electronics development is leader in development based on CAE techniques and it can be useful to transfer huge practical experience that this domain already has, in other domains As evidence, it may be mentioned the rate of increasing the amount of transistors in modern CPU.

4 Objectives of the BroSAnT development

The development of **BroSAnT** is based on multi purpose basic:

- Personal education
- Keeping company know-how
- Part of project documentation
- Advantage in competence over competitors
- Project insurance at acquisition/development phase
- Proofing of new concepts on development phase
- Error analysing in mass production
- Saving manufacture costs by avoiding errors

The personal education in mechatronics area is not a short-period task and many as human so also hardware resources should be involved in it. With the help of virtual system, the learning personal can be quicker and cheaper to reach a higher professional state, which is a necessary condition for successful business today.

With permanent personal education, the company know-how is transferred from experts to younger generation and this know-how should be kept, also taking into account fast changing market situation and development state in the world independent from human resources form. Every effect modelled in the system should be remained in data base of company knowledge independent from human presence. It does not reduce the role of human factor but makes insurance factor higher.

Also corresponding documentation process belongs to consequent project loop. In documentation process it is important to have some possibility to recover simulation state and results for possible analysis in a case the manufacture problems

By rising globalisation in the world, the number of manufactures grows rapidly, almost every month, and the problem to choose the right component supplier becomes of great meaning. One of factors to distinguish among huge proposals on global commercial market is competence degree of the component manufacture. And from this point, a good simulation model is yet “another brick in the wall”.

It is not to wonder that wishes are rising quicker than performance. The customer should be consulted not only emotionally (“we can do it”) but rather technically (let analyse the simulation results). Sometimes, the customer requirements are contradictory but without technical analysis it is invisible. The system analysis tool is a good helper in this case to avoid excessive discussion.

Innovation by development is a locomotive of the progress but only flawless development and production lead to commercial success and sure business. To avoid errors by new development is one of the main tasks of every company. The virtual development is one of many steps in this trend. That is why, there are so many simulation concepts are proved at present. **BroSAnT** gives the possibility to prove the technical innovations at early development phase without big expensive or with relative low costs.

Every error in development brings additional costs. However, it is impossible at all to develop without some errors which occur sometimes at different phases of manufacture: it can be connected with some changing of technological process or supplier. Some improvements in the manufacturing tool can have side effects which are not taken into account and so on. Any simulation tool can not find the reason itself but rather it can be served as assistant instrument to indentify cause for this failure. It should be clear, that any tool is only facility but rather "wonder solution" for all problems. The quality of using tool is the same important as the professionalism of the personal working on it and tight cooperation between simulation and domain experts.

It is very important to use simulation analyses at right development phases. The simulation costs also money and time, which means at the end also money.

It is well known that the costs of error correction being close to start of production rise almost exponentially. That is why the failure analysis (failure prediction) at early development or application phases is almost obligatory for every commercial project.

Last but not least are the expenses for simulation tool. To be profitable, the simulation tool should be used as extensive as possible, which is reachable either with many experts, constantly using this tool, or with organization of some calculation task pool for it. The software manufactures are trying to sell as many software as possible, promising the increasing the quality of development, but by meanwhile they are silent about expanded costs for personal educational, maintenance and so on. The complexity of tools is enhanced and the average efficiency of using the tools, without correct usage, is diminished. The huge problem in developing software tool is different representation between industrial demands by end-user of this tool and possibilities by developer of software.

5 Short description of BroSAnT conception

From reason above mentioned, the server-client concept is ideally appropriate for realisation of system simulation. To give domain experts some possibility to use the simulation tool, it is necessary to hide complexities of simulation and modelling techniques from them. The client part should be independent, as possible as it can, from simulation environment. The focus of analysis should be laid on domain problem but rather simulation nuances.

According to this concept, the grey boxes do not have any environment of simulator and almost do not connect with peculiarities of some simulation knowledge.

During the task preparing for simulation, technical knowledge of simulation object is desirable. The default setting for component is suitable for reasonable functionality of basic model. The parameter changing requires the exact understanding about what will be done.

The most investigations are done in the time range because the dynamical behaviour is the focus of analysing typical mechatronics system.

The client part (Figure 1) is written in Tcl/Tk [4] and principally consists of two parts: results analyser and simulation configurator. The first part presents ordinary twodimension curve browser with zoom and measure facilities. The second part is responsible for simulation content: from choosing model of whole mechatronics system up to changing component parameters and necessary settings for output. At the beginning of building simulation, the macro model (device) for simulation should be chosen the first of all. After this step, the user has possibility to build the selected system from different components available in data base.

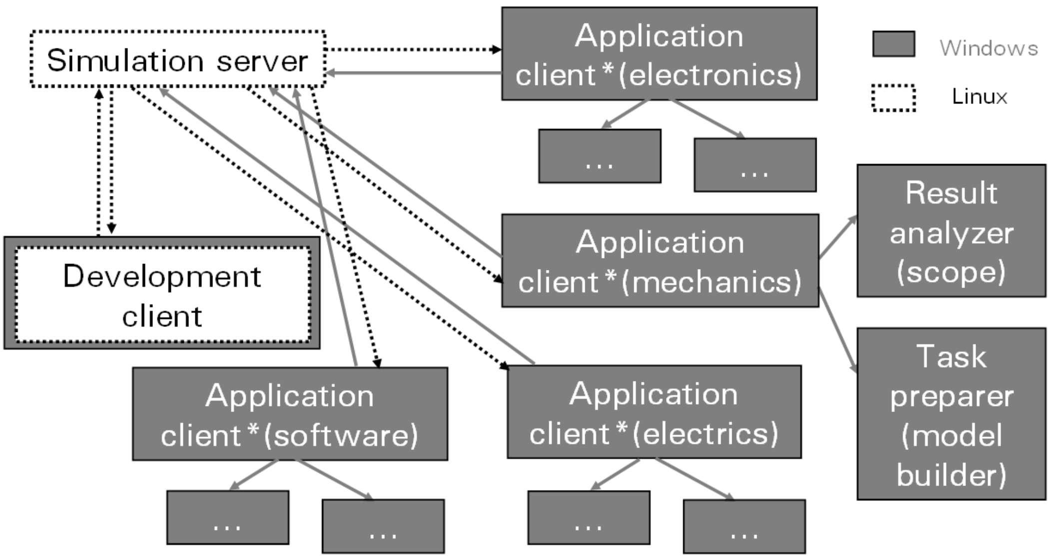

Figure 1: Simulation concept of **BroSAnT**

The main focus of application is modelling electrical power window regulator. It is classical example for modelling mechatronics system, because the window regulator possesses all components that are needed for mechatronics system: electronics, drive, mechanics and control part.

According to Albert Einstein "… we should make things as simple as possible, but not simpler", there are controlling module and also power source for it. In controlling part, the switching sequence as time or target function should be given.

Figure 2: Toolbar of BroSAnT

There are also a number of simulation protocols for tracking the simulation runs. The big amount of simulation possibilities should be cleared through analysing history of the parameter changing. The possibility to save user-tuned component model with help of parameter is realized by means of user library concept. All date base and personal library be kept at server space and can be use from any place if the access to calculation server is granted. On the toolbar (Figure 2) beside zoom/measure and different log elements, there are two important managers: simulation manager and file manager, which allow users to work with simulation results and also with measured data.

The server part is implemented in Linux, the best tradition of server, and is responsible for simulation runs/queues. To server task belongs also the service for component data base, model administration and the protocol of whole simulation processes. Thanks to the server-client solution, calculation duration is independent from client side. All calculations are performed at server and the single restriction on calculation time is the amount of required results. Practically, simulation duration is comparable with “real time”. Of course, it depends on simulated model and required precision.

At present the whole modelling is implemented in MAST [5] modelling language and is planed to transfer on VHDL-AMS language. With it, higher free degree of freedom in selecting simulator manufacture can be reached. The simulator developer does not really support such simulation concepts, because today almost all simulators are domain oriented. Hence, there are two schema of using the simulation methods: the domain specialist makes the simulation itself based on own demand (as a rule in electronics and in algorithm development) or simulation results are produced by simulation experts according to demands from domain specialists (usually other domains). At present, Saber simulator from company Synopsys is used as simulation engine.

6 An example of simulation with BroSAnT a electrical power window regulator model

By choosing a power electrical window regulator as mechatronics system, the

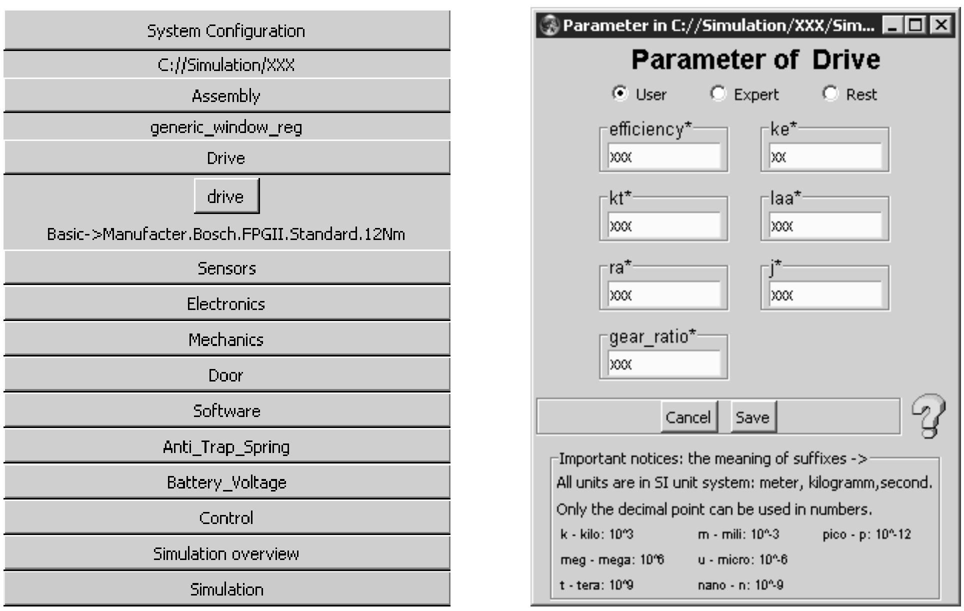

Figure 3: Component’s part of the **BroSAnT** configurator

simulation configurator gives all necessary components to combine the whole system : different drives with sensors, mechanics on the cable basis or cross arm construction, electronics part with switches, corresponding sets of possible software algorithms, carrying door model with anti trap measure gauge, source of power like car battery and necessary controlling switch sequences. At the Figure 3 this part of configurator can be seen and one component has its open parameter set. It is drive consisting of DCPM (Direct Current Permanent Magnet) motor and worm gear.

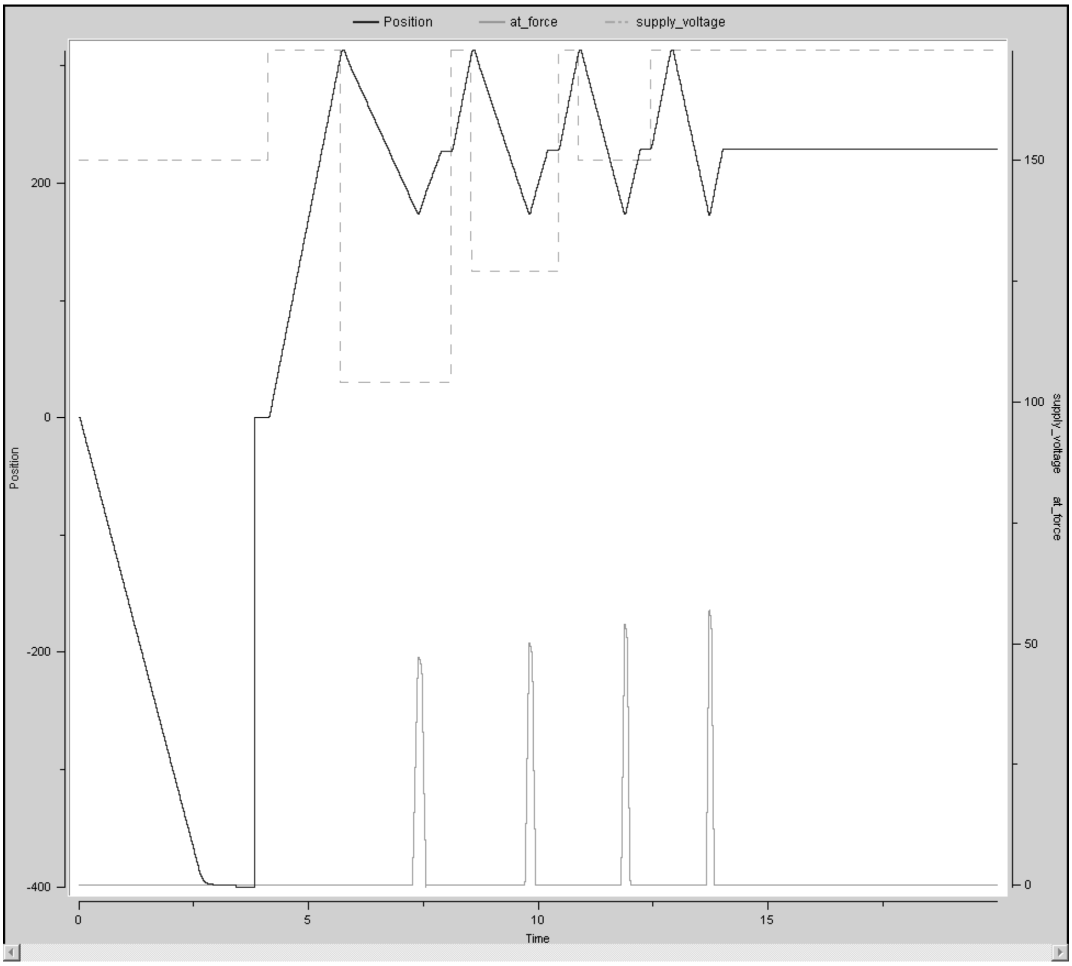

Figure 4: Result of the window regulator working against obstacle (10N/mm)

By choosing a power electrical window regulator as mechatronics system, the simulation configurator gives all necessary components to combine the whole system : different drives with sensors, mechanics on the cable basis or cross arm construction, electronics part with switches, corresponding sets of possible software algorithms, carrying door model with anti trap measure gauge, source of power like car battery and necessary controlling switch sequences. At the Figure 3 this part of configurator can be seen and one component has its open parameter set. It is drive consisting of DCPM (Direct Current Permanent Magnet) motor and worm gear.

After simulation run the results is delivered from server to client and at the Figure 4 shows the position (in electronics unit) at left axis and maximal reached anti trap force for this system under such condition. There are four anti trapping cases and it can be seen the four different value of anti trap force before the window glass was reversed (right axis). It is connected with different supplying voltage by anti trapping (car board voltage - 10 12 14 16 Volt). The supplied voltage to the motor is shown as dashed curve in conventional units on right axis. From this curve it can be concluded that voltage compensation module in algorithms is switched off. The simulation results reflect the behaviour of the system in reality. Such tests can be made with the real hardware too but sometimes it is cheaper and quicker to prove some ideas virtually, especially, if hardware does not exist yet.

If the system is combined, there are many possibilities to test it from various domains. An expert from a domain can study what consequences can be happened under one or another changing in his subsystem/component or how stable his component remains under different working condition.

With practical experiences using the **BroSAnT**, it can be stated that different domains have each own imaginations about what is the most important in system and how it should be studied. A number of practical rules can be concluded:

- simulation results without exact information, about under which condition simulation was done, are not only useless but rather harmful
- the same with validity of test data, it is only reasonable to analyse measured data if the reliability more or less was confirmed
- the most useful information is what is known from daily business and on the contrary, what is not known is no more then interesting but not especially useful

Also after cooperative working, the system simulation developer and product management, a couple “important” remarks come in sight:

- attention to possible problems rises exponentially with rising number of failures in the field or by mass production
- avoided errors and failures will not be taken into account by profit calculation
- every end user wishes each own interface

7 Acknowledgements

I would like to thank Joachim Haase, Roland Kalb, Detlef Ruß and Xin Zhao for their valuable feedback and advices. My great acknowledge to George A. Howlett and ActiveState team for their excellent package for 2D graphics - BLT /RBC

8 Literatur

[1] A Unified Object-Oriented Language for Physical System Modelling: Language Specification Version 3.2. Modelica Association, March 24, 2010. URL: https://www.modelica.org/documents/ModelicaSpec32.pdf

[2] Behavioural languages - Part 6: VHDL Analog and Mixed-Signal Extensions. IEC Std 61691-6/IEEE Std 1076.1, Dec. 14, 2009. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5465882&isnumber=5465881

[3] SAE J2546_200202- Model Specification Process Standard. SAE Electronic Design Automation Standards Committee, February 15, 2002. URL: http://standards.sae.org/j2546_200202

[4] ActiveTclHelp8.5.chm

[5] Saber®MAST Language, Book 2, User Guide, Version W-2004.12, December 2004. URL: http://bbs.dianyuan.com/bbs/u/27/1111050123.pdf