Proceedings
of the 4th International Modelica Conference,
Hamburg, March 7-8, 2005,
Gerhard Schmitz (editor)

C. Nytsch-Geusen et. al.
Fraunhofer Institutes, Germany
MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics
pp. 527-535

Paper presented at the 4th International Modelica Conference, March 7-8, 2005,
Hamburg University of Technology, Hamburg-Harburg, Germany,
organized by The Modelica Association and the Department of Thermodynamics, Hamburg University of Technology

All papers of this conference can be downloaded from
http://www.Modelica.org/events/Conference2005/

MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics

Christoph Nytsch-Geusen[1]
Thilo Ernst[1] André Nordwig[1]
Peter Schneider[2] Peter Schwarz[2]
Matthias Vetter[3] Christof Wittwer[3]
Andreas Holm[4] Thierry Nouidui[4]
Jürgen Leopold[5] Gerhard Schmidt[5]
Ulrich Doll[6] Alexander Mattes[6]

[1]Fraunhofer Institute for Computer Architecture and Software Technology
Kekuléstr. 7, D-12489 Berlin, christoph.nytsch@first.fhg.de

Fraunhofer IIS/EAS[2], Fraunhofer ISE[3], Fraunhofer IBP[4], Fraunhofer IWU[5], Fraunhofer IPK[6]

Abstract

The current GENSIM project, which is being conducted by a consortium of six Fraunhofer Institutes, is developing the generic simulation tool MOSILAB for the analysis of mixed time-continuous / time-discrete (hybrid) models of heterogeneous technical systems. One major innovation here in terms of simulation technology is the mapping of state-dependent changes in the model structure (model structural dynamics). This enables, for example, simulation experiments to be conducted with models of variable modelling depth. The modelling description language in the project MOSILA is based on Modelica, which was extended syntactically in terms of an adequate description of the model structural dynamics. The simulation tool is composed of a kernel and an integrated development environment and will be available in spring 2005 as a first prototypical implementation. The usability of the simulation tool is tested and evaluated in the GENSIM project by means of three use cases in the application areas fuel cell systems, hygrothermal building analysis and cutting tool systems.

Keywords: MOSILAB; Generic simulation tool; Model structural dynamics; object-oriented

1 Introduction

A heterogeneous technical system shows in dependency of its state a different physical behaviour. For example the physical behaviour of a starting plane in the different phases of rolling, taking off and flying can be described with different sets of physical effects, like the air and roll friction on the earth and the aerodynamic laws in the sky. An adequate simulation model for such a technical system needs also a high level of flexibility and adaptation in its model structure and in its equation system.

The innovative goal of the GENSIM project is to develop a new generic simulation tool for hybrid models, which supports model structural dynamics. Model structural dynamics in this context means, the model structure (the number and types of equations) can change during the simulation experiment in dependency of events, which are triggered from the state of the model self or its environment.

The object- and equation-oriented simulation language Modelica (http://www.modelica.org) offers in principal a good language concept for modelling technical systems with structural dynamics. For this reason Modelica was chosen as the language basis for the GENSIM simulation tool. Because the actual specification of Modelica is limited to fixed model structures during the simulation experiment, some syntactical extensions were made in GENSIM to obtain the possibility for describing model structural dynamics in a compact form.

2 Modelica Language Extension

The modelling description language MOSILA (**Mo**delling and **Si**mulation **La**nguage), which is specified and used in the GENSIM project, is based on Modelica. From the view of the modeller MOSILA is mainly an extension of Modelica.

Thereby existing models and also the disposable Modelica standard library can be reused within the GENSIM simulation tool directly or with a small effort of adaptation.

However the means of expressions of Modelica, particular for the description of variable model structures, are not powerful enough yet for using special simulation technologies, e.g. variable modelling depth. Therefore some syntactical extensions are added in MOSILA. These extensions where influenced by the UML^{H} [1], an adaptation of the UML [2] for the context of hybrid systems.

2.1 Dynamical object structures

Dynamical object structures were introduced to represent variable models during the simulation experiment. Thus, it becomes possible to extend the static model tree with dynamical objects during discrete phases of an experiment, which them self can spawn complex model trees. Since objects represent state attributes and behaviour in form of equations, the underlying equation system can be changed in size and quality when a structural change takes place. After such changes a new equation system will be derived for the following continuous phase.

2.2 Object-oriented Statecharts

To ease the description of structural changes, an adequate syntax for the control of discrete model switches were realised on the base of object oriented statecharts. Figure 1 shows a statechart controlling the mode switches of a landing device:

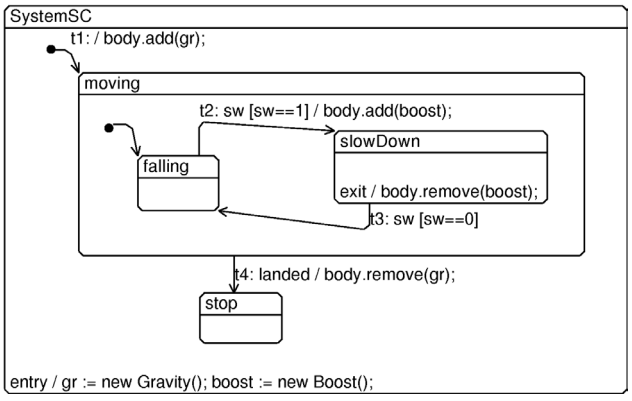

Figure 1: Statechart of a landing device

At the beginning the device enters the mode “moving” and within its sub-mode “falling”. If the booster is enabled (depending on the decend speed), than it enters the “slowDown” mode until the booster is disabled. When the device reaches the ground the “moving” mode is left and “stop” is entered.

To simplify the modelling process direct support for these statechart descriptions was introduced in MOSILA as a special section of a model class. This extension is based on the Modelica type system: Each state is introduced by a state declaration. Within such declarations sub-states and transitions between these sub-states can be specified. Depending on the base type of a state two kinds of state compositions can be declared: Within an active XOR state only one direct sub-state is active at each time instant. Within an active AND state all its direct sub-states are active. Therefore, parallel and sequential processes can be comfortably modelled. Actions which have to take place during switching transitions are defined within the associated transition definition. Figure 2 shows the MOSILA implementation of the above introduced statechart:

```
model System
...
statechart
 state SystemSC extends State;
  state Moving extends State;
   state SlowDown extends State;
    exit action
     body.remove(boost);
    end exit;
   end SlowDown;

   State falling, start(isInitial=true);
   SlowDown slowDown;

   transition start -> falling end transition;

   transition t2 : falling -> slowDown
    event sw guard sw==1 action
    body.add(boost);
   end transition;

   transition t3 : slowDown -> falling
    event sw guard sw==0
   end transition;
  end Moving;

  State stop,start(isInitial=true);
  Moving moving;

  transition t1 : start -> moving action
   body.add(gr);
  end transition;

  transition t4 : moving -> stop
   event landed action
   body.remove(gr);
  end transition;

 end SystemSC;
end System;
```

Figure 2: Implementation of a statechart for the control system of the landing device

The state space for action statements, e.g. assignment, is given by the surrounding type definitions, and thus the statechart acts on the attributes of the associated class.

2.3 Dynamical behaviour

Further, the extended language concept offers an infrastructure, which enables the extension of (basic) model by effects in form of behavioural objects. For this purpose the action language is extended by the operations "add()" and "remove()", which connect/disconnect the given argument (behaviour object) to the target object (base object). To extend underlying balance equations, a new connector type "sum" is introduced. The semantics of this connector type is like zero sum ("flow"), but with a negatively signed base attribute. Thus, balance equations can be extended by terms, which are encapsulated by objects. The following implementation (Figure 3) shows the environment of the landing device:

```
connector FPort
 sum Real F=0;
 Real m=0;
end FPort;
partial model BodyInterface
 FPort p;
end BodyInterface;
model Body extends BodyInterface(p.m=100);
 Real a=0, v=0, s=100;
equation
 der(v) = a; der(s) = v; a = p.F / p.m;
end Body;
model Gravity
 extends BodyInterface;
 parameter Real g=9.81;
equation
 p.F = - p.m * g;
end Gravity;

model Boost
 extends BodyInterface;
 discrete Boolean empty=false;
 Real m;
equation
 p.m = m;
 empty = (not m>20); // = if m>20 then false else
true;
 der(m) = if empty then 0 else -10;
 p.F = if empty then 0 else 1200;
end Boost;
model System
 Body body;
 Gravity gr;
 Boost boost;
 event discrete Integer sw=0;
 event discrete Boolean landed=false;

equation
 sw = if body.v < -5 then 1 else if body.v >= 0
then 0 else pre(sw);

 landed = ( body.s <= 0 ); // = if body.s <=0
then true else false;
...
end System;
```

Figure 3: Implementation of the dynamical behaviour of the landing device

The base model "Body" and the effects "Gravity" and "Boost" have the same interface "BodyInterface", which introduces the variables to connect/disconnect during add/remove operations. The attribute "F" has "sum" quality since it is used to model a (dynamically changed) balance equation. The events "sw" and "landed", which drive the above introduced statechart, are modelled within the top level class "System" as special discrete variables.

3 The MOSILAB Simulator

3.1 MOSILAB Architecture

The GENSIM simulation tool MOSILAB (**Mo**deling and **Si**mulation **La**boratory) includes the simulation kernel (consisting of a model compiler, a runtime system and a numerical solver framework) and an IDE (Interactive Development Environment), the interface to the user of the simulation system. It supports him both in the modelling process with the help of graphical UML and text editors and during the simulation experiment.

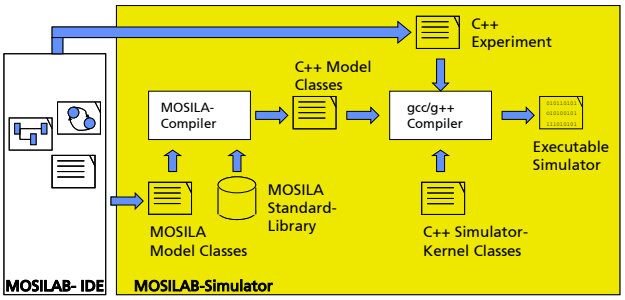

Figure 4: Data flow within MOSILAB

Figure 4 shows the data flow within the MOSILAB tools: Beside experiment definitions, the models also developed within the IDE are stored as MOSILA model classes. Together with the MOSILA standard library, these MOSILA models are compiled to C++ classes by the MOSILA compiler. Using the GNU gcc/g++ compiler, the executable simulator is built from these C++ representations and the simulator kernel classes.

3.2 Numerical solver framework

The numerical solver framework of MOSILAB features general functions such as the construction of the numerical model based on the modelling description, the main simulation control loop and is able to integrate different numerical algorithms.

The simulator kernel library contains some basic algorithms for solving nonlinear implicit differential-

algebraic equations (DAE): the EULER backward formula and trapezoidal rule as simple methods from standard textbooks, and IDA, a powerful public-domain DAE solver for sequential and parallel computers [3]. IDA is a successor of DASSL, a well established DAE solver [4, 5]. IDA provides a routine that computes consistent initial conditions from a user's initial guess for a class of problems - a very important feature for simulating systems with model structural dynamics. The integration method in IDA is a variable-order, variable-coefficient BDF method (backward differentiation formula). The nonlinear systems are solved with some form of NEWTON iteration.

But the library is an open one: according to the requirements of a dedicated MOSILAB implementation or of special problem requirements, additional solvers may be implemented, e.g. data-flow or event-oriented methods, or simplified solvers for linear state-space equations in explicit form.

Such a tailored algorithm, which is implemented in the GENSIM project, is the Plug-Flow method [6, 9]: A so called plug flow model uses finite mass elements $\Delta m=m'\cdot\Delta t$ and finite energy elements $\Delta Q=Q'\cdot\Delta t$ with a fixed time step, allowing the mass flow through closed and opened networks to be traced. This guarantees a fast calculation of the object chain. The plug is initiated in a "pump" or "ventilator"-object, then shifted through the branched network and is returned to the origin (pump or ventilator). This mechanism allows simple modelling of flow delay effects and mass balance at the second call of the pump/ventilator in one time step. Due to the decentralised solver method the state equations of each object are calculated with the updated mass flow of the previous object.

3.3 MOSILAB Configurations

MOSILAB can be configured in to act in three modes:

a) The generated simulator is represented by a single, monolithic C/C++ application. This option has the smallest memory footprint and only few dependencies on the underlying platform, so it is most useful e.g. for embedded applications. However, the functionality w.r.t. dynamic parameterization at runtime is limited.

b) The simulator is represented by a shared object file which can be dynamically linked to a main program which controls the simulation. MOSILAB uses the Python language and interpreter (http://www.python.org) as its central mechanism for experiment control. The simulator is loaded as an "extension" into the interpreter, and "experiment scripts", written in Python, access the simulator API via a Python-level interface.

c) The simulator acts as a service. In this mode, the simulator is linked with appropriate libraries to publish its API via standard TCP/IP-based protocols such as SOAP [5] in a web or grid services framework (e.g. the upcoming release 4 of the Globus Toolkit [6]). In this mode, the simulator can easily be controlled in protocol-based, platform-independent manner, and it is easy to deploy multiple (and potentially large numbers of) "simulator service instances" in a coordinated way in a heterogeneous network or Grid, for instance to solve an optimization problem. Python-based experiment control support is available in this mode as well – a (Python) client library is used to talk to the simulator's API over the network in this case. The simulator maintains a run-time representation of the model object hierarchy (as defined in the source and evolving according to the structural variability of the model). This run-time model can be inquired via introspection features of the simulator API, so (using the synchronisation features offered by this API, too) experiment scripts are able to follow the structural changes over the entire course of a simulation run. This way, if special reactions to model structure changes are needed, which cannot be formulated in the model itself due to their complexity, such reactions can easily be implemented in the experiment script.

3.4 Simulator Coupling

Besides the service-oriented approach to coarse-grained coupling of simulation components described in item c) above, MOSILAB also supports simulator coupling on a fine-granular level. In addition to implementing the standard external function interface defined in Modelica, special interface support is being developed to support coupling with the widely used simulator MATLAB/Simulink and certain specialized simulators relevant to the pilot applications (e.g. CFD and FEM tools). These developments, too, build on the flexible simulator API offered by MOSILAB'S simulation kernel.

4 The MOSILAB Development Environment

The MOSILAB Development Environment (MOSILAB-IDE) supports the user during the modelling process and the simulation experiment.

In the modelling mode the user can choose between three graphical UML^{H}-editors (class diagrams, collaboration diagrams and statecharts) and a text editor. While the graphical views give the user an intuitive overview about the structure and the logic of a complex model, the text editor offers the user features like syntax highlighting for implementing the MOSILA/Modelica models.

In the experiment mode of the MOSILAB-IDE the user can define the root model for the simulation experiment, can parameterize model variables and can choose and configure a suitable numerical solver. Furthermore he can define a subset of model variables, which should be observed during the simulation experiment. The observed variables are the basis for different types of post-processing. Figure 5 shows a screenshot of the prototypical implementation of the MOSILAB-IDE.

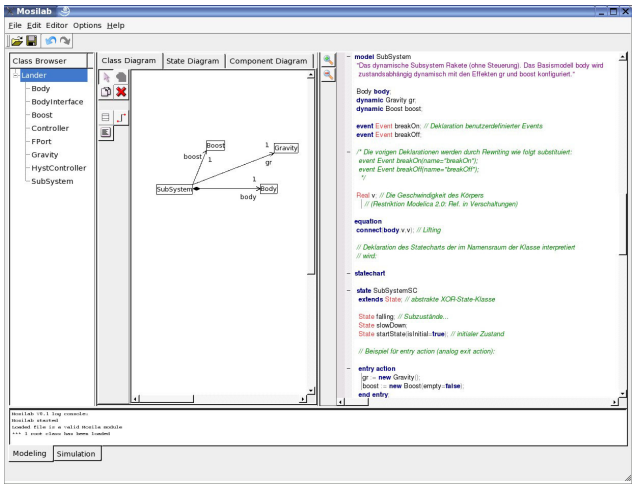

Figure 5: MOSILAB-IDE in the modelling mode

5 Applications

In the GENSIM project model libraries for the three technical application areas fuel cell systems, hygrothermal building analysis and cutting tool systems are developed. On their basis different use cases should be analysed. In each use case the methodological possibilities of the model structural dynamics will be evaluated: For example cutting tool system models are developed, which can activate different physical sub models for tools und working pieces in dependency of the system state during the simulation experiment (e.g. contact between the tool and the working piece exists or not).

5.1 Fuel cell systems

The future structure of power grids will consist of a huge fraction of decentralised power generators. Especially the low voltage grid will be penetrated by small and medium photovoltaic systems, medium combined heat and power units based on natural gas or bio fuels as well as residential fuel cell cogeneration power systems. Dynamic simulation of the entire low voltage grid offers the possibility of analysis and optimisation of the grid in terms of dimensioning and system management.

If cogeneration systems (e.g. residential fuel cell systems) are regarded, thermal aspects have to be considered. The ecologic and economic evaluation of these innovative energy supply systems needs efficient models due to seasonal effects and the necessity of simulation runs in the range of one year [9, 10].

Model structural dynamics allows the investigation of a huge number of grid connected residential fuel cell cogeneration systems in combination with other decentralised energy systems such as photovoltaic, small wind turbines, bio mass systems, etc. in a very efficient way. In this approach the model depth is defined by the operating point and the operating behaviour respectively.

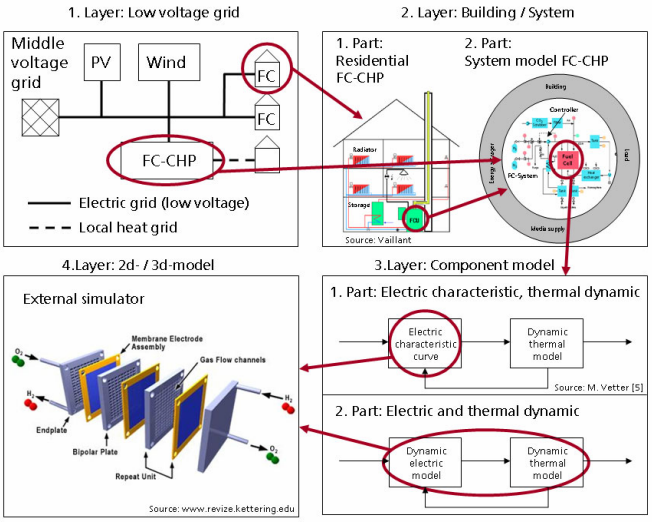

Figure 6: Definition of the single layers of the residential fuel cell system representing the model depth. The current model depth will be switched in dependence of boundary conditions and control actions.

As showcase the model structural dynamics of a fuel cell system is described in the following section. In a

first approach four layers of abstraction are defined, shown in figure 6. The current layer will be switched, if discrete events (e.g. caused by a state variable crossing a threshold) require the change of the model depth. In case of a stationary operating point of the fuel cell system, it is sufficient to represent this device with a simple characteristic curve. As soon as boundary conditions (e.g. cooling temperature) are changing or the operating point is changed by a controller, more detailed models are needed to reproduce the transient behaviour in an accurate way.

Furthermore, critical system states can require detailed models, with which even single cells can be investigated. As an example flooding effects at low operating temperatures or at high load currents shall be mentioned. If a system simulation tool is able to reproduce even such effects, efficient control strategies can be developed to reduce or even avoid system failure. In this project the coupling with external FEM tools are planned to fulfil these demands.

5.2 Hygrothermal building analysis

In the area of Building Physics hygrothermal models of building envelopes to compute coupled transport processes of heat and moisture for one- or multidimensional cases are widely used. In those models however the boundary conditions of heat and moisture have to be user-defined before starting the simulation.

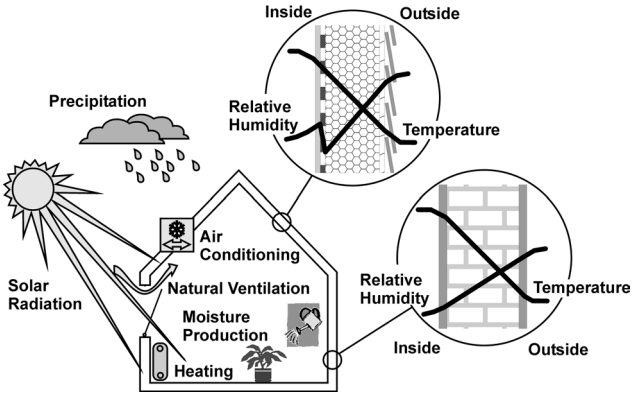

Figure 7: Coupled hygrothermal physical effects in the building envelope

A model that would take into account a multizonal building or even only a single room and the building envelope in detail – thus rendering the definition of the inner boundary conditions for the envelope unnecessary – is still to be defined. Such a model would allow analysing cases with strong reciprocal effects between the climate in the room and the behaviour of the building components (see figure 7). For example the influence of moisture buffering and non buffering surfaces of the components in combination with different ventilation strategies can be investigated to consider the efficiency of thermal drying and ventilation strategies to keep the indoor climate (especially the humidity) in a favourable range.

For this reason Fraunhofer IBP and FIRST started the development of such a new hygrothermal building model within the GENSIM project. Fraunhofer IBP can make use of its extensive experiences with the development and experimental validation of the simulation tool WUFI [11] for the detailed simulation of hygrothermal behaviour of building components. Fraunhofer FIRST has long-year modelling experience in the area of thermal building simulation with the generic and object-oriented simulation environment SMILE [12].

The goal in GENSIM is the development of a Modelica /MOSILA model library, which will contain models of one- and of two-dimensional coupled heat and moisture transport within wall constructions, a thermal/optical window model, a hygrothermal air volume model, a thermal/optical room model, an environment model for the climatic boundary conditions as well as an inhabitant model. From these models it is possible to set up configurations of rooms or whole buildings in a very flexible way by using the object-oriented modelling method. For example figure 8 shows UML^{H}-class diagram for an outside thermal wall model, which is a part of the model library building for hygrothermal building simulation.

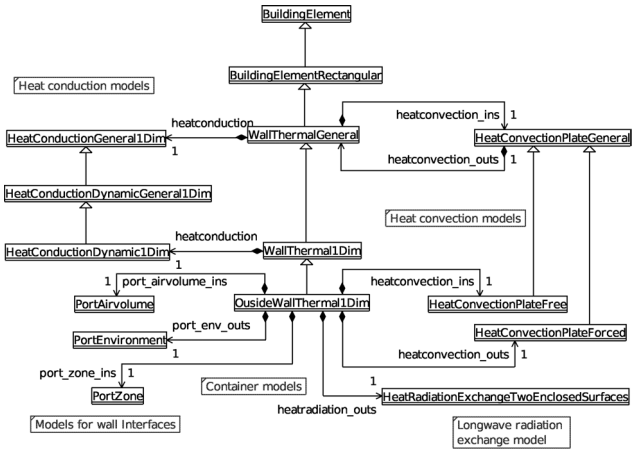

Figure 8: UML^{H}-Class diagram for an outside thermal wall model

The efficiency of the model structural dynamics should be evaluated for the coupled transport processes of heat and moisture in wall constructions: For example, if the gradient of temperature or moisture becomes greater then a limit-value, the level of dis-

cretisation of the wall model will be set on a higher value or the other way around.

During the project the hygrothermal building model will be validated on test houses. Two rooms which are identical both geometrically and in respect of solar gain and outdoor climate, however differ extremely in the sorption behaviour of their wall surfaces are used to validate the building model (see figure 9).

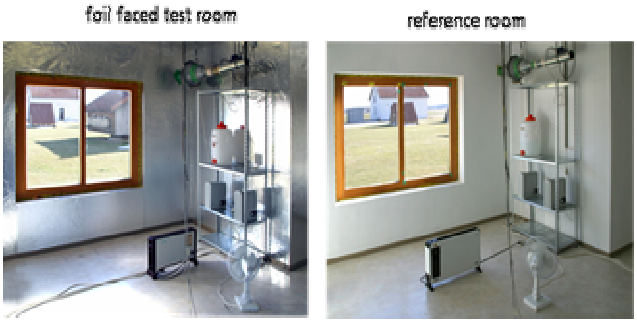

Figure 9: Test rooms for validating the hygrothermal building model

This is to be done by measurements of the energy and moisture balance in both rooms during cycles of heating and cooling as well as humidification and dehumidification.

5.3 Cutting tool systems

The development of high performance cutting processes requires, along with suitable machine tools and clamping devices also specially balanced and designed cutting tools. Safety and precision are the essential criteria at the judgment of tools for the high speed processing. Numerical simulations offer the possibility to evaluate different variants already in the outline process without existing samples. A proved method is the analysis of the tool behaviour under operating conditions with the finite element analysis (FEA).

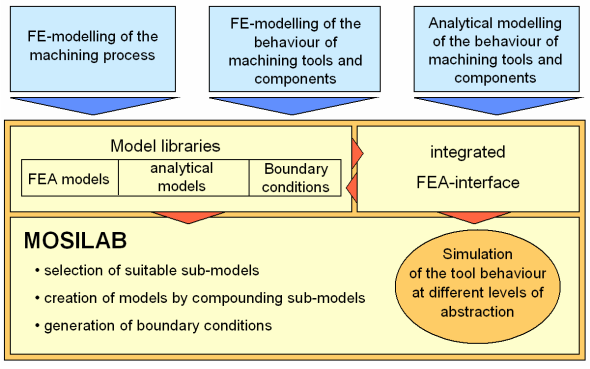

Figure 10: Cutting tool modelling in the framework of MOSILAB

The complexity of requested models depends on workpiece and the shaped elements and also from the type of the tool and its loading. Complex cutting tools consist of several components. Due to relative motions of the components under high centrifugal force load, cutting forces and clamping stresses, the tool models are highly non-linear and heterogeneous.

Different stages of the loading by clamping, centrifugal and cutting forces cause structural dynamic model behaviour and require corresponding changes of model parameters or even switching between different types of models (Figure 10). In the GENSIM project analytical models of complex cutting tools will be developed to be integrated in MOSILAB. FEA will deliver the parameters of these models. Respecting the complex structural dynamics different sets of sub-models are required to compile adequate cutting tool models. In addition to this, a special interface for FEA and MOSILAB data transfer will be developed for these types of cutting tools, which can not be simulated by a homogenous analytical model.

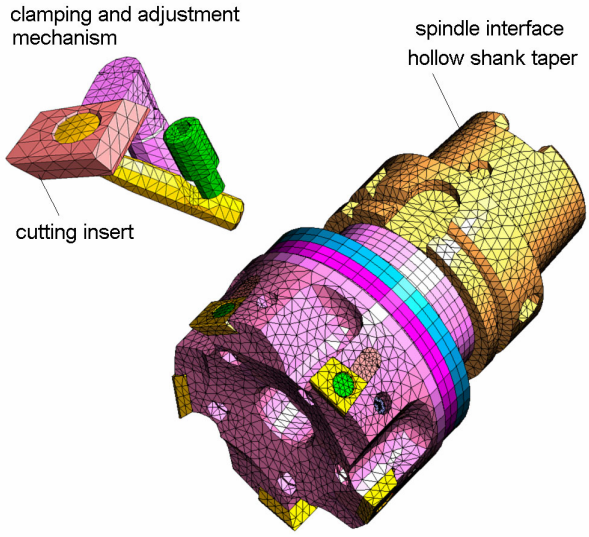

Figure 11: Example of a complex cutting tool

The simulation of cutting tool behaviour in the GENSIM project covers the complex tool – starting from the cutting edge up to the spindle interface (see figure 11). The behaviour of the cutting tool in use is determined by the statically and dynamical components of the cutting force. These loads are essential boundary conditions to investigate the structural dynamics of the cutting tool itself.

To get these loads, the cutting process itself is simulated using another FE-model. Here a small section of the cutting process is represented in which the chip formation happens. As seen in figure 12 the outer edge of the cutting tool and the upper layer of the workpiece are modelled.

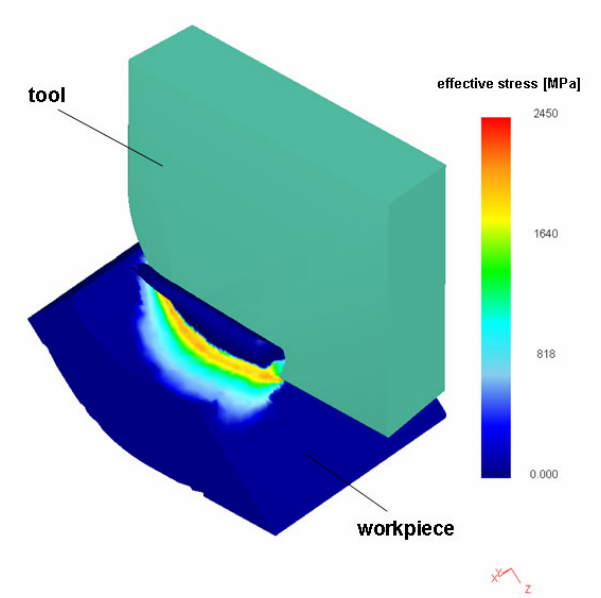

Figure 12: Simulation of the machining process

Through the simulation continuous or segmented chip formation can be described and the distribution of the resulting cutting forces can be calculated connected to the simulation time. Here from a three-dimensional load vector is determined, which acts on the cutting edge of the tool.

Within MOSILAB the simulation models should be combined to a simulation tool which describes the model structural dynamic of the interaction between cutting process and tool holder. For that a library with different combinations of tool holder, cutting tool material and workpiece material will be provided by MOSILAB. If in the cutting process the cutting force reaches limits, which indicate a meaningful influence on the tool holder (vibrations for example) the simulation depth can be changed [13]. This enables analysing changes in behaviour of the tool holder caused by microscope events happening in the region of chip formation. Also the results of the tool holder simulation are useful to include changes in the boundary conditions of cutting process caused by displacement of the tool holder itself.

6 Conclusions

The important results of the GENSIM project are the Modelica language extension for an adequate description of model structural dynamics, the new generic simulation tool MOSILAB and three model libraries of different technical applications:

a) **Modelica language extension MOSILA**: The new modelling description language MOSILA, which is mainly an extension of Modelica, is able to describe simulation models with a time depending model structure during the simulation experiment. This was realised by using dynamical object structures together with object-oriented statecharts for the language specification. An example for this new simulation technology in MOSILA is an adaptive simulation model containing a set of physical sub-models, from these some are activated or not in dependency of the state of the model self.

b) **Simulation tool MOSILAB**: The new developed generic simulation tool for hybrid systems, which includes a model compiler, a runtime system and a numerical solver framework, is able to translate MOSILA or Modelica models to an executable simulation program. Hereby the user is supported by the MOSILAB development environment, which offers possibilities for the graphical and textual modelling process, for simulation experiments and for post-processing. On the one hand the scalable software architecture of MOSILAB can generate small simulators as monolithic C/C++ applications. On the other hand simulator configurations with more flexibility for the simulation experiment are possible by loading MOSILAB and the compiled model libraries as an extension in a python interpreter, while the simulation experiment is formulated in the script language Python. Current research activities in the GENSIM project will show that MOSILAB can also act as a service in web or grid frameworks and can by coupled with other simulation tools like MATLAB/Simulink.

c) **Model Libraries**: Three model libraries for the technical applications fuel cell systems, hygrothermal building analysis and cutting tool systems are developed and validated in GENSIM. In each application area the efficiency of the model structural dynamics are analysed. In relation to its time dynamics, the analysed systems in GENSIM overlap the millisecond- to second-scale (cutting tool systems), the second to hour-scale (fuel cell systems) and the hour to year-scale (hygrothermal building analysis). For these reasons these application areas together are also a suitable test bed for the numerical basis of the simulation tool MOSILAB.

References

[1] Nordwig, A.: Integration von Sichten für die objektorientierte Modellierung hybrider Systeme, Verlag dissertation.de, ISBN 3-89825-692-8, 2003.

[2] Rational: Unified Modeling Language, Version 1.3, 1999.

[3] Hindmarsh, A. C. et al.: "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers," ACM Transactions on Mathematical Software, 2005. Also available as LLNL technical report UCRL-JP-200037, http://www.llnl.gov/CASC/sundials/

[4] Petzold, L.R.: A description of DASSL: A differential/algebraic system solver, in IMACS Trans. Scientific Computing Vol. 1 (1993), pp. 65-68.

[5] Gear, C.W.; Petzold, L.R.: ODE methods for the solution of differential/algebraic systems, SIAM Journal on Numerical Analysis, 21 (1984) 4, 716 – 728

[6] Wittwer, C.: ColSim – Simulation von Regelungssystemen in aktiven solarthermischen Anlagen. Dissertation Universität Karlsruhe (TH), 1999. www.ubka.uni-karlsruhe.de.

[7] http://www.w3.org/TR/soap

[8] http://www.globus.org

[9] Vetter, M.: Modellbildung und Regelstrategien für erdgasbetriebene Brennstoffzellen-Blockheizkraftwerke, Dissertation Universität Karlsruhe (TH), erscheint Anfang 2005.

[10] Muche, L.; Schneider, P..; Vetter, M.; Wittwer, C.: Modellierung und Simulation der Energieversorgung von Gebäuden mittels Brennstoffzellensystem. Proc. 5. GMM/ITG/GI-Workshop "Multi-Nature Systems", 18. Februar 2005, Dresden, 2005.

[11] Künzel, H.M.: Simultaneous Heat and Moisture Transport in Building Components. - One- and two-dimensional calculation using simple parameters. IRB Verlag, 1995.

[12] Nytsch-Geusen, C.; Bartsch, G.: An Object Oriented Multizone Thermal Building Model based on the Simulation Environment SMILE. Proceedings of Building Simulation 2001, International Building Performance Simulation Association, Rio de Janeiro, 2001.

[13] Nytsch-Geusen, C.; Doll, U.; Leopold, J.: Anwendung generischer Simulationstools zum Werkzeugdesign, Chemnitzer Produktionstechnisches Kolloquium, 2004.