Simulation of complex systems using Modelica and tool coupling

Roland Kossel, Wilhelm Tegethoff, Michael Bodmann, Nicholas Lemke

TLK-Thermo GmbH
Hans-Sommer-Straße 5, 38106 Braunschweig, Germany
r.kossel@tlk-thermo.de

Abstract

Basically there are two approaches of modeling and simulating complex systems composed of subsystems. Approach A is based on established simulation tools of the respective domain in order to conduct a co-simulation by tool-coupling. Approach B consists of modeling the whole system in only one suitable language like Modelica. Using approach A the existing software and the in-house and external expert know-how can be applied. Approach B has significant advantages because of the description of the complete system in one standardized and open language like Modelica and employing an efficient simulation environment such as Dymola. When simulation environments which support approach B are coupled with existing software based on approach A, the advantages of both approaches can be utilized. Considering for example the simulation of a passenger compartment, the feasibility of a coupling of Modelica/Dymola, Flowmaster2, Simulink and THESEUS-FE as well as different utility programs is demonstrated in this presentation.

Keywords: co-simulation; tool coupling; middleware; TISC

1 Simulation of thermal systems in an automotive environment

When simulating a vehicle, its thermodynamic subsystems like engine, engine-cooling system, oil cooling, air conditioning cycle, HVAC unit and passenger compartment are connected with each other especially by fluid- and heat-flows. In order to describe the thermodynamic behavior of the complete system, the models of the subsystems have to be coupled using the variables heat flow rate, mass flow rate, temperature and pressure and if necessary other, non-thermal variables like motor rotation speed. It makes sense to choose the bested suited simulation tool for each subsystem. The following list gives a fragmentary overview of tools that may be used for the respective problems:

Air conditioning cycle	Modelica-Dymola
HVAC unit	CFD / Simulink
Passenger compartment	CFD / THESEUS-FE
Cooling cycle	Flowmaster2 / KULI
Oil cooling	Flowmaster2 / KULI

In the literature, several different approaches are presented for describing the complete system. The approaches range from simple couplings of simulation tools over a co-simulation with different simulation programs to modeling the whole system in one software environment [1], [2].

Figure 1 shows the typical current state of simulation tool coupling in the area of automotive thermal systems. For transferring CFD-information to a 1d-simulation program files are used in most cases [3]. In case of a bidirectional coupling this approach has limitations especially concerning synchronization. When only two programs are coupled, different techniques for tool integration like DLLs or communication channels through COM or CORBA can be employed. As Figure 1 shows, using pair-oriented software coupling quickly meets its limitations. The number of pair-oriented interfaces increases greatly with every integrated tool. Without a standardization and control of the tool-coupling, this technique will be difficult to use. An appropriate synchronization seems to be hard to introduce. Additionally, the implementation and maintenance of the numerous interfaces involves high costs (see [4]).

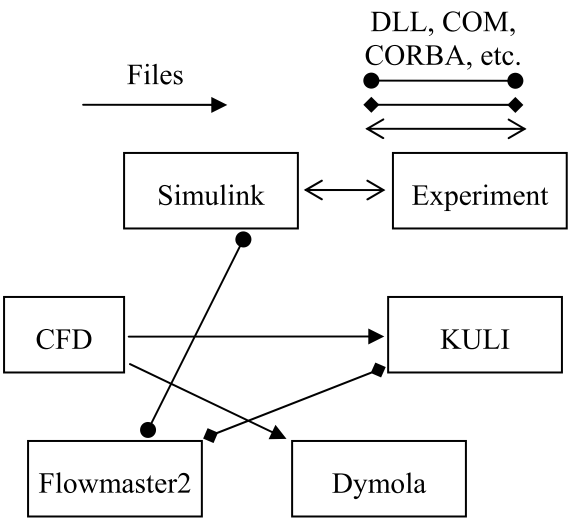

Figure 1: Exemplary current state of tool coupling of simulation programs and HIL-applications

It follows from the above, that the coupling technique for simulation programs must be standardized. The following chapter describes a possible way for this. Afterwards an alternative way describing the thermal behavior of a vehicle holistically is presented.

2 Coupling of simulation tools

As pointed out in Chapter 1, special coupling software (middleware) should be employed when simultaneously coupling more than two programs (see Figure 2). The middleware provides the systematic coupling of the different simulation programs and takes care of the synchronized data communication. In difference to the pair-oriented technique from Chapter 1, only one interface is needed for every program when using a middleware concept.

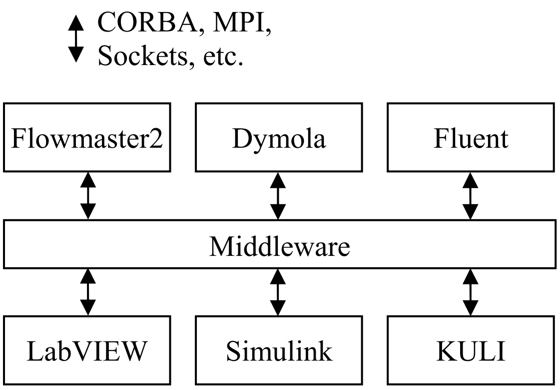

Figure 2: Coupling of simulation tools and data acquisition software through a middleware

A typical middleware is for example the program EXITE [5], which is widely used in the field of vehicle electronics. EXITE's data exchange is based on CORBA for normal and MPI for HIL applications. In the field of thermal management Puntigam [1] describes the middleware of the VIF in Graz. This software relies on CORBA and has numerical means for coupling the differential equations of the coupled programs. The MPI based middleware MpCCI [6] provides a solution for coupling 3D programs. MpCCI aims at mesh-mesh-coupling. At the present date MpCCI support only the coupling of two applications at one time.

The configuration displayed in Figure 2 is currently not realizable with any of the described middleware. Due to the need of a short-term availability and special demands concerning the data synchronization and platform independent data exchange, the middleware TISC was developed.

TISC allows platform independent exchange of commands (strings), floating point numbers (double) and integers as scalar values as well as vectors and matrices. The data exchange is realized through sockets. As Figure 3 shows, TISC consists of a server and the respective clients, which are integrated into the coupled programs. For this integration, interfaces are available in C, C++, C# and FORTRAN. A connection over COM is also supported and used for example for Flowmaster2 and KULI.

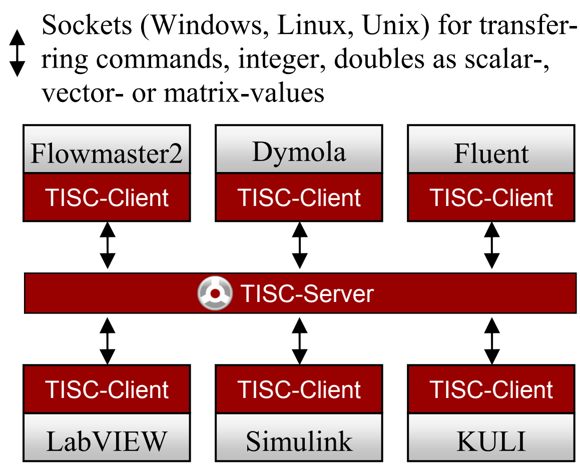

Figure 3: Coupling of simulation tools and data acquisition software with TISC

TISC-integrations are currently available for the following programs:

- Flowmaster2
- Fluent
- KULI
- LabVIEW
- Matlab/Simulink
- Modelica/Dymola
- StarCD
- THESEUS-FE
- Trnsys

The following interfaces are currently under development:

- Ansys
- Excel
- CFX
- COM

Furthermore utility programs exist for visualizing the exchanged data. TIM – the **T**ISC **I**nformation **M**onitor – receives and saves all data sent by the coupled programs. The StateViewer displays a pressure-enthalpy-diagram at run time of the co-simulation or the measurement. A special client for controlling the next time increment (if the coupled programs allow this) and for coupling the variables numerically efficient is in development (see [1]).

2.1 Advantages of coupling simulation tools

When coupling simulation tools with a middleware, several advantages arise besides from the saving of time. Employing a middleware for the simulation of the complete system allows the utilization of present expert knowledge which is bound to existing software. Using this knowledge can speed up integrating the examination of the complete vehicle into the product development process. Beyond this, the best tool can be chosen for describing each subsystem. By using a standardized definition of the interface variables, the single simulation tools can be easily exchanged (compare Figure 4). Even if the approach of using a unified language for building the models is used, it can make sense to couple sub models with a middleware to decouple the time Steps of the solver.

2.2 Arising problems

Besides the mentioned advantages, there are also drawbacks to consider. Since events have to be triggered to exchange the data between the simulation tools, the solvers of those programs have to be stopped and restarted at every exchange time. Since the received variables show a discrete behavior, this restart can become difficult. Further on numerical oscillations can occur, when the interdependence of the coupled variables is very high. To reduce these effects, interpolation methods are currently explored in order to convert the discrete behavior of the received variables into a continuous one or to achieve a pseudo multistep integration method.

3 Unified simulation with Modelica

An alternative way to the coupling of simulation tools is describing the complete system including all subsystems in one unified programming language and mathematics. In the following it is assumed that the subsystems and therefore the complete system can be described by hybrid systems of differential equation (ADE-systems). A hybrid ADE-systems consists of the following three equation types:

1. Ordinary differential equations (ODE) with the respective differential variables whose derivatives explicitly appear in the system of equations.
2. Linear and non-linear ordinary algebraic equations (OAE) with the respective algebraic variables.
3. Event equations with the respective discrete variables whose values can only be changed at an event. The corresponding simulation technique is highly developed in digital electronics. Because of the combination with the continuous ADE-system it is referred to as hybrid ADE-system.

By assembling the sub models to the complete thermal system, a big hybrid ADE-system is created which can be solved with appropriate numerical techniques. Methods for coupling partial differential equations like FEM or finite volume methods form a separate technique. From the authors point of view this technique can not be integrated into the ADE-systems by a standardized description with a unified programming language at this time. Coupling using co-simulation seems to make sense in this case.

The thermal models for engine, engine cooling cycle, oil cooling, air conditioning cycle, HVAC unit and passenger compartment can be described using hybrid ADE-systems. Highly developed computer languages have been developed for this purpose. Besides VHDL-AMS [7] which is mainly used in the field of electronics, the language Modelica [8, 9] is currently widely propagated. The language Modelica is developed by the non-profit Modelica Association with the goal to describe hybrid ADE-systems from diverse physical and engineering domains. Already numerous commercial and non-commercial libraries are available. For example the AirConditioning library [10] is being used by the A.K.2.4.3 of the German OEMs with participation of Audi, BMW, DaimlerChrysler and Volkswagen for unified steady state and transient simulation of air conditioning cycles. In this context the OEMs require the suppliers previous to the project assignment to build the respective component models in the language Modelica [11].

The physical equations are set up in the component models. The transformation of the equations as well as the numerical solution is left to the used working environment. With means of object-oriented modeling, component libraries can easily be built up. Defining standard interfaces for the components allow a simple model exchange.

At the moment, Dymola is the most powerful simulation environment which supports the language Modelica [12]. Other simulators like Mosilab [13], Amesim [14] and SimulationX [15] are being extended to also support Modelica. The free working environment OpenModelica can be employed for a subset of simulation problems [16].

The open and unified programming language allows an optimal communication and standardization between departments and suppliers. By using the object-oriented techniques of Modelica, the work in teams is eased. Extending and reusing existing models of the component library is also eased. A substantial training of the employees is necessary, because otherwise the object-oriented design can easily cause the opposite effect. Because of the open language the sustained realization of the own expert knowledge is possible with less dependence of software companies and engineering service providers. By employing the right numerical and object-oriented techniques, shorted times for method development and calculation can be achieved. Furthermore the goal of a working environment providing the pre-processing, simulation and post-processing of the complete system can be reached more quickly.

In order to profit from the mentioned advantages and the benefits of a co-simulation – and to be capable of working without a complete Modelica model library – it makes sense to combine a co-simulation with a unified programming language.

4 Example: Passenger compartment

The tool-coupling using TISC is demonstrated for a simulation of the cool down of a passenger compartment. Figure 4 shows the programs participating in the co-simulation. The central element of the co-simulation is the TISC-server to which every program connects via its interface, the so called TISC-client.

The passenger compartment is being simulated with THESEUS-FE [17]. The model has been provided by P+Z Engineering and describes a British sedan including the driver.

The compartment is being cooled by an R134a air conditioning cycle built in Modelica using the Air-Conditioning library.

A PID-controller assembled in Matlab/Simulink tunes the temperature within the compartment.

Using TLK's utility program StateViewer, the ph-diagram for the air conditioning cycle can be displayed while simulating. The TISC Information Monitor – short TIM – logs all data exchanged between the different programs over TISC.

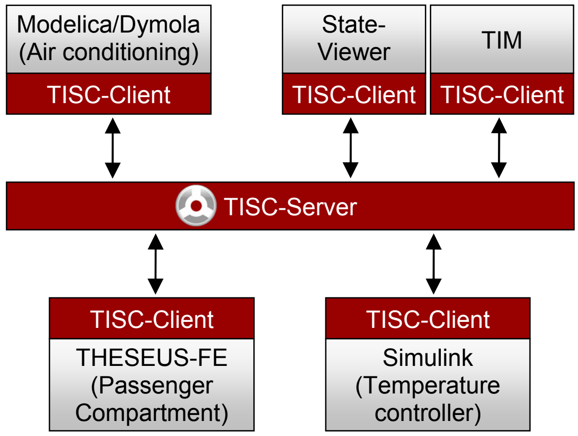

Figure 4: TISC-setup for co-simulation example

As Figure 4 shows, the single programs are not connected among each other but only with the TISC-server. During initialization the simulation tools register the variables they are sending and receiving at the server. The data exchange and synchronization during the simulation is handled by the server. By using this communication setup, an exchange of the simulation tools can be easily arranged – as long as the exchanged tools send and receive the same variables, the other tools' configurations do not have to be altered. For example, the model of the passenger compartment could be changed this way to describe a different vehicle.

Figure 5 lists the variables sent and received by the simulations tools for the co-simulation.

	Sends	Receives
Dymola	Output evaporator (T, mdot)	Input evaporator (T, mdot)
		Relative displacement
Simulink	Relative displacement	Signal of temperature sensor
THESEUS-FE	Input evaporator (T, mdot)	Output evaporator (T, mdot)
	Signal of temperature sensor	

Figure 5: Variables exchanged in co-simulation

The Modelica-model describing the air conditioning cycle is shown in Figure 6. The TISC-interface is integrated as a single block into the example "OrificeTubeCycle" of the AirConditioning library.

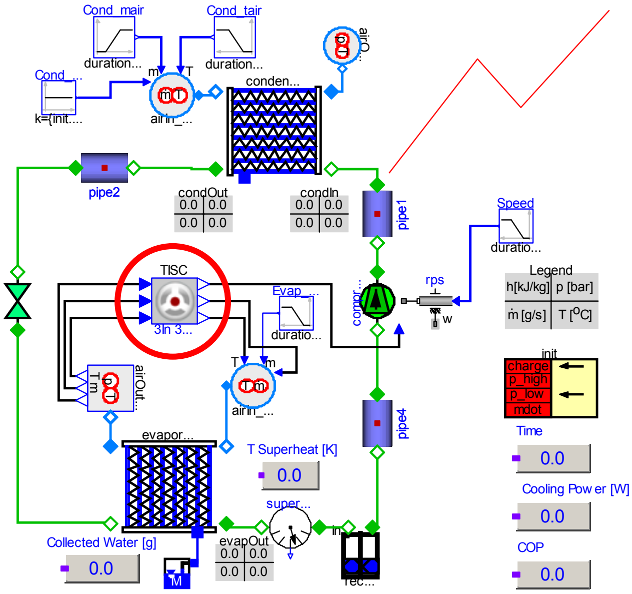

Figure 6: Modelica model in Dymola for co-simulation example

The TISC-block is a composition of single blocks for the different functions of the tool coupling. These are

- Creating an event when data needs to be exchanged
- Managing the connection to the TISC-server
- Sending variables
- Synchronizing with the TISC-server
- Receiving variables

Figure 7 shows the internal composition of the TISC-block from Figure 6.

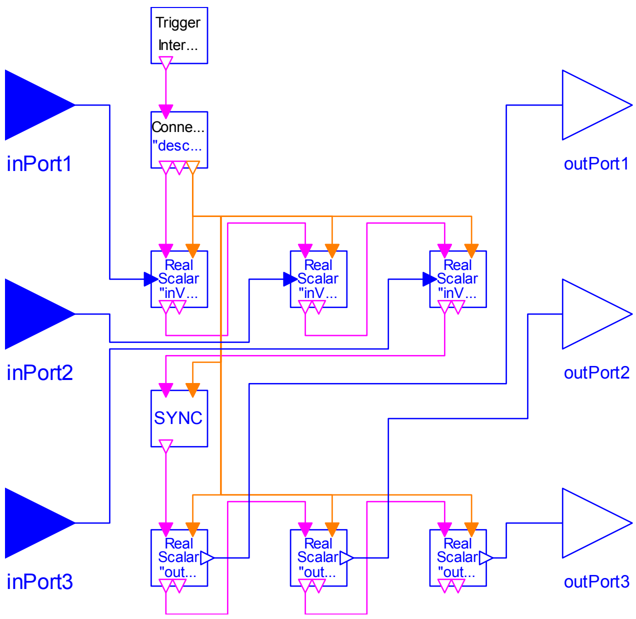

Figure 7: Internal composition of TISC-block

The block sends three variables to the TISC-server and receives three variables from it. Every block in the assembly stands for one function. A trigger token is passed from one block to the next one to assure the correct execution sequence of the blocks. The token is created in the "Trigger"-block every time the variables need to be exchanged. The sequence shown in Figure 7 is:

1. Create Trigger-token
2. Manage connection to TISC-server
3. Send three variables
4. Synchronize
5. Receive three variables

Every block sending or receiving a variable allows the user to specify an offset and a factor. By doing so, unit conversions between the coupled programs can easily be conducted.

Figure 8 shows the passenger compartment at the beginning of the cool down co-simulation. The initial conditions are calculated with a heat soak simulation with one hour of solar irradiation at 1000 W/m^2.

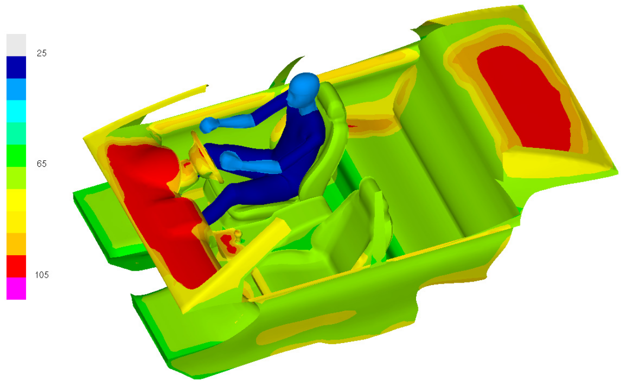

Figure 8: THESEUS-FE result at simulation start

The temperature distribution in the passenger compartment after the coupled cool-down simulation of 2400 s duration is shown in Figure 9.

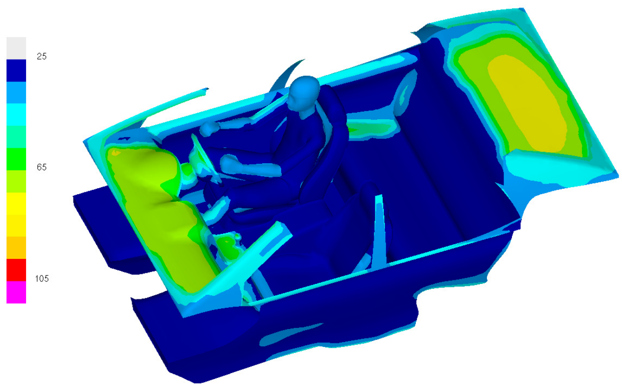

Figure 9: THESEUS-FE result at simulation end

5 Conclusion

When simulation tools supporting the approach of a unified programming language are coupled with existing software using a middleware, the advantages of both approaches can be utilized. Using co-simulation, established programs and the respective knowledge can still be used and combined with standardized code replacing non-satisfying software or describing new functionality.

In the process of moving to a unified language, co-simulation can make sense as the first steps. In the presented example, the temperature controller modelled in Simulink can be replaced by one modelled in Modelica. The next step of integrating the controller in the air conditioning system is not far. Only the co-simulation of models with extremely different 3d approaches like the passenger compartment using the finite element method in THESEUS-FE or the 0d and 1d simulation of the air conditioning cycle in Modelica can hardly be replaced. Even if all models are described in one language, a co-simulation can still be employed i.e. to decouple systems with different time constants.

To allow the definition of company- and tool-spanning interfaces for co-simulation that ensure the numerical and physical compatibility, techniques for example from the CapeOpen standard [18] could be used as suggestions. The CapeOpen standard is commonly used in the field of process engineering and describes the coupling of partial hybrid ADE-systems. However this mathematically very ambitious method has no practical application yet.

Concluding we like to thank P+Z Engineering for providing the model used in the shown example and for the help in developing the TISC-interface for THESEUS-FE.

References

[1] Puntigam G., Balic J., Almbauer R., Hager J.; Transient Co-Simulation of Comprehensive Vehicle Models by Time Dependent Coupling, SAE 2006-01-1604

[2] Duhme M., Flögel H.-H., Braun M., Säger U.; Parameterstudie im thermischen Übertragungspfad Motor – Kabine; in: Wärmemanagement im Kraftfahrzeug III; Hrsg. Deussen N. ; S.1 - S.11; Essen; expert Verlag 2002

[3] Pollack J.; CFD Simulation des Motorraums als Beitrag zur Verbesserung des Wärmemangements; in: Wärmemanagement im Kraftfahrzeug III; Hrsg. Deussen N. ; S.1 - S.11; Essen; expert Verlag 2002

[4] Betz, J., Anzenberger T., Kobs T.; Entwicklungstendenzen der Wärmemanage-mentsimulation im Bereich Kühlung und Klimatisierung bei Audi; in: Wärmemanagement im Kraftfahrzeug IV; Hrsg. Steinberg P.; S.126 - S.140; Essen; expert Verlag 2004

[5] Extessy AG; Exite Manual; Version 1.4.3.; Wolfsburg, 2005, http://www.extessy.com

[6] http://www.mpcci.de

[7] Hessel E.; Standardisierte Modellbibliotheken für die Automobilindustrie – Aktueller Stand der Arbeiten des AK30 in der FAT; Vortragsband der Virtual Vehicle Creation, Stuttgart 2006 und http://fat-ak30.eas.iis.fraunhofer.de

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

[9] Fritzon P.; Object-oriented Modelling and Simulation with Modelica 2.1; Wiley 2004

[10] Tummescheit H., Eborn J.; Prölß K.; Försterling S.; Tegethoff W.; AirConditioning: Eine Modelica Bibliothek zur dynamischen Simulation von Kältekreisläufen; in: PKW-Klimatisierung IV; Hrsg. Schlenz D.; S.196 - S.214; Starnberg; expert Verlag 2005

[11] Schneider F.; Bunzel A., Hofhaus J., Braun M., Limperich D., Cäsar R., Arntz K.-D., Schröter A., Specht B., Mönkediek T.; Entwicklung und Einführung eines einheitlichen Kältekreislaufsimulationsprogramms; in: PKW-Klimatisierung IV; Hrsg. Schlenz D.; S.185 - S.194; Starnberg; expert Verlag 2005

[12] http://www.dynasim.se

[13] http://www.mosilab.de

[14] http://www.amesim.com

[15] http://www.iti.de

[16] http://www.ida.liu.se/labs/pelab/modelica/OpenModelica.html

[17] http://www.theseus-fe.com

[18] www.colan.org