M. Tiller, P. Bowles, H. Elmqvist, D. Brück,
S. E. Mattson, A. Möller, H. Olsson, M. Otter:
Detailed Vehicle Powertrain Modeling in Modelica.
Modelica Workshop 2000 Proceedings, pp. 169-178.

Detailed Vehicle Powertrain Modeling in Modelica

Michael Tiller and Paul Bowles
Ford Motor Company, USA

Hilding Elmqvist, Dag Brück, Sven Erik Mattson, Andreas Möller and Hans Olsson
Dynasim AB, Sweden

Martin Otter
German Aerospace Research Establishment (DLR), Germany

Abstract

This paper describes of a detailed vehicle model in Modelica, consisting of the 3D vehicle chassis, engine, automatic gearbox and hydraulics to control the gearbox. Furthermore, simulations of these models were done using Dymola. Emphasis is given to the engine modeling, because it required many new Modelica components to be developed. This feasibility study shows that it is possible to use Modelica for large models assembled from complex structured subsystem models.

1 Introduction

The work presented in this paper was done to evaluate the capability of the Modelica [2, 8] modeling language and Dymola [7] for modeling and simulation of complex automotive systems. There were three aspects to this evaluation. The first aspect was whether existing models, with very complex behavioral descriptions, could be expressed using the Modelica language. The second aspect was how reusable these models could be made. Finally, the last aspect of the evaluation was to determine if the Modelica approach would scale well for very large problems.

Initially, the model development started with the development of mechanical models for the engine and soon included gas properties and combustion models. Several tests were conducted to validate the behavior of both motoring (no combustion) and firing engines.

The next stage in the project was to develop a mechanical transmission model. The development of this model was greatly facilitated by the existence of the Modelica standard library's 1D rotational package. Once a mechanical model of the transmission had been constructed, the hydraulic subsystem was realized.

In order to study the powertrain system it was necessary to create a model of the vehicle chassis. The Modelica multi-body systems library [3] contains a collection of 3-dimensional mechanical components from which a complete vehicle chassis model can be created. However, creation of a complex vehicle chassis model by hand would be a tedious and error prone task. A model of the vehicle chassis was already available in ADAMS [1]. For this reason, an ADAMS translator was developed which automatically translates ADAMS models into Modelica representations which use the Modelica MBS library.

Finally, the reusability and scalability of the Modelica language and the Dymola environment [7] were tested by integrating various combinations of the engine, transmission, transmission hydraulics and chassis.

This work benefited greatly from the domain-neutral nature of the Modelica language as well as the extensive set of available components in the Modelica Standard Library. The significance of both of these factors will be very important in the success of the Modelica modeling language in industrial engineering applications.

The emphasis in this paper is on the development of the engine models because most of those models needed were not already available and had to be developed. More details about the results and validation of the transmission and chassis work can be found in [6]. Additional background information is available for the rotational library [4], hydraulics library [5] and multi-body components [3].

2 Engine Models

2.1 Background

The first system modeled was a multi-cylinder internal combustion engine. The focus of such a model is to capture the behavior of the thermodynamic effects inside the cylinder. The models presented in this section are referred to as **cycle simulation** models [9]. This means that they are capable of representing the transient response of the engine. In contrast, "cycle averaged" models predict the nominal, steady-state torque output of an engine.

In order to model the engine, the connector definitions and the underlying behavioral models had to be developed from scratch. Initially, there were not many models available which could be used in building a model of an engine. However, once the fundamental models were developed a wide variety of systems could be built by reusing fundamental ones.

2.2 Connector

The first step in building up thermodynamic models for the engine was to decide on the `connector` definition. One of the very useful features of the Modelica language is the fact that a connector can have multiple *through* and *across* variables. For thermodynamic systems, the state of the working fluid is represented by pressure and temperature. In addition, the composition (*i.e.,* chemical mixture) of the working fluid is represented by a vector containing the mass fractions of the various species. In addition, each connection represents a path for energy and mass (both total mass and mass of a particular chemical species) to move through the system. Taking all of these issues into account, the following `connector` definition was formulated:

```
connector Thermo "Therm. connection"
  package SI=Modelica.SIunits;
  parameter Integer nspecies=4;
  SI.Pressure     P;
  SI.Temperature  T;
  SI.MassFraction X[nspecies];
  flow SI.Power   q;
  flow SI.MassFlowRate m_dot;
  flow SI.MassFlowRate xm_dot[nspecies];
end Thermo;
```

While this `connector` definition has proved very useful, it still has several shortcomings. First, there is redundancy in this connector definition because the quantity `m_dot` should always be equal to the sum of the components of the `xm_dot` vector. In addition, there is no capacity to handle momentum flowing across such a connector. Representing momentum is important when modeling other engine processes (*e.g.,* manifold dynamics).

When formulating the `connector` definitions, it is sometimes necessary to consider implications to the underlying medium models (see Section 2.3.3 for more details).

2.3 Thermodynamic Components

2.3.1 Thermodynamic State

At the heart of the thermodynamic models is the application of the first law of thermodynamics. Typically, the first law for a control volume is represented by an equation of the form:

$$\frac{dU}{dt} = \sum_i \dot{m}_i h_i + Q - P\frac{dV}{dt} \qquad (1)$$

where U is the total energy within the control volume, $\dot{m}_i$ is the mass flow rate into the control volume via the i^{th} path, h_i is the specific enthalpy of the fluid into the control volume via the i^{th} path, Q is the net heat into the control volume, P is the pressure of the control volume and V is the volume of the control volume.

The other conservation law that is used is conservation of mass which can be written as:

$$\frac{dm_t}{dt} = \sum_i \dot{m}_i \qquad (2)$$

where m_t is the total mass within the control volume. This conservation law can in turn be applied to each of the chemical species present which yields:

$$\frac{dm^c}{dt} = \sum_i \dot{m}_i^c \qquad (3)$$

which can be applied for each chemical species (represented by the superscript c).

In addition to the first law, there are several constitutive relationships used in the thermodynamic models. For example, the ideal gas law is used to express the relationship between the temperature in a control volume and the pressure of the control volume (for a given mass and volume). Other important phenomena (*e.g.,* flow through the valves) have their own constitutive relations. All of these are straightforward to implement in Modelica because these are represented by straighforward equations.

There are several modeling issues that appear in the thermodynamic domain that are uncommon in other domains. The first is the implicit nature of the problem. Rewriting Equation (1) to include explicit references to the temperature of the control volume gives:

$$\begin{aligned} \frac{dU}{dt} &= \sum \dot{m}_i h_i + Q - P\frac{dV}{dt} && (4) \\ U &= m_t u(T) && (5) \end{aligned}$$

where $u(T)$ is the specific internal energy of the control volume and m_t is the mass within the control volume. Note that it is essential to compute the temperature since it is used in evaluating mixture properties and it appears in many constitutive equations.

There are several ways to deal with this equation. The first is to choose U as the state variable. In this case, the differential equation for U is integrated and the solution of U is available directly. The temperature, T, is then computed by applying a non-linear solution algorithm (typically Newton-Raphson) to Equation (5). Note, that within a differential-algebraic equation solver, these two steps are combined. The requirement of invoking a non-linear solver could negatively affect the performance of the integrator. The other approach is, to add an equation which computes the derivative of the temperature, so that temperature can be used as a state during the simulation:

$$d_T = \frac{dT(t)}{dt} \qquad (6)$$

This equation introduces a new (dummy) variable d_T which is computed by the derivative of T. Since we would like to transform the equations (at least locally) into state space form, i.e., $\frac{dT(t)}{dt} = f(T)$, the derivative of T has to appear in the equations, which is performed by introducing (6). Equations (4, 5, 6) are now three equations to compute the three unknowns U, T, d_T. Due to (5), there is a constraint equation between the potential states T and U and therefore only one of them can be the "actual" state.

Modelica has been constructed such that these situations can be solved automatically by using the algorithm of Pantelides [11] together with the dummy derivative method [10]. The result is, that T is selected as a state, that U is computed from (5), $\frac{dU}{dt}$ is computed from (4) and $\frac{dT}{dt}$ is computed by the differentiated equation of (5). Therefore, with this approach, no nonlinear equation has to be solved. From a modellers perspective, only equation (6) has to be added, and the rest is performed automatically with an appropriate Modelica tool, such as Dymola [7].

2.3.2 Numerical Issues

Another problem that can occur in thermodynamic systems is that the mass within a given control volume might vanish. This can happen for a number of reasons. For example, the control volume may contain liquid fuel which completely evaporates. The difficulty in handling this case is that no equation exists for the temperature as the mass vanishes and the problem becomes under constrained. In some cases, it should be possible to solve for the limit of the temperature solution as the singular case is approached.

The vanishing of mass within the control volume is also a problem because thermodynamic problems contain numerous intensive variables (*i.e.,* quantities which are normalized with respect to mass). For example, the specific internal energy, $u(T)$, shown in Equation (5).

Finally, when dealing with thermodynamic models it is often necessary to use an iterative algorithm like Newton-Raphson. Such algorithms require initial guesses for quantities like temperatures and pressures. It is important to make sure that a reasonable value (from an engineering perspective) is provided for the `start` attribute of any variables that represent thermodynamic states. In the worst case, the definitions from the `Modelica.SIunits package` would be used where the value of the `start` attribute is zero by default (a very non-physical value for engineering problems).

2.3.3 Medium Models

As pointed out in the previous section, the enthalpy and energy of the working fluid must be represented in the fundamental thermodynamic equations. Tied to the definitions of enthalpy and energy are other properties such as density, molecular weight and specific heat capacities. A consistent set of these properties is known as a **medium model**.

When modeling thermodynamic systems, it is possible for a wide variety of medium models to be available and/or required. Different medium models may represent different working fluids or different levels of compositional detail for a given working fluid. For this reason, it is very useful to build component models in such a way that they can be used with a variety of medium models. Idioms for representing medium models in Modelica are still being developed and evaluated [12]. Medium models are important in many engineering domains (*e.g.,* hydraulics, heat-transfer, fluid flow).

Modelica supports the ability to have external functions compiled in other languages (*e.g.,* C or FORTRAN). This is very useful to allow the incorporation of existing source code for implementing medium models.

2.3.4 Summary

Overall, the representation of thermodynamic systems is relatively straightforward as this Modelica control volume model shows:

```
partial model ControlVolumeBase
  package SI = Modelica.SIunits;
  parameter Integer nsp=4 "# of species";
public
  Ford.Interfaces.Thermo n(nspecies=nsp);
protected
  SI.Mass     total_mass;
  SI.Mass     mx[nsp];
  SI.Volume   vol "Volume";
  Ford.Engine.Properties.PreferredPropBlock
     props(T=n.T, P=n.P, X=n.X) "Media";
  Real dT;
equation
  // Conservation of mass
  der(total_mass) = n.m_dot;
  der(mx) = n.xm_dot;

  // First law of thermodynamics
  der(props.u) = n.q - n.P*der(vol);
  dT = der(T);

  // Ideal gas law (equivalent to P*V=m*R*T)
  n.P*vol = total_mass*(props.h - props.u);

  // Compute mass fractions
  n.X = mx/total_mass;
end ControlVolumeBase;
```

Note the similarities between the equations in this model and Equations (1, 2, 3).

2.4 Modeling Combustion

A single zone model of combustion only requires that mass of one species in the mixture be converted into another. Such a conversion results in a tremendous increase in the temperature of the mixture. The temperature increases because the mixture composition lowers the specific internal energy of the gas but the total energy changes very little during the combustion process (*e.g.,* due to heat transfer or work). The result is that the gas must have a higher temperature in order to contain a nearly constant total energy.

Multiple zone models are more complicated because mass is exchanged between two different zones. The multiple zone case brings up several complications. The first is that at all other times during the combustion process only one zone is present in the combustion chamber. Because Modelica does not allow the number of equations to change during the simulation, equations must be present for both zones even when only one really exists. This leads to the problem of having zero mass in the control volume (as discussed in Section 2.3.2).

Finally, because the combustion process is cyclical it is necessary to be able to reinitialize the state of the system at the start of each cycle. For example, in the multiple zone case one zone is typically called the “burned zone” and the other is the “unburned zone”. At the start of a cycle, all mass is in the unburned zone[1]. As combustion proceeds, all the mass is transfered to the burned zone (along with a compositional change). When the next cycle starts the burned mass from the previous cycle becomes the initial unburned mass for the next cycle. For this reason it is necessary to reinitialize the states of the burned and unburned zone (*i.e.,* the mass and energy must be instantaneously relocated).

2.5 Mechanical Components

The one area where some component models already existed and could be reused was in the mechanical aspect of the engine. For example, the piston model is a translational force component and it uses the translational connector definition from the translational motion library[2]. Because it uses the same connector definition as the existing components in the translational library, translational components like masses, springs, dampers and friction elements can be connected to the piston to model a variety of important effects inside the engine.

Obviously, rotational components also play a large part in the modeling of an engine. The rotational motion library[3] provides models to represent inertias, damping, friction, gear effects, *etc.*

Finally, an important thing to represent among the mechanical components is ”experimental instrumentation”. In other words, it is important to be able to represent macroscopic information about the performance of the engine. Using some of the hybrid modeling capabilities of the Modelica language, many important cycle averaged results can be computed. For example,

---

[1] The name “unburned zone” is a misnomer in this case because the unburned zone invariably contains some combustion products left over from the previous cycle.

[2] Modelica.Mechanics.Translational

[3] Modelica.Mechanics.Rotational

by integrating the instantaneous torque over two complete revolutions of the crankshaft (*i.e.*, one engine cycle), commonly used metrics like cycle-average torque and mean effective pressure can be computed.

2.6 Complex Assemblies

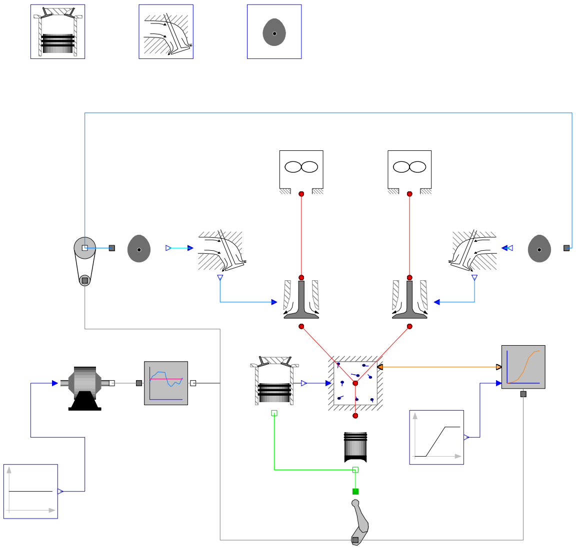

Figure 1: Single cylinder engine

So far, the basic building blocks of the engine models have been described. Once these building blocks are in place, complex assemblies can be created. Figure 1 shows a model of a single cylinder engine. This basic model includes piston, crank mechanism, timing belt, cams, valves and an ideal dynamometer.

Of course, building an entire six cylinder engine the way the single cylinder engine is constructed in Figure 1 would be very tedious. For this reason, an individual cylinder can be create which includes only the per-cylinder components. A model of such a cylinder is shown in Figure 2.

Using the individual cylinder model shown in Figure 2 a six cylinder engine model like the one shown in Figure 3 can be constructed.

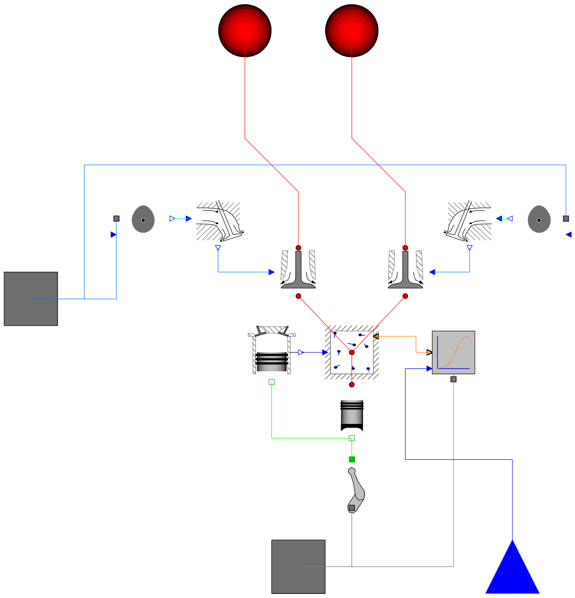

Figure 2: An individual cylinder

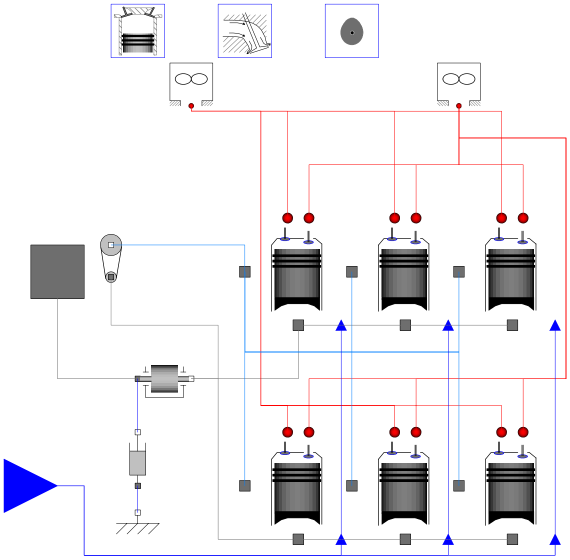

Figure 3: Six cylinder engine

2.7 Future Directions

Predictive combustion models would also benefit from the ability to simulate “impulses”. In this context, an impulse is an instantaneous flow of conserved quantities from one location to another. An example of an impulse is an elastic ball bouncing on a hard surface. It is often convenient to model the collision as an instanteous event. The same sort of functionality is useful in initiating the combustion process (*i.e.*, the spark).

Another effect that is useful to capture that has not been modeled is the thermal warmup of the engine. Since a thermal component library is currently in development, the hope is that such a warmup model could easily be developed from the components that will be available in that library.

The models presented here have been designed to allow for multiple medium models. Unfortunately, only one medium model has been tested. In the future, more detailed medium models will become available and the robustness and reusability of the medium model approach will be tested more thoroughly.

These models have only demonstrated the uncontrolled operation of the engine. A more detailed model would be equipped with a wide range of sensors and actuators. The sensors and actuators would be connected to a control system and the closed-loop performance of the engine could be tested. The ability to test the effectiveness of sensors, the response of actuators and the overall performance of the control system is extremely valuable in current engineering processes.

Comparisons were done between existing Ford in-house engine modeling tools and the models developed in Modelica. There was good agreement between both tools. However, the Ford in-house tools are still farther advanced than the Modelica models in their ability to predict combustion characteristics. Future development work may focus on bringing the Modelica models up to the level of existing Ford in-house cycle simulation models.

3 Transmission Models

3.1 Mechanical System

A transmission model was developed for the work described in this paper. A schematic of the mechanical subsystem is shown in Figure 4. The existence of a rotational library of components made the development of this model significantly easier. Many of the components needed were already available and the ones that were not available were easily built from the connectors and partial base classes available.

The mechanical system by itself was validated against a set of Ford in-house tools. There was “line on line” agreement between the results from Dymola and the results from the Ford in-house tools.

3.2 Hydraulic System

Because of the complexity in the hydraulic subsystem of the transmission, only a subset of the hydraulics were modeled. Specifically, only the valves required for a shift from first to second gear were included. The development of these libraries was facilitated by the existence of a commercial library of hydraulic components [5]. The subsystem model that was developed is shown in Figure 5.

Just as with the mechanical subsystem, the hydraulic subsystem was validated by comparing simulation results between Dymola and our in-house tools.

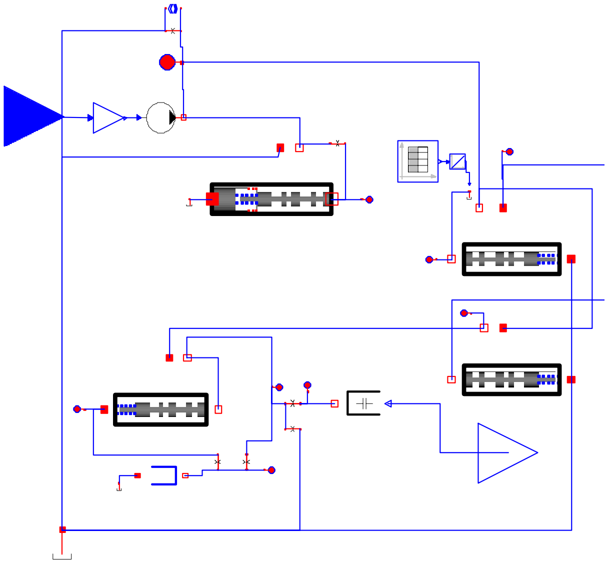

Figure 5: Hydraulic Subsystem

Once again, the comparison between the numerical results showed “line on line” agreement between the trajectories.

3.3 Combined System

The real test of the mechanical and hydraulic subsystem models was to see if they could be connected together and function as a complete subsystem. Building the integrated transmission model was surprisingly straightforward. There was no significant difference between connecting two simple components and connecting two complex subsystems. As with the previous models, comparisons were made with existing Ford in-house tools and there was excellent agreement.

3.4 Summary

Just as with the engine, the ability to model impulses would be very useful in development of the transmission models. In the mechanical system there are numerous sources of backlash and many of these models would be easier to express using impulses. Within the hydraulic system, it is common for the spools inside the spool valves to collide both with the surrounding walls as well as with each other. Once again, the modeling of these collisions would be easier with the ability to express impulses.

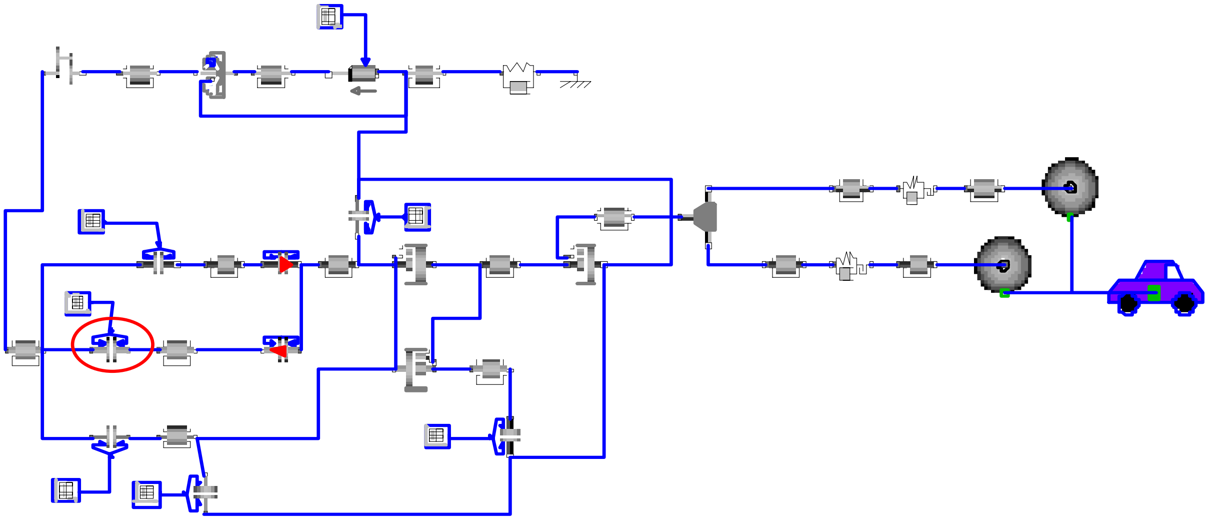

Figure 4: Mechanical Transmission Subsystem

4 Vehicle Chassis

For this study we chose to use a detailed model of a minivan chassis which had already been created using ADAMS, a program for three-dimensional mechanical simulation from Mechanical Dynamics Inc. [1]. In order to integrate the chassis, engine and transmission into one Modelica model, the ADAMS chassis model was converted to Modelica by a newly developed translator which is described to some detail in the rest of this section.

4.1 Multibody Systems

Prior to this work, a library for modeling of multi-body systems in Modelica had already been developed. Figure 6 shows the sublibrary for joints. Other sublibraries contain parts, forces and sensors. The newest version of this library is available from `http:\www.Modelica.org\library\library.html` as ModelicaAdditions.MultiBody.

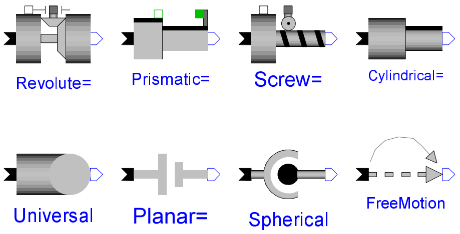

Figure 6: MBS sublibrary for joints

4.2 ADAMS to Modelica translator

To simplify the translation of the ADAMS model, a new library of ADAMS compatible models was developed based on this MBS library, see Figure 7. It contains realizations of ADAMS elements such as Ground, Part, Revolute, Sforce, Coupler, Bushing and Field.

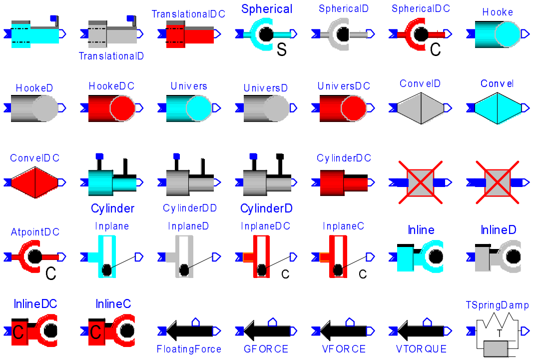

Figure 7: Subset of ADAMS compatible library

An example is shown in Figure 8, where the ADAMS JPRIM-INLINE joint is realized. This joint has 4 degrees-of-freedom such that the right connector moves along the z-axis of the left connector. This new joint class is simply built-up by connecting an available translational and spherical joint with each other.

Additionally, a translator from the ADM file format of ADAMS to Modelica was realized. This translator reads an ADM file, stores all information of the ADM file in an internal data structure, analyzes the data and generates appropriate Modelica code. Most

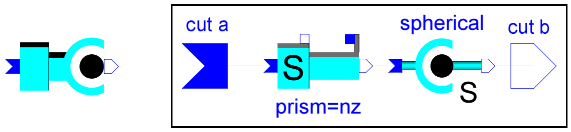

Figure 8: ADAMS JPRIM/INLINE joint (left) and the individual joints used to construct it (right)

statements have corresponding models in the Modelica package `Adams`, resulting in a one-to-one translation of the ADM file. Many properties which are represented as references to MARKER statements are converted to parameters of part and joint components. For example, the center of mass of a part is represented by a marker,

```
PART/2, MASS = 4.116014532, CM = 5
MARKER/5, PART = 2, QP = -200, 450, 0
```

but in the Modelica model, it is given directly as a parameter:

```
Part P2(MASS = 4.116014532,
      CM_QP={-200, 450, 0}, ...);
```

Finally, the model topology is represented in Modelica by connections between Part, Joint and Force objects. This information is extracted by visiting each joint and force element in the data structure and looking up the corresponding marker and part elements. For example, the following ADAMS statements (slightly abbreviated):

```
PART/1, GROUND
MARKER/1, PART = 1, QP = -200, 450, 0
PART/2, MASS = 4.116014532, CM = 4
MARKER/5, PART = 2, QP = -200, 450, 0
JOINT/1, REVOLUTE, I = 5, J = 1
```

yield the following Modelica code:

```
  Ground P1(...);
  Part   P2(...);
  Revolute J1(...);
    ...
equation
  connect (P1.b, J1.a);
  connect (J1.b, P2.a);
```

Using the ADAMS-to-Modelica translator, an available ADAMS model of a minivan was translated to Modelica. This chassis model consists of 73 parts, 32 revolute joints, 13 translational joints, 14 other joints, 22 bushings, 10 fields, 67 other force elements, and 205 graphical elements. The resulting 3D composition after translating this model from an ADAMS ADM-file to its Modelica representation is shown in Figure 9. Simulation in Dymola for a 10 second interval with the numerical solver DASSL and a tolerance of 10^{-4} took 17 min (in ADAMS it took 15 min).

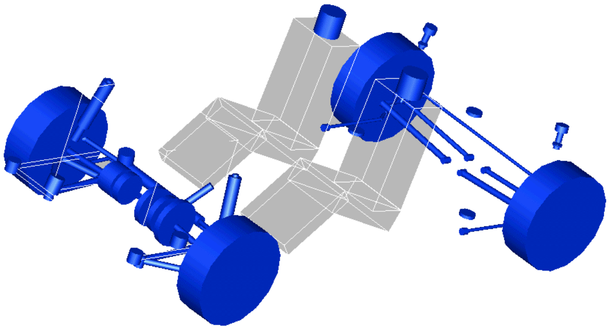

Figure 9: Modelica model of chassis.

5 Integration Results

In the previous sections favorable comparisons were shown between Modelica models and the existing Ford in-house analysis tools. The primary goal was that the independently developed and validated subcomponents, such as the transmission and the chassis should be assembled together to arrive at an overall vehicle model. The ability to do this integration is important for several reasons: First, it makes collaboration between different modeling efforts within the same organization easier. The other reason is to leverage work done by third parties (e.g. suppliers, tool vendors, universities) regardless of the specific toolset used.

5.1 Engine and Transmission

In order to test the integration features in Modelica, the engine and transmission models presented in the previous sections were combined. By using a more detailed engine model, the transmission receives a more widely varying torque compared to a cycle average engine model . The results of the analysis using the detailed engine and transmission models can be seen in Figure 10.

The ability to combine the engine and transmission models together allows analyses of complex interactions that may occur between the engine and transmission. Some examples where the combination of detailed engine and transmission models would be useful include neutral rollover noise, gear chatter and body boom among others.

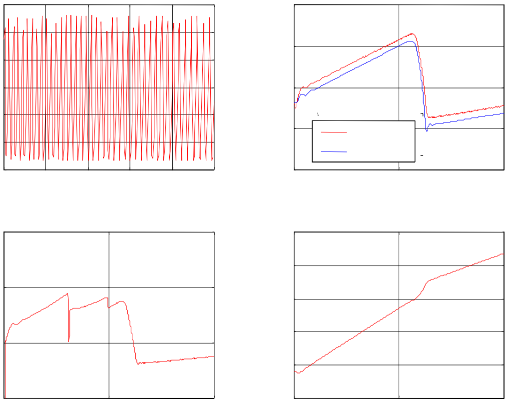

Figure 10: Detailed Engine and Transmission Analysis

5.2 Engine, Transmission and Chassis

The chassis and transmission including hydraulics model, as well as a simplified engine model were integrated to finally arrive at a detailed overall vehicle model. This model contains about 3500 parameters, 25000 nontrivial scalar equations, and 320 state variables. The equations are analyzed for systems of simultaneous equations (algebraic loops). Nonlinear systems of equations corresponding to the position equations of kinematic loops are found. Linear systems of equations correspond to inversions of the mass matrix and to velocity equations of kinematic loops. The symbolic manipulation takes just a couple of minutes to perform on a PC with a Pentium 500 MHz and 256 Mbytes of main memory. In this case, no comparisons of simulation results could be made, since such an integrated model did not exist in any other tool. The simulation time increased considerably due to the complex dynamics of the hydraulics. Simulating a start from zero velocity and a gearshift from first to second gear was performed over a 10 sec interval and took 125 min.

6 Conclusion

The `connector` object in Modelica is quite powerful. For the engine and chassis modeling, the ability to carry numerous signals on a single connector is essential. For the transmission and hydraulics work, the use of common `connector` definitions allowed several different libraries developed independently to be used together with no significant reworking.

While the model development described was very successful, there are many other applications for these models that have yet to be explored. The potential to do an even more comprehensive model of the powertrain system still exists and the development of less detailed, control system oriented models seems straightforward. In fact, several of the models that were developed were successfully exported to Simulink[4] as S-functions which enables plant models to be developed using the acausal approach available in Modelica.

A few issues were raised in this study about the suitability of Modelica for such a wide variety of systems. Overall, Modelica was very capable of expressing the behavior of all the models mentioned. While there is room for improving the Modelica language (*e.g.*, representing impulses) such improvements are not required in order to describe the behavior of all of the systems discussed here.

7 Acknowledgments

The authors would like to thank George Davis and Nizar Trigui who initiated and guided this feasibility study and who provided the necessary resources.

We would also like to thank the following people for the technical support of this study: Bill Tobler, Greg Pietron, Yuji Fujji and James McCallum assisted in the development of the transmission component models (both mechanical and hydraulic). In addition, they provided sample component data and helped us to create benchmark cases to compare Dymola results with the existing Ford in-house tools.

Cleon Davis helped develop the initial engine component models used in this study. Charles Newman and Hubertus Tummescheit assisted in the development of the engine subsystem models by helping to identify and explore the common difficulties that arise when developing engine and thermodynamic models. In addition, they assisted in collecting data and running validation cases.

Kevin Martus helped in understanding the ADAMS chassis model and some of the peculiar ADAMS component models.

References

[1] ADAMS. Mechanical Dynamics Inc., Homepage: http://www.adams.com/.

[2] Modelica Association. Modelica Language Specification (Version 1.3), http://www.Modelica.org, 1999.

---

[4] Simulink is a registered trademark of The MathWorks, Inc.

[3] Modelica Association. ModelicaAdditions.MultiBody, http://www.Modelica.org/library/library.html, 2000.

[4] Modelica Association. Modelica.Mechanics.Translational, http://www.Modelica.org/library/library.html, 2000.

[5] P. Beater. Modeling and Simulation of Hydraulic Systems in Design and Engineering Education using Modelica and HyLib. In *Proceedings of the Modelica 2000 Workshop*, Lund, Sweden, October 2000. Modelica Association.

[6] P. Bowles, M. Tiller, H. Elmqvist, D. Brück, S.E. Mattsson, A. Möller, H. Olsson, and M. Otter. Feasibility of detailed vehicle modeling. In *Proceedings of the SAE 2001 World Congress*. The Society of Automotive Engineers, Inc., 2001.

[7] Dymola. Dynasim AB, Lund, Sweden, Homepage: http://www.dynasim.se/.

[8] H. Elmqvist, S.E. Mattsson, and M. Otter. Modelica — A Language for Physical System Modeling, Visualization and Interaction. In *Proceedings of the 1999 IEEE Symposium on Computer-Aided Control System Design, CACSD'99*, Hawaii, August 1999. IEEE Control Systems Society.

[9] John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, 1988.

[10] S.E. Mattsson and G. Söderlind. Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives. *SIAM Journal of Scientific and Statistical Computing*, 14(3):677–692, May 1993.

[11] C.C. Pantelides. The Consistent Initialization of Differential-Algebraic Systems. *SIAM Journal of Scientific and Statistical Computing*, 9:213–231, 1988.

[12] H. Tummescheit, J. Eborn, and F. Wagner. Development of a Modelica Base library for Modeling of Thermo-Hydraulic Systems. In *Proceedings of the Modelica 2000 Workshop*, Lund, Sweden, October 2000. Modelica Association.