

# Feasibility of Detailed Vehicle Modeling

**Paul Bowles, Michael Tiller**

Ford Research Laboratory, Ford Motor Company, USA

**Hilding Elmqvist, Dag Brück, Sven Erik Mattsson, Andreas Möller, Hans Olsson**

Dynasim AB, Sweden

**Martin Otter**

German Aerospace Research Establishment (DLR), Germany

Copyright © 2000 Society of Automotive Engineers, Inc.

## ABSTRACT

A feasibility study is presented concerning detailed vehicle modeling, including submodels for engine, transmission mechanics and hydraulics, as well as three-dimensional chassis behavior. The study was conducted jointly by Ford Motor Company, Dynasim AB and DLR. The results demonstrate that complex behavioral models of each subsystem can be developed, used and validated independently from each other, and finally assembled together to an overall model. Therefore, this approach could be the basis to establish modeling standards that allow collaboration between model developers throughout the automotive industry.

## INTRODUCTION

One of the challenges in powertrain design is to evaluate not just the behavior of a particular subsystem (e.g., an engine) but also to evaluate the behavior of the entire vehicle and powertrain system. Typically, different modeling and simulation tools are used for each component. The use of different tools can be driven by several factors. For example, some tools are highly specialized for a particular subsystem which gives them very good predictive, design-oriented capabilities. Other tools may sacrifice predictive capability and rely more heavily on empirical relationships in order to run simulations with quick turnaround times or predict system level response directly. In practice, the result is that many tools are used and each has slightly different models with varying levels of detail.

Traditionally, detailed vehicle modeling has been considered impractical because no single tool was considered sufficient to analyze such large systems containing subcomponents from diverse domains. The results of our study presented below demonstrate that detailed vehicle modeling can be achieved in one simulation environment by

1. using Modelica [1,2], a general purpose modeling language not tailored to any particular engineering domain, to describe the models of all subsystems in a convenient, component-oriented way.
2. using Dymola [3] to transform, even large, Modelica models automatically into a form which can be solved by standard numerical methods. Either Dymola's simulation engine can be used to determine the solution or in many cases the transformed model can be incorporated into the block-diagram simulator SIMULINK [4] as a S-function MEX block.

For this study, first-principles Modelica models have been developed for the engine, transmission mechanics and hydraulics, driveline components, suspension and chassis. For the latter three-dimensional mechanical components, a translator from ADAMS [9] to Modelica was realized, to draw on available ADAMS models at Ford. The results from Dymola are compared with results obtained from several validated Ford tools that have been specialized to simulate particular subsystems. The same results could be reproduced for each subsystem with the additional benefit of being able to combine the subsystems without having to introduce any additional tools or use slow and unreliable co-simulation techniques.

The purpose of this paper is to emphasize that technologies, like the Modelica modeling language and the Dymola software, can solve real world problems and at the same time help streamline the modeling process when moving from concept to production.

## MODELICA

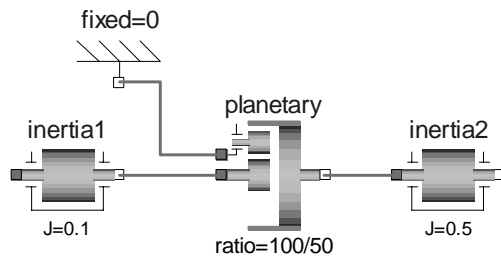
Modelica is a freely available object-oriented modeling language developed by the Modelica Association [1]. It is similar in spirit to VHDL-AMS [5] and Verilog-AMS [6], but with more emphasis on multi-domain modeling (i.e. not developed primarily for electrical systems), as well as

on graphical representation of models based on schematics. The Modelica modeling language could be implemented by a variety of simulation tools allowing models to be shared across simulation environments. Currently Dymola is the only simulation program available that implements Modelica.

## CAUSAL AND ACAUSAL MODELING

One important property of the Modelica modeling language is its ability to handle both dataflow and first-principles approaches to modeling. Dataflow formulations are commonly used to represent controller and simple plant behavior. However, complex plant behavior is often not practical to represent using dataflow formulations because such component models need to be developed with *a priori* assumptions about causality that are not consistent with all possible model uses.

A typical example of a Modelica model is shown in Figure 1, a screen shot of Dymola's schematic editor. It shows a section of a drive line, consisting of an ideal planetary gearbox where the carrier is fixed to the bearings and the sun and ring wheels are connected to inertias. All components are available from the Modelica standard library, which is developed together with the Modelica language.



**Figure 1: Modelica model of planetary gear with inertias**

All the details of a model, including the equations, graphical representation and documentation are defined using Modelica. For example, the model of Figure 1 has the following representation in Modelica<sup>1</sup>:

```

model DriveLine
  Inertia      inertia1(J=0.1);
  Inertia      inertia2(J=0.5);
  IdealPlanetary planetary(ratio=100/50);
  Fixed        fixed;
equation
  connect(inertia1.flange_b, planetary.sun);
  connect(inertia2.flange_a, planetary.ring);
  connect(fixed.flange_b, planetary.carrier);
end DriveLine;

```

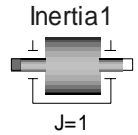
The second line in the model means that the new component `inertia1` is an instance of model class

`Inertia` and the value of the inertia is set to 0.1 kg m<sup>2</sup>. In the lower part of the model, it is defined how the components are connected together, e.g., the flange called `flange_b` of component `inertia1` is rigidly attached to flange `sun` of component `planetary`. Model class `Inertia` has the following representation:

```

model Inertia
  parameter SIunits.Inertia J=1;
  Flange_a flange_a;
  Flange_b flange_b;
  SIunits.Angle phi;
  SIunits.AngularVelocity w;
equation
  phi = flange_a.phi;
  phi = flange_b.phi;
  w = der(phi);
  J*der(w) = flange_a.tau + flange_b.tau;
end Inertia;

```



Library `SIunits` contains predefined types, e.g.,

```

type Inertia = Real(unit = "kg.m2");
type Angle   = Real(unit   = "rad",
                    displayUnit="deg");

```

which are used in model `Inertia` to define the physical variables with appropriate units. The second line in model `Inertia` defines the inertia `J` as a parameter of type `SIunits.Inertia` which is constant during simulation. The next two lines define in which way an `Inertia` component can be connected to other elements with appropriate interfaces. Specifically, `flange_a` is an instance of the connector class `Flange_a`:

```

connector Flange_a
  SIunits.Angle phi;
  flow SIunits.Torque tau;
end Flange_a;

```

Variable `phi` in the connector is the absolute angle at this point, whereas `tau` is the torque at the mechanical connection represented by `Flange_a`. The `flow` attribute indicates that a zero-sum equation is generated for the torques of all flanges which are directly connected together. The definition of connector `Flange_b` is identical to `Flange_a` with the only exception that the graphical representation is different, to distinguish them in a schematic.

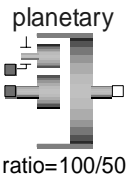
The lower part of model `Inertia` contains the component equations. Basically, the equilibrium condition is stated between the derivative of angular momentum and the sum of the torques of the two flanges. In addition the angles of the two flanges are defined to be identical. Operator `der(...)` in the equations represents the time derivative of the variable in parenthesis. Note, that the equations in a model are treated as mathematical equations and not as assignment statements. Finally, model `IdealPlanetary` is defined as:

<sup>1</sup> For brevity, the graphical information for the models in this section was omitted.

```

model IdealPlanetary
  parameter Real ratio=100/50;
  Flange_a sun, carrier;
  Flange_b ring;
equation
  (1+ratio)*carrier.phi =
    sun.phi + ratio*ring.phi;
  ring.tau = ratio*sun.tau;
  carrier.tau = -(1 + ratio)*sun.tau;
end IdealPlanetary;

```



Parameter *ratio* is the ratio of the number of ring teeth to the number of sun teeth. In the equation section it is stated that the angles of the three flanges of the gearbox are constrained and the equilibrium conditions of the flange torques are given. The planetary gear is a good example of a component which can have several different kinds of causality. For example, the first equation in the equation section can be used to solve for either of the three angles, depending on how the planetary gear is connected and which state variables are used for the description. An acausal representation allows the planetary gear to be used in a variety of contexts because it does not require an *a priori* assumption about the computational causality.

Acausal models are allowed in Modelica since the low level description is composed of equations, i.e. the model developer does not have to solve the fundamental equations for any particular variable. Dymola incorporates computer algebra to perform this task automatically using specialized algorithms which allow handling of large sets of equations efficiently. Note, that the algorithms used in general purpose computer algebra programs, such as Maple or Mathematica, are more general than those used by Dymola and as a result they tend to be slower and poorly suited for large systems of equations.

Solving a system such as the drive line in Figure 1 directly with a standard numerical integration method, gives rise to severe numerical problems because every inertia has one degree of freedom, but by connecting them together via the planetary gear, the overall system has only one degree of freedom and not two (this model is a so-called index-3 DAE system; most numerical integration routines can only solve index-1 and only few can solve index-2 systems).

For this reason, Dymola automatically differentiates the constraint equation between the flange angles of the planetary gear twice and uses these newly generated equations to remove the states of, say, component *inertia2*. If initial conditions are given for *inertia1*, the states of *inertia2* are computed algebraically from the states of *inertia1*. Such types of symbolic transformation techniques are essential to translate a Modelica model into a form which can be solved reliably and efficiently by standard numerical integration methods.

## REUSABILITY

Another goal of the Modelica modeling language is to promote the development of reusable component libraries. This has been accomplished by object-oriented techniques in Modelica and by allowing true equations. Several free Modelica libraries are available and can be downloaded from the internet<sup>2</sup>. These libraries provide ready-to-use components for electrical and electronic systems, 1-dimensional translational and rotational systems, 3-dimensional mechanical systems, hydraulic systems, power systems, heat flow and input/output blocks (sources, controllers, discrete systems etc.). A number of additional free libraries are currently under development for other domains such as thermo-fluid systems.

In the remaining part of this paper it is shown how all components of a detailed vehicle model have been realized in Modelica and how the subcomponents have been validated.

## ENGINE

The engine model used in this study includes detailed thermodynamics of the combustion event (*i.e.*, gas properties, reaction of charge into products and varying chemical composition). Many of the models contain equations which can be found in various textbooks [7,8].

All the models related to the engine subsystem were developed as a hierarchical library of components. This library could easily be passed on to other users and/or stored in a central location for others to use.

## MECHANICAL COMPONENTS

The mechanical components of the engine (some of which can be seen in Figures 2 and 3) developed for this library include an ideal dynamometer, timing belt, crank, connecting rod, piston and valve. Other models, such as the crankshaft and camshafts, can be represented using models already available in the Modelica Standard Library.

The mechanical models that were developed use translational and rotational "connector" definitions provided by the Modelica Standard Library. Therefore these models can be seamlessly connected to components other Modelica which use the same connector definitions.

## THERMODYNAMICS

In addition to the mechanical components, several models related to the thermodynamic processes of the engine were required. Besides combustion, models which characterize flow over the engine valves, manifold filling and emptying, throttle behavior, infinite reservoirs

<sup>2</sup> <http://www.modelica.org/library/library.html>

and heat transfer were developed. Some of these can be seen in Figures 2 and 3 as well.

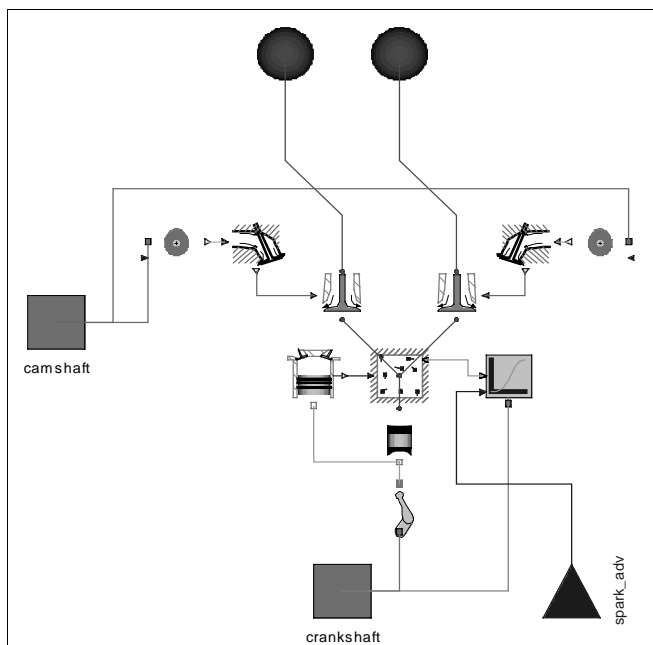
## CONFIGURATIONS

The first step in developing the library was to create individual component models. For example, starting with engine valves and ideal reservoirs, it was possible to create simple tests that exercise and validate these models. Once these tests are complete, more complex systems could be built.

As part of this study, many subsystem models have been realized (*i.e.* models composed of other smaller components). For example, individual cylinders composed of:

- Engine valves.
- Ports (*i.e.* geometry effects).
- In-cylinder control volume.
- Combustion model.
- Crank, rod and piston.
- Camshaft and crankshaft connections.

This collection is grouped together as a single subsystem model. The advantage of creating such a configuration is that it can be duplicated many times. A schematic of an individual cylinder is shown in Figure 2.

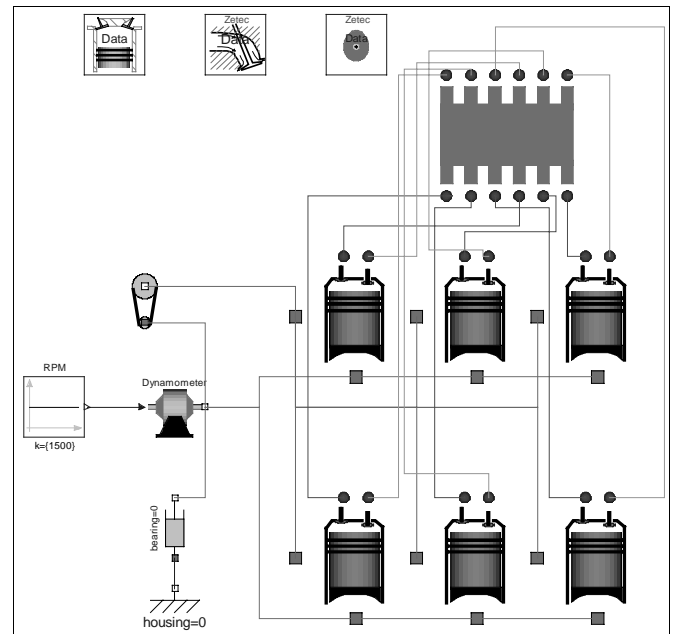


**Figure 2: Individual Cylinder Schematic**

It is used to build a V6 engine model which, in addition to six of the individual cylinder, also adds:

- Crankshaft and camshaft.
- Timing belt.
- Manifold conditions.

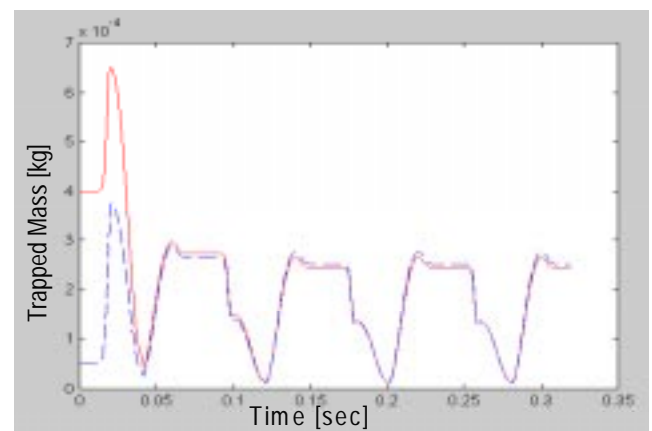
The schematic of the V6 assembly is shown in Figure 3. Each of the cylinder icons in Figure 3 represents a grouping of the components shown in figure 2.



**Figure 3: V6 Engine Assembly**

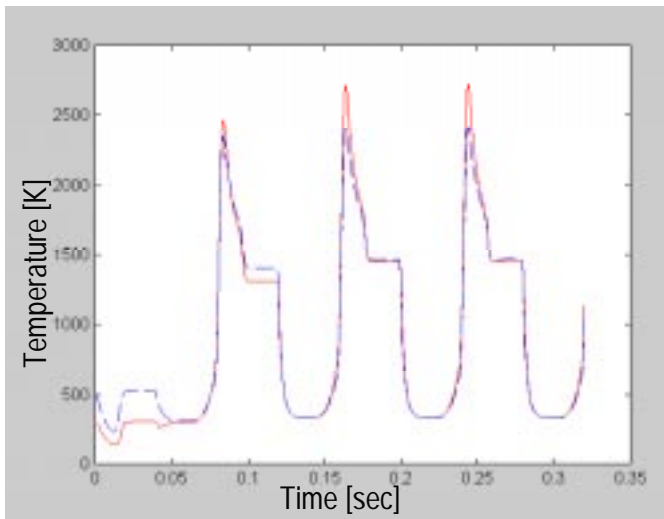
## RESULTS

To validate the engine models, the results were compared against validated Ford in-house analysis tools. Figure 4 shows the trapped mass in a side-by-side comparison of the two programs. While the two simulations do not start at the same initial conditions, they do converge to the same curve after a few cycles. The slight difference in converged results is attributed to the fact that the gas properties of the working fluid were not identical.



**Figure 4: In-cylinder Trapped Mass**

Figure 5 shows a side-by-side comparison of in-cylinder temperature for the same analysis<sup>3</sup>. Again, the initial conditions are different but the results converge after only a few cycles.



**Figure 5: In-cylinder Temperature**

There are two important caveats to mention regarding these results. The first is that not all of the capabilities of the Ford in-house tools have been implemented due to time constraints. Additionally, it is much easier to add new models and explore advanced powertrain concepts in this environment.

## TRANSMISSION

An automatic transmission can be divided into mechanical and hydraulic subsystems. The mechanical subsystem includes all of the rotational inertias of the transmission, gearsets, clutches, and axle halfshafts. In order for the transmission component to be simulated as a stand-alone model, simple models for the engine, wheels and longitudinal vehicle dynamics have to be added. The hydraulic subsystem consists of all components that principally affect the behavior of the transmission hydraulics. This includes the hydraulic flows, valves, accumulators, and solenoids.

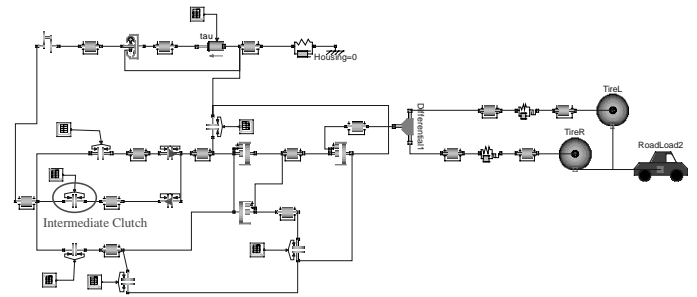
### MECHANICAL

The mechanical subsystem model, shown in Figure 6, consists of the following components:

- Multiple lumped inertias throughout the transmission and driveline;
- Engine model, consisting of a relative torque (both pulsating and non-pulsating) between the crankshaft and case inertias;
- Engine mount with torsional stiffness and damping;

<sup>3</sup> The peak in-cylinder temperatures are larger than expected because there is no in-cylinder heat transfer in these simulations (for comparison purposes).

- Three-port torque converter based on steady state test data;
- Multiple clutches and one-way clutches throughout the transmission;
- Multiple planetary gearsets;
- Differential gear;
- Halfshaft backlash, torsional stiffness and damping;
- Tires;
- Vehicle road load model.



**Figure 6: Transmission Mechanical Subsystem**

The benchmark maneuver for the transmission feasibility study was a 1-2 shift. This maneuver consists of engaging the intermediate clutch; all other clutches and bands remain either locked or unlocked throughout the shift. The intermediate clutch is highlighted in Figure 6.

There are a number of potential challenges that the model in Figure 6 poses for a simulation program. The first is the ability to handle multiple lumped inertias with kinematic constraints. For example, the two rotational inertias downstream of the torque converter are rigidly connected through a fixed gear ratio. There exists only one degree of freedom and Dymola is able to associate all of the relevant inertias with this degree of freedom. This requires symbolic transformations including differentiation. Similar constraints exist with the interconnection of the planetary gears.

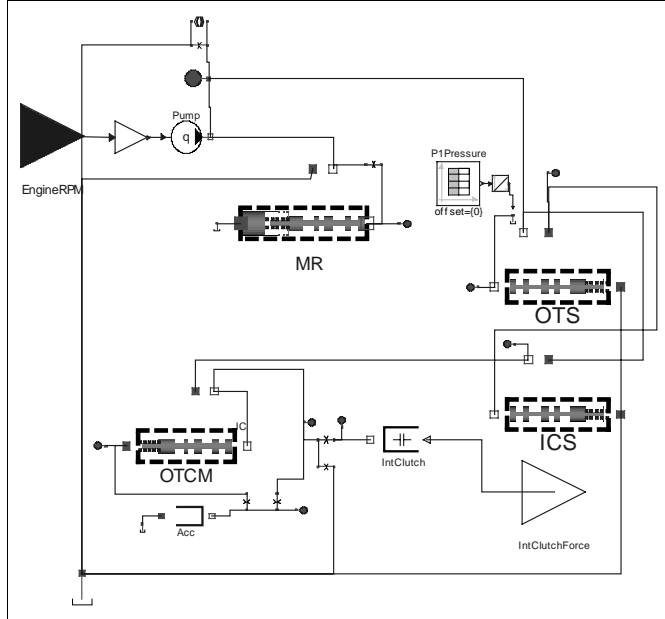
### HYDRAULICS

Hydraulic transmission systems pose a different set of modeling challenges than their mechanical counterparts. Apart from being a different physical domain with its own set of governing equations, hydraulic AT systems are often modeled as extremely stiff and nonlinear and have tightly coupled component dynamics. This is due to a number of factors including: small spool masses; rapid changes in line pressures; small line volumes; orifice sizes which can be very large relative to the volumes they connect; and steep gradients in the orifice equation near zero pressure drop.

Being able to simulate AT hydraulics is very important in capturing the shifting dynamics. In many automatic transmissions, small differences in the behavior of the hydraulic subsystem can have a significant impact on the shift dynamics.

A complete hydraulics model was not possible for this study due to time constraints; however, such detail is not necessary to demonstrate feasibility. The idea for the study was to capture the essential kinds of dynamics that are present in a hydraulic system, without necessarily building a complete model. In light of this, the hydraulics were simplified to a minimum subsystem required to simulate a 1-2 shift. Such a subsystem consists of four spool valves, an accumulator and the intermediate clutch. Because of the model simplification, the dynamic behavior of this subsystem is not expected to accurately represent that of the complete hydraulics performing a similar maneuver. What is important is that the Modelica models be able to represent a multi-valve system, and that it reproduces results that are close to those predicted by HCS, the Ford in-house hydraulic systems simulation tool.

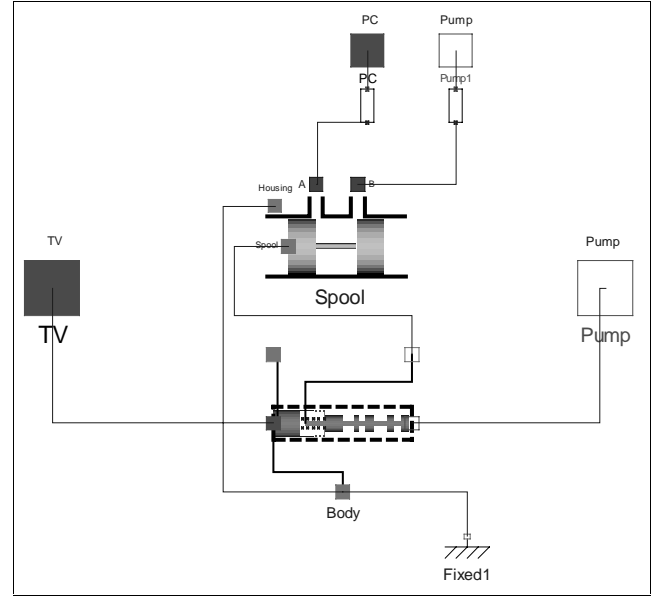
Figure 7, shows the four valves modeled in this subsystem. These are the main regulator (MR), 1-2 shift (OTS), 1-2 capacity modulation (OTCM) and intermediate clutch shuttle (ICS) valve. The 1-2 shift maneuver is initiated by stroking OTS; this is accomplished by rapidly increasing the left-side pressure on that spool, P1Pressure. The flow through OTS changes the position and flow through ICS and OTCM, which causes the accumulator and intermediate clutch to stroke. Engagement occurs towards the end of the intermediate clutch stroke.



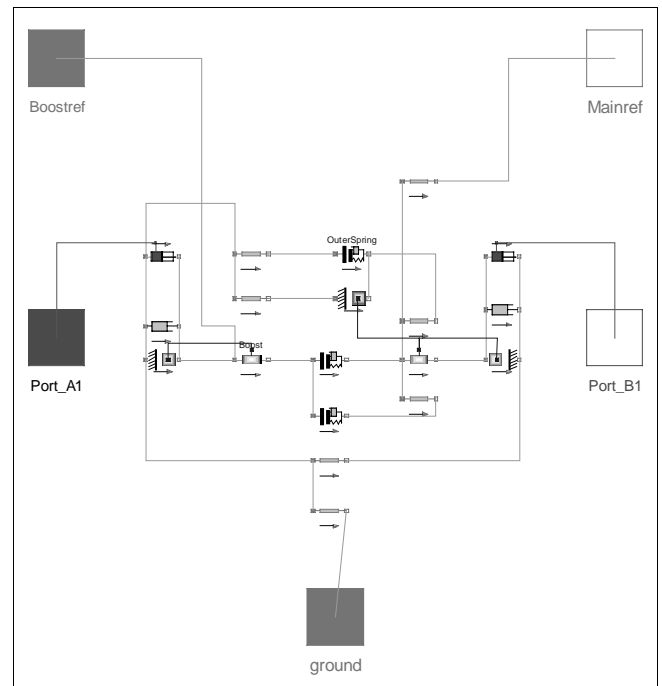
**Figure 7: Hydraulic Subsystem**

Each one the valves is represented by a separate model. The main regulator valve model is shown in Figure 8. The model is split into two main submodels. All of the moving valve parts, associated springs and pressure chambers, and all geometry not directly related to the flow through the valve is contained in the Body model. The actual spool valve orifice model, including the orifice

equation and all spool and port geometry pertinent to fluid flow through the valve is contained in the Spool model.



**Figure 8: Main Regulator Valve Assembly**



**Figure 9: Main Regulator Valve Body Model**

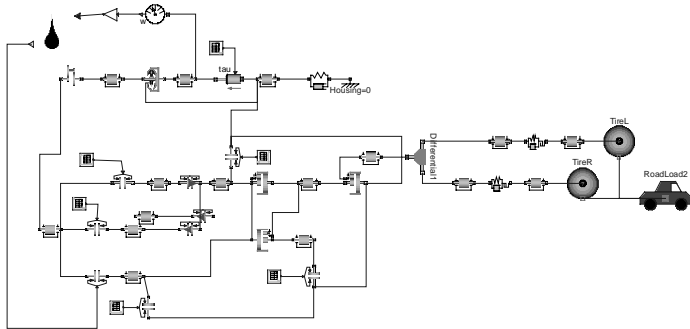
Figure 9 shows the internals of the Body model for the main regulator valve. Two masses are contained therein; they are the boost and regulator valve spools. Collisions between the valve spools and the valve chamber are modeled as coefficient of restitution collisions. The collision between the two spools was modeled as a stiff spring/damper. There are two springs in the main regulator valve, as well as viscous damping for each valve. On either side of the valve chamber, pressure chambers are present that can exert an inward force on the boost and regulator valves. These chamber models

are connected to the pressure connectors on either side of the model. `Bar` models (slender bars in Figure 9) allow for reference points on the valves to be used, such that springs and collision interfaces can be established where they physically exist. In this way, the model bears some meaningful physical resemblance to the valve schematic.

## COMPLETE SUBSYSTEM

The mechanical and hydraulic layers of the transmission have been discussed separately. In reality, the mechanical and 1-2 shift hydraulic subsystems are coupled: Firstly, the engagement of the intermediate clutch is dictated by the transmission hydraulics. The pressure acting on the clutch dictates the clutch capacity based on the component geometry. Secondly, the transmission volumetric pump is mechanically connected to the engine crankshaft. This means that the speed at which fluid is being pumped through the transmission depends on the engine speed.

Coupling between the mechanics and hydraulics is only achieved if the two models are combined and simulated together. To verify that Dymola would be able to simulate both models together as well as their interaction, the hydraulics model was imported into the mechanical subsystem model.



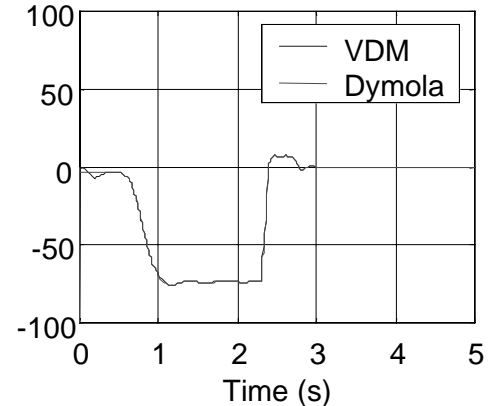
**Figure 10: Complete Transmission Subsystem**

Figure 10 is very similar to Figure 6. The difference is that the 1-2 shift hydraulics in Figure 6 have been iconified into a black oil drop and connected to the mechanics. The physical connections from the hydraulics to the engine crankshaft and intermediate clutch are clear from represented by the lines entering and leaving the hydraulic subsystem. It should be pointed out that the connection of the hydraulics was straightforward; the hydraulics model was inserted into the mechanics model and connected, and this new resulting model was recompiled and simulated.

## RESULTS

### Mechanical Subsystem

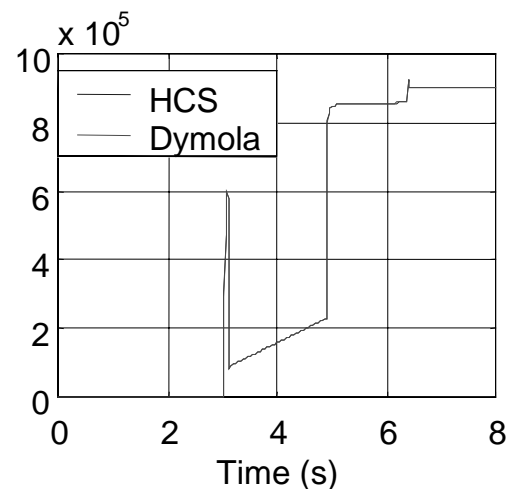
Figure 11 compares the mechanical subsystem response from VDM (a validated Ford in-house tool) and the Modelica models for the same 1-2 shift maneuver. This is one of many plots that show excellent agreement between the two sets of results<sup>4</sup>. At 0.5 sec, a rise in the torque output of the engine results in an increase in engine, turbine and vehicle speed, as well as increases in the forward clutch, low-intermediate band and halfshaft torques.



**Figure 11: Forward Clutch Torque (Nm)**

The intermediate clutch is engaged at about 2.4 sec. Clutch lock-up occurs around 2.7 sec, when the intermediate clutch torque drops off, carrying only what is required to maintain its locked state.

### Hydraulic Subsystem



**Figure 13: Hydraulic Only IC Pressure (Pa)**

As with the mechanical subsystem results, there is excellent agreement between Dymola and the Ford in-house simulation software (HCS in this case). From 1 sec to 2 sec, the pump speed is raised, which causes a rise in the pump line pressure and a change in main

<sup>4</sup> In fact, the agreement in most of these plots is so good it is difficult to distinguish the two signals.





After this step, the data structure contains a faithful representation of the ADM file.

In the second step of the translation, the model is analyzed. The data structure is augmented with synthesized information stored as additional attributes. The analysis is focused on model topology and markers used by PART, JOINT and FORCE statements.

The third step is to generate Modelica code for the ADAMS statements. Most 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(...);

connect(P1.b, J1.a);
connect(J1.b, P2.a);
```

ADAMS provides several special functions that operate on marker numbers. For example, in force elements like SFORCE, it is possible to specify a force expression including functions like DX(i, j, k) which operate on marker numbers. The function DX calculates the actual distance between marker i and j in X-direction of marker k. In addition to the static data associated with markers, dynamic data thus needs to be associated. This is accomplished by making connectors correspond to markers since the connectors of the MBS library have a representation of position, orientation, forces and torques:

```
connector MBSCutBase
  Position r0[3];
  Real S[3, 3]; // Orientation
  flow Force f[3];
  flow Torque t[3];
end MBSCutBase;
```

It is thus easy to implement a Modelica function, DX, operating on connectors:

```
function DX
  input MBSCutBase I, J, K;
  output Real x;
algorithm
  x := (I.r0 - J.r0)*K.S*{1,0,0};
end DX;
```

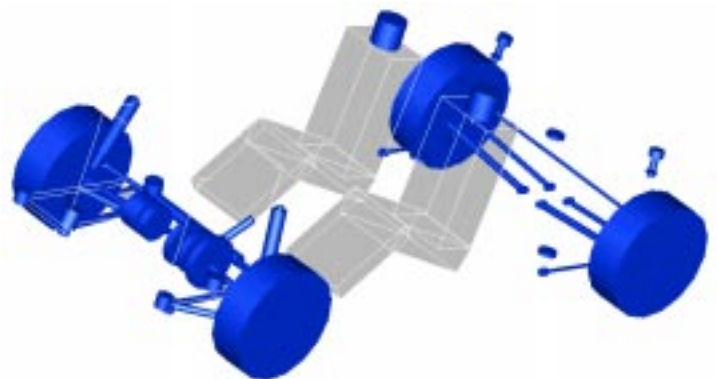
The ADAMS calls to the DX function contains marker numbers. In the translated Modelica model, there is no mechanism to refer to the corresponding connectors by number. Instead, the translator needs to analyze all expressions and recognize all functions like DX operating on marker numbers, look up the corresponding connector and generate Modelica code with hierarchical names like:

```
Real dx = Adams.DX(P2.a, P3.b, P2.a);
```

Functions for spline interpolation were also implemented in the Modelica language.

## RESULTS

The 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 is shown in Figure 18.



**Figure 18. Modelica model of chassis**

Certain simulation experiments have been performed on the model of the vehicle chassis both in ADAMS and with Dymola using the output from the ADAMS to Modelica translator. Similar results were obtained after setting appropriate maximum step size for ADAMS, since tightening the tolerance gave problem with convergence.

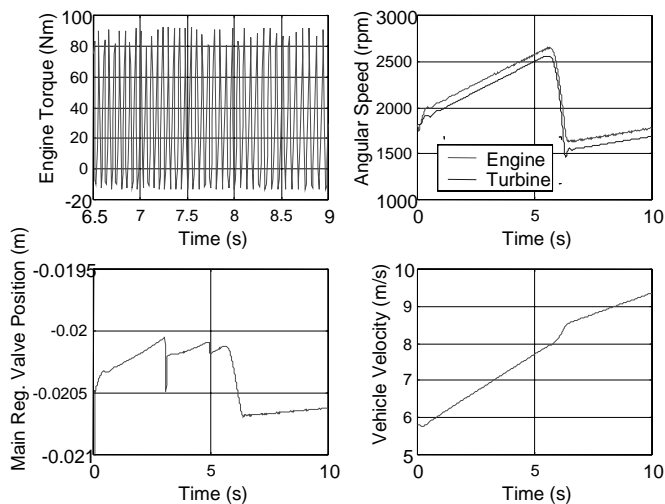
The simulation of a 10 sec interval took 15 min in ADAMS. Simulation in Dymola with the integration method DASSL and a tolerance of  $1e-4$  took 17 min. For a tolerance of  $1e-6$ , the simulation time was about 3 times longer.

## INTEGRATION

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.

## 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 the previous cycle average engine model (shown in Figure 6). The results of the analysis using the detailed engine and transmission models can be seen in Figure 19.



**Figure 19: Detailed Engine and Transmission Analysis**

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.

## 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 3 500 parameters, 25 000 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 corresponds to inversion 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 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.

## CONCLUSION

This work demonstrates that technologies like the Modelica modeling language and the Dymola translator and simulator are capable of handling detailed vehicle models including three-dimensional mechanics, transmission and engine submodels. Such technologies enable greater collaboration between model developers. This will not only streamline processes within a company but can also streamline processes involving suppliers and external researchers. Open specifications, like Modelica, are the key to a future where *models are treated as easily exchangeable commodities*.

## 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 Fujii and James McCallum helped to develop 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. Modelica, Modelica Association. Homepage: <http://www.Modelica.org/>
2. H. Elmqvist, S. E. Mattsson, M. Otter: "Modelica — A Language for Physical System Modeling, Visualization and Interaction", Plenary presentation by H. Elmqvist, The 1999 IEEE Symposium on Computer-Aided Control System Design, CACSD'99, Hawaii, August 22-27, 1999.
3. Dymola - Dynamic Modeling Laboratory, Dynasim AB, Lund, Sweden. Homepage: [www.Dynasim.se](http://www.Dynasim.se)
4. SIMULINK, MathWorks. Homepage: <http://www.mathworks.com/products/simulink/>
5. "IEEE Standard 1076.1: VHDL-AMS". Homepage: <http://vhdl.org/vi/analog/>
6. "Verilog-A/MS Version 2.0 Manual", Open Verilog International, <http://www.ovi.org/>
7. Heywood, JohnB., "Internal Combustion Engine Fundamentals", McGraw-Hill, Inc., 1988

8. Wark, Kenneth J., "Thermodynamics", 5<sup>th</sup> Edition, McGraw-Hill, Inc., 1988
9. ADAMS, Mechanical Dynamics Inc., Homepage: <http://www.adams.com/>

## CONTACT

Paul Bowles ([pbowles@ford.com](mailto:pbowles@ford.com)) and Dr. Michael Tiller ([mtiller@ford.com](mailto:mtiller@ford.com)) work in the Engine and Processes Department of the Powertrain and Vehicle Research Laboratory at the Ford Research Laboratory. Dr. Tiller is a Technical Specialist at Ford and treasurer of the Modelica Association.

Dr. Hilding Elmqvist ([Elmqvist@Dynasim.se](mailto:Elmqvist@Dynasim.se)) is president of Dynasim AB, the company which develops Dymola. He is also past chairman and current secretary of the Modelica Association. Dag Brück ([Dag@Dynasim.se](mailto:Dag@Dynasim.se)), Dr. Sven Erik Mattsson ([SvenErik@Dynasim.se](mailto:SvenErik@Dynasim.se)), Andreas Möller, and Dr. Hans Olsson ([Hans@Dynasim.se](mailto:Hans@Dynasim.se)) work in the development department of Dynasim AB.

Dr. Martin Otter ([Martin.Otter@dlr.de](mailto:Martin.Otter@dlr.de)) is research engineer at DLR, the German Aerospace Research Establishment. He is the current chairman of the Modelica Association.