Proceedings
of the 3rd International Modelica Conference,
Linköping, November 3-4, 2003,
Peter Fritzson (editor)

John Batteh, Michael Tiller and Charles Newman
Ford Motor Company, USA:
Simulation of Engine Systems in Modelica
pp. 139-148

Paper presented at the 3rd International Modelica Conference, November 3-4, 2003, Linköpings Universitet, Linköping, Sweden, organized by The Modelica Association and Institutionen för datavetenskap, Linköpings universitet

Simulation of Engine Systems in Modelica

John Batteh **Michael Tiller** **Charles Newman**

Ford Motor Company, Powertrain Research Department
Dearborn, MI USA
{jbatteh, mtiller, cnewman}@ford.com

Abstract

This paper details the use of the Modelica modeling language for the simulation of engine systems. The first part of the paper briefly outlines some of the challenging, multi-domain components of engine system modeling and is followed by a discussion of some of the connectors, interfaces, and model templates that enable robust, efficient model development. The remainder of the paper presents selected modeling examples with particular attention to the structure and implementation of the models that promotes model flexibility and re-use.

1 Introduction

As automobile manufacturers face increasing pressure to reduce emissions, increase fuel economy, reduce development costs, and enhance vehicle performance and driveability, it has become especially crucial to consider optimization opportunities at the system level. While it is conceptually possible to obtain system improvements via prototype hardware fabrication, this process is inefficient, costly, and sub-optimal. With the development of modeling tools that allow robust, multi-domain, system-level simulations, it is becoming increasingly attractive to perform this optimization process in the virtual environment.

Engine systems, in particular, contain a wide range of multi-domain physical modeling challenges [1]. Table 1 contains a partial list of physical processes and modeling domains that could be considered in the modeling of a spark-ignited (SI) engine system depending on the particular analysis and desired level of detail. Due to the wide variety of physical processes and modeling domains along with the inherent interactions, it is imperative to have a descriptive language that is capable of modeling across the different physical domains. This need only increases as more of the overall vehicle system and associated attributes (*e.g.* NVH, safety, *etc.*) are included.

Table 1. Physical processes and modeling domains for an engine system

Physical Process	Modeling Domain(s)
Intake and exhaust valve actuation mechanisms	M, F
Intake and exhaust flow past the valves	T
Piston and crankshaft motion	M
Manifold dynamics in the intake and exhaust systems	T, F
Injection and transport of liquid fuel and fuel vapor	T
In-cylinder fluid motion	T, F
Ignition and flame propagation in the combustion chamber	T, Ch
Heat transfer between the gas, fuel, coolant system, and metal surfaces	Th
Frictional effects in engine, valvetrain, and powertrain	M, Th
Emissions formation and mitigation	T, Th, Ch
Thermal response of the intake system, engine, and exhaust system	Th
Coolant and lubrication flow	F
Powertrain, chassis, and mount dynamics	M
<u>Legend</u> *Ch = Chemical* *F = Fluid (distributed)* *M = Mechanical* *T = Thermodynamic* *Th = Thermal*	

Modelica[1] [2] with its high-level, acausal, declarative formulation for physical modeling is an ideal language for multi-domain system simulations. The Modelica standard `Mechanical`, `Rotational`, `MultiBody`, and `Thermal` libraries contain the connector definitions, interfaces, and basic models that provide the framework for the modeling of engine systems. The sections that follow discuss the use

---

[1] Modelica is a trademark of the Modelica Association

of these standard libraries along with the supplemental connectors and associated models that enable the formulation and simulation of engine system models.

2 Physics Overview

For each of the physical processes described in Table 1, models of varying level of detail can be formulated. Due to the number of component models used in a typical engine systems simulation, it is impractical to discuss the physics of particular models in detail. This section is meant to give a very brief overview of some of the physics involved in engine systems modeling.

Mechanical modeling in an engine system includes a combination of 1D and multi-dimensional dynamics. Typically, the multi-dimensional dynamics are of interest in detailed models of the vehicle dynamics and mounting systems. A 1D approach is often used in modeling the engine itself. Within the 1D framework, the model of the valve actuation mechanism can either include kinematic relationships (*i.e.* cam motion constrained to the motion of the crankshaft with valve lift prescribed as a function of the cam motion) or dynamic behavior (see [3] for a discussion of a dynamic, camless valve actuator model). Similarly, the piston can be modeled as massless using kinematic relationships between the piston, crank-slider, and crankshaft or can include the effects of piston mass from a force balance.

Modeling the thermodynamics is a crucial part of engine systems modeling. Typically several control volumes are formulated for which fundamental equations for energy and mass conservation are applied:

$$\frac{dU}{dt} = \dot{Q} - \dot{W} \tag{1}$$

$$\frac{dM}{dt} = \dot{m} \tag{2}$$

A typical engine model might include one (or several) control volumes in the cylinder, the intake system, and the exhaust system with mass and energy exchange between the volumes. Flow past the valves in an engine is typically modeled using isentropic relationships for flow past an orifice with an experimentally determined discharge coefficient [1]. The calculations of the requisite thermodynamic properties come from models with varying treatments of the species (*i.e.* fuel, fresh air, *etc.*) and levels of detail (*i.e.* constant c_p and c_v, polynomial property functions, chemical equilibrium mixture calculations [4], *etc.*). Fluid modeling is similar to thermodynamic modeling but usually involves a larger number of distributed control volumes and may involve the conservation of momentum as well. For example, accurately capturing the pressure dynamics of the flow in induction and exhaust systems requires a high level of discretization, perhaps even with specialized numerical techniques for shock capturing.

Heat transfer and thermodynamics are intimately linked in engine systems via Eq. (1). Convective heat transfer between the gas and the metal surfaces affect the volumetric efficiency of the engine, heat losses during the power stroke, heat losses in the exhaust system, and the thermal response of the engine and exhaust system components. The convective heat transfer is modeled from the fundamental constitutive equation:

$$\dot{Q} = \overline{h} A \left(T_g - T_w \right) \tag{3}$$

where the average convective heat transfer coefficient comes from experimental correlations. Cold start thermal response of the engine components is key from the standpoint of both mixture preparation and emissions formation and mitigation.

Combustion is a highly complex process involving thermodynamics, heat transfer, fluid motion, and chemical kinetics. Combustion models come in many flavors and with varying levels of fidelity. The combustion process can be simplified to a prescribed heat release process, such as a Wiebe function [1] for mass fraction burned. More detailed, predictive combustion models typically can account for multi-zone combustion and heat transfer, the effects of charge motion on the combustion process, variations in the laminar flame speed for different cylinder conditions, *etc.* (see [4] and the references therein for a description of a detailed combustion model in Modelica).

3 Interfaces

Standard interfaces are a key element for developing flexible models. Experience has shown that the most powerful and flexible Modelica libraries are based on solid connector definitions. The remainder of this section discusses some of the modeling elements that comprise the engine architecture.

3.1 Thermal Architecture

The heat transfer process plays a significant role in engine systems modeling. The interaction between the air in the cylinder and the metal surfaces in the intake, exhaust, and cylinder affects the liquid fuel preparation process along with the volumetric efficiency, performance, and emissions of the engine.

One challenge in modeling the thermal effects in the engine is the variety of different models that can be used to represent the thermal response of the various pieces. For example, an engine thermal response model could be formulated on a cylinder-by-cylinder basis or could be a lumped model at the engine level. To allow for both of these formulations and to minimize the number of connections between the engine or cylinder and the thermal models, the special thermal connectors in Figure 1 were developed. Modelica code fragments for these connectors are shown in Figure 2. The `CylinderTemperatures` connector is a "mega connector"- a connector that is an aggregate of other connectors- and can be thought of as a thermal bus. It contains a number of thermal and friction connectors that comprise the pre-defined standard thermal cylinder architecture. This architecture defines the elements that are included in every cylinder thermal response model and is represented graphically in Figure 3. This breakout box explicitly shows all the connectors that are lumped into the single `CylinderTemperatures` connector and is used in the low-level cylinder heat transfer models to facilitate the graphical connection of the individual elements of the heat transfer model. The `ThermalEnvironment` connector is the engine-level connector and is an array of `CylinderTemperatures` connectors. This parametric representation scales with the number of cylinders being modeled and, by consolidating the signals onto one connector, allows for a single connection between the engine and the engine thermal response model at the top level. The cylinder and engine connectors will be seen repeatedly in the standard interfaces that follow.

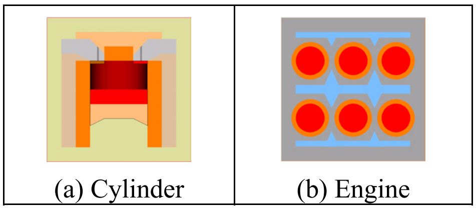

(a) Cylinder
(b) Engine

Figure 1. Thermal connectors

```
connector CylinderTemperatures
  import HeatTransfer=Modelica.Thermal.HeatTransfer;
  outer parameter Ford.Types.EngineTopology
    engine_topology;
  HeatTransfer.Interfaces.HeatPort_a head;
  HeatTransfer.Interfaces.HeatPort_a intake_valves[
    engine_topology.intake_valves];
  HeatTransfer.Interfaces.HeatPort_a block_coolant;
  HeatTransfer.Interfaces.HeatPort_a cylinder_liner;
  HeatTransfer.Interfaces.HeatPort_a piston;
  HeatTransfer.Interfaces.HeatPort_a oil;
  Ford.Engine.Interfaces.Friction valvetrain;
…
end CylinderTemperatures;
```

```
connector ThermalEnvironment
  outer parameter Ford.Types.EngineTopology
    engine_topology;
  CylinderTemperatures
cylinder_temperatures[engine_topology.cylinders];
end ThermalEnvironment;
```

Figure 2. Excerpts from the thermal connectors models

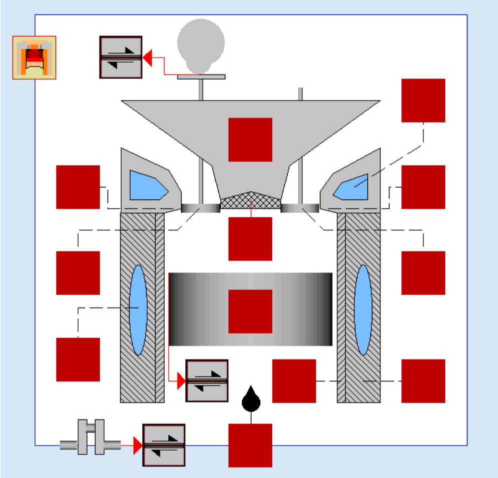


Figure 3. Breakout box showing elements of `CylinderTemperatures` connector

The thermal architecture in the engine provides the framework for the interactions between the cycle simulation models and the engine temperature models, thereby allowing independent selection of the either model. Roughly speaking, the cycle simulation models are responsible for computing the "metal-gas" thermal interactions while the engine temperature models calculate the "metal-fluid" interactions.

3.2 Cylinder Interface

The cylinder interface defines the framework for the cylinder implementation process. The standard interface is shown in Figure 4 and defines the exterior connection points for the cylinder. The `partial model` contains three 1D rotational connectors, one each for the crankshaft, camshaft, and engine block. The connection to the engine block allows for the rotational motion of the engine on the mounts. The interface also includes the previously discussed `CylinderTemperatures` connector for the cylinder thermal environment along with thermodynamic connectors for both the induction and exhaust systems. The thermodynamic connectors contain pressure, temperature, species mass fraction, species mass flow rates, and convected energy along with information related to fluid properties. It is anticipated that these thermodynamic connectors will be replaced with those from the Modelica standard fluids library currently under development [5].

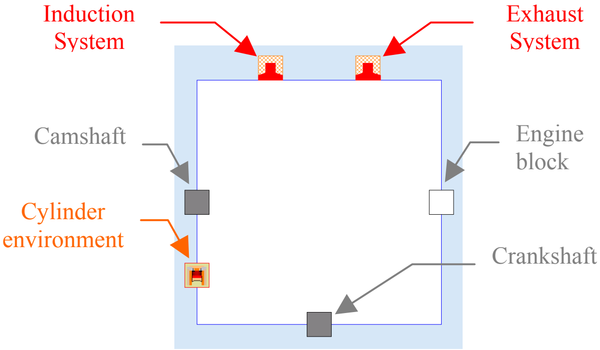


Figure 4. Cylinder interface

3.3 Engine Interface

The standard engine interface is shown in Figure 5. This `partial model` contains two 1D rotational connectors, one each for the crankshaft and the engine block. In addition, the interface contains a `ThermalEnvironment` connector to represent the engine thermal behavior. Note the absence of the induction and exhaust system thermodynamic connectors in the engine interface. These connectors have been omitted from the interface definition so that derived models can define their own plenum configurations (*i.e.* single plenum, dual plenum, *etc.*). Section 4.2 describes models that extend from this engine interface and instantiate the needed components (*i.e.* cylinders, *etc.*) for a complete engine implementation.

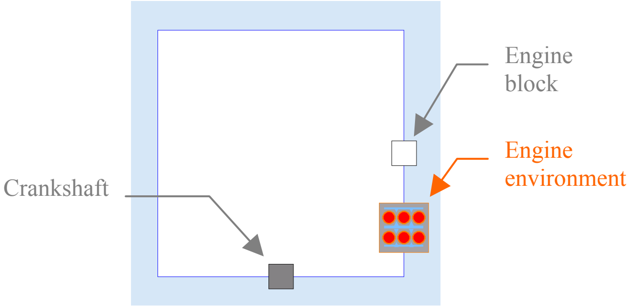


Figure 5. Engine interface

3.4 Medium Models

The working fluid is defined using the MediumModel idiom [4]. This approach defines a consistent set of models, functions, constants, and connectors that contain all the medium-specific information and thus define a particular implementation of the MediumModel idiom. For example, the material property calculations, equations of state, chemical species representation, combustion chemical kinetics, and associated helper functions could be included in the formulation. Implemented via replaceable packages, the MediumModel idiom enables the orthogonal development of property models and the components that use them (*i.e.* the decomposition of medium and machine) and provides an organized, consistent framework for the development of models with varying levels of detail.

Because the medium-specific information is contained wholly within the replaceable package, the working fluid specification can be changed at a single place at the highest level of the model with a consistent application of the change reflected throughout the model hierarchy. This "flip of a switch" flexibility is enhanced by the addition of the `choices` annotation in the Modelica language. The MediumModel concept is currently being used in the development version of the Modelica standard fluids library [5].

3.5 ModelData Structure

Populating hierarchical model structures with consistent data is a non-trivial task, especially considering the different data required for models of the same type but with varying levels of fidelity. To ensure a consistent application of data throughout the modeling structure, the MediumModel concept [4] has been adapted to organize data required for the engine models. A new ModelData package has been created to serve as the repository for the data required for the various models in the main library. Inside this package are sub-packages that correspond to the various subsystems in the vehicle (*e.g.* Engine, Transmission, *etc.*). Finally, packages exist that contain the particular data for a given entity (*i.e.* a vehicle, specific transmission, *etc.*). The various components that use the model data contain a `replaceable package` called EngineData from which specific elements are instantiated. Thus, a single redeclare of the EngineData package at the top-level of the model hierarchy populates the entire hierarchy with a consistent data set for simulation of a particular system. The `redeclare` is simplified by the support for the `choices` annotation in the Dymola[2] [6] GUI.

3.6 SignalBus Concept

The SignalBus concept [7, 8] is used to pass control signals throughout the model hierarchy. This concept uses the `inner` and `outer` semantics to propagate the control signals without requiring connections at every level in the model hierarchy. This technique facilitates the propagation of the control signals for replaceable components which typically require varying control signals for different levels of model fidelity. The SignalBus concept requires a top-level definition that represents the union of all the control signals and is coupled with selective definition and use of the control signals at the lower model levels. The interested reader is referred to [7, 8] for more discussion of the implementation of the SignalBus idiom.

4 Model Templates

While the standard interfaces discussed in the previous section provide a nice framework for a flexible, reusable modeling system, it is highly desirable to have more extensive models pre-built to establish a higher-level starting point for the model developer. This section provides some sample template and configuration models with a focus on the key Modelica language features that contribute to the flexibility. Additional details of the templates and configuration options are given in [8].

4.1 Cylinder Configurations

The majority of the work in engine modeling is focused on establishing the proper model for the cylinder. This process involves choosing the intake and exhaust system models (including the valve actuation mechanism), the combustion and heat transfer models, and populating the models with the appropriate data (*i.e.* bore, stroke, compression ratio, valve timings, *etc.*). To streamline the effort in assembling the cylinder design model, it is desirable to create a baseline cylinder model that can be used as the starting point for many different variants via the Modelica `replaceable` feature. Figure 6 shows the `MinimalCylinder` model that serves as a base model for various cylinder designs (note the components from the cylinder interface shown in Figure 4). An excerpt of the Modelica code is provided in Figure 7. Note the extensive use of `replaceable` types. Currently, the modifiers are applied to the instantiated components to ensure that the modifiers are picked up during a subsequent `redeclare`. In Modelica 2.1, the semantics of `redeclare` have been defined more explicitly to address the issue of modifiers with `replaceable` and `redeclare`. The combustion and heat transfer models are not included in `MinimalCylinder` and are left to be instantiated in an extending model. The `MinimalCylinder` template provides a flexible platform for creating cylinder models from different configurations and fidelity levels.

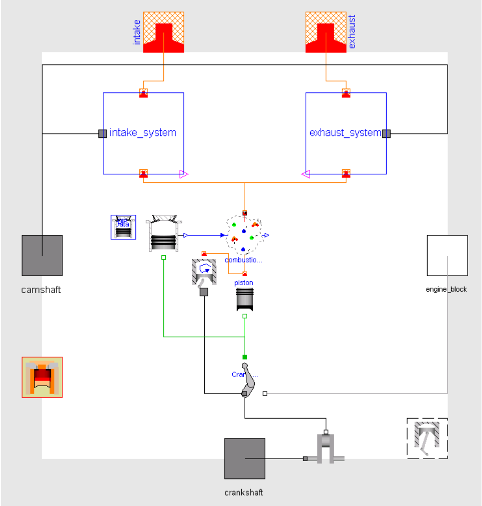


Figure 6. `MinimalCylinder` template model

```
partial model MinimalCylinder
  extends Ford.Engine.BaseClasses.Cylinder;
  replaceable model ControlVolume =
    Thermodynamics.VariableControlVolume;
  Control Volume combustion_chamber(modifiers);
  replaceable model Piston=Drivetrain.MasslessPiston
    extends Ford.Engine.Interfaces.Piston;
   Piston piston(modifiers);
  Mechanical.Crank crank(modifiers);
  InCylinder.ChamberVolume chamber_volume(modifiers);
  replaceable model IntakeSystem =
    Ford.Engine.Interfaces.IntakeExhaust;
  IntakeSystem intake_system(modifiers);
  replaceable model ExhaustSystem =
     Ford.Engine.Interfaces.IntakeExhaust;
  ExhaustSystem exhaust_system(modifiers);
…
end MinimalCylinder;
```

Figure 7. Code excerpt for `MinimalCylinder`

Figure 8 shows such an extension of the `MinimalCylinder` model with the intake and exhaust systems redeclared to be conventional, fixed valve timing models and the instantiation of

---

[2] Dymola is a trademark of Dynasim AB

Wiebe [1] combustion and Woschni-type [9] heat transfer models. Taking advantage of the `replaceable` components allows model variants to be quickly created with a minimum amount of model re-wiring, configuration, and code duplication. This sort of "plug and play" flexibility allows model assembly via simple `redeclare` statements for existing components. In terms of the valvetrain models, model variants exist to account for different valve actuation mechanisms, timing and phasing strategies, and configurations. The ideal piston could be replaced with a model that accounts for the effects of piston mass. Liberal use of the `replaceable` components is the key Modelica language feature for establishing these sorts of template models for "plug and play" configuration.

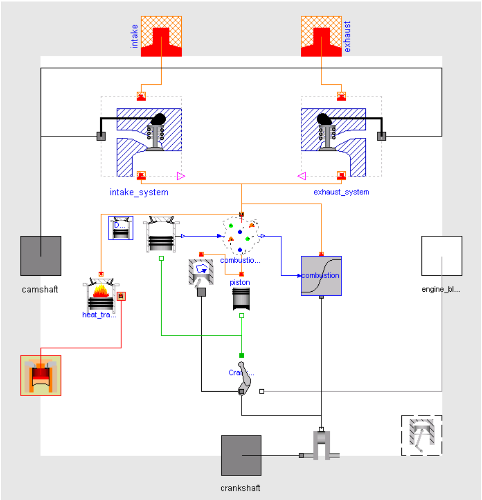
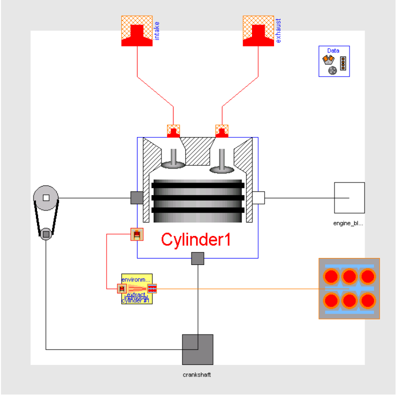


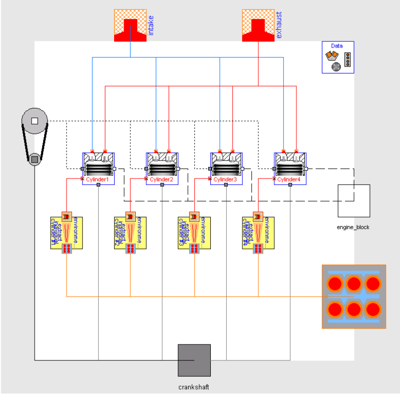
Figure 8. Fixed timing, Wiebe cylinder model

4.2 Engine Templates

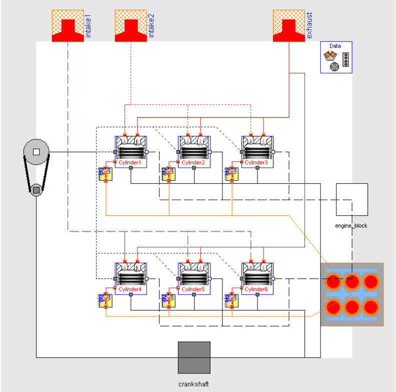
Having established a flexible framework for the cylinder design process, it naturally follows that templates should be established for the various engine configurations. Again, these templates help to minimize the modeling effort for assembling model variants, which at the engine level means building an engine model using a new cylinder design. Templates exist for various engine/plenum configurations (*i.e.* single cylinder, I4, V6, V8, *etc.*) as shown in Figure 9. Each template extends from the engine interface in Figure 5 and includes all of the connections between the cylinder(s) and the external interfaces. The key feature in each of the engine configurations is the `replaceable` `CylinderModel` shown in the code excerpt in Figure 10. This `CylinderModel` is then instantiated repeatedly for multi-cylinder engines. Therefore, creating a stand-alone engine model is simply a matter of extending from the appropriate engine template and redeclaring the `CylinderModel`. This single redeclare of the `CylinderModel` type is then used for the instantiation of each cylinder in the engine.



(a) Single cylinder engine



(b) I4 engine



(c) V6 engine with two intake plenums

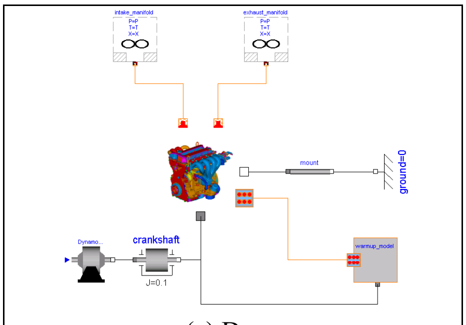
Figure 9. Engine configurations

```
...
  replaceable model CylinderModel =
      Interfaces.Cylinder;
  CylinderModel Cylinder1(shift=crank_shift[1],
redeclare package MediumModel
      = MediumModel)
...
```

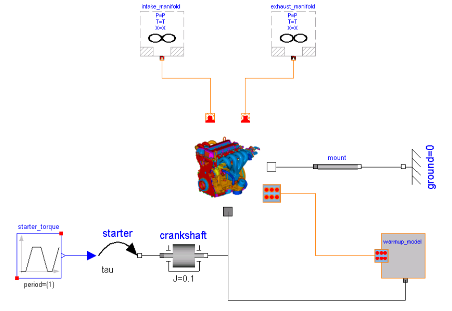
Figure 10. Code excerpt for engine templates

4.3 Experimental Templates

Extending the template abstraction even further, templates have been created for common types of simulation experiments. Figure 11 shows examples of an experimental setup for an engine on a dynamometer (a) and for a cranking engine (b). Code excerpts from the template base class are shown in Figure 12. These generic templates can be simulated for a particular engine configuration and cylinder design by simply extending from the appropriate template and adding a `redeclare` for `Configuration` and `CylinderModel`. This technique allows single templates to be used for every existing engine configuration and cylinder design that conforms to the interfaces in Figures 4-5.



(a) Dyno



(b) Cranking

Figure 11. Templates for dyno and cranking experiments

```
...
  replaceable model CylinderModel =
     Interfaces.Cylinder extends
     Ford.Engine.Interfaces.Cylinder;
  replaceable model Configuration =
     Interfaces.Engine;
  replaceable Configuration engine(modifiers);
...
```

Figure 12. Code excerpt from experimental template base class

5 Model Examples

This section presents some examples of engine system simulations. These examples illustrate the use of the experimental templates and also show how models of increasing complexity can be built using the modeling framework discussed previously. Each model was simulated using Dymola [6].

5.1 Engine Cranking

The key-on crank of the engine is a complex, dynamic process involving the electrical system and controls, along with the actual engine itself. Controlling and optimizing the engine cranking behavior is crucial from the standpoint of both emissions and customer feel. This section shows some results from a detailed, multi-domain model of a cranking engine.

The crank model shown in Figure 13 is built upon the cranking template in Figure 11b. The `Configuration` has been defined as a single-cylinder engine with a `CylinderModel` that includes detailed, multi-zone, predictive combustion [4]. The intake reservoir has been replaced by a dynamic model of the manifold and throttle. The engine warmup model is a simple, fixed temperatures model. The control and electrical systems have been simplified such that the starter applies the commanded torque for 0.5s at 0.25s. The treatment of the engine friction is simplified in this model to a constant opposing torque starting at 0.5s. In this simulation, the throttle is closed to represent idle conditions. During the cranking process, the liquid fuel dynamics are extremely important since mixture preparation is inhibited at low speeds, high manifold pressures, and under cold conditions. While these effects can be considered within this modeling framework [3, 10], they are not included in these simulations.

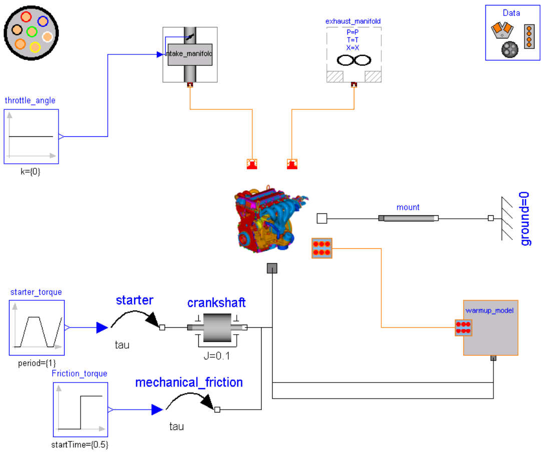


Figure 13. Model for cranking engine simulation

This single-cylinder cranking simulation with fixed metal temperatures has 207 components, 1033 time-varying variables, 1120 non-trivial equations, and 53 states. Figures 14-15 show the response of the engine speed and manifold pressure during the first 3s of the cranking simulation. The starter begins to spin the engine up at 0.25s. The manifold starts at approximately ambient pressure and then begins to pump down due to the emptying and filling process between the upstream intake reservoir (the ambient) and the engine. Note the "gulping" from the manifold due to the single-cylinder engine. A multi-cylinder engine results in the smoothing of the pumping down of the manifold due to the more frequent breathing from the multiple cylinders. The engine speed increases rapidly during the first few firing events since the manifold pressure is still high, resulting in a large amount of combustible mass in the cylinder. The engine speed starts to drop as the manifold pumps down and starts approaching a steady idle speed of 1700 RPM.

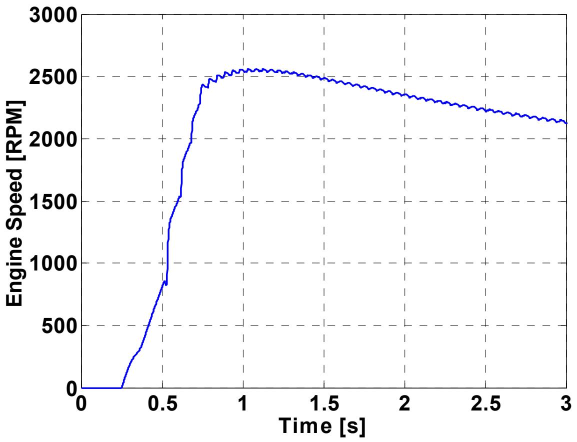


Figure 14. Engine speed response

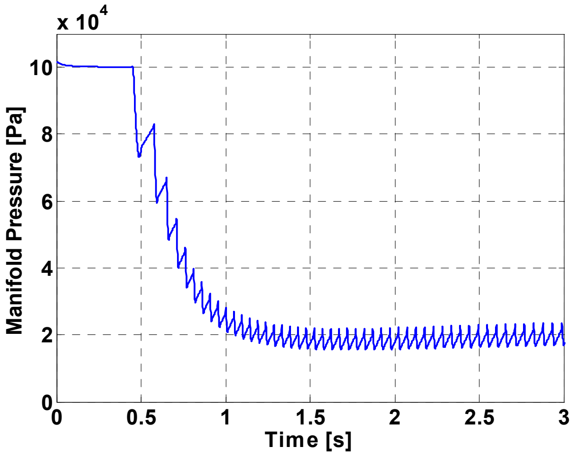


Figure 15. Manifold pressure response

5.2 Exhaust System Warmup

Vehicle thermal management is a critical issue in light of the recent legislation mandating lower emissions levels. The optimization of the engine system, from start-up strategy to component design of the intake, cylinder, and exhaust systems, is a key enabler to meeting more stringent emissions standards by reducing engine-out emissions and light-off time for the three-way catalyst. This section shows an engine system, cold start simulation from crank for evaluation of the thermal response of the exhaust system.

The model used in this simulation extends from the cranking engine model discussed previously (Figure 13). This version replaces the fixed temperatures model for the engine with the dynamic thermal response model shown in Figure 16. This model is extended from the work in [11] and includes models for the warmup of the piston, head, block, and valves along with a simplified representation of the oil and coolant loops. This simulation also includes a model, shown in Figure 17, of the exhaust system, including the exhaust manifold and downpipe leading to the catalyst. This model is based on [12] and includes distributed models for the thermal interaction between the exhaust gas and the pipe wall. The effects of forced convection between the gas and the wall, conduction along the pipe wall, and natural convection between the pipe outer wall and the ambient are included.

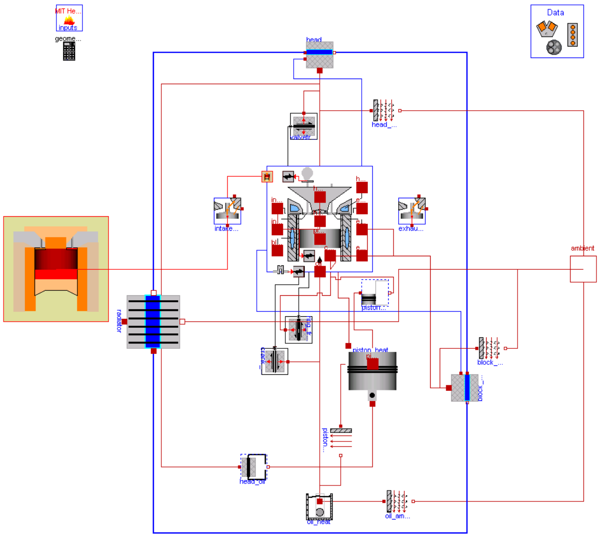


Figure 16. Engine thermal response model

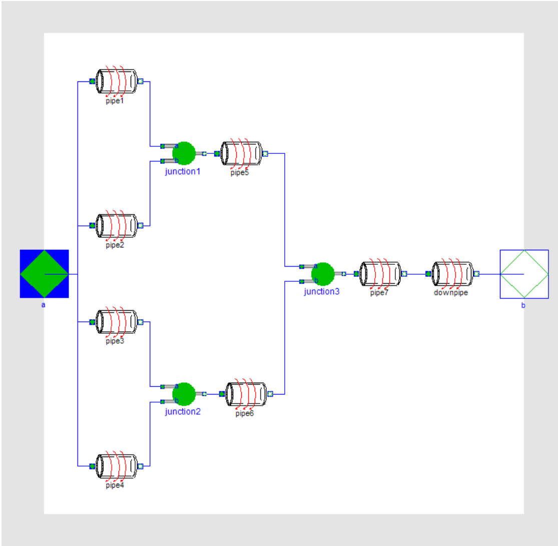


Figure 17. Exhaust system model

The cold start, cranking model with variable metal temperatures and an exhaust system has 924 components, 5476 time-varying variables, 5825 non-trivial equations, and 221 states. The increase in the number of equations and states from the cranking simulation discussed in Section 5.1 results mainly from the inclusion and discretization of the pipes in the exhaust system. Each of the 8 pipes in Figure 17 was divided into 10 elements along its length, and each element has 2 states (one each for the temperature of the exhaust gas and the temperature of the pipe wall in the element).

To simulate the start of the FTP drive cycle test for emissions, the model was run for approximately 20 seconds. This test begins with a cold crank and idle until approximately 20 seconds when the first acceleration occurs. Figure 18 shows the thermal response of some of the components in the engine thermal model (Figure 16). Note that the components that receive heat directly from the gas in the cylinder (*i.e.* piston, head, liner) start to warm first. The piston has a lower thermal capacitance than does the liner and the head so it warms more quickly. The temperature rise from ambient is fairly modest due to the large thermal capacitance of the engine and the short simulation time (typical engine warm-up occurs over several minutes).

The temperature of the exhaust gas as it traverses the exhaust system is crucial as the thermal energy in the gas is responsible for warming the three-way catalyst to the elevated temperatures at which it becomes effective. Figure 19 shows the transient temperature of the exhaust gas as various points in the system. The highest temperatures are at the entrance to the exhaust port (just past the exhaust valve) with temperatures decreasing along the system due to heat loss to the cold pipe walls. The highest exhaust gas temperature occurs roughly at the maximum speed (see Figure 14) where maximum amount of combustible mass is trapped in the cylinder due to the high manifold pressure. Note the large drops in temperature throughout the system. Minimizing the amount of energy lost in the exhaust manifold and piping leading to the catalyst during a cold start is crucial for minimizing catalyst light-off times. This sort of engine system model can be used to effectively and efficiently evaluate different engine startup strategies and hardware designs and their effects on exhaust system thermal response.

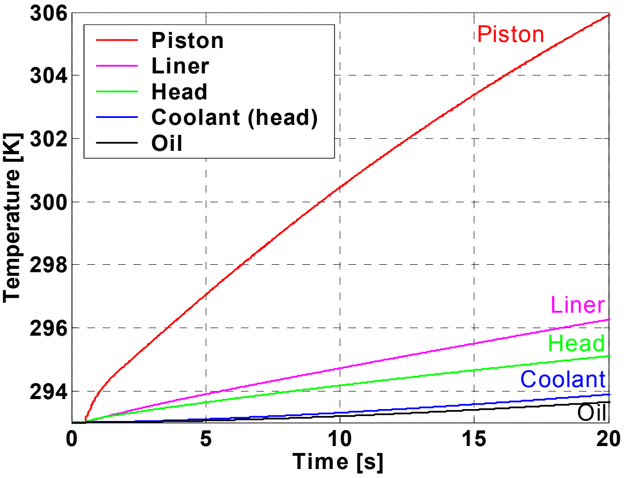


Figure 18. Thermal response of engine components

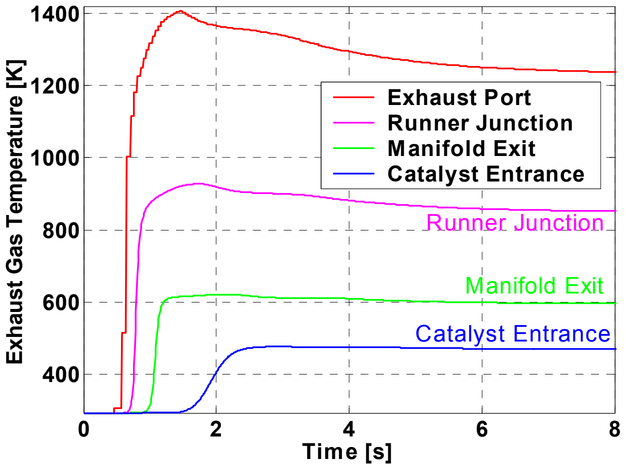


Figure 19. Thermal response of exhaust gas

6 Conclusions

This paper describes the use of the Modelica modeling language for engine system simulations. A robust, flexible, and re-usable modeling framework of connectors, interfaces and templates is described for multi-domain engine system modeling. Results from the detailed simulations of the engine cranking process yield some insight into the types of models that can be realized using this framework and the vast amount of information that can be obtained from these types of simulations. These multi-domain models are well suited for the evaluation and optimization of hardware design and control strategies, especially during the early concept assessment stage of the design process. Future work will focus on the validation of the individual submodels and system-level models.

References

1. Heywood, J.B., 1988, *Internal Combustion Engine Fundamentals*. McGraw-Hill.
2. Modelica Association, 2002, "Modelica Language Specifications (Version 2.0)", http://www.modelica.org
3. Puchalsky, C., *et al.*, 2002, "Modelica Applications for Camless Engine Valvetrain Development", *2nd International Modelica Conference Proceedings*, pp. 77-86, http://modelica.org/Conference2002/papers/p11_Puchalsky.pdf
4. Newman, C., Batteh, J., and Tiller, M., 2002, "Spark-Ignited-Engine Cycle Simulation in Modelica", *2nd International Modelica Conference Proceedings*, pp. 133-142, http://modelica.org/Conference2002/papers/p17_Newman.pdf
5. Elmqvist, H., Tummescheit, H., and Otter, M., 2003, "Principles of Object-oriented Modeling of Thermo-Fluid Systems", *3rd International Modelica Conference Proceedings*.
6. Dymola. Dynasim AB, Lund, Sweden, http://www.dynasim.se
7. Tiller, M., Tobler, W.E., and Kuang, M., 2002, "Evaluating Engine Contributions to HEV Driveline Vibrations", *2nd International Modelica Conference Proceedings*, pp. 19-24, http://modelica.org/Conference2002/papers/p03_Tiller.pdf
8. Bowles, P. and Batteh, J., 2003, "A Transient, Multi-Cylinder Engine Model Using Modelica", *SAE-2003-01-3127*, Society of Automotive Engineers.
9. Woschni, G., 1967, "A Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine", *SAE-67-0931*, Society of Automotive Engineers.
10. Batteh, J.J. and Curtis, E.W., 2003, "Modeling Transient Fuel Effects with Variable Cam Timing", *SAE-2003-01-3126*, Society of Automotive Engineers.
11. Kaplan, J.A. and Heywood, J.B., 1991, "Modeling the Spark Ignition Engine Warm-Up Process to Predict Component Temperatures and Hydrocarbon Emissions", *SAE-91-0302*, Society of Automotive Engineers.
12. Laing, P.M., Shane, M.D., Son, S., Adamczyk, A.A., and Li, P., 1999, "A Simplified Approach to Modeling Exhaust System Emissions: SIMTWC", *SAE-199901-3476*, Society of Automotive Engineers.