

Integration of CATIA with Modelica

Partha Bhattacharya Nick-Ange Suyam Welakwe* Ranga Makanaboyina
Adithya Chimalakonda

DaimlerChrysler Research and Technology India

*DaimlerChrysler Research and Technology Germany

*Partha.Bhattacharya@DaimlerChrysler.com Nick-Ange.Suyam_Welakwe@DaimlerChrysler.com
Ranga.Makanaboyina@DaimlerChrysler.com Aditya.ch@DaimlerChrysler.com*

Abstract

Virtual Prototyping is becoming increasingly popular in the automotive and other industries, especially in the context of concurrent engineering processes. CATIA [1] is a powerful CAD tool for modeling and Modelica based tools are increasingly becoming popular for analysis using multibody simulation. In this work, we developed an interface that integrates Modelica into CATIA to facilitate product data information exchange. This enables automatic extraction of pertinent information from CAD representation of the physical components to Modelica models for multibody simulation. The integration will reduce the design cycle significantly by providing immediate feedback to the designer with minimal intervention of simulation and modeling specialists. We studied three different implementation approaches for integration. The results are presented for a mechanism that opens the car-hood.

Keywords: Multibody Simulation, Concurrent Engineering, Virtual Prototyping, Interface Tools, Joints, CATIA, Integration, Modelica.

1 Introduction

Two important phases in Virtual Prototyping, an integrated part of the modern product development process, are design and analysis. While design deals with form, analysis deals with behavior of the system components. Form can be represented by CAD models (Design) and behavioral models, at different accuracy levels, can be simulated (Analysis) to compare with the product function specifications. Multibody Simulation or motion analysis is an important part of the analysis phase. Generally, design and analysis, are done independently. Designer and Dynamic simulation expert are different persons. But Simulation experts and designers need to exchange data in order to validate their models. This data exchange is made most of the time manually. There is no guarantee that the exchanged data are right and up

to date. So tools are needed to exchange information between these two processes so that designers can take a holistic design approach [2] [3].

The main benefit of the interface we developed is to provide a user friendly framework to support the concurrent engineering process between dynamic simulation experts and designers taking some specific organizational process constraints into account. With these constraints in place, the interface contributes significantly to the improvement of the data exchange process. Here the goal is not to provide an integrated multi body simulation environment in CATIA but to keep CATIA and Modelica separate and just enable the data exchange between CATIA models and corresponding Modelica models.

In fact, most of the commercial MBS softwares provide interface to standard CAD softwares. For example SIMPACK [4] has an interface, CATSIM, to CATIA. SimMechanics [5] provides translator from SolidWorks models, SimDesigner [6] interface enable multi body simulation within CATIA V5.

Modelica is becoming increasingly popular in the automotive industry, at least because of object oriented multidomain modeling and simulation capabilities. A typical application area of Modelica is in modeling of mechatronic systems. In these systems, multi body simulation plays a significant role. The objective of making an interface between CATIA and Modelica is to provide a process and user friendly framework for the exchange of kinematics data.

People have already integrated CAD data into Modelica simulation [7]. MathModelica, one of the popular Modelica tool, offers interface to SolidWorks [8]. Our work is different from the past works in that we are not proposing to automatically translate a CAD model directly which might not be very suitable for MBS, though its one of the options discussed in here (Option I), but we are proposing a semi-automatic data extraction appropriate for MBS.

2 CATIA Data required for MBS simulation in Modelica

A MBS model in Modelica mostly consists of rigid bodies connected by different kind of joints. These bodies interact via Joints and respond to different forces i.e. gravity. Joints define the degrees of freedom between bodies. To define Bodies, three sets of information are required. They are,

- A. **Physical Information:** Mass, centre of mass and inertia tensor of the bodies. CATIA models of the corresponding bodies store mass and inertia of the assembly or individual parts which can be extracted automatically.
- B. **Geometrical Information:** Modelica uses ‘Mechanical Pin’ or ‘frame_a/b’ which is nothing but the ‘connecting’ point of the body. Getting this information automatically is challenging. Fixing a point to define frame_a/b in a 3-D body is difficult. A CAD designer can define points, frame_a/b on CATIA parts which can be chosen interactively. The center of gravity position vector is also required and can be extracted automatically from the CAD model.
- C. **Visualization information:** This data is required for animation. We use BodyShape to represent a body in Modelica. This body can be represented by different available simple shapes or it can refer to CAD data for visual representation. This data can either be in DXF format or STL format. Here we convert CATIA models (CATProduct, CATPart, PartBody) to STL files and use the same in Modelica.

Apart from the bodies, Joints can be defined through a set of points/lines/planes depending on the nature of joints which will be defined by the CAD designers in CATIA components. These points/lines/planes can be chosen by the user and extracted to Modelica.

Component characteristics can be optionally required to simulate some special physical effects. But these characteristics are not usually stored in the CAD model and therefore don't need to be extracted.

3 Implementation Approaches

We have three possible scenarios for integration.

- I. **Translation of CATIA V5 defined Multi-Body structure to Modelica model.** CATIA offers a workbench called DMU Kinematic Simulator in which a mechanism can be defined with different kind of joints. A MBS structure thus constructed

can be directly translated to Modelica. Because, for example, a revolute joint in CATIA can be replaced with a revolute joint in Modelica. This seems to be an easy and obvious choice. But DMU Kinematic Simulator is suitable for kinematic analysis only. In practice, using the same kinematic topology for dynamic analysis is not always possible.

- II. **Development of an add-on to the CATIA interface to define the MBS structure.** To overcome the above drawbacks and to replace the function of DMU Kinematic Simulator, we can develop an add-on to the CATIA interface to define the MBS structure where a modeler can define a MBS structure using connection diagram in a visual editor (like Dymola [9]) which supports drag and drop. This is done in Dymola model editor. Another example of this kind of editor is DynaFlexPro from Maple [10]. This is going to be heavy on the CAD designer who might not have the MBS fundamentals. The best thing will be to see the mechanism already defined and just relate it to the CATIA model which leads us to the next option.
- III. **Existing Modelica model integrated to its CATIA counterpart.** The MBS modeler makes a Modelica MBS skeleton/structure. He also sits with the CATIA designer to define a configuration / mapping file. This mapping file related BodyShapes and bodies to CATParts or CATProducts. It also has information about the joints in terms of lines, planes and points. If additional points are required CATIA designer defines them. With the mapping file and the CATIA model, automatic extraction of appropriate data takes place. The Modelica skeleton gets filled with information. Here for the first time we are talking about starting the flow from Modelica. This definitely lacks the flexibility, user interaction on the CATIA side and most importantly visibility.

4 Implementation Concept

Taking into consideration all the advantages and the drawbacks of the above approaches, we have converged to the final option which we demonstrate with a scenario here.

1. To simulate a mechanism, a kinematic skeleton is constructed with Modelica. The geometric models of the components in this skeleton exist in CATIA.

2. The CAD model of the components is opened in CATIA. A toolbar is created with CATIA Product Design workbench. Using one of the buttons in this the user imports the corresponding Modelica file to CATIA workspace.
3. Using another button the user invokes a graphic user interface (GUI) where all the details of the Modelica components (BodyShape and Joints) are shown.
4. User selects the appropriate parts from CATIA interface to relate it to the proper objects in the diagram.
5. All the data are then extracted and assigned to proper objects. The extracted data is exported back to Modelica script.

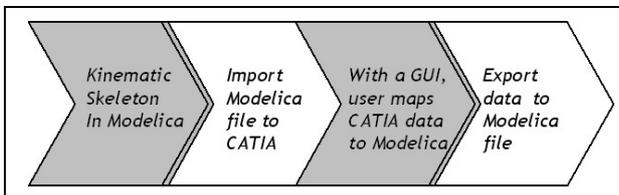


Figure 1: Process flow for the Implementation

Figure 1 gives an overview about the Steps described above.

5 Implementation Results

CATIA allows a set of C++ Application Programming Interfaces (API) as part of Component Application Architecture (CAA) to access functionalities of CATIA models. We have chosen CAA as our development platform. To implement the solution we need core CATIA functionalities with graphical user interface programming. CAA has ability to program graphic interfaces but it's limited. So we used CAA for accessing CATIA's core functionalities. The implementation consists of the following steps:

5.1 Design the kinematic skeleton

The product is decomposed into body shapes P1, P2, P3 and P4, and prismatic joint PJ1, revolute joints RJ1, RJ2 and RJ3 as shown in figure 2.

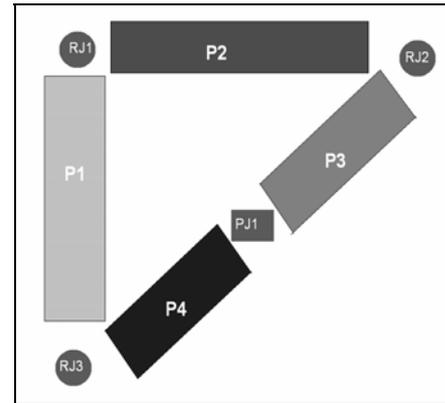


Figure 2: Kinematic skeleton

5.2 Convert the skeleton to Modelica model

Figure 3 shows the corresponding model of the kinematic skeleton in the Dymola editor.

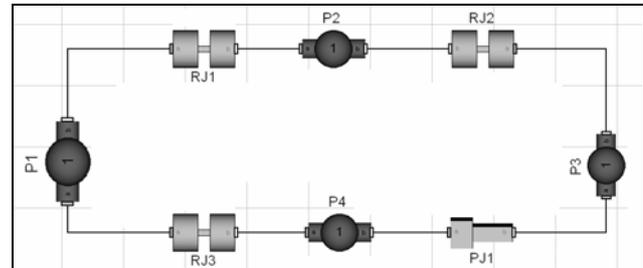


Figure 3: Corresponding Modelica model

5.3 Import Modelica model to CATIA workspace

Our algorithm scans the Modelica script and picks up the relevant components like BodyShape, different Joints etc. and lists them.

5.4 Extract the CATIA data

The next step involves opening the corresponding CATIA model and starting data extraction (figure 4).

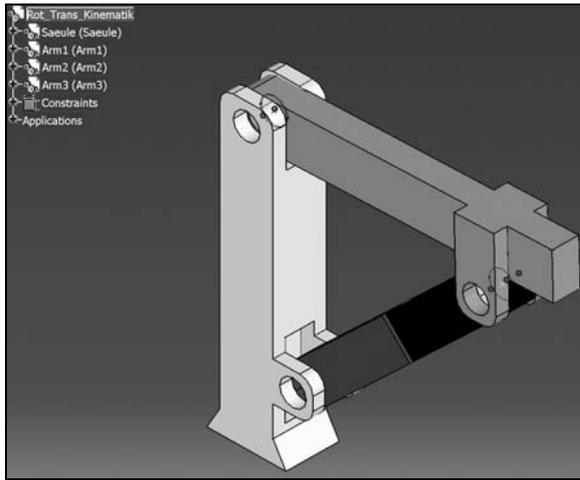


Figure 4: Related CATIA Model

5.4.1 For BodyShapes:

When any BodyShape component is selected in the GUI user can specify a CATPart or a CATProduct to relate it to the BodyShape component. For any BodyShape component we need mass, inertia tensor and centre of mass information. This information is already stored in CATIA. We need to specify the shape information in terms of frame a/b. User can choose points on the component to specify this information (figure 5). Visualization information is exported by converting the CATPart/CATProduct into an STL file.

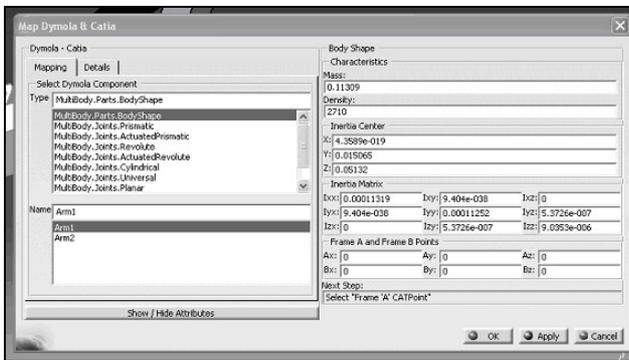


Figure 5: Interface for BodyShape

5.4.2 For Joints:

When a joint is selected in the GUI, details of its connecting frames are shown. User can now choose the required data by choosing the corresponding geometric element in CATIA as shown in figure 6. For example, when a revolute joint is selected in the GUI, details of its connecting frames are shown (figure 6). User can now choose the axis of revolution by choosing a line on the components.

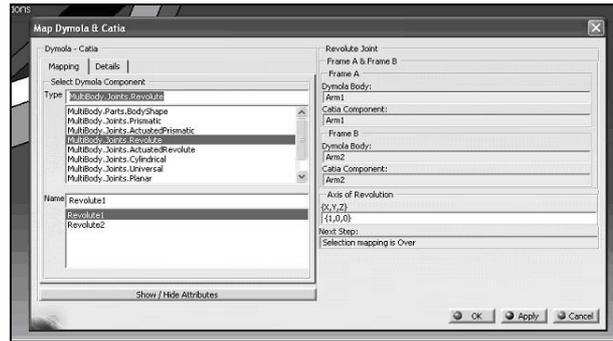


Figure 6: Interface for Revolute Joint

5.5 Update Modelica model

Figure 7 is the Modelica script before the data export. Figure 8 is the Modelica script after the extracted data is appended to the Modelica script for the corresponding components. It may be noted that the mass, centre of mass and inertia tensor values, and geometrical shape values are updated in the Modelica script.

```
World world #;
.Revolute Revolute1 #;
.ActuatedRevolute ActuatedRevolute1 #;
BodyShape Saeule #;
FixedTranslation FixedTranslation_Saeule #;
BodyShape Part2 #;
BodyShape Part3 #;
BodyShape Part4 #;
.Prismatic Prismatic1 #;
.Revolute Revolute2 #;
timeTable(table=[0,0; 1,Modelica.Constants...
```

Figure 7: Modelica Script of kinematic skeleton

```
Modelica.Mechanics.MultiBody.Parts.BodyShape Part2(m=0.043511,
r={0.000000,0.000000,0.000000},
r_CM={0.000000,0.146040,0.021204},
shapeType="1",
I_11=0.000012,
I_22=0.000002,
I_33=0.000011,
I_21=0.000000,
I_31=0.000000,
I_32=0.000000);
Modelica.Mechanics.MultiBody.Parts.BodyShape Part3(m=0.074444,
r={0.000000,0.000000,0.000000},
r_CM={0.000000,0.174394,0.062528},
shapeType="2",
I_11=0.000042,
I_22=0.000002,
I_33=0.000042,
I_21=0.000000,
I_31=0.000000,
I_32=0.000000);
```

Figure 8: Updated Modelica Script after Data Export

6 Conclusion

We have presented an integrated interface between CATIA V5 and Modelica. The interface supports the exchange of kinematics data between CATIA model and Modelica model. It provides further a framework to support the concurrent engineering process between dynamic simulation experts and designers taking following organizational and OEM process constraints into account:

- Designer and dynamic simulation expert are different roles
- Designer only works with CATIA and the simulation expert with Dymola
- Design process is the master.

The use of the interface requires minimal process changes from the user for data exchange and thus guarantees high process flexibility.

References

- [1] www.3ds.com
- [2] R. Sinha, C.Paredis and P.Khosla, “Integration of mechanical cad and behavioral modeling”, Proceedings of the IEEE/ACM Workshop on Behavioral Modeling and Simulation, 2000.
- [3] V. Engelson, “Tools for design, interactive simulation and visualization for dynamic analysis”, Linkoping Electronic Articles in Computer and Information Science, 5 (2000).
- [4] <http://www.simpack.com>
- [5] <http://www.mathworks.com>.
- [6] <http://www.mscsoftware.com>
- [7] P. Bunus, V. Engelson, and P. Fritzon, “Mechanical models translation, simulation and visualization in Modelica”, in Modelica Workshop 2000 Proceedings, pp. 199–208.
- [8] <http://www.mathcore.com>.
- [9] <http://www.Dynasim.se>
- [10] <http://www.maplesoft.com/>