

Exploring the Possibilities of Co-Simulation with CATIA V6 Dynamic Behavior Modeling

Peter Smolek¹, Bernhard Heinzl², Horst Ecker¹, Felix Breiteneker³

¹Vienna University of Technology, Institute of Mechanics and Mechatronics

²dwh simulation services, Vienna, Austria

³Vienna University of Technology, Institute for Analysis and Scientific Computing

peter.smolek@tuwien.ac.at

Abstract

This study shows how to utilize the CATIA V6 Dynamic Behavior Modeling (DBM) software environment for the purpose of co-simulation of physical models. The implementation of a co-simulation using Building Controls Virtual Test Bed (BCVTB) is demonstrated and the pros and cons are discussed. Further the methods for FMI export and import in CATIA are explored with respect to implementing a co-simulation either in CATIA itself or other host programs. The two approaches are displayed by implementing simple examples. Ultimately possible applications for an advanced tool to link 3D geometric data and systems simulation, with the potential to perform co-simulation, are presented.

1 Introduction

The approach of using system simulations results in new opportunities for integrated virtual product development. Putting the product itself in the center of focus and building the different tools around it allows connecting the different disciplines of development, such as Computer Aided Design (CAD), Computer Aided Engineering (CAE) and system simulation. Co-simulation describes the process of using multiple programs working on the same simulation, sharing data with each other. This grants the flexibility to use specialized software to describe specific parts of the model, for example analytical, data-based, static or dynamic components. This enables to examine complex problems like multidomain simulations, stiff systems and many more. In addition it allows the developers to use the software tool best suited for their field of study, collaborating simultaneously on the same model. Thereby using different

model descriptions and multiple numerical algorithms can be used to study heterogeneous systems.

With the CATIA V6 release, an environment for systems simulation has been implemented – the Systems Dynamic Behavior Modeling (DBM) [1] – which in itself consists of an altered Dymola build. Therefore the CATIA DBM also uses the Modelica standard [2], and the subsequent libraries. The layout has been adjusted to fit the general CATIA logic, the functionality of the Dymola software remains for the most part unaltered. The main difference with respect to Dymola structure is that the DBM is integrated in the CATIA Product Lifecycle Management (PLM) solution, therefore the data is no longer stored locally but on a server. On this remote computer the files of multiple users are stored and the license management is performed. As a result, libraries need to be imported and propagated to the server before use. Modelica files can be imported as well as exported, however the common approach is to work with the DBM projects through the server.

As part of the CATIA PLM, combining physical data with the functionality of the model is realized through the Requirements, Functional, Logical and Physical (RFLP) environment, which can be accessed through the Functional and Logical Workbench seen in figure 1. It is possible to define logical references, which interact through connections and can be arranged hierarchically. The references in turn can be given a DBM model. At the top of figure 1 an example of a logical hierarchy is shown. The small icon in the lower right corner of the logical references indicates that a DBM model is implemented. Further, a geometric representation can be linked to the logical reference to represent the physical properties of the logical

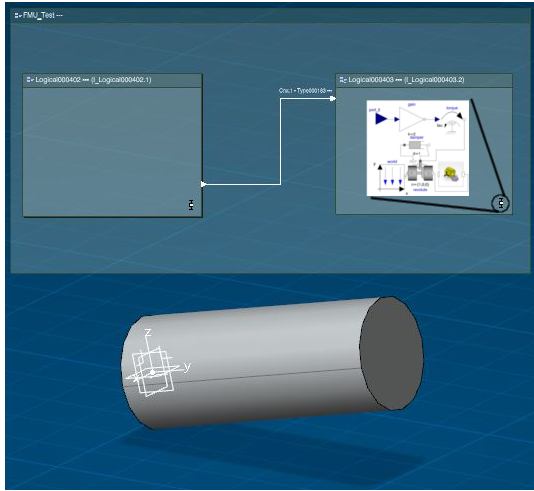


Figure 1: CATIA Functional and Logical Workbench with logical references containing a DBM model and a 3DRepresentation.

component, as seen at the bottom of figure 1. As it stands, it is not possible to use general 3DParts in the RFLP environment but special 3DRepresentations. This however is announced to be resolved in future releases. The whole simulation can then be run from the Functional and Logical Workbench where all the underlying DBM models are executed and data is exchanged according to the logical connections.

2 Co-simulation with BCVTB

To prepare CATIA DBM for co-simulation, a software solution that is compatible with Dymola is chosen. The Building Controls Virtual Test Bed (BCVTB) [3] delivers that in combination with an interface provided in the Modelica Buildings Library [4]. The BCVTB was originally used for buildings simulation, but the ability to link to the general purpose tools Dymola and MATLAB enables the use in other domains. Since this is a Modelica library, importing it into the DBM environment is easily achieved. Models can be built very similarly as in Dymola, since the functionality of the two is basically the same. BCVTB controls the communication between the backbone and CATIA and relies on BSD sockets [5] for runtime data exchange. To minimize startup time, an existing CATIA instance is accessed. The running instance can be contacted through a COM server, which CATIA already runs natively. This can be achieved via a Visual Basic script since CATIA handles macros internally with the same lan-

guage. Therefore further control over the CATIA instance is possible, especially remote execution of simulation runs is practical.

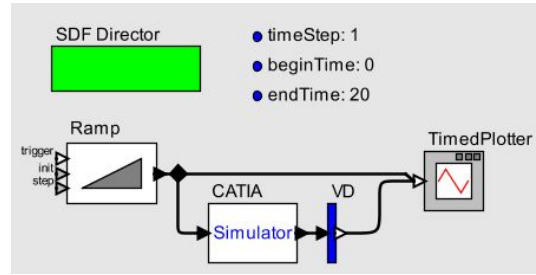


Figure 2: Model in BCVTB; the CATIA simulator block is responsible for the communication with CATIA.

To demonstrate the basic data exchange, a simple model was implemented consisting of a ramp signal, defined in the BCVTB environment. The data signal is then sent to the simulator actor block, which implements the BSD socket communication with the CATIA DBM environment, as seen in figure 2. The DBM model simply consists of a gain element, which gets the data value from BCVTB, multiplies it by two and returns the result to the backbone. The model is shown in figure 3, figure 4 is demonstrating an example simulation run and depicts the input and output signal (timed plotter in figure 2). The red graph represents the ramp output, the blue is the returned data of CATIA, which is zero per default during the initial data exchange. This simple example is solely sup-

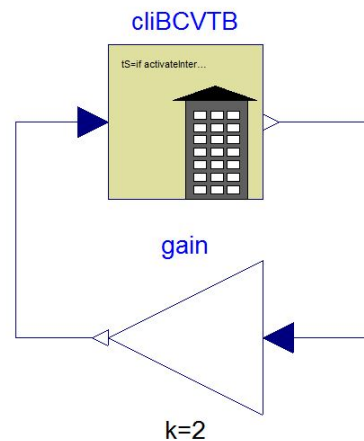


Figure 3: Modelica model implemented in CATIA DBM with the BCVTB Modelica block; the data is received from BCVTB, amplified and returned to BCVTB.

posed to demonstrate the communication between BCVTB and CATIA, no differential equations are solved and the connection to a second simulation software is missing.

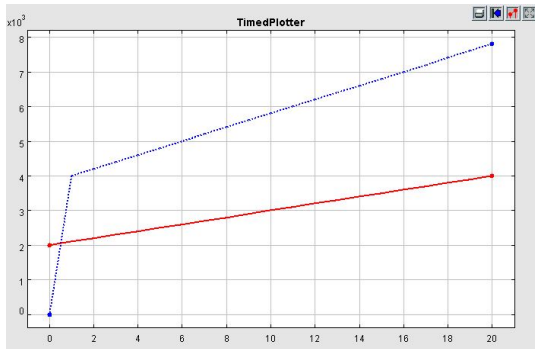


Figure 4: Output of the timed plotter of BCVTB; red: the ramp, blue and dotted: simulator output.

With this method of employing co-simulation, coupling models implemented in different simulation tools with CATIA is achieved. In order to take full advantage of what system simulation in combination with a CAD software offers, it is necessary to couple BCVTB not with the DBM, but with the Functional and Logical Workbench. Unfortunately, with the current release of CATIA V6 that could not be achieved. The RFLP environment cannot execute a simulation successfully with the BCVTB block as a part of it, even though the underlying DBM model in itself can be run. The problem seems to be, that the linked C-functions are not called properly. A solution could not yet be found since CATIA does not allow insight into the internal processes. It is to be expected, that this issue will be resolved in the future, enabling real advantages for co-simulation with CATIA.

3 Co-simulation with FMI

Another interesting possibility to realize co-simulation with CATIA V6 is by employing the Functional Mockup Interface (FMI) [6]. This standardized interface allows exporting models as Functional Mockup Units (FMU), which in turn can be imported into other simulation tools. It is possible to either export the model alone – the solver of the host then calculates the imported model – or the solver algorithm can be exported with the model, therefore allowing a co-simulation with a modelica host, which is further discussed.

CATIA supports the export and import of FMUs

in the DBM as well as the RFLP environment. For this study, the FMUs to be imported are created with Dymola 2014. First of, the DBM Workbench offers the same functionality as does Dymola in its current release. It is possible to import FMUs, link them via appropriate interfaces to the DBM model and run the simulation. The export of a model, as well, works with exceptions unproblematic. The more interesting subject is the FMU export and import of the Functional and Logical Workbench. As it stands, there is no method for exporting models as FMUs. The option that is available is only for exporting already imported FMUs again and therefore is not suitable for using RFLP content externally. The import however is implemented as an extra method of defining DBM models for logical references. This is a convenient way of linking FMUs to logical references, which can be connected to the rest of the logical hierarchy for data exchange. The second possibility of importing a FMU is to import it into a DBM model. This method is more robust, even though some issues regarding updating an imported FMU do exist. Both methods grant access to co-simulation in the RFLP environment.

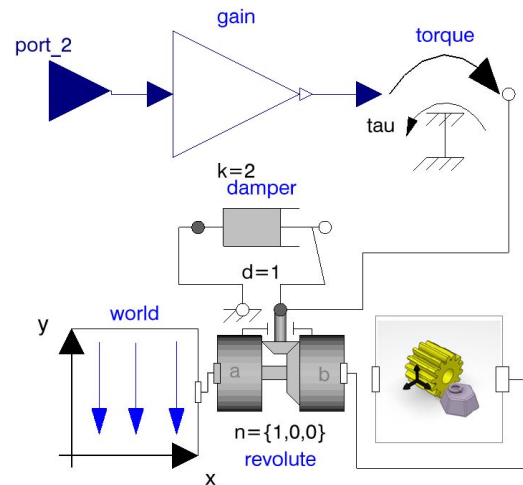


Figure 5: DBM model of a revolvable cylinder; the data from the FMU is received through the input signal port, driving the rotation.

For demonstration purposes, a simple setting was chosen. In the RFLP environment, two logical references have been defined (figure 1). One holds a normal DBM model, containing solely the imported FMU. In the second reference a model of a damped rotation of a cylinder is implemented, see figure 5. A torque drives the rotation and receives

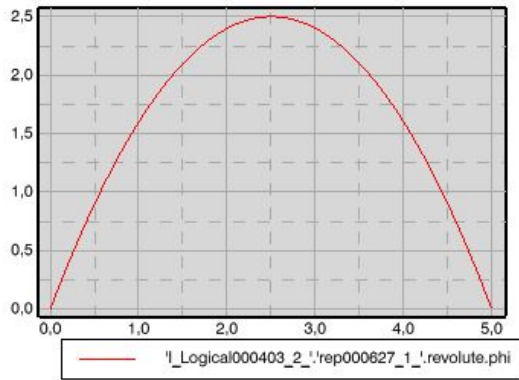


Figure 6: Plotting window of CATIA in the Functional and Logical Workbench; red: the angle of the cylinder over time.

its amplitude from the FMU. Using the appropriate block in the DBM model and linking it with a 3D representation realizes the mechanical arm. Figure 1 shows the logical hierarchy and the cylinder in motion. Figure 6 is an example of a plotter window showing the angle of the mechanical arm plotted over time.

With the presented methods, setting up a co-simulation using FMUs in CATIA is possible. However, the potential is constricted by the existence of a way to export a model via FMI. It is to hope, that with further development of the FMI standard more simulators will add support for exporting FMUs, thus allowing a flexible choice of software for co-simulation.

4 Outlook

As the situation currently presents itself, the unique opportunities of combining CATIA V6 with a system simulator cannot be efficiently utilized for co-simulation yet. Once the software is further developed, the usability will be enhanced and integration of the physical data will enable efficient knowledge management across various domains. Furthermore, whole new options arise from the possibility of not only being able to drain information of CAD data but to alter the geometry data itself. It is conceivable to build a model of a mechanical systems simulation inside an optimization loop. The optimization, for example a genetic algorithm, can then adapt the geometry to optimize e.g. the distribution of cross sections of a shaft.

One can imagine the possibility to automatically extrapolate a mechanical model from CATIA's As-

sembly Design, since the program already has the information of the relationship between different components. Therefore realizing dynamic analysis of complex components in a short timeframe would become realistic. Testing a product virtually before prototypes are built has the benefit, that complex interactions of parts are incorporated in the development. This results in reduction of development time and cost of mechanical parts.

References

- [1] Dassault Systemes: CATIA V6R2013X Portfolio, 2013, <http://www.3ds.com/products-services/catia/portfolio/catia-version-6/v6-portfolio/>
- [2] Modelica Association: Modelica Specification, version 3.3, 2012, [modelica.org](http://www.modelica.org)
- [3] M. Wetter: Building Controls Virtual Test Bed User Manual Version 1.3.0, 2013, simulationresearch.lbl.gov/bcvtb
- [4] M. Wetter et al.: Modelica Buildings Library, 2012, *Journal of Building Performance Simulation*, (pre-print) 2013
- [5] M. Wetter: Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed, 2011, *Journal of Building Performance Simulation*, 4(3):185-203, 2011.
- [6] Modelica Association: Functional Mock-up Interface for Model Exchange and Co-Simulation, 2.0 Beta 4, 2012, www.fmi-standard.org