

Middleware based Hardware in the loop Simulation for the Steel Plant Industry

Prof. Dr. Ing. C. Bruce-Boye
FH-Lübeck University of Applied Science
Mönkhofer Weg 239
23562 Lübeck, Germany
cecil@bruce-boye.com

Rüdiger zum Beck
cbb software GmbH
Isaac-Newton-Straße 8
23562 Lübeck, Germany
ruediger.zum_beck@cbb.de

Abstract

The paper presents a concept and an implementation of Modelica simulation interaction using the windows based middleware application LabMap[®].

The middleware LabMap[®] is used as a transport layer for distribution and synchronization of different data sink and sources and a Dymola simulation model which is running on a Microsoft Windows Operating System.

A simulation setup is presented using an intelligent energy and media management model for an integrated steel mill. The feasibility and advantages of the utilization of a middleware are shown, through the synchronization of the simulation model to interlinking with various plant data bases, allowing components to operate as hardware in the loop system (HIL).

1. Introduction

The traditional approach of simulation tools is to describe physical processes or objects in or mathematical or physical models. These systems are described with discrete equations or with continuous algebraic and differential equations. The simulation environment uses a solver algorithm which generates a solution for the system model considering the initial values.

In some cases the user or other applications have to interact with the simulation, for example to:

- Start or interrupt the simulation
- Change parameters or variables during the simulation run time
- Communicate with other applications, for example a database as data source
- Exchange data with input/output cards or a peripheral communication interface
- Build up a hardware in the loop simulation environment

HIL is the integration of real components and system models in a common simulation environment [1]. This means that some parts of a system, which should be

tested, are modeled and other parts are real. HIL simulations are an important method for the development of mechatronic systems.

An important advantage of HIL is that it allows function tests of mechatronic systems or components of such systems under simulated real conditions. Therefore HIL helps to save cost and time compared to conventional test runs on a real prototype.

There are three important considerations for the implementation of a hardware-in-the-loop simulation:

- The simulation of a dynamic system, we assume the mathematical or physical model must be processed in real-time
- There must be synchronization between the time in the real world (the so called real time) and the digital simulation-time of the simulation tool.
- The simulation tool must be able to communicate e.g. with other applications or an I/O communication interface.

The cbb software GmbH developed an Energy and Media Solution (EMS) for the Steel industry for that they acquiring know-how in the simulation of integrated steel plants and its processes.

A major part of the simulation models was built up in Modelica 3.2 and simulated with Dymola7.4 [2], [3]. In order to verify and validate the implemented models and the developed system components, a connection to a hardware-in-the-loop simulation environment has to be implemented.

In this way, an interaction of a Modelica/Dymola simulation with components outside of the simulation tool is shown in this paper.

At this proposed HIL system the Dymola application runs on a Microsoft Windows Operating System (XP or 7).

Actually three important processes run on the simulation PC:

- one process is the Dymola application
- Second process provides the input and output functionality (called as middleware).

- The third process is the Energy and Media Solution (EMS) application which controls the Dymola simulation.

Details on the implementation of these tasks are given below.

2. The System Setup

Figure 1 shows the global system

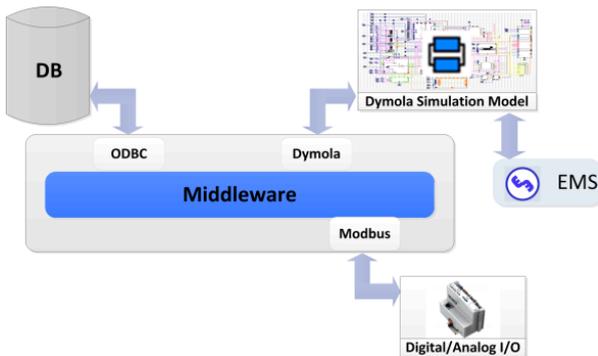


Fig 1. Global System

2.1. Energy and Media Solution (EMS)

The EMS software is used to configure/start the Dymola simulation and to visualize the simulation results.

2.2. Dymola simulation model

The model calculates the online energy and material flows of an integrated steel plant. The inputs for the model verification are provided by various steel plant databases and other A/D inputs.

Figure 2 visualize the base components and flows of the integrated steel mill.

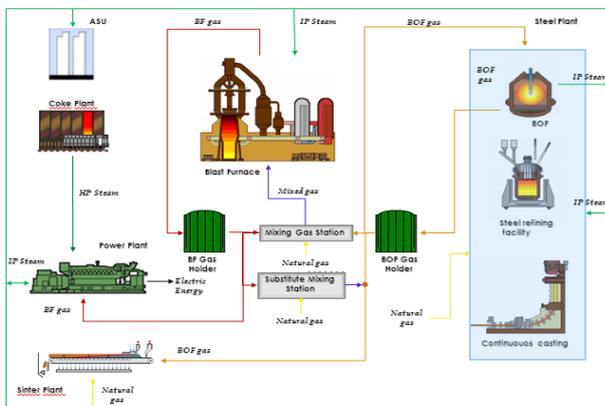


Fig 2. Steel Plant

2.3. Middleware

The internal functionality of the here used middleware LabMap is discussed in earlier works [4, 5, 6] and will not mentioned here.

In this approach the middleware is used as data handler between Dymola and external data source and sinks. In this case the LabMap ODBC interface and Modbus interface is used to acquire data from the integrated steel mill processes that are locate on databases and an digital and analog I/O's.

For other data sources, LabMap provides different Interfaces to Hardware and Software components.

Following a list of supported Hardware and Software:

- Virtual registers
- AK (AK interface)
- DMX (Digital MultipleX device interface)
- FLG (Driver's aid interface)
- IAVCAN (IAV CAN iteface)
- ModBus (MODBUS/TCP interface)
- Net (Network TCP/IP interface)
- NICAN (National Instruments' CAN-bus interface)
- ODBC (Open DataBase Connectivity interface)
- PGMNet (Network Pragmatic Generic Multicast interface)
- PLU (PLU 401 interface)
- SPS (OPC interface)
- User (User registers interface)
- VCI (Virtual CAN interface of IXXAT)
- Virt (Virtual registers interface)
- VXL (CAN interface through XL driver library Vector Informatik GmbH)

And a list of Software components that can be used with LabMap:

- Matlab
- LabView
- WinFact

For programming languages like Modelica, LabMap provides an Application Programming interface (API) in ANSI C. So that Modelica can be interconnected with LabMap and further hardware and application.

To get access from the Dymola model to LabMap, an external function library in ANSI C is implemented in the Modelica language that wraps the LabMap API functions.

With this external function library, the Dymola model has access to the data points configured on the middleware LabMap.

3. Middleware configuration

To allow Dymola to read or write a specific external data point from the middleware, the middleware must be configured first. This can be implemented dynamically or static.

Each data point has a unique identifier, a name, a data direction, a unit and hardware interface specific settings.

The external Modelica library use the unique data point identifier as parameter to access the data point on the middleware.

4. HIL Simulation of the Model

As preparation the Dymola model must be compiled for hardware-in-the-loop real-time simulation. Otherwise an algorithm must be implemented to synchronize the simulation runtime to real time See [6].

Dymola generates after compilation the `dymosim.exe`. This application contains the executable runtime simulation model.

The simulation executable (`dymosim.exe`) is started by the EMS application and the simulation is initialized.

At each simulation step the required external parameters for the model are requested by a `get`-function from the external Modelica library within the unique identifier.

A Modelica `get`-function call activates the underlying LabMap function to request the value from the data point with the matching identifier and requests the value from the configured hardware interface. In this case: values from a database and other digital and analog I/O.

After the specific hardware interface returns, the requested value from the external Modelica `get`-function returns the value to the simulation model.

The simulation step results can be written to the middleware so that other applications can access the data, for e.g. for further analysis or statistics.

5. Conclusion

The Modelica hardware-in-the-loop simulation approach using middleware communication was shown in this paper. This proposed system does not need an expensive “true” real-time platform. It bases on a common Windows PC with a Dymola simulation environment interlinking with the middleware LabMap which is an application under the operating system.

This application can therefore also be used in Systems where the process data does not change so rapidly for e.g. chemical process systems and so on.

References

- [1] Verein Deutscher Ingenieure, VDI Richtlinie 2206, Entwicklungsmethodik für mechatronische Systeme – Design methodology for mechatronic systems, June 2004.
- [2] Dymola, Dynamic Modeling Laboratory, User’s Manual, <http://www.Dynasim.com>: Dynasim AB, 2004.
- [3] P. Fritzson, Principles of Object-Oriented Modelling and Simulation with Modelica 2.1. Piscataway, NJ: IEEE Press, 2004.
- [4] C. Bruce-Boye and D. Kazakov, “Distributed data acquisition and control via software bus,” Proceedings CSMITA’04, pp. 153–156, Sep 2004.
- [5] C. Bruce-Boye, D. Kazakov, Quality of Uni- and Multicast Services in a Middleware. LabMap Study Case,“ Conference CIS2E 06 (International Joint Conference on Computer, Information and System, Science and Engineering”) 2006, IEEE, 4-14 December 2006
- [6] C. Bruce-Boye, D. Kazakov, “Distributed data acquisition and control via software bus”, International Industrial Ethernet Development High Level Forum 2004 (IEHF 2004) in Peking, Automation Panorama No. 5