

# A Common User Interface for Offline and Real-time Simulation of Transients

M. Giroux, J. Mahseredjian

**Abstract**—This paper presents EMTP-RT, a real-time toolbox developed for use with EMTP-RV – an application for the simulation and analysis of electromagnetic transients. EMTP-RT enables users to share a single, unique graphical user interface (GUI) for both offline and real-time hardware-in-the-loop (HIL) simulation of transients. The growing need to conduct real-time HIL studies presents new challenges to users of classic offline transient simulation applications. These challenges include the requirement for models to be separated and distributed across multiple processors to reach the computational performance required for real-time simulation, for extra tools to be added for data visualization and synchronization, and for the design interface to contain an HIL interface for model communication with actual analog and digital signals.

**Keywords:** real-time, EMTP-RV, Graphical user interface, EMTP-RT

## I. INTRODUCTION

MANY power system simulation tools are available on the market for use in different types of studies, including Load-Flow, Short-Circuit, Stability, and Transient studies. Other simulation tools are available for solving circuit equations in real-time and for interfacing digital and analog I/Os to physical control devices, thereby enabling accurate simulation of FACTS, HVDC, photovoltaic systems or drives. Although most simulation tools use similar models and modeling concepts, they each use different modeling user interfaces and proprietary data formats. Consequently, this can create major delays for users who wish to use more than one tool to run different types of studies. Moving from one tool to another typically requires meticulous manual data translation. In addition, since each interface has its own functionality, the engineer may be forced to spend additional time adapting to and learning new modeling environments.

This paper presents a real-time toolbox that combines two simulation tools in a single GUI. The real-time toolbox enables EMTP-RV [1] users to perform real-time HIL studies with models previously built for offline transient simulation only. The real-time toolbox is more than a data translator since it

enables real-time functionality within the EMTP-RV GUI. Therefore, users can perform offline transient and real-time studies using a unique modeling interface, without the need to adapt to new software.

Real-time simulation requires a platform that enables parallel processing on multi-processor targets, as well as I/O conditioning. The real-time simulator platform which this toolbox is designed to work with is eMEGAsim from Opal-RT Technologies [2][3][4]. eMEGAsim is built entirely from commercial-off-the-shelf (COTS) hardware and software components. It uses QNX as a real-time operating system, and a target PC built on two Intel Quad-Core processors, yielding the computation power of eight CPU cores. Simulink [5], an application for multidomain simulation and Model-Based Design of dynamic and embedded systems, and its associated SimPowerSystems [5] library, are used for the modeling of power system and control devices. The RT-LAB software platform enables distributed real-time simulation of the Simulink models over the 8 processor cores.

## II. A COMMON MODEL DATA FILE

EMTP-RV's GUI, EMTPWorks, is totally decoupled from the application's computational engine. User-defined model data is transmitted to the EMTP-RV solver via a text file named netlist. The EMTP-RV computational engine reads and processes the model data, generating and solving model equations in both steady-state and time-domain. Simulation results are then output to a binary file (see

Fig. 1 for an overview of the entire process). The EMTP-RV computational engine can also run load-flow and frequency-domain studies.

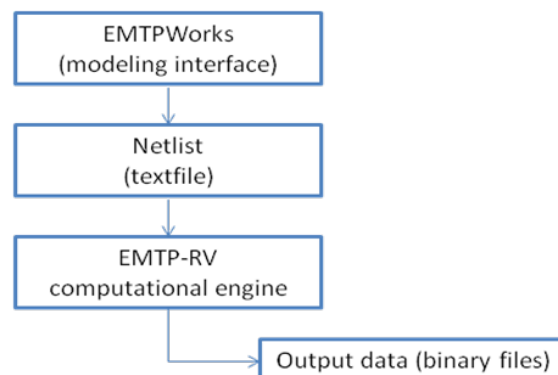


Fig. 1. EMTP-RV program main components

M. Giroux is with Opal-RT Technologies Inc., Berlin, Germany (mathieu.giroux@opal-rt.com)

Jean Mahseredjian is with École Polytechnique de Montréal, Campus Université de Montréal, 2900, Édouard-Montpetit, Montréal (Québec), Canada, H3T 1J4 (corresponding author, 514-340-4711 (4870); e-mail: [jeanm@polymtl.ca](mailto:jeanm@polymtl.ca)).

Fig. 1 illustrates the basic architecture of EMTP-RV. Other functions, not shown here, allow extracting and exchanging of

objects from EMTPWorks. It is also possible to direct communications to EMTPWorks from the computational engine.

The netlist file contains all the data required by the computational engine; simulation option parameters, as well as details on design devices, including their type, their parameters and interconnections. The EMTP-RV netlist file uses a simple proprietary format that yields fast writing and reading. The file generation process is very fast; less than a second, even for large power system models with different subcircuit levels. The netlist file is not encoded. Therefore, it can be read and understood by external applications. Each component present in the design (simulation case) is recorded in the text file using character strings with comma, semicolon and/or equal sign separators. The format is based on object serialization and designed for fast decoding. It includes hierarchical formatting for the representation of multilevel and masked subcircuits.

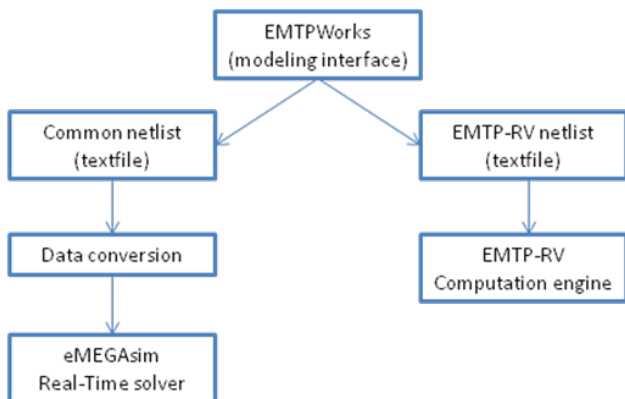


Fig. 2. New netlist for real-time interface

EMTPWorks is an open-architecture package that enables users to modify the format and data contained in the netlist file of the studied design. It achieves this through the use of a special scripting language called *Report Script Language*. The use of a scripting language means that no compilation is required, enabling any high-end user to access the file containing the netlist creation rules in EMTPWorks and edit it to define any desired file format. The script contains a very large collection of commands and offers the freedom to design the most complex netlist formats. EMTP-RT uses this feature to generate a common data exchange file containing all EMTPWorks design data which is then transmitted to the real-time simulator. Since the standard EMTP-RV netlist file only contains data required by the EMTP-RV computational engine, several parameters must be added prior to exporting data to the EMTP-RT real-time interface. These include data on device aspect, such as position, size, and orientation. Such data can be simply grabbed from EMTPWorks objects. The resulting common netlist file between EMTP-RV and eMEGAsim resembles the standard netlist file with the exception of the additional data required for real-time execution. The success of this approach opens the door to using this data format to enable EMTP-RV to share data with other third-party tools, providing the engineer with greater

flexibility in the choice of simulation environment.

Fig. 2 shows how a case study model developed in EMTPWorks can be interfaced with the real-time platform. The same model can be used for conducting offline transient simulations with EMTP-RV or real-time studies with eMEGAsim.

### III. MODIFICATIONS TO THE GUI FOR REAL-TIME

In addition to handling netlist issues, it is necessary to add specific functionality and libraries to EMTPWorks to enable real-time studies. This can be done at the user level without the need for any hard-code changes.

#### A. Model Separation and the Use of ARTEMiS Lines

Real-time simulators achieve real-time performance by distributing models across multi-processor targets. Before being simulated in real-time, models of large power systems are separated into different subsystems. Each subsystem is solved by a unique processor.

Once separated, the subsystems transmit signal data between each other through the use of shared memory inside the same real-time target or via communication links for clustered targets. Model separation introduces a data communication delay between subsystems equivalent to at least one time-step. For instance, a voltage value metered at time  $t$  in subsystem #1 will only be accessible in subsystem #2 at  $t+\Delta t$ , where  $\Delta t$  is the simulation time-step. To maintain simulation accuracy and stability, a power system model cannot be separated randomly.

In some cases the separation rules are easy to follow, in other cases they are not. This depends on the nature of the network being simulated. With real-time simulation of power systems, the natural choice for separation point is the transmission line since it introduces a propagation delay. The real-time toolbox uses the decoupling elements provided with ARTEMiS, a real-time solver for use in the Simulink/SimPowerSystems environment [6][7]. This library contains essential decoupling elements for the parallel distributed solving of system equations across multi-core processors. The real-time toolbox allows users to right-click on any transmission line of the system to determine a separation point. When the separation point is chosen, the program automatically separates the power system model and identifies individual subsystems by attributing a specific color and a name to each. Fig. 3 illustrates how each subsystem is represented by a unique color within EMTPWorks. In the separation process, the real-time toolbox replaces EMTP-RV transmission lines with ARTEMiS decoupling lines. The decoupling is naturally made with distributed parameter line models and their intrinsic propagation delays.

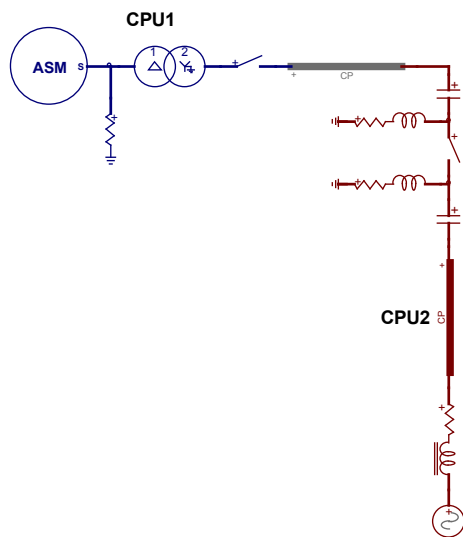


Fig. 3. Separation of a simple EMTP-RV circuit onto two processors

The studied network can be also separated on the control logic block side. Natural separation control points are devices that introduce a delay between input and output. In cases where the model does not contain any transmission line or a control block for separation, the user can still separate the model on a signal. However, this operation will automatically introduce a delay in the network, and consequently an error in the system, or even simulation instability.

There are some exceptions. Model separation can be achieved through artificial means by substituting transformer inductances or shunt and series capacitors with a distributed model. This enables the solution of equations in parallel. However, such an approach typically requires the engineer to have extensive modeling experience and expertise. The impact of each non-natural delay must be evaluated by the user. The network can be separated, not only at the top-level of the design, but also in any subcircuit.

#### B. Monitoring and I/Os

The real-time toolbox adds new libraries to EMTPWorks for real-time modeling purposes. The RT-LAB I/O library contains blocks that enable the model to communicate with real analog or digital signals that may come from external hardware. The Opal-RT OP5110 family of I/O cards are currently supported. Opal-RT OP5110 FPGA-based I/O cards feature 10-nanosecond digital I/O, 1-microsecond D/A converters, and 2-microsecond A/D converters.

Another library called RT-LAB has been added to enable standard real-time operations in EMTPWorks including:

- writing data in a file during simulation
- triggering real-time data acquisition
- monitoring real-time variables (including effective time-step and number of overruns,)

#### C. Initial conditions

Initial system conditions, including control states and machine speeds, can be pre-calculated in EMTP-RV and

transmitted to the real-time simulator.

Steady-state and load-flow studies are performed using the EMTP-RV solver. EMTP-RV outputs load-flow solution data (P, Q, V,  $\theta$ ) for each bus through an HTML file. EMTP-RT reads the data and calculates the initialization parameters required by SimPowerSystems synchronous machines.

#### D. RT-EVENTS

RT-EVENTS [8], also developed by Opal-RT, is a Simulink library which enables the simulation of event-based systems. This toolbox solves the problem of inaccurate real-time simulation with fixed-step algorithms by using interpolation techniques to compensate for events that occur between two simulation time-steps. The library contains 2-level and 3-level IGBT bridges that enable precise simulation of high IGBT switching frequencies in real-time simulation. When deployed, the real-time toolbox adds RT-EVENTS library blocks to EMTPWorks. Users can then drag and drop any RT-EVENTS block into their EMTPWorks design. Currently, the event compensation functionality of RT-EVENTS is for use in real-time simulation only. When the user is conducting offline simulation with EMTP-RV, RT-EVENTS blocks are automatically replaced by equivalent EMTP-RV control blocks.

### IV. IMPORTING EMTP-RV DATA INTO THE REAL-TIME ENVIRONMENT

From EMTPWorks, the EMTP-RV network data must first be imported into the Simulink environment (see Fig. 4). Accordingly, each EMTP-RV device must have an equivalent in SimPowerSystems or within other Simulink control blocks. Most blocks can be found in both simulation tools. Before converting the model under study, EMTPWorks evaluates the current schematic and warns the user, if necessary, of any existing unsupported block types or parameters.

The program then reads the generated netlist file and finds an equivalent for each block. EMTPWorks' netlist parsing capability includes algorithms that take into account the recursive aspect of the model design, allowing users to develop models on different layers (subcircuits).

For each device, EMTP-RT applies conversion, if necessary, to make model data fit from one interface to another.

Finally, one Simulink/SimPowerSystems network model is automatically generated per defined subsystem (CPU). The models are ready for real-time simulation, i.e.: they contain ARTEMiS blocks and the appropriate format required by the RT-LAB real-time simulation platform.

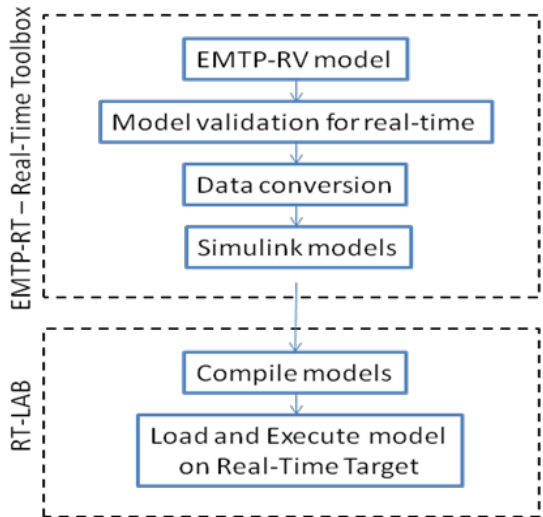


Fig. 4. From an EMTP-RV model to Real-Time simulation

## V. DEMONSTRATION EXAMPLE

The demonstration example is a 500 kV 23-bus transmission network currently operated in Asia. The system is comprised of seven 1000 MVA generators supplying power to 17 loads over 45 overhead lines (see Fig. 6).

To enable comparison of results between EMTP-RV and the real-time simulator, no real hardware is connected in this demonstration example.

The model is built in EMTPWorks and is separated into 4 subsystems. Each subsystem has been assigned to one processor node.

The EMTP-RT real-time toolbox is run and the network is automatically translated into 4 equivalent Simulink/SimPowerSystems subsystems. The system is then compiled and loaded onto the real-time targets using RT-LAB. The system is set to run in real-time with a time-step of 60  $\mu$ s.

### A. Real-time performance

The time-step of 60  $\mu$ s is a user-defined value. From EMTPWorks, a block from the RT-LAB library, called OpMotoring, is added onto each subsystem. This block can monitor different real-time simulation parameters. It is used to monitor the actual time-step required by the eMEGAsim real-time simulator to solve each subsystem. For instance, in this example, the subsystem CPU1 communicates with the other processor cores and solves in 15  $\mu$ s. The processor core then remains idle until the remainder of the user-defined time-step has passed (an additional 45  $\mu$ s). This user-defined time-step includes communication time for data exchange between processors.

TABLE I shows the performance results obtained for each processor. It appears that some processors are used more than others. If the size of the real-time simulation time-step is an important issue, there are two options for reducing the time-step value:

- distribute parts of Subsystems 1 and 3 on CPU1 while

keeping the number of CPUs to 4;

- distribute the network on additional CPUs, since 8 CPUs are available on the real-time simulator.

### B. Comparison between EMTP-RV offline and real-time results

Both an EMTP-RV offline simulation and a real-time simulation (using ARTEMiS) of the same network are run using EMTP-RT. The results are compared by superimposing the measurements taken from the two simulation environments.

TABLE I  
REAL CALCULATION TIME-STEP FOR EACH PROCESSOR ON REAL-TIME TARGET

Subsystem Name	Content	Real Time-Step
CPU1	Machine Controls	15 $\mu$ s
CPU2	Subsystem 1	53 $\mu$ s
CPU3	Subsystem 2	37 $\mu$ s
CPU4	Subsystem 3	42 $\mu$ s

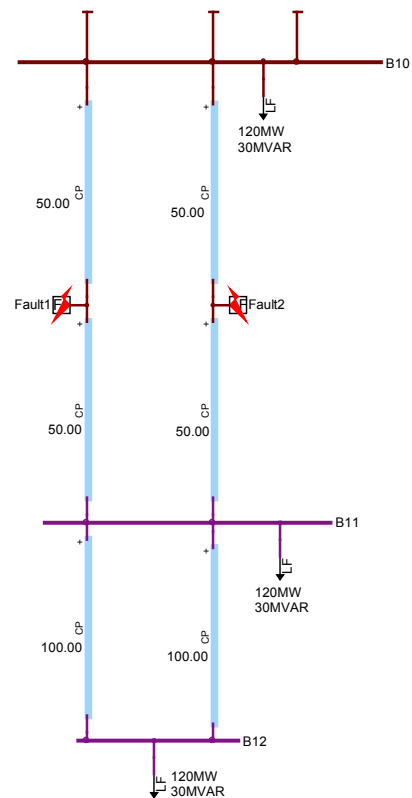


Fig. 5. Buses B10, B11, and B12 represented in EMTPWorks

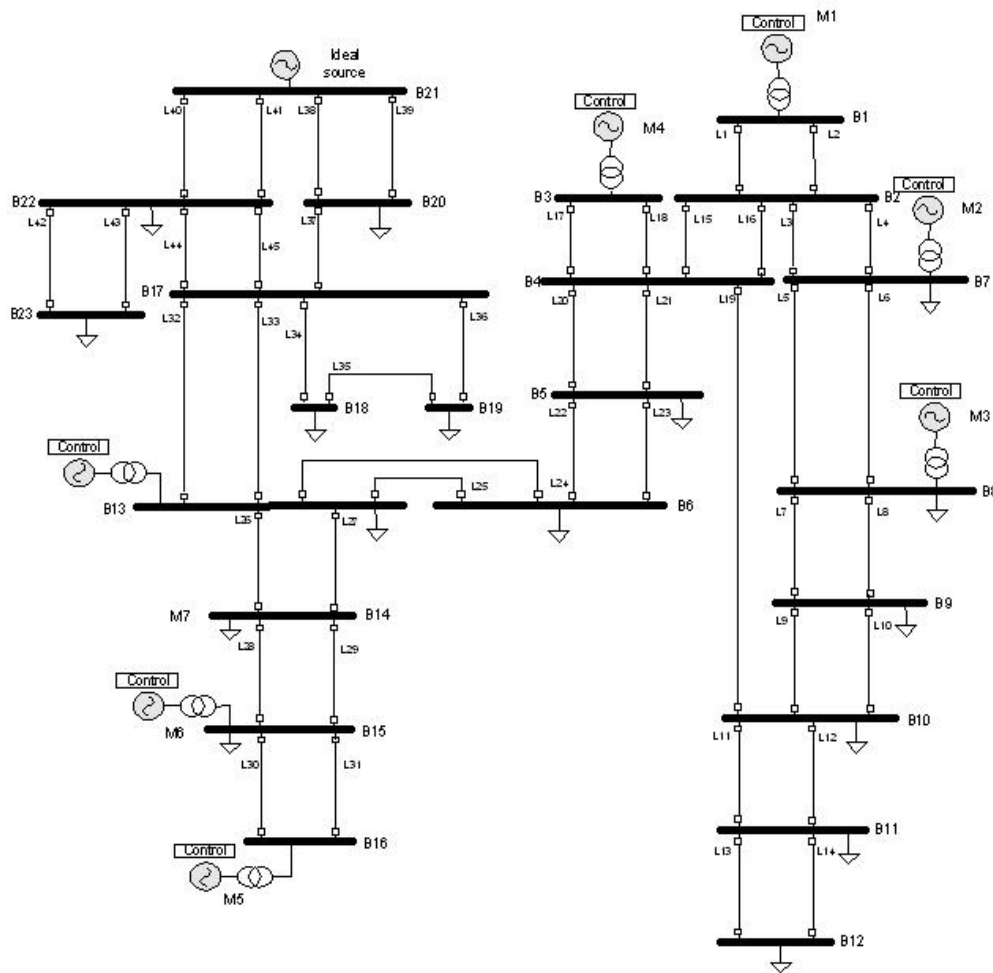


Fig. 6. The demonstration example network

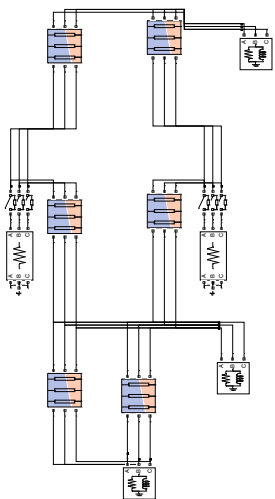


Fig. 7. Conversion from EMTPowerWorks to Simulink

First, load-flow and steady-state studies are run. These studies are run to initialize the model state variables and the synchronous generators. The load-flow data is transferred to the translated Simulink model, automatically initializing the SimPowerSystems generators. For the time-domain

comparison tests, a three-phase fault is applied on the lines between buses B10 and B11 (see Fig. 6). A 50 millisecond fault is applied when the voltage on bus B11 crosses zero. A zoomed view for buses B10, B11 and B12 in EMTPowerWorks is shown in Fig. 5. The corresponding Simulink/SimPowerSystems (translated) network is illustrated in Fig. 7. This translation is automatic.

Power and voltage measurements are compared at bus B11 when the transient occurs. The results are presented in Fig. 8, and Fig. 9.

The results demonstrate the accuracy of the two simulation tools and the performance of the real-time toolbox data converter.



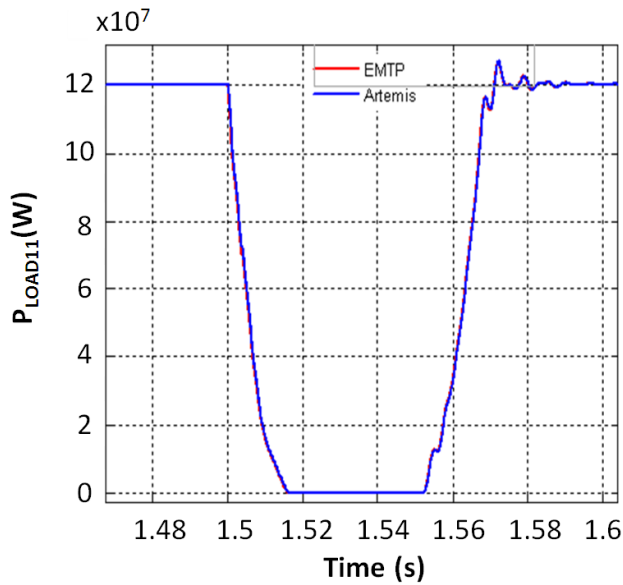


Fig. 8. Bus B11 active power measured during the phenomenon

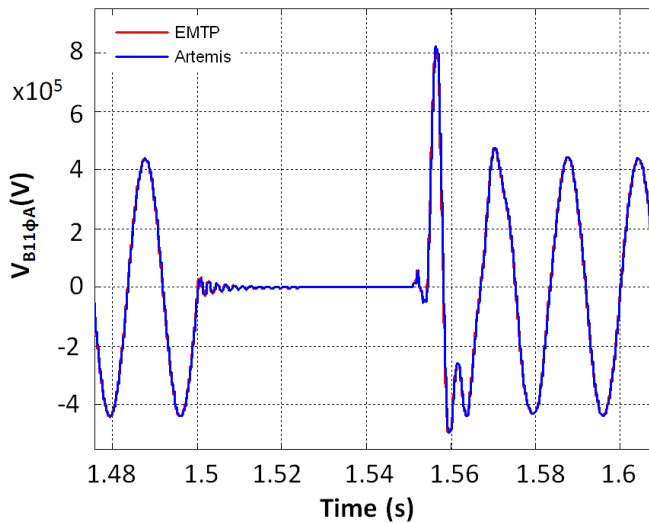


Fig. 9. Bus B11 Phase-A Voltage measured during the transient

## VI. CONCLUSIONS

This paper must be viewed as an experiment with portability of data using open architecture concepts for the creation of a common graphical user interface for both offline and real-time simulation methods. Such a concept has a dramatic time-saving impact for conducting advanced studies, since from the same design it is feasible to generate and run simulations in the real-time mode.

Key software development aspects have been identified and used for the presented development. These are mainly open architecture, scripting methods for generating data, scripting methods for automatically placing and populating objects in the receiving software, and methods for automatically complying with the rules of another simulation environment. It is noticed that in this demonstration it was also possible to recreate the complete network diagram in the receiving application, which is an ultimate capability.

## VII. REFERENCES

- [1] J. Mahseredjian, S. Denetière, L. Dubé, B. Khodabakhchian and L. Gérin-Lajoie: "On a new approach for the simulation of transients in power systems". *Electric Power Systems Research*, Volume 77, Issue 11, September 2007, pp. 1514-1520
- [2] [www.opal-rt.com](http://www.opal-rt.com)
- [3] L.-F. Pak, O. Faruque, X. Nie, V. Dinavahi, "A Versatile Cluster-Based Real-Time Digital Simulator for Power Engineering Research", *IEEE Transactions on Power Systems*, Vol. 21, No. 2, pp. 455-465, May 2006
- [4] C. Dufour, G. Dumur, J. Paquin, J. Bélanger "A PC-based Hardware-In-the-Loop Simulator for the Integration Testing of Modern Train Traction Systems" *PESC'2008 Conference Proceedings*, Rhodes, June 2008.
- [5] [www.mathworks.com](http://www.mathworks.com)
- [6] C. Dufour, J. Bélanger, S. Abourida, "Real-Time Simulation of Onboard Generation and Distribution Power Systems," in *Proceedings of the 8th International Conference on Modeling and Simulation of Electrical Machines, Converters and Systems (ELECTRIMACS 2005)*, April 17-20, 2005, Hammamet, Tunisia.
- [7] C. Dufour, J. Bélanger, "Discrete Time Compensation of Switching Events for Accurate Real-Time Simulation of Power Systems", in *Proc. 2001 IEEE Industrial Electronics Society Conference*, pp. 1533-1538.
- [8] C. A. Rabbath, M. Abdoune, J. Belanger, "Effective Real-Time Simulations of Event-Based Systems" *Proceedings of the 2000 Winter Simulation Conference*, 2000.