WM-SIM: A Platform for Design and Simulation of Wireless Mobile Systems

Juan J. Sánchez, D. Morales-Jiménez, G. Gómez, Eduardo Martos-Naya,
Unai Fernández-Plazaola, and J. T. Entrambasaguas
University of Málaga. ETSI Telecomunicación
Campus de Teatinos s/n. Zip Code 29071
Málaga, Spain
<jjsanch,morales,ggomez,eduardo,unai,jtem>@ic.uma.es

ABSTRACT

The design and validation of forthcoming mobile communication technologies (Beyond 3G and 4G) require appropriate tools for building reference models. In this article, a new tool called WM-SIM (Wireless Mobile SIMulator) is presented and compared with others recognized tools as Simulink, Visual System Simulator (VSS) and Ptolemy II. Benchmark results have proved WM-SIM to be more efficient than aforementioned tools in terms of processing time. Finally, a model for the coming cellular technology called Long Term Evolution (LTE) is modelled and simulated with WM-SIM. Simulation results for LTE technology are shown in order to prove our platform as a good option to evaluate this kind of systems.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modelling**]: Simulation Support Systems - Environments

General Terms

Performance, Design, Algorithms

Keywords

Wireless, Performance, LTE, Simulator, Platform

1. INTRODUCTION

Nowadays, there is a growing interest on next generation of mobile communication technologies (Beyond 3G and 4G). The definition of these technologies is a quickly changing process that implies continuous validation of new features and architectures. Therefore, the development of efficient and reliable simulation tools for modelling and evaluating these systems is a hot topic both from academic and business point of view. Several alternatives as Simulink [1], Visual System Simulator (VSS) [2] or Ptolemy II [3], are already available to validate these complex models.

These simulation tools usually provide graphic user interfaces for simplifying the definition of models. However, the design of new systems is often constrained by the available blocks and functionalities. On the other hand, existing tools are general purpose simulators, where neither efficiency nor resources consumption have been optimized. This may be a significant limitation for evaluating actual complex systems that need long simulations.

In this paper, we present the Wireless Mobile Simulator (WM-SIM) Platform [1], a design and simulation tool oriented to the evaluation of wireless communication systems. WM-SIM has been developed to support tailored models for actual systems; thus, integration of real code into a model is a simple task.

The rest of the paper is structured as follows. In section 2, a brief description of WM-SIM is presented, focusing on its basic elements and key features. In order to evaluate the performance of a model created with WM-SIM, section 3 provides a performance comparison between that model and other models built with recognized tools like Simulink, Visual System Simulator and Ptolemy II. An example of use of WM-SIM to model complex system is presented in section 4, where results obtained with a model based on Long Term Evolution (LTE) technology are presented. Finally, section 5 gathers main conclusions.

2. WM-SIM OVERVIEW

WM-SIM is a data-flow oriented platform, where the execution of the different processes is managed by generation and consumption of data. Data flow into a process pipeline made up by several process stages. Interconnection between different stages is managed by sharing memory addresses for writing and reading purposes.

WM-SIM allows for multi-rate models and, therefore, it is possible to combine different processing stages with different data rates. This is a key feature when it comes to model mobile communication systems with several asynchronous processes and complex data transfer between them. The platform also includes an extensible library of behavioral blocks for creating modular systems.

[1]It is the result of a collaborative research project between Communication Engineering Department at the University of Málaga and the company AT4Wireless S.A.

MSWIM'07, October 22–26, 2007, Chania, Crete Island, Greece.

2.1 Basic entities

In WM-SIM *Block* entity is the elementary building element and it represents the minimal executable unit. Specific blocks, instances of this entity, may be created by designers, e.g. an Encoder block to carry out channel coding. A block models a specific process as a 'Black box', for which only its inputs and outputs are known. Configuration of specific block is possible by means of several parameters that may be included in a parametrization file.

A set of basic operations that are predefined for any instance of *Block* entity. For instance, interconnection between blocks (*bind* operation) or loading configuration parameters (*settings*). Besides, operations to manage the life cycle of any instanced block must be carefully defined for each specific block. The most important is the *process* routine, where block's processing and functionality are defined.

Communication among blocks is carried out by means of efficient memory management. Each block is responsible of allocating memory for its outputs and other blocks address these memory positions for reading purposes. Each block's input has an interconnection attachment called *signal* that specifies the memory address to be read. Thus, memory consumption is reduced and inefficient multiple copies of data are avoided.

Following a modular approach, the *System* concept is defined to model systems made up by several functional blocks. A complete simulation model is defined as an overall system that may contain both blocks and other systems (then called subsystems).

2.2 Block design

The platform includes a repository of blocks and utilities called WM-SIM library. However, models created with WM-SIM are not limited to its associated library. New blocks may be defined to be added to the platform's library. Thus, users can create their own block library that may be reused by other users following some basic rules:

- Memory operations must be clearly defined and checked.
- Block's settings and functionality must be defined and implemented in their corresponding routines.

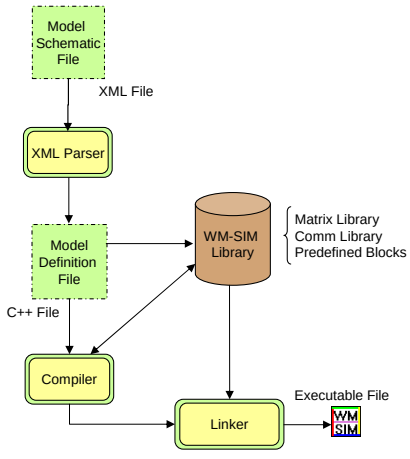

Figure 1: Stages in the creation of an executable file for a WM-SIM model.

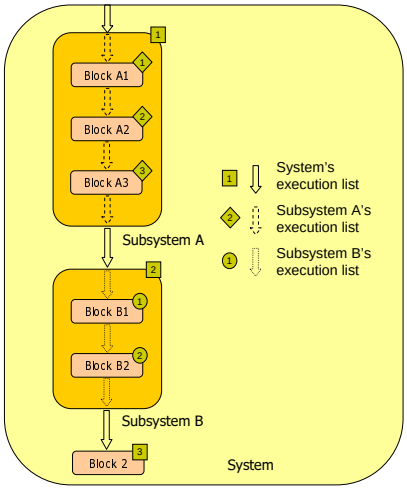

Figure 2: Execution list for WM-SIM example system.

- Finally, a block's behavior in terms of generation and consumption of data must be checked in order to avoid blockages in model execution.

WM-SIM and its associated library have been developed in C++ language. Hence, designers must follow the C++ programming premises when creating a new block in order to assure backward compatibility.

2.3 Stages in WM-SIM model design

The process of creating a simulation model with WM-SIM implies several simple steps, as depicted in Figure 1. Firstly, the designer defines an XML schematic file which specifies all the building elements (blocks and/or subsystems) and the interconnections between them. From this architectural description, the XML parser creates a C++ file (model definition file) automatically.

WM-SIM library gathers all the predefined blocks and new specific blocks created by designers, as well as other auxiliary libraries like Matrix (for vector and matrix operations) or Comm (which offers utilities for communication systems). Model definition file is compiled and references to WM-SIM library are resolved before linking stage. The final result is a stand-alone executable file that makes possible to simulate the model.

Thus, a WM-SIM model only uses the required resources when simulation is performed. This is a significant advantage with respect to other simulation tools that usually load a kernel with many unused functionalities before the simulation starts.

2.4 Model execution

Model execution is automatically managed by the overall system's *Check list*, which contains references to each functional element in the system (block or subsystem). In fact, the *Check list* establishes the order in which blocks are checked for being executed. A block will be executed if it is not blocked, that is, if its inputs are available for being processed and its outputs were already consumed by the following blocks.

Check list is also used to carry out system's creation and

destruction stages. In a complex system, a *Check list* may include other subsystems, each of them having their own *Check list*. Thus, nested lists can be defined in WM-SIM.

In Figure 2, a *Check list* for the previous example is depicted. Three different lists are easily identified: one for the entire model's execution and one for each of the subsystems. Checking order in model's list (marked with squares) is: 1) subsystem A, 2) subsystem B and 3) Block2. Checking order within subsystems A and B is marked with rhombus and circles, respectively.

WM-SIM allows for multi-rate models and, therefore, it is possible to combine blocks with different execution rates as long as buffers are included into these blocks.

3. SIMULATORS BENCHMARK

This section presents a performance comparison among WM-SIM and other available simulation platforms.

3.1 Simulink

Simulink [1], developed by The MathWorks Inc., is a software package for multi-domain simulation and Model-Based Design for dynamic systems. Models are hierarchical and they can be built using both top-down and bottom-up approaches. Simulink has the ability to simulate discrete systems, supporting multi-rate processing and systems that mix discrete and continuous components. The Simulink engine successively computes the states and outputs of the system at intervals from the simulation start time to the finish time, using information provided by the model. Simulink provides a customizable set of block libraries, which can be extended for specialized applications.

3.2 Visual System Simulator

The Visual System Simulator (VSS) [2], developed by Applied Wave Research Inc. (AWR), is a software tool for the design of complete, end-to-end communications systems. VSS is data-flow oriented, i.e., the model blocks are executed only when the required number of data samples are available at their input ports. VSS performs the interconnection between blocks by using first-in-first-out (FIFO) queues supporting asynchronous data transfer and multi-rate processing. A robust library of behavioral blocks is included in VSS platform. Users can incorporate their own algorithm or legacy code into VSS by several ways.

3.3 Ptolemy II

Ptolemy II [3], developed by a group of researchers at U.C. Berkeley, is a software for modelling, simulating and designing of concurrent, real-time embedded systems. It is a Java-based component assembly framework. Ptolemy II includes a growing suite of domains, each of which realizes a model of computation. Some of the implemented domains are: continuous-time modelling, discrete-event modelling, synchronous dataflow and process networks with asynchronous message passing. It also includes a component library, in which most components are domain polymorphic as they can operate in several domains.

3.4 Benchmark Results

Benchmark scenario is a simple Orthogonal Frequency Division Multiplexing (OFDM) system made up by a transmitter, an Additive White Gaussian Noise (AWGN) channel, and a receiver. Transmitted bits are compared with the received ones to calculate the Bit Error Rate (BER). Each simulation consists of the generation of 10000 OFDM symbols and different numbers of carriers have been used, namely: 512, 1024 and 2048. All platforms have been tested using the same computer, an Intel Core Duo processor at 2.4 GHz and 2 GB RAM.

Platform	512 carriers	1024 carriers	2048 carriers
WM-SIM	10.25 s	19.75 s	39 s
VSS	24.8 s	50.6 s	101.1 s
Simulink	66 s	131.3 s	265.5 s
Ptolemy II	225 s	450 s	895 s

Table 1: Time consumption for different compared platforms.

Simulation time for each platform is presented in Table 1. Efficiency gain of WM-SIM with respect to VSS (2.5 times faster) is related to the non-use of FIFO buffers to interconnect blocks, avoiding unnecessary copies of data. Comparison with Simulink is even more favorable to WM-SIM, since the latter is 6 times faster. On the other hand, Ptolemy II has the highest time consumption (over 20 times slower) and it is clearly behind the other tools, which could be due to the fact it runs in a Java Virtual Machine (JVM) [5].

4. WM-SIM LTE DOWNLINK MODEL

In this section, WM-SIM is tested for developing and evaluating an actual and complex model based on LTE technology specifications. Modelled scenario consists of a Base Station (evolved-Node B or eNode B in LTE terms) transmitting data to several User Equipments (UEs). Only downlink direction of the Physical and Medium Access Control (MAC) layers has been implemented and evaluated.

4.1 LTE Overview

LTE introduces important changes in the radio interface to improve the spectral efficiency and it is expected to become a high-data-rate, low-latency and packet-optimized radio-access technology [4]. Main LTE features are:

- Scalable bandwidth from 1.25 MHz up to 20 MHz.
- Peak data rate up to 100 Mbits/s in downlink and 50 Mbits/s in uplink (for 20 MHz bandwidth).
- Support latency below 5 ms in the user plane and below 20 ms in the control plane.
- Support for Multiple-Inputs Multiple-Outputs (MIMO) up to 4x4.
- Adaptive Modulation and Coding (AMC) allowing keeping the average Bit Error Rate (BER) below a predefined target value.
- It is based on Orthogonal Frequency Multiple Access (OFDMA), which allows frequency-dependent link adaptation for dynamic resources allocation.

4.2 Simulation Results

Several simulations have been carried out in order to analyze the implemented LTE downlink system. Different scheduling algorithms and transmission modes (SISO or MIMO 2x2) have been evaluated, as well as the impact of UE speed

on system performance. Simulation parameters are listed in Table 2.

Parameter	Value
Carrier frequency (fc)	1.8 GHz
Sampling frequency (fs)	30.72 MHz
Channel type	Flat Rayleigh
UE speed (v)	11 m/s
Cyclic prefix length	144 samples
Bandwidth	20 MHz
Number of data carriers	1200
FFT size	2048
Coding rate	1/3
Feedback delay	0.5 ms
Antenna correlation	0

Table 2: Simulation Parameters

Figure 3 shows delay results for different UEs and both for RR and BC scheduling algorithms. Average SNR values for each UE follows a log-normal distribution with 20 dB mean and deviation of 3 dB. With BC algorithm, delay for a particular UE depends on channel conditions and, therefore, users with a worse channel will experience longer delays.

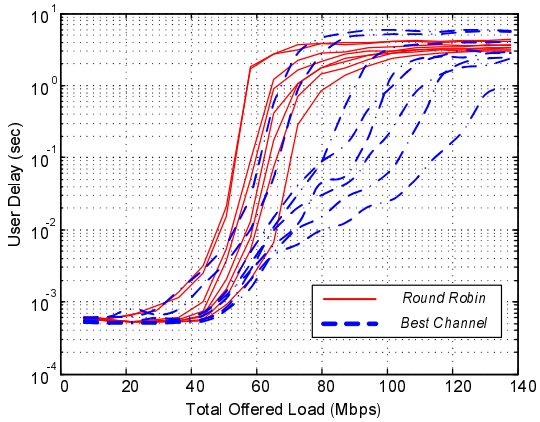

Figure 3: Delay evolution for 5 UEs with RR and BC scheduling.

SISO and MIMO 2x2 with Spatial Division Multiplexing (SDM) transmission schemes are compared in terms of spectral efficiency in figure 4. Spectral efficiency for MIMO-SDM doubles the value for SISO mode for the same BER_T (10^{-2}) value. Efficiency decreases gradually as BER_T is more restrictive.

5. CONCLUSIONS

WM-SIM platform has been shown as an useful and efficient tool for implementing and simulating complex communication systems. It includes a library of behavioral blocks and some auxiliary libraries with useful functionalities for signal processing and communication systems.

Benchmark results show that WM-SIM outperforms VSS by a factor of 2.5, and comparison with Simulink and Ptolemy II are even more favorable. Better performance of WM-SIM platform is due to the following features:

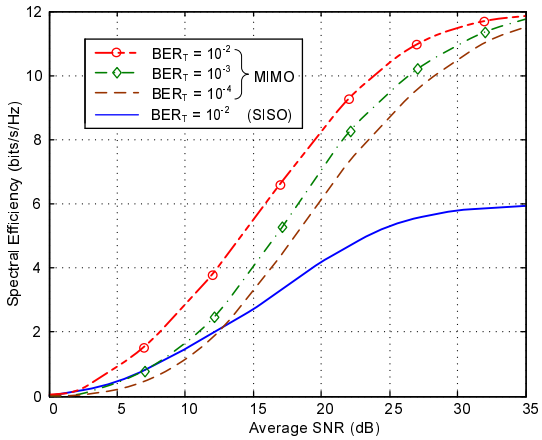

Figure 4: Spectral efficiency versus SNR for SISO and MIMO 2x2.

- Copies of data between blocks are avoided since interconnections are managed by memory references.
- Models are built into an stand-alone executable file that only uses the required resources.
- WM-SIM is dataflow oriented, i.e. execution is managed by generation and consumption of data, which is suitable for systems that involve signal processing algorithms.

Finally, several simulation results for a LTE downlink model built on WM-SIM have been presented as an example of evaluating wireless mobile systems.

6. ACKNOWLEDGMENTS

This work is partially supported by the Spanish Government and the European Union under project TIC 2003-07819 (FEDER) and by the company AT4Wireless S.A.

7. REFERENCES

[1] The MathWorks, Inc. . Simulink 6 Reference. Revised for Simulink 6.6 (Release 2007a). *MathWorks, Inc.*, March 2007. http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/slref.pdf

[2] Applied Wave Research, VSS 2006 Datasheet, *Applied Wave Research*, 2006. http://web.appwave.com/Products/pdfs/vss_simulator-collateral1.pdf

[3] C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, and H. Zheng, editors. Ptolemy II - Heterogeneous Concurrent Modeling and Design in Java, volume 2. *Department of Electrical Engineering and Computer Sciences University of California at Berkeley*, 2003. Memorandum UCB/ERL M03/29, version 3.0.

[4] 3rd Generation Partnership Project. UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE). *3rd Generation Partnership Project* http://www.3gpp.org/Highlights/LTE/LTE.htm

[5] N. Matloff and T. Fifield. The Java Virtual Machine *University of California at Davis*, December 11, 2006 http://heather.cs.ucdavis.edu/ matloff/50/PLN/JVM.pdf

[6] A. Goldsmith. Wireless Communications. *Cambridge University Press*, 2005.