Formal Verification and Validation of DEVS Simulation Models

Soremekun Ezekiel Olamide
African University of Science and Technology,
Abuja, Nigeria
lammyday@gmail.com,

Traoré Mamadou Kaba
LIMOS, CNRS UMR 6158, Université Blaise Pascal,
Clermont Ferrand 2, France
traore@isima.fr

Abstract - **We present a model-based verification technique built on selective and pragmatic use of formal methods, by using simplified model checking tools that focus on error detection rather than formalized proofs. This framework is to check and confirm that the trajectories and events of DEVS-Driven Modeling Language (DDML) simulation models and that of the real system agree in order to achieve replicative, predictive and structural validity through the lightweight application of formal methods. This is to reduce and ease the Simulation model verification efforts while increasing the coverage of the process, in order to verify the transformational accuracy of the model development process, increase confidence in the simulation model and allow for performance evaluation of simulation models. This framework provides a model refinement iterative procedure that helps to enhance the DEVS Simulation Model, correct errors or adapt to changing contextual requirements. This refinement procedure is applicable to evolutionary software development and systems requiring rapid prototyping, in order to meet up with changing requirements of such systems with the aid of iterative refinement. Furthermore, we present a case study example of a GSM telecommunication system to reveal the ability of this framework to not only formally verify system but also refine their models.**

Keywords: DEVS, DDML, Formal Methods, Model Verification and Validation, Model- Refinement, GSM

I. INTRODUCTION

System verification is a methodology used to establish that the design or product under consideration possesses certain properties or not [Baier and Katoen 2008]. It involves checking some properties of a designed system through a prototype or product in order to discover bugs or certify it bug-free. In the industry, peer reviewing and testing are the popular software verification techniques used, while emulation, simulation and structural analysis are the major hardware verification techniques [Baier and Katoen 2008]. However, these techniques have their limitations; the major pitfall of these methods is that they cannot exhaustively test all possible behavior or scenarios of a system, thus the motivation for the application of formal methods.

Moreover, the importance of accurate simulation model design cannot be over emphasized, in order to apply simulation models in the study of real life system, accurate knowledge is required. It is important to note that unless developed simulation models are demonstrated to be valid and certified to be credible, the predictions and explanations resulting from simulation experiments will carry the high risk of leading to the dissemination of inaccurate knowledge and decisions [Mackenzie et al 2002]. We therefore address the following issues in this work:

1) The interpretation and application of V & V in the context of Modeling and Simulation (M & S) Application Domain and the clarification of the relationship between M&S and V&V and the development of a fidelity framework [Sargent et al 2000].
2) Verification problem: pragmatic application of formal tools at different level of abstraction to perform efficient property checks.

Thus, we develop a fidelity framework that enables effective application of V&V in the study of Simulation Models. This is because, a comprehensive Verification and Validation (V&V) of Simulation Model is important to ensure safety, reliability and low cost in systems management and operational decision making. Besides, accurate system modeling through efficient V & V would enhance DEVS Simulation Model reusability and benchmarking.

II. OVERVIEW OF MODEL V & V, DEVS, DDML AND FORMAL ANALYSIS

Formal analysis, evaluation, Verification and Validation (V&V) methods are necessary to measure and predict certain design quality metrics from a Simulation model. Hence, it is important to understand these concepts and their applicability to DEVS Simulation Modeling.

Model Verification is the process of checking that a model correctly generates its behavior according to the system's specification. Model verification checks for the internal consistency between the simulation

978-1-4673-5943-6/13/$31.00 ©2013 IEEE

model of a system and the system requirements [Mackenzie et al 2002]. It helps the modeler to answer the question: is the model correctly implemented? Model-based verification techniques are based on models describing the possible system behavior in a mathematically precise and unambiguous manner [Baier and Katoen 2008]. This provides the basis for various verification techniques ranging from an exhaustive exploration (model checking) to simulation experiments and testing.

The Discrete Event System Specification (DEVS) is a modeling formalism that has become important and highly preferable in modeling and simulation; this is because other formalisms have been proven to have an equivalent DEVS representation. It has been shown that Discrete Time Events Systems and Differential (continuous) Event Systems can be easily converted to DEVS [Ziegler et al 2000]. DEVS is a semi formal modular formalism that promotes the separation of components/concerns (model, simulator and experimental frame).

DEVS modeling language (DDML) is a graphical and hierarchical formalism that was developed to construct models of dynamic systems for simulation. DDML was developed to capture the structure and behavior of systems by focusing on the three levels of abstraction, the Input Output System (IOS) level, the Coupled Network (CN) level and the Input Output Relation Observation (IORO) level [ighoroje et al 2011].

III. APPROACH

We introduce an approach that satisfy Simulation model conceptualization using the DDML formalism and formal methods to validate and refine models by developing a chosen well-formed formalism of software engineering models to capture the relevant properties and behavior of systems. DDML is a well defined paradigm that includes mathematical and logical concepts permitting deduction and reasoning by using model checking formal tools. The application of formal tools is used to reason about the behavior of system components at higher levels of abstraction with the selective application of formal methodologies and the focus on error identification and correction (refinement).

Formal Methods apply formal process of symbol manipulation and inference according to well defined proof rules of the utilized specification language [Mackenzie et al 2002]. These techniques are sought to reduce and ease the verification efforts while increasing their coverage [Baier and Katoen 2008]. Formal methods offer a large potential in the V&V of Simulation models, it helps:

a. to obtain an early integration of verification in the design process,
b. to provide more effective verification techniques and
c. to reduce the verification time [Baier and Katoen 2008]..

Model Checking is an automated technique that, given a finite-state model of a system and a formal property, systematically checks whether this property holds for (a given state in) that model [Baier and Katoen 2008]. Formal methods such as model checking can help model designers prove the satisfiability of certain functional as well as non-functional properties defined in the conceptual model, it exploits the finiteness property of the system by performing an exhaustive analysis of the transition system's states [Mackenzie et al 2002].

We carry out V&V of the DDML Simulation models through the application of well known formal tools, with adaptation to DDML; we use these tools to perform rigorous model checking. Through reuse and adaptation of useful software verification techniques from earlier research, we propose a way to perform model verification and refinement using formal tools. The approach verifies both the behavior and structure of the system in a hierarchical way:

1) *Behavioral verification*: running property check on the entire behavior space of the system at both the atomic and coupled levels.
2) *Structural Verification*: through the hierarchical behavioral verification at each level of abstraction;
   a) *Individual component model verification:* perform automated model checking on the system atomic models.
   b) *Coupled Network model verification:* perform automatic model checking on the system coupled model(s), with emphasis on the coupling, interactions and relationships among system atomic models.

IV. V & V MODEL DEVELOPMENT FRAMEWORK

The goal of linking model development process with V & V is to present the main life cycle phases and indicate how the verification and validation process using formal tools can be integrated with the model building life cycle. The Model Development with V & V Framework Steps includes the following:

1. *System Reasoning*: This stage involves brainstorming about the system to discover its

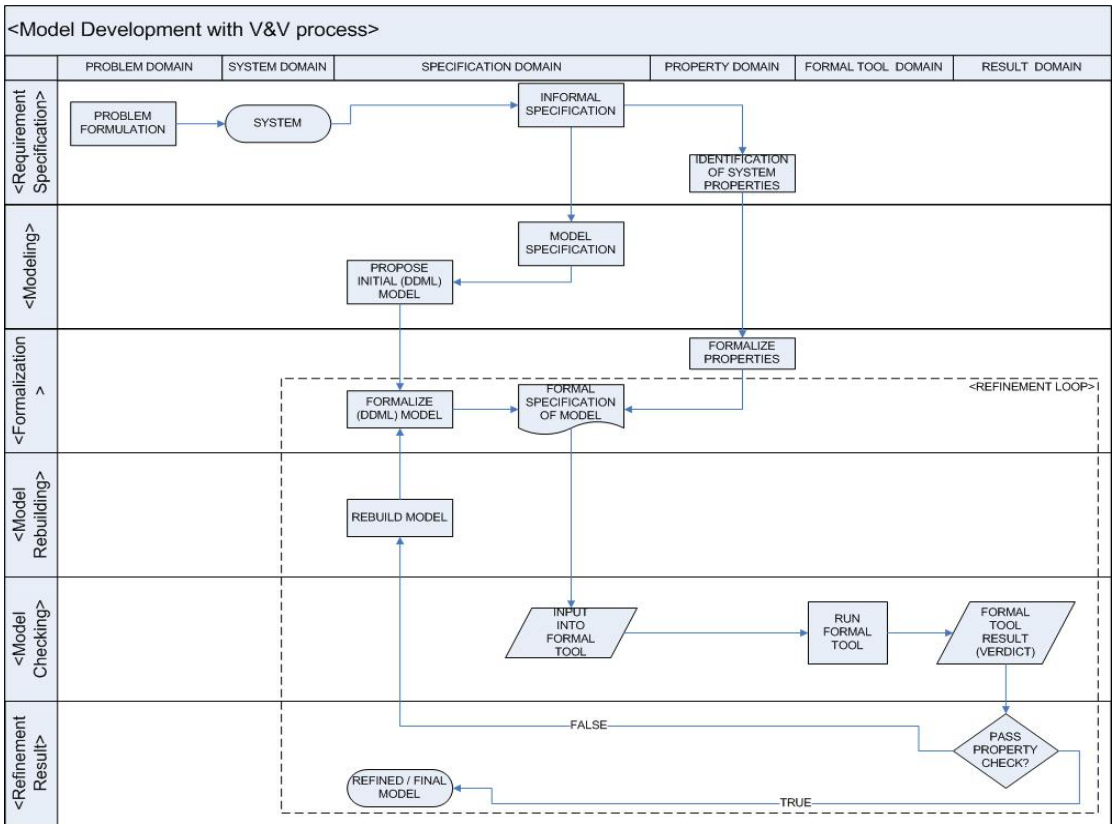

Fig 1: Flowchart Showing the Linked V&V Model Development Process

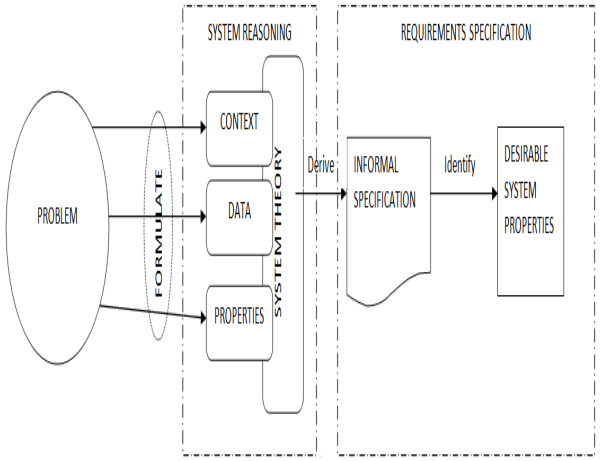

Fig 2: System Reasoning and Requirement Specification phase

behavior in order to build an initial model of the system and also outline its desirable properties (required constraints on the behavior of the system).

2. *Requirements Specification:* At this stage the model and the properties gotten from the system reasoning stage are formalized, translated to the formal language of the required formal tool. The transformation from the system reasoning stage to this stage is illustrated in figure 2 above.

3. *Modeling*: this involves all the activities that take place from the conceptualization of the model to the final DDML diagram specification of the model

4. *Verification*: the verification process is benchmarked on the use of formal methods, especially the application of formal tools in order to aid the basic (non-expert) users. Formal tools include model checkers, model verifiers, and theorem provers; these tools apply a strong mathematical framework to help users check their system (models) for properties.

5. *Model Rebuilding*: If the model passes the verification performed by the tool, then it is acceptable as the final model. This means it meets the system property requirements. However, if it fails, we rebuild the model to meet the defined properties, then re-formalize and perform verification again with the aid of the formal tool iteratively until it pass the property check and we obtain a refined model.

6. *Refinement:* The refinement technique is built on the use of formal methods, by using simplified model checking techniques and tools that focus on error detection. It detects inconsistency, incoherence and behavioral defects at the conceptualization and model specification stage; this helps to improve the modeling and simulation development life cycle and encourage model reusability.

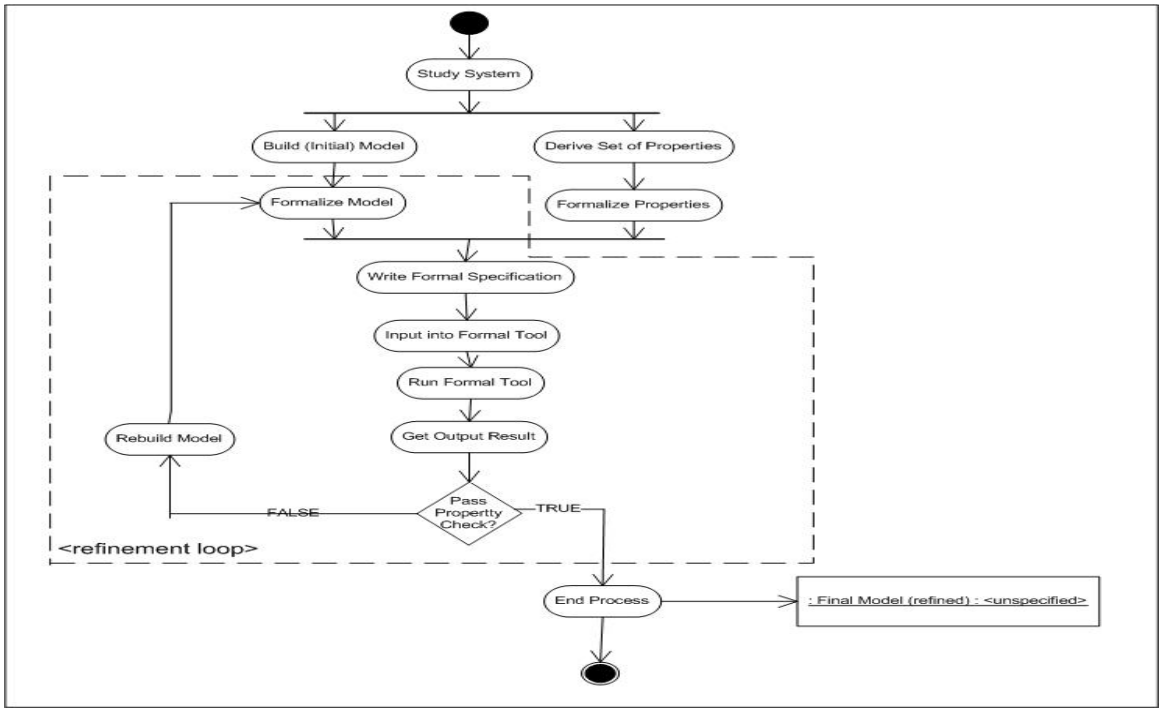

Fig 3: Activity Diagram for the V & V process with emphasis on the refinement stage

V. CASE STUDY

Our case study example models the GSM telecommunication system showing the interaction between the Network and a handset Phone as well as the individual behaviors of both entities. The system shows the concurrency of state events between the two atomic models (phone and network) and their interactions in the Coupled Network model, in order to achieve two major phone activities: calling and messaging.

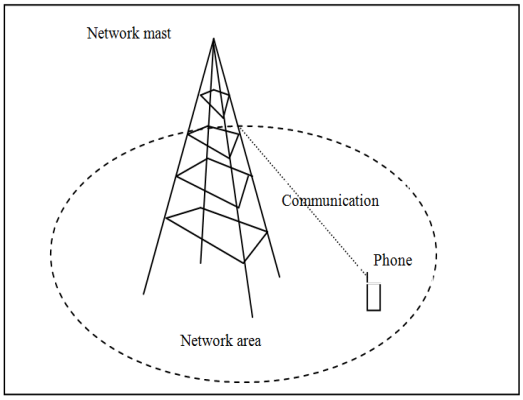

Fig 4: GSM illustration for the model Case Study

We model the activities of a simple handset phone and that of its service provider network as illustrated in figure 4 above. We enumerate the following permissible singular states events.

Phone System state events:

{PHONE_ON, MAKE_CALL, IDLE, PHONE_RING, RECEIVE_CALL, SMS_READING, SMS_SENDING, SMS_COMPOSITION, PHONE_OFF}

Network System state events:

{PHONE_ON, PROVIDE_SERVICES, MAKE_CALL, CALL_ALLOWED, SMS_SENDING, SMS_ALLOWED, SMS_SENT, DISABLE_SERVICE, PHONE_OFF}

Through the application of the Process Analysis Toolkit [Yan et al 2007], we developed the Communicating Sequential Processes (CSP) specification of this model and the Real Time System (RTS) specification of the refined time delay model of the system, we check for model refinement properties of the system by comparing both models and we verify the following properties of the phone, the network and the GSM system atomic model: deadlock freeness, divergence freeness, non-termination and deterministic behaviors.

VI. RESULTS

After careful probing of the initial model of the specified GSM system in accordance with the refinement technique specified in section IV of this paper and illustrated with the activity and flowchart diagrams in figure 1 and 3 respectively, we got the Real Time System (RTS) refined specification shown in figure 5 below. The RTS specification below showing two specifications of the system: the initial coupled network model (without timing) and the timing delay specification, the latter being the refined specification after careful application of the proposed V&V Simulation Modeling Method proposed above. The defined properties check and refinement comparison is simulated as shown in figure 7 below and verified using the PAT formal tool, verification results are as shown in figures 6 below.

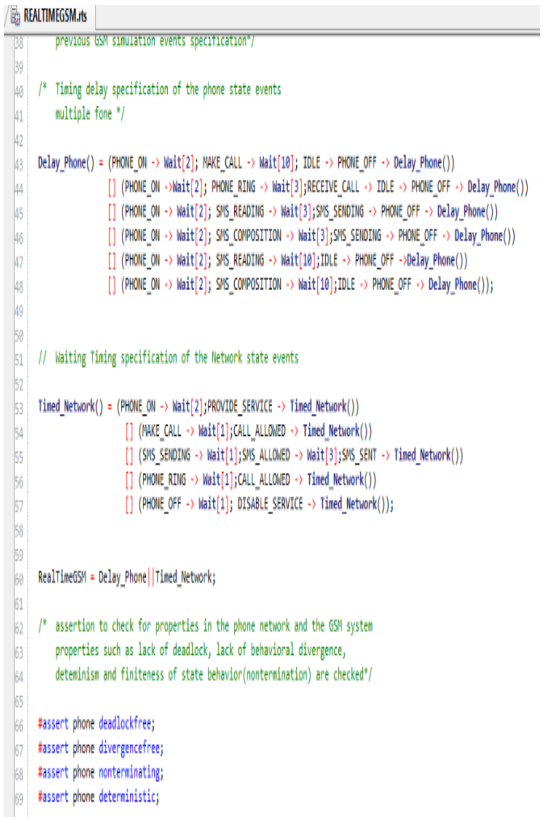

Fig 5: GSM RTS specification Code snippet in PAT

VII. CONCLUSION

The importance of V&V in Modeling and Simulation cannot be over emphasized, it is important to integrate V&V activities with the model development lifecycle in an organized, efficient and well documented manner. This is to ensure accuracy and correctness of Simulation results. The need to have

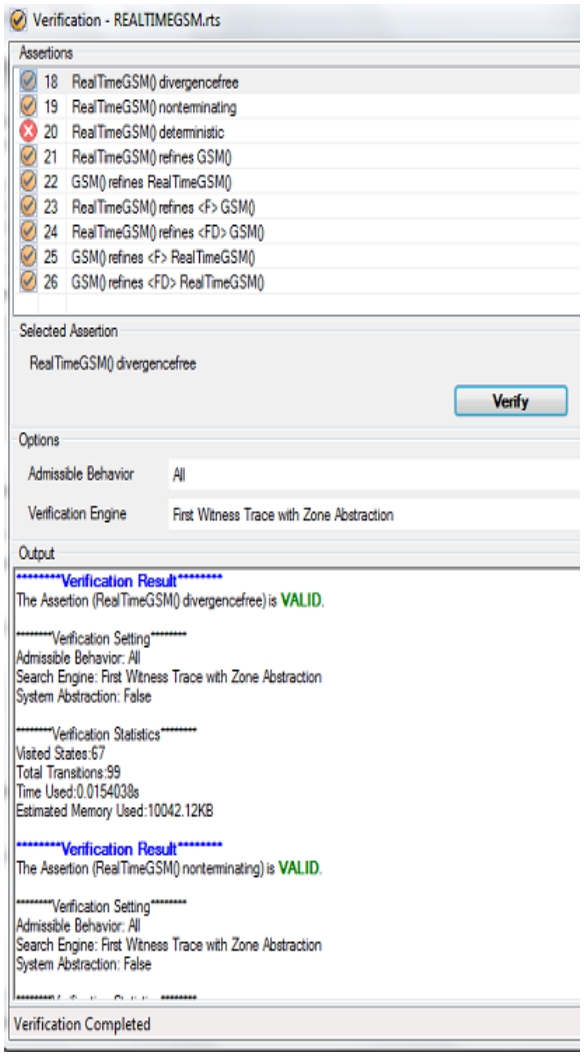

Fig 6: Real Time GSM specification Verification

a well defined technique and a strong fidelity framework to support the model development process has been addressed and developed in this paper, this is to improve the quality of Simulation models and provide a means to effectively check for system behavioral and structural properties, as well as operational constraints of Simulation Models. The importance of V&V in Simulation experiments for both industry and academia are numerous, and ensuring this through the application of formal methods and tools that are easily usable provides an interesting prospect for the V&V of DEVS Simulation Models. Furthermore, work to ensure that the framework can be further extended to perform and ensure model benchmarking and operational performance measurement is in progress.

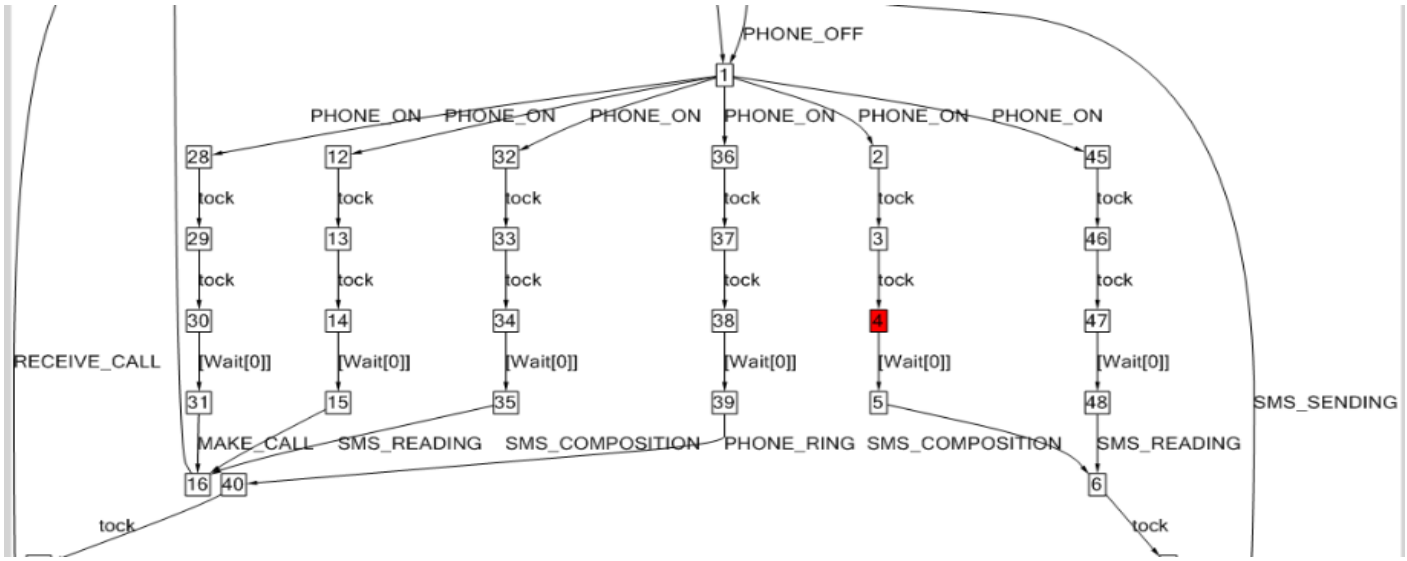

Fig 7: Simulation Graph of the GSM Phone System in PAT

REFERENCES

Axel van Lamsweerde. Formal Specification: a Roadmap. 2000. Proceeding ICSE '00 Proceedings of the Conference on The Future of Software Engineering Pages 147 – 159. ACM New York, NY, USA.

Belinfante, A.F.E. 2010. . JTORX: A Tool for On-line Model Driven Test Derivation and Execution. In: *Tools and Algorithms for the Construction and Analysis of Systems*, 16th International Conference, TACAS '10: 266- 270. Lecture Notes in Computer Science 6015, Springer Verlag.

Christier Baier and Jooster-Pieter Katoen. Principles of Model Checking. Massachusetts Institute of Technology. 2008.

Gabriel A. Wainer. Discrete-Event Modeling And Simulation . Taylor & Francis Group, LLC. 2009.

Garth R. MacKenzie. Verification Technology Potential with Different Modeling and Simulation Development and Implementation Paradigms. Foundations for V&V in the 21st Century Workshop (Foundations '02). The Johns Hopkins University, Laurel, Maryland (USA). October 22-24, 2002.

Ighoroje, U.B., Maiga, O., and Traoré, M.K. 2011. Formal Framework for the DEVS-Driven Modeling Language. *Proceedings for the 23rd European Modeling and Simulation Symposium:* 669- 674. September 12-15, 2011, Rome. (Rome, Italy).

Magee, J., Kramer, J. 2006. *"Concurrency, State Models and Java Programs".* 2nd Edition.

Robert G. Sargent et al Strategic Directions in Verification, Validation and Accreditation Research.. Winter Simulation Conference Proceedings. 2000.

Yan, L., Sun, J., Dong, J.S.. 2007. *Process Analysis Toolkit (PAT) 3.4 User Manual.*1st Edition.

Zeigler, B.P., Praehofer, H., Kim, T.G., 2000. *Theory of Modeling and Simulation*. (2nd Edition. Academic Press. Inc., Orlando, FL, USA).

BIOGRAPHY

Soremekun Ezekiel Olamide is a researcher at the Simulation Network Lab of the African University of Science and Technology, Abuja and an Assistant Lecturer at the Nigerian Turkish Nile University, Abuja (Nigeria). His research focuses on Formal Modeling, Simulation, Formal Methods and its application in telecommunication.

Traore Mamadou Kaba is an Associate Professor in Computer Science at the Blaise Pascal University of Clermont-Ferrand (France). His research focuses on formal specification, symbolic manipulation and automatic code generation of simulation models.