**Visual Modeling and Formal Verification in Telecom**

Soremekun Ezekiel Olamide

African University of Science and Technology, Abuja, Nigeria

[lammyday@gmail.com](mailto:lammyday@gmail.com),

Traoré Mamadou Kaba

LIMOS, CNRS UMR 6158, Université Blaise Pascal, Clermont Ferrand 2, France

[traore@isima.fr](mailto:traore@isima.fr)

**ABSTRACT**

Telecommunication protocols are critical and extensively interactive; a protocol’s process and system interaction requires rigorous verification to assure performance and optimum Quality of Service (QoS). In this paper, we present the study of a GSM call management protocol by applying a visual modeling technique that presents an avenue for behavior simulation and formal analysis of property specifications. We show through visual modeling its dynamic behavior, and the verification of its properties through the application of formal tools.

**Keywords**: Telecommunication, GSM, Visual Modeling, formal tool, Verification

1. **INTRODUCTION**

A communication protocol is the standard sets of rules and requirements that allow electronic components in a telecommunication connection communicate, protocols specify the interactions between the communicating entities [Rouse 2007]. A protocol defines the steps, interactions and rules dictating the subcomponent interactions in a telecommunication system.

The GSM (Global System for Mobile Communication) Call Control management protocol defines the process and technology interaction in Mobile to Mobile Call. This involves the establishment of channel connections on call making request and their disconnection on call end request between two mobile users in a GSM system. This protocol is important for the authentication and ciphering of mobile subscribers, equipment validation, call setup, handovers and call release [RTTC 2008]. The verification of this protocol is therefore highly important in order to provide security, privacy, fault prevention and provision of optimum QoS.

The quality of GSM call control management in Africa has been poor. This is evident in the QoS reports of Communication Commissions in member states, according to the Nigerian Communication Commission (NCC) the Call Set-up success Rate (CSSR) for all operators was estimated to be 97.07 percent, less than the 98 percent target of the commission. Also, the Call Completion Rate (CCR) for all operators was 95.78 percent based on the Commission’s Audit Report for March and April, 2012 [NCC 2012]. In addition, for the financial year 2011/2012, the Communications Commission of Kenya reported 71.88 percent compliance of eight QoS parameters (which includes Completed Calls, Call Drop rate, CCR, CSSR, Call Block rate and speech quality), while the commissions target was benchmarked on 80 percent for the year completed [CCK 2012].This low QoS can be blamed on lack of an efficient Call Control Management.

1. **APPROACH**

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.

Generally, simulation and Formal analysis are the major verification techniques used in the study of communication systems [Baier and Katoen 2008]. Simulation is a paradigm that provides a way of reasoning about problems in order to proffer a solution or a method to absorb relevant data to solve such problems. Process and hardware simulation in communication systems help to study and analyze these systems to gather data and to verify such systems. Formal Verification apply formal process of symbol manipulation and inference according to well defined proof rules of the utilized specification language to study and verify systems [Mackenzie et al., 2002].

We apply visual modeling in the study of this GSM call control protocol, which allows at the same time simulation and formal analysis. The DEVS-Driven Modeling Language (DDML) is the visual modeling paradigm used. This modeling language is based on DEVS (Discrete Event Simulation Specification) [Ighoroje et al., 2011]. We are able to simulate dynamic behavior because the DDML simulation algorithms are those defined by DEVS.

Besides this, we have a modeling framework called SimStudio from which  
we automatically generate the simulation code from the DDML Models and run it for observation and dynamic property checks. SimStudio is an operational framework that must serve to capitalize theoretical advances in Modeling and Simulation (M&S) as well as to gather M&S tools and make them accessible through a web browser [Traoré 2008]. It includes a middleware for the federation of simulators and the collaborative building of simulations. With this toolkit we were able to get the simulatable code of the protocol from its DDML Model and study the simulation trajectories.

Furthermore, the logical semantics of DDML provides a mapping of the model to other formal specification languages in order to perform efficient formal evaluation of static system properties through pragmatic use of formal tools. This logical semantic mapping is shown in figure 2 below and illustrated in section 3 of this paper.

Thus, this approach combines both property evaluation techniques in order to study both the dynamic fine-grained properties of the system as well as the static coarse-grained properties. With the aid of DEVS based Simulation we evaluate dynamic fine-grain properties of the protocol such as: completeness, lack of redundancies, precedence rules and trajectory evaluations. While, the formal verification efforts help to study the static coarse-grained properties such as: safety, progress, lack of deadlock, divergence freeness, determinism and termination.

1. **DDML AND FORMAL METHODS**

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 functional behavior, the dynamic structure and the functional structure [Ighoroje et al., 2011].

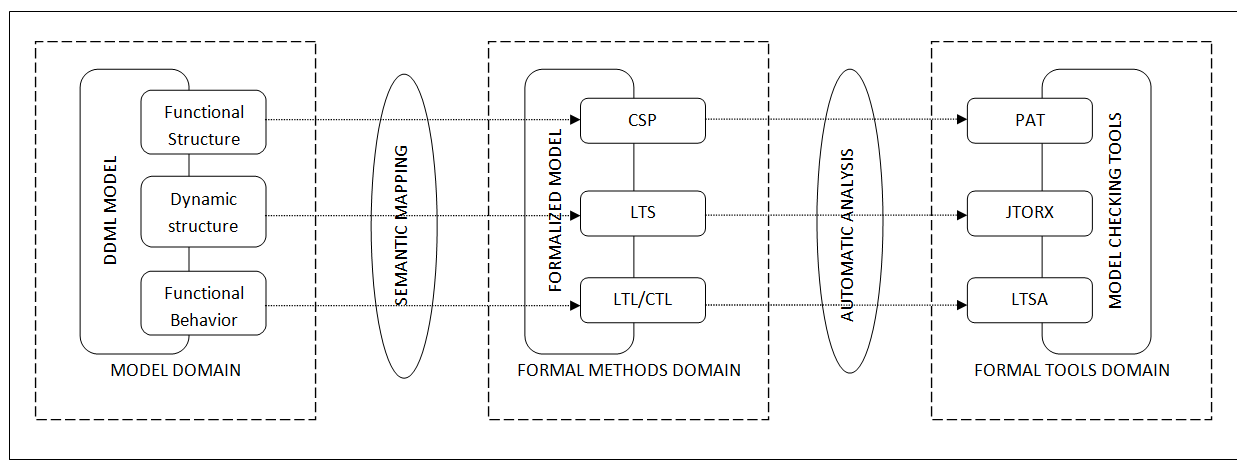
DEVS-Driven Modeling Language (DDML) provides a graphical notation for DEVS. DDML focuses on three hierarchical levels of abstraction. They are:

* The Dynamic Structure is concerned with system dynamic behaviors, especially the change from one state to another and the input stimuli causing such changes as well as the output of the system.
* The Functional Behavior is concerned with traces and trajectories of the system, i.e., the input/output pairs resulted from the dynamic behaviors.
* The Functional Structure is concerned with the structural properties and functional couplings of the system components.

Formally, the DDML logical semantics are mapped into these three levels of abstraction in order to capture the relevant information important at each level. In order to assess system properties at each level, different formal methods adaptive to the specialization of each level are employed to help formally express the DDML framework. In view of this, the different tools that correspond to each level of abstraction and semantic definition would also be applied at each level as shown in figure 1 below and discussed in section 4 below.

This hierarchical system specification at each level is intended to capture the important information of the DDML model. At the dynamic structure level, the DDML semantics are mapped to the labeled transition system (LTS) formal method in order to capture the transition based formal definition of the model. At the functional behavior level, the DDML semantics are mapped to the Linear Temporal Logic (LTL) and the Computation Tree Logic (CTL) semantics in order to capture the temporal logic based formal definition of the modeling language, the functional structure level of DDML is mapped to the Communicating Sequential Process (CSP) formal method in order to capture the process algebraic form of the functional structure model and the interaction of its component parts [Ighoroje, et al., 2011].

Furthermore, with DDML, we can visually represent the functional structure as well as the dynamic structure in the same modeling graph, we can visually represent the system’s functional behavior and dynamic behavior(the latter being the trajectories of system state, the former being the trajectories of system input/output) in the same result graph.



***Figure 1*** *Semantic Mapping of DDML semantics and tool-based amenability to formal analysis*

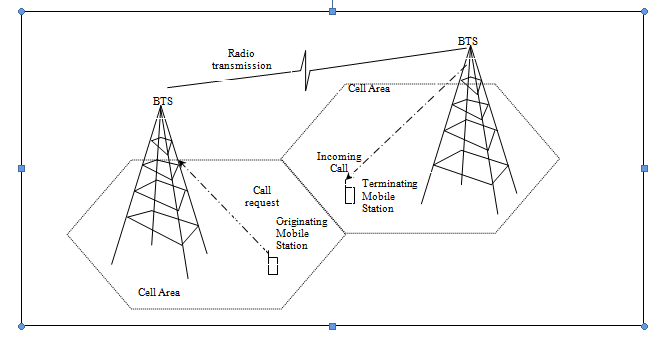
1. **FORMAL TOOLS**

Formal tools present automated reasoning software that take in user specification and properties in a simplified formal language, perform automated analysis and property checks on these specifications, simulate and provide visualization of the specified system processes and return a property check verdict.

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].

**4.1. JTORX:** JTORX was developed and introduced by the University of Twente’s Formal methods (FM) group in the Netherlands in 2000. JTORX is a tool to check whether there exists an IOCO (input output conformance) testing relation between the specification of a system and its implementation [Belinfante 2010]. JTORX checks the behavior of the implementation in relation to the specification and states its correctness judgment (fail or pass conformance). As shown in figure 1 above, the JTORX formal tool would be used to analyze DDML dynamic structure models through their LTS formal translation.

**4.2 Labeled transition State Analyzer (LTSA):** LTSA is formal tool developed by Jeff Kramer and Jeff Magee at the Imperial College London [Magee and Kramer 2006]. The LTSA tool is designed to help modelers formally analyze LTS models (labeled transition systems) written in FSP (finite state processes) to check for safety properties, deadlocks and progress. Furthermore, with the aid of the Fluent LTL specification of the LTSA, some LTL related properties such as mutual exclusion, safety properties, deadlocks and progress are checked. LTSA is used in the course of this work to analyze both the DDML dynamic structure and functional, through their LTS and CTL/LTL semantic mapping as shown in figure 1 above.



***Figure 2*** *Diagrammatic illustration of the Mobile to Mobile Call scenario*

**4.3 Process Analysis Toolkit (PAT):** Process Analysis Toolkit (PAT) is a self-contained tool used to support composing, simulating and reasoning of concurrent, real-time systems and other possible domains; it comes with user friendly interfaces, featured model editor and animated simulator [Yan, et al., 2007]. Process Analysis Toolkit (PAT) is a symbolic model checker tool developed at the National University Singapore (NUS) in 2007. It was designed to check for properties in systems through specifications written in Communicating Sequential Processes (CSP) and assertions written in LTL and CTL to check properties such as divergence, deadlock, reachability and eventuality.

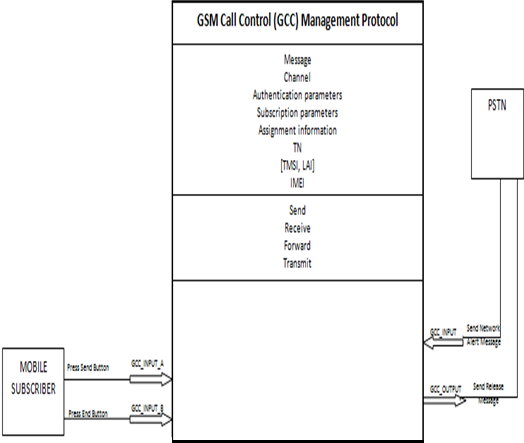
**CASE STUDY**

A GSM call Control Protocol handles the call management of a Mobile to Mobile Call. This involves the interaction between the Mobile Subscribers’ phones and the Network architecture. We model the procedure, its interacting components and the signaling messages involved using DDML, this provides a means to simulate the protocol to study behavioral properties and at the same time perform efficient formal evaluation of its static properties.

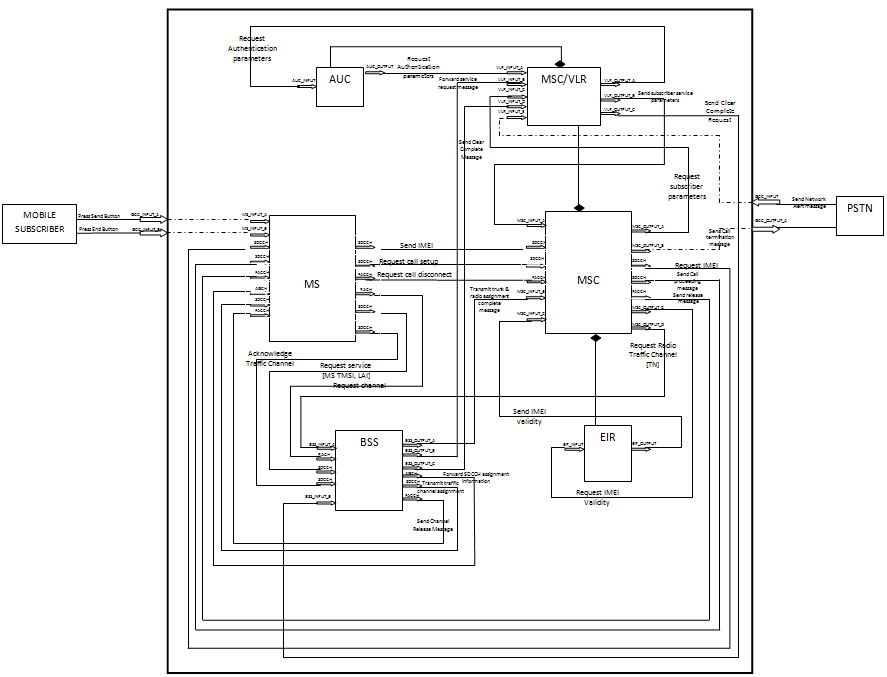
A Mobile subscriber initiates a call request by typing digits and pressing the “send” button. When this call request is done, the GSM call Control procedure performs the following processes: service request, authentication, ciphering, equipment validation, call setup and handovers [RTTC 2008]. Finally, on the mobile subscribers request to end the call by pressing the “end” button, the call release procedure is initiated. The diagram in figure 2 below briefly illustrates the call setup architecture and technologies illustrating the GSM Call Control process sequence.

With the aid of DDML we model the subsystems involved in this procedure and the interaction amongst them, as shown in the functional structure diagram in figure 3 and 4 below. The subsystems involved includes: the Mobile Station (MS), the Mobile Switching Center (MSC), the Base Station Subsystem (BSS), Equipment Identity Register (EIR), Authentication Center (AUC), Visitor Location Register (MSC/VLR) and the Public Switched Telephone Network (PSTN). In addition, the interactions among these subsystems includes: messaging (such as send, request, receive, forward and transmit messages), authentication, validation and functional (such as call release, channel assignment, channel clearance and call setup).

Furthermore, we modeled the GSM Call Control Management System using the visual modeling power of DDML. This provided the foundation to apply the SimStudio Modeling framework, where the DDML Simulation models were automatically converted to DEVS code and simulated to observe both the functional and the dynamic behavior of the system. Thus, DDML enabled us to model the system, it provided the basis for Simulation through the application of SimStudio in order to study the dynamic fine-grained properties of the system. Furthermore, through the DDML Logical semantic mapping, the pragmatic application of formal tools for the formal evaluation of the static coarse-grained properties of the system was achieved.

**

***Figure 3*** *DDML Functional Structure Diagram of the GSM Call Control Management system*

**

***Figure 4*** *Details of the DDML Functional Structure of the GSM Call Control Management system*

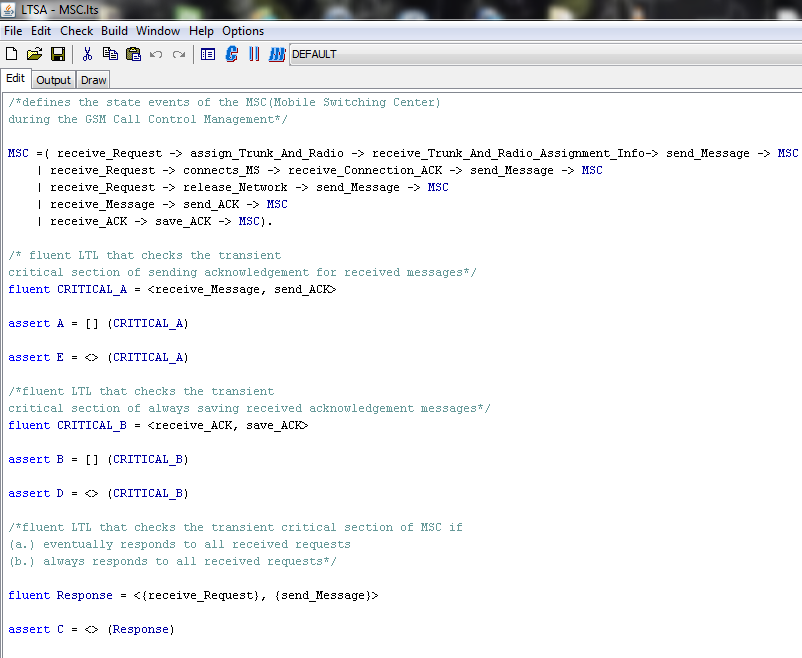
1. **FORMAL ANALYSIS**

Two of the major formal evaluation activities carried out in the course of studying the GSM Call Control (GCC) Management Protocol are discussed in this section, this includes:

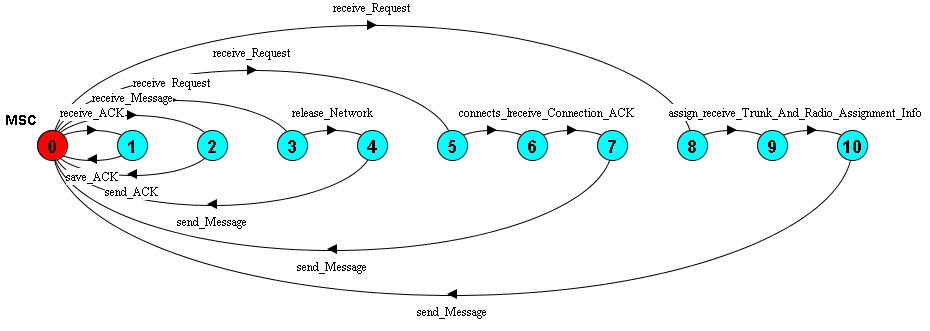
1. Study of the functional behavior and the dynamic structure of the Mobile Switching Center (MSC) in the GCC system using LTSA
2. Study of the Major Channels (Port) management of the Protocol using PAT

**6.1. Functional and Dynamic Structure Verification of the MSC using LTSA**

The dynamic structure of the MSC component is semantically mapped to LTS and defined in the Finite State Process (FSP) specification language (the input language of the LTSA tool). A snapshot of the code and its LTS diagram is shown below in figures 5 and 6 respectively. In addition, this specification passed safety and progress properties check, which can not be showed graphically due to space limitation.



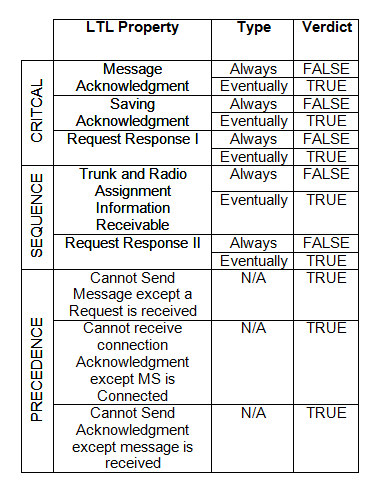
***Figure 5*** *Dynamic structure FSP code in LTSA*

**

***Figure 6*** *LTSA diagram of the dynamic structure in LTSA*

The formal evaluation of the dynamic structure and the functional behavior of the MSC in the GSM Call Control Management protocol are illustrated above in figures 5 and 6, and the verification results obtained are documented in table 1. Consequently, from these results, the following can be deduced from the structure and behavior of the MSC in this system:

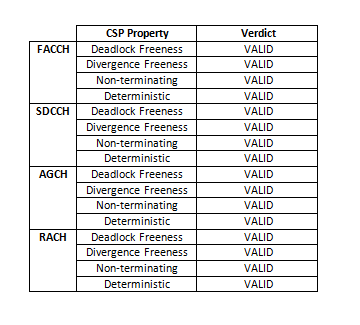
1. All sequential behavior and defined critical behavior of the MSC eventually occurs as shown in the eventuality type verdicts in rows 1-5 of table 1 below.
2. All sequential and critical behaviors of the MSC do not always occur, as shown in the always type verdicts in rows 1-5 below, which implies the likelihood of some of the scheduled sequential and critical events of the MSC such as message acknowledgment, saving acknowledgment and request response are not assured to always take place, especially in case of an interruption.
3. The MSC conforms strictly to important precedence rules that define the order of its events and activities, especially some activities that cannot have to be preceded by some action, as shown in the verification success verdicts shown in rows 6, 7 and 8 of table 1 below.

Thus, from the above deduction from our verification efforts, we recommend that more efforts be made to ensure the occurrence of the system critical sequential processes of the MSC**. ***Table 1*** *Showing MSC Property Checks performed with LTSA Tool (N/A means Not Applicable)*

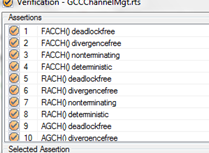
**6.1. Major Channel Verification using PAT**

There are some major channels in the GSM call Control Management system that carry critical messages between the interracting components of the system, such channels include: Stand-Alone Dedicated Control Channel (SDCCH), Access Grant Channel (AGCH), Random Access Channel (RACH) and Fast Associated Control Channel (FACCH). These channels convey important messages, requests and information among communicatiing components such as the MSC, the BSS and the MS. Therefore, it is important to verify the delivery of all received messages, the possibility of a deadlock along message conveyance, the deteministic behavior of the channel interactions and the non-terminating properties of the channels during conveying processes.

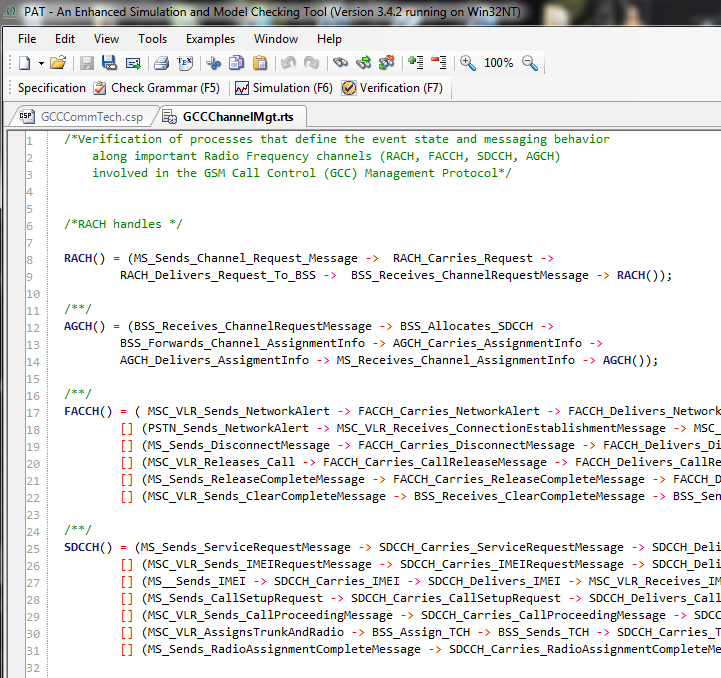
In order to perform the verification of these channel interaction based properties, from the GCC functional structure diagram in DDML from figure 4 above, we semantically mapped the channel interaction process of the GCC protocol into CSP and perform formal verifcation through the application of PAT. A snapshot of the CSP code of this channel interaction is shown in figure 8 below and the success results of the verification exercise is shown in figure 7 and illustrated in table 2 below.

***Table 2*** *Showing Channel Property Checks performed with PAT Tool*

Thus, from the success verification results obtained in figure 7 below and the table results shown above, we can assert that the channel interaction behavior defined in figure 8 is correct and all the channels and the locked synchronization of their functional behavior pass the deadlock freeness, divergence freeness, nonterminating and determintistic properties. Therefore, the channel interaction of the GSM Call Control Protocol ensures the safety of message conveyance and their delivery.

**

***Figure 7*** *PAT results showing the success results of the verification exercise*

**

***Figure 8*** *A snapshot of the CSP code of the GSM Call Control channel interaction in PAT*

1. **CONCLUSION**

DEVS Driven modeling language (DDML) is a visual language for expressing systems in a simple and expressive way. It has provided an intuitive way to understand, specify, study and analyze systems. With its extensive DEVS-based logical semantics, DDML provides a means to simulate and formally analyze systems in order to check for important dynamic behaviors, properties, structure and operations. The combination of DDML and formal methods in the study of systems provides an analytical, mathematical and logic approach to study possible behaviors of a system and inferring important information about such systems.

The importance of verifying Telecommunication protocols cannot be over emphasized. We have provided a means to do this through a visual modeling paradigm that provides property evaluation through simulation and formal analysis. Through the effective visual modeling of the GSM call Control system using DDML we have studied the dynamic behavior and the static structures of this system and verified some important properties. With the aid of the SimStudio toolkit we were able to get the simulatable code of the protocol from its DDML Model and study the simulation trajectories. In addition, the Logical Semantics of DDML and its amenability to formal methods (being easily mapped to formal specification languages and easily analyzed through effective application of formal tools) provides a means to effectively study both fine-grained dynamic properties as well as the coarse-grained static properties of a communication system.

Furthermore, tools applicable to the formal specification languages are consequently used to assess the DDML framework at each level using the GSM Call Control system case study. Properties such as safety, progress, eventuality, reachability, completeness, precedence and lack of deadlock are analyzed through both simulation and formal evaluation.

It is pertinent to note that this approach through the combination of Simulation and formal analysis proffers a foundation to study many dynamic and reactive systems by verifying models of systems to achieve verification, validation and accreditation. This is being strengthened and standardized by integrating the different tools used at each abstraction level in order to provide a single tool that can easily analyze systems automatically.

Integrating the different formal tools provide a plug-in that can easily and automatically analyze any DDML case study on the run and provide significant information about its properties, structure and behavior at all levels of abstraction. This plug-in would be a part of the SimStudio framework, it would form the Automation layerfor translation and interfacing, allowing to map multi-perspective models into the universal formal semantics of the platform, to generate simulation codes and trajectories and, to allow analysis and visualization plug-ins to operate on models, simulators and simulation traces [Traore 2008]. In specific terms, the plug-in tool would be able to easily translate DDML formally to relevant formal specification languages as well as assess the different properties available at each level of abstraction, and it would automatically generate codes from DDML to the different specification languages such as LTS/FSP, Kripke Structure, LTL, CTL and CSP.

**REFERENCES**

Baier, C. and Katoen, J.P. 2008. Principles of Model Checking. MIT.

Belinfante, A.F.E. 2010. “JTORX: A Tool for On-line Model Driven Test Derivation and Execution”. In TACAS’10. Springer Verlag. 266- 270.

Communication Commission of Kenya (CCK), 2012. Report on the QoS Performance Assessment for Cellular Mobile Networks for Period 2011/2012. Nov. 2012. Nairobi, Kenya. <http://www.cck.go.ke/consumers/other_info/downloads/QoS_Report_2011-12.pdf>

Ighoroje, U.B., Maiga, O., and Traoré, M.K. 2011. “Formal Framework for the DEVS-Driven Modeling Language”. In Proc. for the 23rd EMSS. September 12-15, 2011, Rome. 669- 674.

MacKenzie, G.R. 2002. “Verification Technology Potential with Different Modeling and Simulation Development and Implementation Paradigms”. Foundations for V&V in the 21st Century Workshop. Maryland (USA). October 22-24, 24.

Magee, J. and Kramer, J. 2006. “Concurrency, State Models and Java Programs”. 2nd Edition.

Margaret Rouse. April 2007. Network Tutorials and Technical guidelines: Protocol. <http://searchnetworking.techtarget.com/definition/protocol>

Nigerian Communication Commmission (NCC), 2012. QoS KPI Report for GSM Operators, <http://www.ncc.gov.ng/index.php?option=com_docman&task=doc_download&gid=243&Itemid>=

Regional Telecom Training Centre (RTTC). 2008. Training Course Material of ‘BSNL Certified GSM RF Engineer.

Sargent, R.G. 2000. “Strategic Directions in Verification, Validation and Accreditation Research”. In Proc. Of WSC.

Stevenson, D.E. 2003. “From DEVS to Formal Methods: a Categorical Approach”. Proc. of SCSC’03, 1-6, July, Montreal. (Montreal. Canada), 20-24.

Traoré, M.K., 2005. “Combining DEVS and Logic”. Proc. of the 1st OICMS. June 12-15, Clermont-Ferrand. (Clermont-Ferrand, France), 307-317.

Traoré, M.K. 2008. “SimStudio: a Next Generation Modeling and Simulation Framework”. In Proc. of SimuTools’08. Marseille, France, March 3-7.

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

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

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