Distinguishing Automata Machine By Using UPPAAL As A Model Checker

Yogeswaran Nagarathinam, Dr. Nor Fazlida Mohd Sani

Abstract: Upon to the evolvement of technologies, electronic commerce and other online businesses are exposed to vulnerability hence invoking damages and untraceable fraud to the end users. Software engineers in the moment by moment, tracks the design and the analysis so that they can ensure the safety of the overall process from the root itself. Besides that we have proposed model checking to check on the behavior of a design. Thus our research has identified and differentiate the best of two methods of model checking which is Finite State Automata and Non Deterministic Pushdown automata. For the purpose of simulation, UPPAAL tool has been used over a part of Online Shopping system case study.

Index Terms: model checking, electronic payment, Finite State Automata, Non Deterministic Pushdown Automata, UPPAAL, Online Shopping system, possible traces .

1 Introduction

Finite State Automata known by its achievement on simulating the design of a software system. Finite State Automaton has set of states which rely or response on external “inputs”. It is clearly reads sequential from start state to the end state. Finite State Automata can be categorized into two which is ‘Deterministic and Nondeterministic. Deterministic ensure the every state is in control example Deterministic automata travels only one at a time where else for non deterministic automata can be more than one state per time. These are some characteristics of Nondeterministic Finite Automata, a set of state with one start state and few final or end states. An epsilon ∑ for all possible inputs. Allowing more than one possibility states in Nondeterministic. It is said that Finite State Automata act as the most powerful way to implement logic on the applications. The fundamental is proven in many ways example checking the behavior in software design, on lexical analyzer, scanning web pages and verifying system such as transaction, stock market and for the distribution system. However they are some limitation in imposing Finite State Automata such as it can be difficult to manage in a context of a larger system without well designed example “spaghetti factor”. Secondly this process has consume lots of time since some of the design needs to rebuild again and again from scratch. Thirdly Finite State Automata has trouble in dealing with concurrency when running on multiple machine state in parallel. Apart from that, it is discovered that Finite State Automata is weak even to ensure that input string was a form of X^nY^n, a comparative study on Pushdown Automata has been done related to infinite words.

- *Yogeswaran Nagarathinam is currently pursuing masters degree program in software engineering in University Putra Malaysia, Country,Malaysia*
  E-mail: yogiran@hotmail.com
- *Assoc Prof Dr.Nor Fazlida Bt Mohd Sani is currently Associate Professor in faculty computer science at University Putra Malasysia , Country Malaysia.*
  E-mail: fazlida@upm.edu.my

On checking the emptiness likely Pushdown Empty Stack [1] Buchi has developed a pushdown automaton which accept infinite words. Prior to this, pushdown automata used to validate XML documents which are helpful in electronic commerce. In other word applying pushdown automata is to check infinite size of stack and to understand some non regular languages. Even though Deterministic Finite Automata can implement regular expression, Pushdown Automata likely to implement context free grammar. Pushdown Automata can store any information in stack and process it continuously with the information on top of the stack. However nondeterministic pushdown automata able to digest deterministic context free languages. It is able to move on again to the same state with similar inputs. The remainder of the paper is prescribed as follows : - Section 2 would be related work, Case Study will be in Section 3 followed by Simulation in Section 4 and finally Section 5 Conclusion.

2 Related Work

Since Business Process Execution Language for Web Service always with complexity therefore semantically unclear, the researcher proposed Timed Automata [2]. A part of checking the design of particular issue pertaining to electronic commerce, model checking such as Non Deterministic Finite State used to ensure the correctness of web services. It helps to check on the design which any of transaction correlate with a time constraint. As pointed out in [2] Hybrid Timed Automata with the combination UPPAAL model used to check the Secure Electronic Transaction in regards of its capacity to check the correctness at a real time. Time based Automata also used to check in real time and probabilistic behavior in securing electronic transaction [2]. This both applies on Deterministic Finite Automata and also Non Deterministic Finite Automata. Several studies have been carried out to check the correctness of the system and the design by using model checkers. Previously researchers conduct the study of correctness by using UML statechart diagram [3]. The UML Statechart diagram is used because it can determine all possible paths of an object in the entire operation. Deepak [3] has proposed a methodology transforming UML to Finite Automata emphasizing regular grammar. To predict the valid inputs in state diagram regular grammar is highlighted. Besides that the above approach has reduced the effort, costing and also helps eliminate anomalies in software development. In addressing the issue on synthesized web services complexity which highlights on equivalence problem, validation problem and the non emptiness problem [4] the

researcher has proposed upper bounds to the complexity of web services. Besides that researcher identify some restriction allowing decidability and static analysis with aggregation on the case study *travel package* [4]. Afef [5] has defined by generating and executing test cases using Web Services Business Processing Language Composition Conformance Tool WSCCT approach to implement conformance testing on Web Services Business Processing Language. The author used Timed Automata to check on the timing behavior of BPEL. Jocelyn Simmonds et al [6] has introduced a framework for monitoring runtime web services using Nondeterministic Finite Automata. The researcher discusses that static analysis will not be an effective element to check on the properties of web services. Since the function of sequence diagram is used to capture the interaction between the object in Unified Modeling Language UML thus in [6] sequence diagram is chosen because of the ability to capture live and safe properties of web services. The Framework has been applied and been compared to the case study of Online Shopping System, The Travel Booking System and The Loan Application System. However Hamidreza et al [7] has used Finite State Automata on use cases with interactive behavior of the states and furthermore impose security on input, operation and output security check. The researcher introduced ISOAS model Interactive Service Oriented Architecture Security model to highlight the secured and flexible electronic commerce. Apart of electronic commerce, in recent studies composite electronic services has been highlighted by Gerede et al [8]. The researcher has proposed roman model which especially running on finite state automata and focusing on mediator that functioning as delegators. The researcher also introduced Wozart as mediating tool. In Wozart the input will be Desired electronic services, Available electronic services and amount of lookahead meanwhile the output will be successful using k-lookahead (k represents a time polynomial) and failure without using the delegator. The researcher applies Wozart on a case study Travel services [8]. Since the contribution of finite state automata deterministic and nondeterministic also regular expression have provided efficient on electronic commerce protocols rather than other simulation approaches. Pushdown automata is a part of model checkers which handles better checking rather than finite machine. Qi He [9] has address violation as an important issue in business process. The violation is analyzed by artifacts of data and services from business processes. In this paper researcher applies pushdown automata to define the context free lifecycles and decidability for valid artifact. However Alex Thomo et al [10] focuses on extensible mark up language XML, whereby XML from defined sources wraps the data. Furthermore when wrapping is in the process, it tends to wrap diversify information from other sources. Hence researcher defined visibly pushdown automata that supports visibly pushdown languages on approaching XML rewritings. Table 1 shows the summary of review of model checking process in electronic commerce. The summary explains the deterministic finite automata and pushdown automata which has been helping out the software engineer to do research in electronic commerce design.

3 METHODOLOGY

3.1 CASE STUDY

In this case study section is presented in following to explore the verification method. The case study would be a part of the Online Shopping system that behavior are illustrated as such : - The client at the initial place "Homepage" and the system direct into "SelectItem" or second option "Login" . To illustrate further the flow starts from HomePage to SelectItem and directed to AddToCart page here the user can do many times "SelectItem" and "AddToCart" hence it direct to to "CheckOut". Once again here the user can do the transaction many times starting from "SelectItem" to "AddToCart" and then "CheckOut". Hence finally "LogOut" then can choose to go to the initial state "HomePage". There are also another option after the "CheckOut" the user can "Login" to access his own account. However for another option the user from the state of "HomePage" to the "Login" state and then if "LoginSuccessful" user can perform the task simultaneously "SelectItem" to "AddToCart" then to "CheckOut" finally "LogOut". Another Option if the User has perform wrong "Login" the system direct the user to the "LoginFailed". Furthermore if "LoginFailed" three times, the system will "BlockUser". In the below transition diagram model of an abstract online shopping system as given in **Figure 1** , elaborated as a Finite State Machine and defined as a 5 tuple as follows ;

$$\mathbf{M = (S, \sum, s, F, T)}$$

S S = {S0,S1……S8} where the states are:-
S0 – Homepage, S1 – SelectItem, S2 – AddToCart, S3 – CheckOut, S4- Logout, S5-Login,S6 LoginSuccessful, S7-Loginfailed, S8-BlockUser

∑ is set of event whereby the system accept:- ∑ = { e0,e1……e15} where the events are is set of event whereby the system accept, ∑ = { Homepage, SelectItem, AddToCart, LogOut, Login,, LoginSuccessful, Login Failed, BlockUser} where the events are ∑

s start of state system, s is the initial state s = S0 – Homepage

F set of "final" or " accepting states" s = S4 Logout, S7 LoginFailed, S0 Homepage

T is the "transition function", Si+1, and for event ei where
T = f : S x ∑ S

TABLE 1

NO	YEAR	TITLE	PROBLEM STATEMENT	SOLUTION/HYPOTHESIS	METHODOLOGY USED
1	2010	Modeling and Verifying Web Service Applications with time constraints	BPEL4WS with time restriction is not appropriate, to verify the processes happening in web services	This paper present a formal approach to verify time related Web Service applications defined by timed automata using Uppaal tool to simulate and verify the correctness of the system.	case study :- airline reservation system which using time constraint. -Nondeterministic finite state (Timed automata) used to ensure correctness of web services and simulate by Uppaal
2	2012	Semantic for UML Model Transmission and Generation of Regular Grammar	How to verify the correctness of the design diagram and how to detect the anomalies in UML diagram	FSA with generation with regular grammar be used in checking the correctness in UML state diagrams	Finite State Automata with regular grammar
3	2008	Complexity and composition of synthesized web services	Complexity of decision problem and composition synthesis are found on web services	Proposed synthesized web services to uniformly manage characterize FSA and transducer abstract of web services	-case study:- booking travel package -FSA
4	2013	WSST: A Tool for WS-BPEL Compositions Conformance Testing	In execution of web services, BPEL codes become crucial and correctness still becomes as issue	WSCCT tool allows online tracking test execution for correctness BPEL and Time Automata used as underlying formalism	WSCCT and Timed Automata
5	2009	Runtime Monitoring of Web Service Conversations	Web services are dynamic of their properties, so its hard to check the correctness behavior	Introduced a framework for monitoring runtime web services using Nondeterministic Finite Automata	-Framework for runtime monitoring web services -Nondeterministics finite automata
6	2008	A New Approach on Interactive SOA Security Model based on Automata	Security model built into application may not be appropriate when the application exposed as services that used by other applications.	ISOAS model Interactive Service Oriented Architecture Security model to highlight the secured and flexible electronic commerce	-Proposed Security model called ISOAS -Finite state machine -web services
7	2004	Automated Composition of E-services: Lookaheads	Complexity of constructing delegators in e services	General class of delegators called “lookahead” investigate complexity of constructing such delegators if they exist	Wozart, automated mediator

8	2012	Recognizing valid artifacts in business processes	Recognizing violation in business rules which happen in business process	By considering (data)Artifacts and activities (services), identify decidability and undecidability of valid artifacts then present it on pushdown automata	Pushdown automata
9	2008	Rewriting of Visibly Pushdown Languages for XML Data Integration	Problem in rewriting XML data integration since data when its wrap from other source may contain diverse information	visibly pushdown automata that supports visibly pushdown languages on approaching XML rewritings.	Visibly pushdown language and visibly pushdown automata
10	2005	An approach to handle Real Time and Probabilistic behavior in ecommerce: validating the SET Protocol	Model checking on real time behavior and probabilistic way cant be handled by Uppaal tool alone.	Uppaal and Rapture tool allow to check the probabilistic and real-time behavior of security protocols such as Secure Electronic Transaction.	Time based automata Uppaal and Rapture used to verify on SET (Secure Electronic Transaction

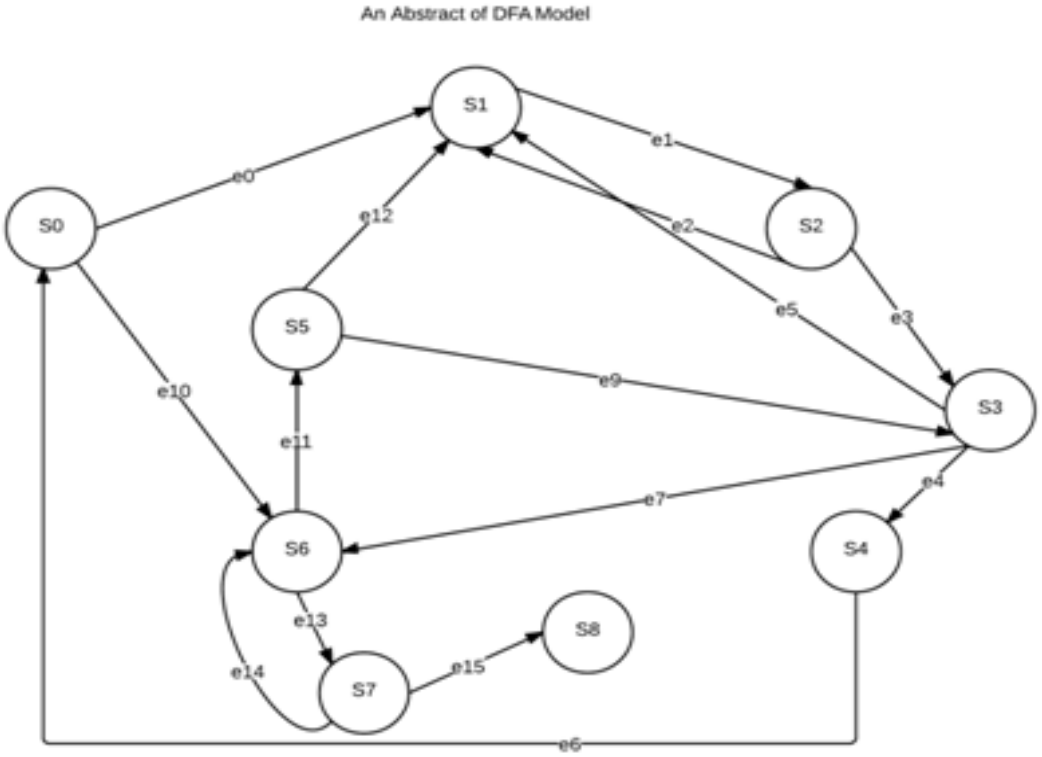


FIGURE 1

TABLE 2

Present State	Next State	Output (Possibilities Traces)
S0 Homepage	- S1,S2,S3,S4,S0 -S6,S7,S8	SelectItem,AddToCart,CheckOut,LogOut,Homepage Login,LoginFailed,LoginFailed,LoginFailed,BlockUser
S1 SelectItem	S2,S1,S2,S1,S2,S3	AddToCart,SelectItem,AddToCart,SelectItem, AddToCart,CheckOut
S2 AddToCart	S3,S4	CheckOut,LogOut
S3 CheckOut	-S4 -S1,S2,S3 -S6,S5,S3	LogOut SelectItem,AddToCart,CheckOut Login,LoginSuccessful,CheckOut
S4 LogOut	-S0	Homepage
S5 LoginSuccessful	-S1,S2 -S3,S4	SelectItem,AddToCart CheckOut,LogOut
S6 Login	-S7,S6,S7,S6, S7,S8 -S5	LoginFailed,Login,LoginFailed,Login,LoginFailed,BlockUser LoginSuccessful
S7 LoginFailed	-S6,S5 -S6,S7,S6,S5 -S6,S7,S6,S7, S6,S7,S8	Login,LoginSuccessful Login,LoginFailed,Login,LoginSuccessful Login,LoginFailed,Login,LoginFailed,Login,LoginFailed,BlockUser
S8 BlockUser	Deadlock	-

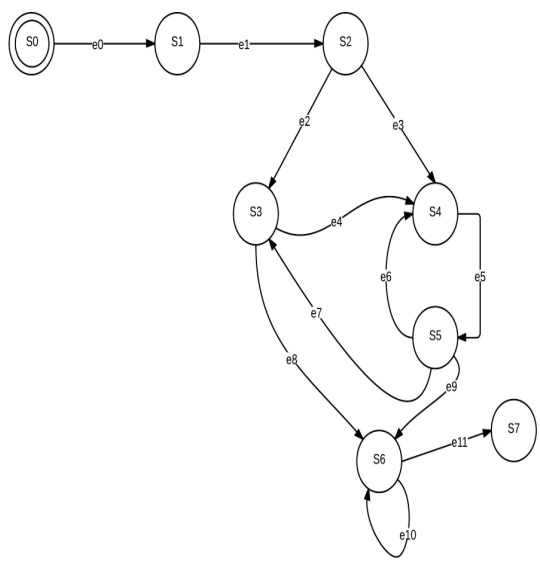
In the above **Table 2** shows that all possibilities traces which is known as accepted inputs were found. These 16 accepting inputs compute from the model which is build shows on **Figure 1.** Furthermore the projection of inputs is depend from the model build by software engineers to determine the correct model which would be used in future electronic commerce. In this automaton the accepting state would be S0 Homepage, which is also initial state, S4 LogOut, S7 LoginFailed and S8 for Deadlock. The reading from Table 2 (possibility traces), shows that the user only have limited boundaries which can read. However the model reject the following path which is from S6(Login) to one time S7 (LoginFailed) to S8 (BlockUser). S6 (Login) to S8 (BlockUser). S1(SelectItem) to S8(BlockUSer). S1(SelectItem) to S6 (Login). S3(CheckOut) to S7 (LoginFailed).

3.2 CASE STUDY OF NON DETERMINISTIC

PUSHDOWN AUTOMATA APPROACH

To distinguishing and compare the DFA automaton model, this section present case study on Non Deterministic Pushdown Automata (NDPDA). From the part of DFA illustration shown on **Figure 1** the NDPA works as such :- The client have two optional Login and Select Item or Select Item and Login. From the Select item client can do multiple transaction Add To Cart and Select Item back and finally Check Out. On the other hand client Login then Select Item and Add To Cart hence Check Out. The client also can Select Item then Add To Cart then Login back. The **Figure 2** below shows how the abstract NDPDA model been developed.

FIGURE 2



A Non Deterministic pushdown automata consist of 7 tuple, whereby:-

M = (Q, ∑,r,T, q0,Z0, F) which we would map as T (x,y,z) = (S,e)

Q is a set of States, here we considered only two states S = {S0….S7}

∑ is set of finite input alphabet event whereby the

system accept, $\sum = (\sum \cup \{\epsilon\})$
1 = User_Login, 2 = Select Item, 3= Add To Cart , 4 = CheckOut

r finite alphabet symbols which is {x,y,z}

T is the "partial transition function", $Q \times r \times \sum \epsilon \rightarrow Q \times r^*$
T(x,y,z) = {(S0,e0),(S1, e1)…….(S7,e11)}

S0 starting or initial states

Z0 in r (gamma) symbol on the pushdown

F contain in K is set of final states

In the below Table 3, the states in NDPDA is defined as 7 Transition which would be:-

TABLE 3

Transition number	State transition	Input,read & pop
From Transition 1	(S0 → S1)	ϵ
From Transition 2	(S1 → S2)	ϵ, ϵ
From Transition 3	(S1 → S2)	ϵ, ϵ → ϵ
	(S2 → S3)	ϵ,x → ϵ 1,x → 1
	(S2 → S4)	ϵ,y → ϵ 2,y → 2
From Transition 4	(S3 → S4)	ϵ,x → 2 2,y → 2
	(S3 → S6)	4, ϵ → 4
From Transition 5	(S4 → S5)	3,y → 3 3,z → 3
	(S5 → S4)	2,y → 2 2,z → 2
From Transition 6	(S5 → S3)	1,z → 1 1,x → 1
	(S5 → S6)	4, ϵ → 4
	(S6 → S3)	4, ϵ → 4
From Transition 7	(S6 → S6)	check pop every input:- 1,x,y → 1, 2,y,z → 2, 3,y,z → 3, 4, ϵ → 4

In this automaton the accepting state would be S5 and S6. S0 is initial state. The reading from **Table 3** (possibility traces), shows example of five accepted inputs as such :-

1) SelectItem&&AddToCart&&SelectItem &&AddToCart &&User_Login && CheckOut.

2) User_Login && SelectItem && AddToCart &&SelectItem && AddToCart && UserNPDA.SelectItem &&AddToCart &&CheckOut.

3) SelectItem && AddToCart &&User_Login && CheckOut.
4) User_Login&&SelectItem && AddToCart && CheckOut
5) SelectItem&&AddToCart&&User_Login&&SelectItem&&AddToCart && CheckOut. However the five rejected possibilities would be :-

1) CheckOut && not.User_Login && AddToCart && User_Login && SelectItem && AddToCart
2) AddToCart && CheckOut && not.User_Login && AddToCart && User_Login && SelectItem.
3) CheckOut && SelectItem && AddToCart && not .User_Login
4) SelectItem && not.AddToCart && User_Login
5) SelectItem && not AddToCart && CheckOut

After considering the languages in NDPDA and the mapping, the next section would be evaluated by UPPAAL simulation.

4 SIMULATION & DISCUSSION

In this paper, we have conduct automatic verification by using the UPPAAL tool. UPPAAL was founded by BRICS at Aalborg University with the Department of Computer Systems at Uppsala University. Apart of other model checker , UPPAAL has more benefits and significance in model checking. The UPPAAL tool consists of components such as variables, inputs , outputs, and the states. UPPAAL uses Java for its GUI and its supports 3 file formats for models: XML, XTA & TA. It uses .q for query and for trace files it uses .xtr. UPPAAL is known for real time simulation and model checking for deterministic and non deterministic process [11]. The significant with real time that UPPAAL relates is having transitions connected to the location. That transitions contains Boolean, integer, variable and importantly clock. Upon verification it provides fault detection which correlates the dynamics of the system. By establishing properties of Online shopping such as in **Figure 1** and **Figure 2**, it then addresses and set to the periodic task of UPPAAL tool. Then the tool will interact with the automata collection and communicate using the channels. The simulation starts when the model developed in the UPPAAL GUI and then some declaration need to be clarify such as Bool and Int with entering the value in UPPAAL Guard, UPPAAL Sync and UPPAAL Update. After establishing this we debug the simulation and check this possibility traces. In the below simulation model **Figure 3** represent Deterministic Finite Automata model and **Figure 4** represent Non Deterministic Pushdown Automata. Furthermore in **Figure 5** and **Figure 6** represent verifier on both Deterministic Finite Automata and Non Deterministic Pushdown Automata. From the studies above, it can be conclude that by using model checker as UPPAAL, that can support concurrency and real time system, properties of each model can be well defined. Furthermore in this paper found that by using deterministic finite automata, it would only read certain limitation of finite inputs. The possible path or inputs could be more as we have to determine every possible steps that can be made. In the results that shown above as in verifier it can read one at per time in one particular transition however by using non-deterministic pushdown automata, the inputs are read and then put into stack before it pops out hence the inputs are read, stored and pop until the first reading which is following the concept of queue.

5 CONCLUSION

As per analyzing the proposed abstract model, it is always to be certain that the early process of design would be adequate for only allowing the valid inputs. Hence the valid inputs can be predicted by finite state automata. However Finite state automata is impossible to be used in the larger context of the design diagram and doesn't have level of accepting

concurrency and non-deterministic process. Therefore to support the concurrency and non-deterministic process, real time based and to identify possible input, this paper suggest non-deterministic pushdown automation. As such implementing varieties of processes which includes the securable one like payment and online transaction, the model that software engineer proposed should be invulnerable to any kinds of exposure or conditions.

FIGURE 3

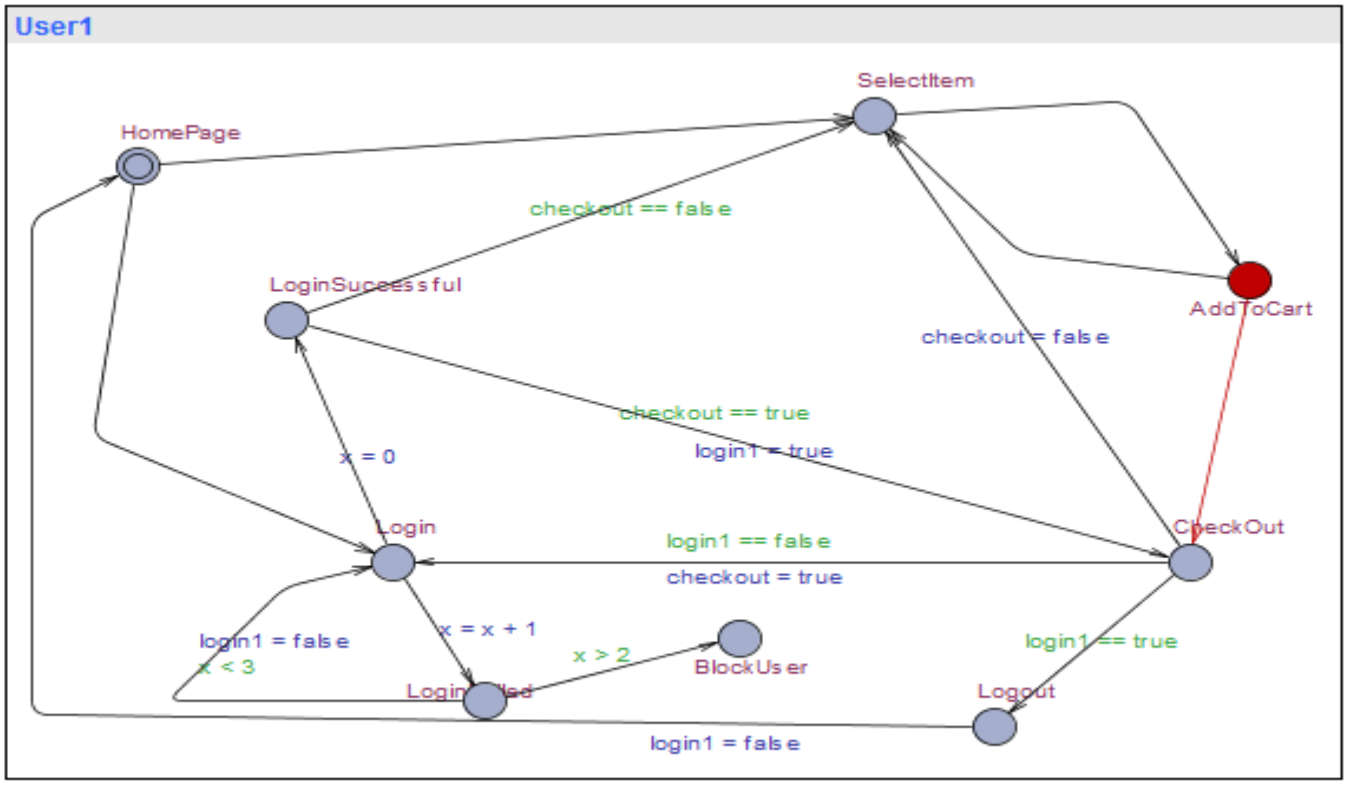


FIGURE 4

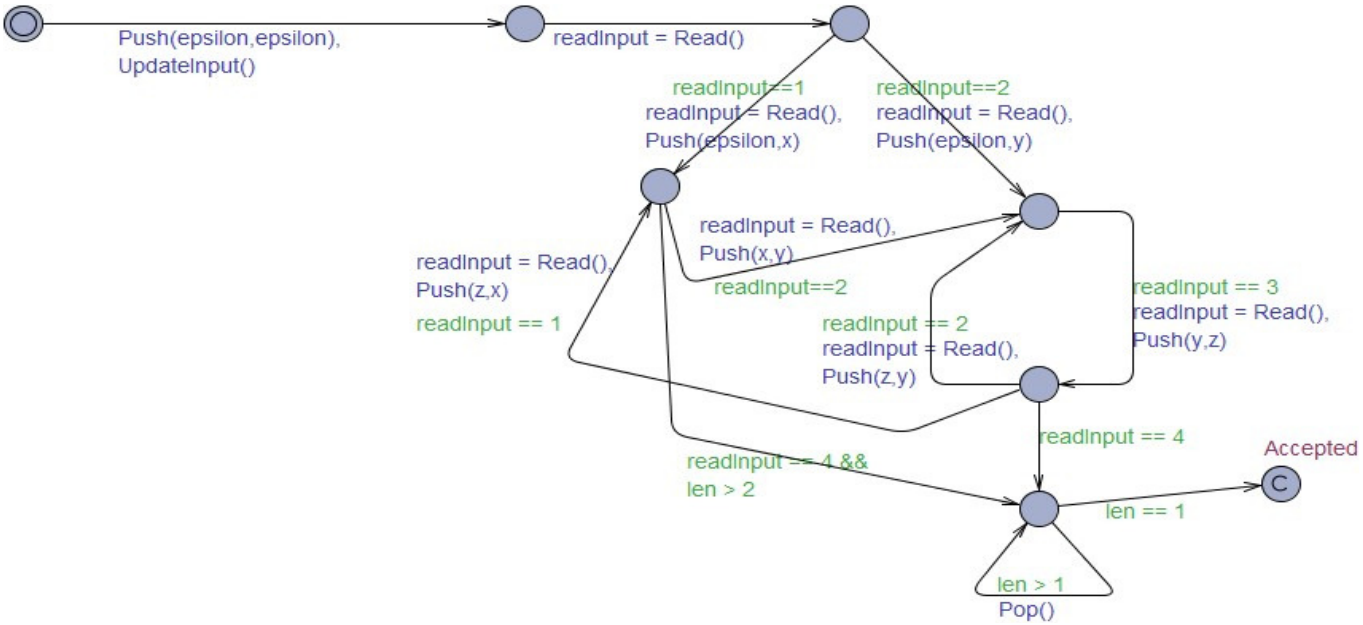


FIGURE 5

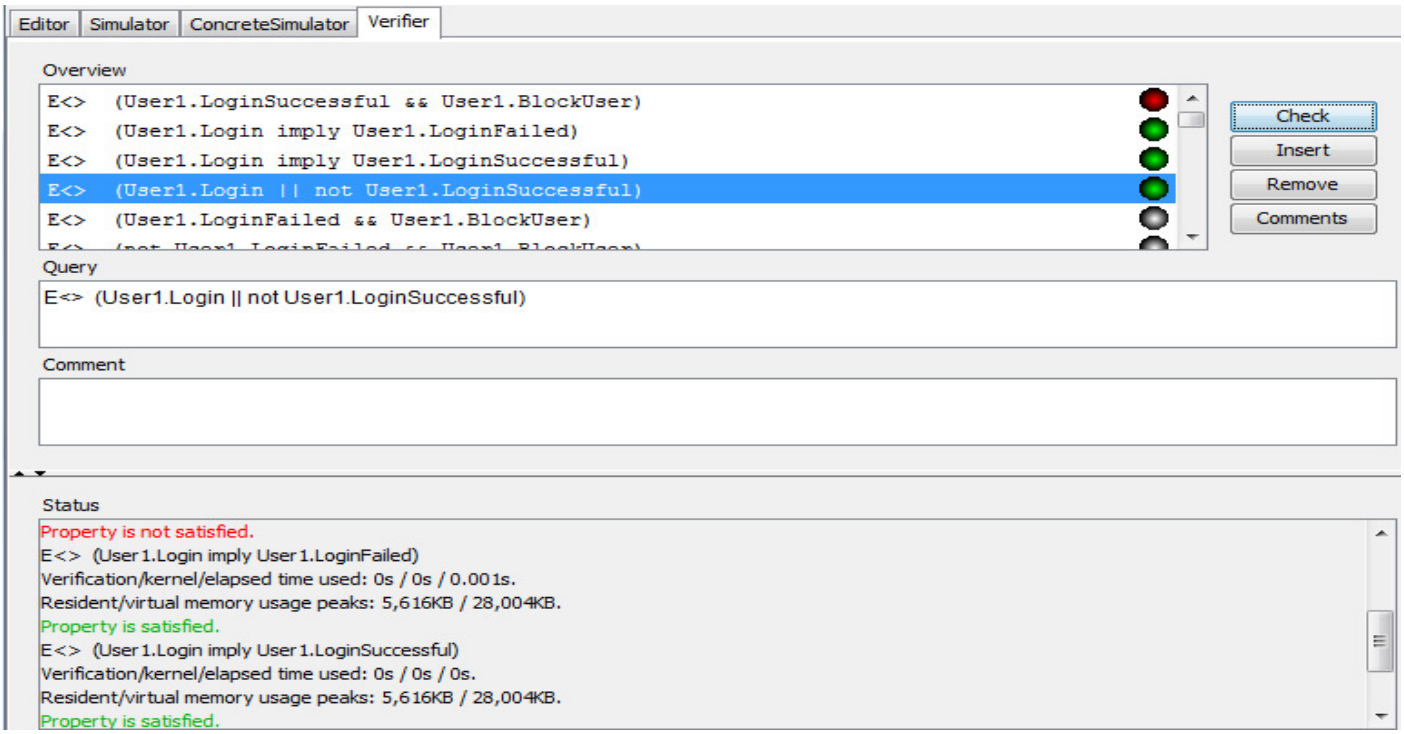
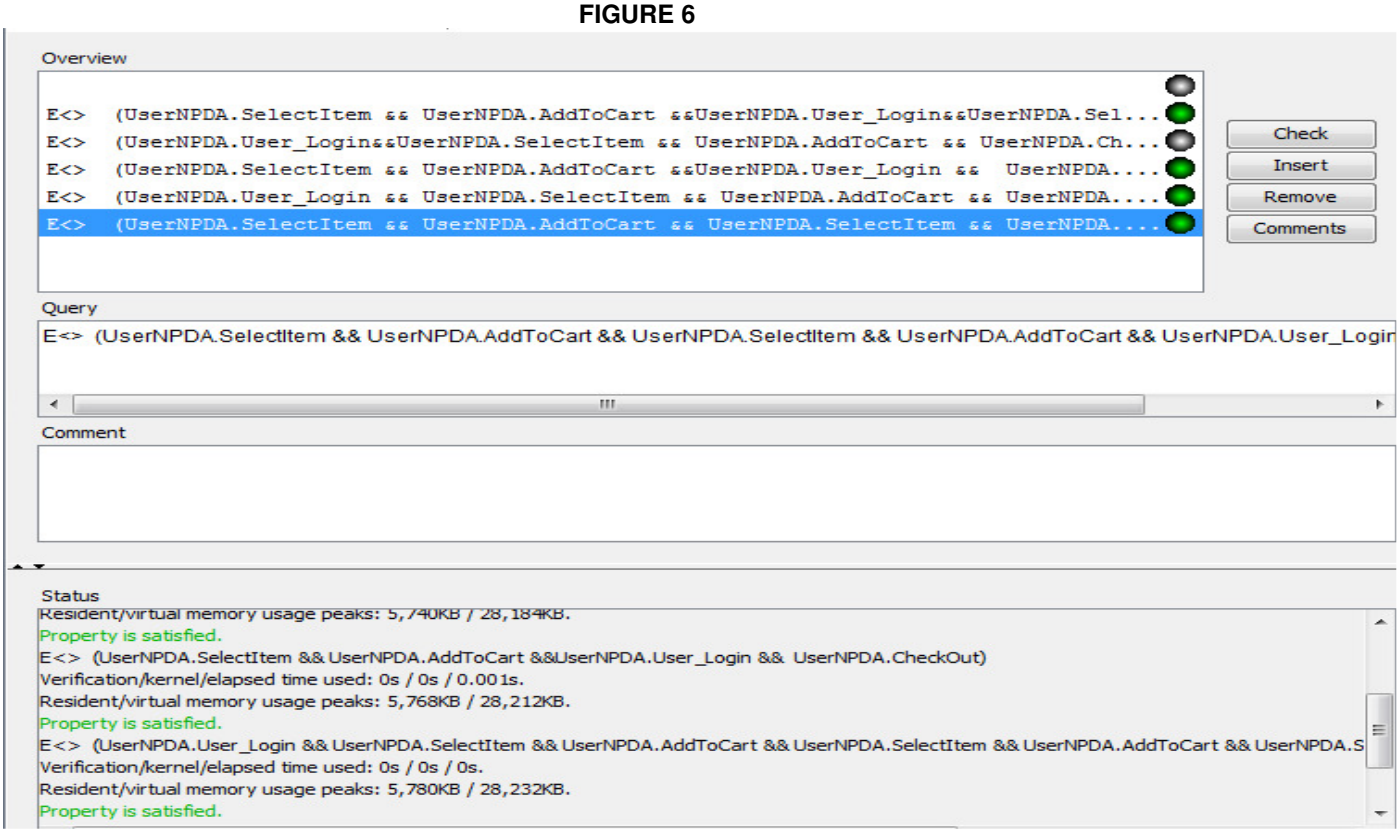


FIGURE 6



ACKNOWLEDGMENT
Special thanks from authors for financial support (Fundamental Research Grant Scheme, FRGS) from the Ministry of Education (MoE), Malaysia via Universiti Putra Malaysia. The principal investigator of this research project is Assoc. Prof. Dr. Nor Fazlida Mohd Sani.

REFERENCES
[1]. Esparza, J., Hansel, D. Rossmanith, P.Schwoon, S.: Efficient Algorithms for model checking Pushdown Systems. In: Emerson, E.A., Sistla, A.P.(eds.) CAV 2000.LNHC, vol.1855, pp. 232-247. Springer, Heidelberg (2000)

[2]. Jia Mei., Huaikou Miao, Qingguo Xu, Pan Liu, : Modeling and Verifying Web Service Applications with time Constraints. School of Computer Engineering and Science and Shanghai Key Laboratory of Computer Software Evaluating & Testing, Shangai, China, 978-0-7695-4147-1/10 IEEE China (2010)

[3]. Deepak Arora., Bramah Hazela, Vipin Saxena, : Semantics for UML Model Transformation and Generation of Regular Grammar. Department of Computer Science & Engineering Amity University, Lucknow, India,http://doi.acm.org/10.1145/180921.2180931 ACM (May 2012)

[4]. Wenfei Fan., Floris Geerts, Wouter Gelade, Frank Neven, Antonella Poggi,: Complexity and Composition of Synthesized Web Services. University of Edinburgh & Bell Labs, PODS'08 , June 9 – 12,2008, Vancouver, BC , Canada ACM (2008)

[5]. Afef Jmal Maalej., Moez Krichen, Mohamed Jmaiel,: WSCCT : A Tool for WS-BPEL Compositions Conformance Testing. ReDCAD Laboratory, University of Sfax Edinburgh & Bell Labs, ACM (March 2013)

[6]. Jocelyn Simmonds., Yuan Gan, Marsha Chechik, Shiva, Bill O'Farell, Elena Litani, Julie Waterhouse,: Runtime Monitoring of Web Service Conversations , IEEE Transaction on Services Computing Vol.2.No.3 (July – Sept 2013)

[7]. Hamidreza Amouzegar.,Shahriar Mohammadi, Mohammad Jaafar Tarokh, Anahita Naghilouye Hidaji,: A New Approach on Interactive SOA Security Model Based on Automata , Seventh IEEE/ACIS International Conference Computer and Information Science, 978-0-7695-3131-1/08 (2008)

[8]. Cagdas Evren Gerede.,Richard Hull, Oscar H.Ibarra, Jianwen Su,: Automated Composition of E-services: Lookaheads, University of California, New York USA, 1-58113-871-7/04/0011 (2004)

[9]. Qi He.,: Recognizing valid artifacts business processes: School of Computer Science , Fudan, China, International Conference on Web Intelligent and intelligent Agent Technology, IEEE 978-0-7695-4880-7/12 (2012)

[10]. Alex Thom., S.Venkatesh: Rewriting of Visibly Pushdown Languages for XML Data Integration: Department of Computer Science, University of Victoria School of Computer Science , Napa Valley, California, USA, ACM 978-1-59593-991-3/08/10 (2008)

[11]. G.D'iaz., K.Larsen, J.Pardo, F.Cuartero, V.Valero,: An Approach to handle Real Time and Probabilistic behaviors in e commerce : Validating the SET Protocol Rewriting of Visibly Pushdown Languages for XML Data Integration: Department of Computer Science, University of CastillaLa, SantaFe, New Mexico, USA, ACM 1581139640 (2005)