Formal Methods and Discrete-Event Simulations

Aznam Yacoub, Maamar El Amine Hamri, Claudia Frydman

Formal Methods and Discrete-Event Simulations

Méthodes Formelles et Simulation à Evénements Discrets

A. Yacoub M. HAMRI
C. Frydman
LSIS UMR 7296
Aix-Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296
13397, Marseille, France
{aznam.yacoub, amine.hamri, claudia.frydman}@lsis.org

Résumé :

Les méthodes formelles (FM) représentent un formidable outil pour la Vérification et la Validation (V&V) de logiciels et de systèmes électroniques, voire de systèmes au sens large, grâce à l'emploi d'un raisonnement logique rigoureux. En particulier, les techniques dites de model-checking explorent de manière exhaustive l'espace d'états de ces systèmes et démontrent leur validité par rapport à certaines spécifications. Toutefois, ces techniques sont inefficaces, et même inapplicables sur des systèmes complexes, tels que les systèmes temporelles ou à événements discrets. A contrario, le domaine de la Modélisation et Simulation (M&S) fournit des outils et des techniques matures pour la V&V de ces systèmes. La mise au point d'un cadre de travail opérationnel combinant FM et M&S semble alors être une approche prometteuse permettant d'améliorer qualitativement la V&V de modèles, et ainsi augmenter la confiance placée aux systèmes étudiés ou conçus. En particulier, nos travaux portent sur la combinaison entre SPIN, un outil reconnu de model-checking, et la simulation à événements discrets pour la V&V de programmes.

Mots-clés :

Méthodes Formelles, Model-Checking, Vérification formelle, Vérification et Validation, Modélisation et Simulation, SPIN, PROMELA, Systèmes à événements discrets.

Abstract:

Formal Methods (FM) are amazing tools for the Verification and Validation (V&V) of software and electronic systems (systems at large). Indeed, these tools use a rigorous logical reasoning. Particularly, model checkers probe the total state space of the verified systems, and they check their validity against specifications. However, these techniques are not efficient on complex systems, such as timed or discrete-event systems. In opposite, the theory of Modeling and Simulation (M&S) provides some powerful methods for the V&V of these systems, by focusing on their semantics. Design a framework combining FM and M&S thus seems to be a good approach to improve the quality of V&V of systems. The confidence put in these verified models is then increased. Especially, our work is about combining a well-known model checker, Simple PROMELA Interperter (SPIN), with Discrete-Event Simulation (DEVS) for the V&V of software and systems.

Keywords:

Formal Methods, Model-Checking, Formal Verification, Verification and Validation, Modeling and Simulation, SPIN, PROMELA, Discrete-event Systems.

1 Introduction

Making reliable software or systems[1] is nowadays becoming more difficult. Systems involve complex behaviours between several components with various specifications. For instance, one of the differences between components typically concerns time representation. When one designs a system or tries to understand it, making a representation of this system is thus needed: this is the modeling process. The question is thus how increasing the confidence put into these models. These last decades, many techniques of design were proposed in the literature in order to answer this question. Verification and Validation (V&V) procedures have been well-defined, and rely on two domains which can appear as opposing: on the one hand, the Formal Verification (FV) methodology groups rigourous mathematical methods which show the correctness of a model by using formal proofs; on the other hand, the Theory of Modeling and Simulation (M&S) provides methodologies and tools allowing accurate representation, verification and validation of systems by focusing on their behaviour. Then, while FV appears as a set of rigorous and exhaustive methods, M&S seems to be similar

[1] We use in this paper the term *system* at large. *System* is thus used to denote software, hardware, algorithms, or any complex systems.

to an empirical experiment, although these domains share many terms and have common objectives. However, FV and M&S both suffer of limits. The proof techniques are limited by the complexity of the systems under study (which can also impact the quality of the design) and by the computational power, while the simulations strongly depend on the played scenarii and on the experimental cases. That is why techniques based on the combined use of FV and M&S have begun to be studied in the literature since few years. This is in this context that we propose in this work a general approach for combining FV and M&S in a same framework, especially by combining model checking tools and discrete-event simulation (DEVS), in order to improve the procedures of V&V. This leads us to define a new formalism, called Discrete-Event Protocol Meta Language (DEv-PROMELA), to represent discrete-event systems (DES). This new formalism allows combining formal verification and simulation-based verification, and is also usable for validation purposes.

The first section of this paper is about the state of the art, explaining why combined methods are interesting in the context of V&V. Next, we introduce the DEv-PROMELA formalism and talk about our contributions, before concluding and presenting the future work.

2 State of the Art

Like said in introduction, V&V is an important process in the development of a system. It increases the reliability that one can put in this system. V&V techniques are various, and based on FV and M&S domains. And if we analyze in depth the literature about FV, M&S and V&V, we can remark these domains are really interlaced. This is thus important to precise definitions before speaking about existing techniques and methodologies.

2.1 Verification and Validation

Verification and Validation are two independent procedures. They are used together to check that a system meets initial requirements and specifications, and does what why it was designed for. The PMBOK guide [Project-Management-Institute2011] gives two standard definitions:

Validation is the assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verification.

Verification is the evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation.

Verification thus answers the question ”Is this system being built right?”, whereas validation answers the question ”Is this the right system which is building?” That meens Validation gives an important information about the correctness of the specifications used to build the system. Among V&V techniques, Formal Verification, Test Cases and Simulation are widely used.

The M&S community provides two others, but close, definitions of V&V [Missile-Defense-Agency2008].

M&S Verification is the process of determining that a computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.

M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

In that sense, verification of simulation models concerns the correctness of the computerized simulation model against a conceptual model (using FV techniques), whereas validation of simulation models gives informations about the accuracy of the representation of the real system. Sargent goes further by precising the relationship between real world and simulation world [Sargent2004]. The conceptual model is thereby obtained by modeling theories and assumptions about the system under study. This conceptual model specifies the simulation model specifications, which are then used to implement the computerized simulation model. M&S Verification is thus done at two levels: between the conceptual model and the simulation model specifications on the one hand; and between the simulation model and the specifications on the other hand.

It is important to keep in mind these definitions. Indeed, our objective is not to improve V&V of simulation models, which is well studied in the the literature, but V&V at large.

2.2 Introduction to Formal Verification

Formal verification is the act of proving or disproving the correctness of a system against properties and specifications, using formal methods. Formal methods (FM) are a set of formal notation and tools that allow a strict and rigourous description of the system under study, with formal semantics and an automatic proof mechanism [Bowen and Hinchey1995].

FM are divided into two families:

- Automated theorem proving methods show that a set of statements of a system can be deducted from another set of statements. Formally, we consider Γ, a set of logical properties describing the system (we called them axioms and hypothesis), and ϕ a set of specifications (that we called conjectures). Theorem proving methods try to find a proof that $\Gamma \vdash \phi$, in other words, that we can syntactically deduce specifications from properties of the system.

- Model Checking methods [Huth and Ryan2005] [Baier and Katoen2008] show that a system satisfies a set of properties. Formally, we consider M, a model (in the mathematical sense) of the system, and ϕ, a set of logical properties. Model Checking methods check whether $M \models \phi$: all models M syntactically and semantically satisfy ϕ. In fact, because the system is generally modelled by a finite automaton, model checking tools systematically explore the entire state space of the system model, inducting to the well-known state space explosion problem, which is extensively treated in the literature [Clarke2008].

In practice, software formal verification with formal methods can be done in several manners. First of them is modeling software using a formal specification language, then translating this model into code. Another fashion consists of extracting the model from the code and doing the verification on the resulting model [Holzmann and Smith1999].

According to these definitions, one can easily understand why FM are considered as powerful methods. But, FM are facing heaviness and are not really applicable on large and complex systems [Heitmeyer1998]. FM indeed require strong assumptions and strong abstractions in order to be efficient. That is why formal methods impose strong restrictions (that involve finiteness for instance) to their modeling language. Their expressive capability is thus reduced. As a result, many systems, like timed or event-based systems, can't be verified against temporal properties.

2.3 Introduction to M&S and Discrete-Event Simulation

M&S domain has been explored since the early 1960s, but was really theorised by [Zeigler1976]. This theory tried to make uniform these two notions used extensively in many disciplines like medicine, physics, etc; it also defines a global and universal framework and methodology that is not dependent on the domain of application. As the name suggests, the two key concepts behind M&S are "Model" and "Simulation". A model is a semantic interpretation of a structure, while a simulation is "executing a model to generate its behaviour" [Zeigler et al.2000], by acting on inputs and parameters of the model. Zeigler defined also an Experimental Frame (EF) as a set of conditions under which the real system is observed. This notion is also important because it implies a certain level of abstraction.

However, Zeigler also introduced an unique and universal formalism to describe discrete-event system in a generic manner. Discrete-Event system Specifications (DEVS) formalism is a symbolic representation of system semantics, unlike syntactic formalisms used in FV approaches. This involves that DEVS models are focusing on the behaviour of the systems which they represent, without any other constraints. Models can also be interpreted in only one way, which is not necessarily the case in FV approaches.

The main problem with the M&S framework is that simulation strongly depends on the EF. Because the model is a result of the view of the real system from the point of view of the EF, the correctness of the simulation essentially comes from the accuracy of the assumptions made under the EF. As a result, simulation-based verification methods cannot guarantee values outside of the domain of the EF.

2.4 Complementarity between Simulation and Formal Methods

Complementarity between Simulation and Formal Methods has been already shown in the literature. The FV community agrees that "In order to improve the quality of the model, a simulation prior to the model checking can take place. Simulation can be used effectively to get rid of the simpler category of modeling errors." [Baier and Katoen2008]. Like viewed in the previous section, M&S Verification is also explored in order to increase the credibility of simulation models [Sargent1998] [Sargent2001] [Kuhn et al.2003]. Many methods to transform certain DEVS subclasses into Timed Automata for purpose of static verification were developed [Dacharry and Giambiasi2007] [Saadawi and Wainer2009]. Other approaches tend to integrate Z into DEVS models [Trojet et al.2009] [Trojet2010] by transformation.

But other approaches [Abdulhameed et al.2014] [Li et al.2005] try to combine formal verification and simulation by deriving specifications into two different specification languages, for simulation and formal verification purposes. However, these approaches don't take into account the heterogeneous aspects of systems.

3 Contributions

3.1 General Approach for V&V using Combined Formal Methods and Simulation

The main contribution of our work is to provide a new generic framework for V&V of systems, and which combines the use of formal methods and simulation. This framework is based on the M&S and DEVS formalism defined by Zeigler. It consists into introducing a clear operationnal semantics in formalisms used by formal methods. That then allows combining FV and simulation by transforming specifications

expressed in the new formalism into specifications expressed in a verifiable formalism on the one hand, and into specifications expressed in a simulable formalism in the other hand. Four steps can be summarized in:

1. Firstly, it is necessary to make a rigorous description of the formalisms in order to understand concepts and notions involved. We call *source formalism* f_s, the formalism used by a formal method (typically a model checker). *Target formalism* f_t is a chosen formalism of simulation. For a given couple (f_s, f_t), we determine a metamodel of each formalism. This allows establishing a relation between the concepts of each formalism and identifying missing notions, which can be related to the semantics employed. It is also in this step that we can easily define a transformation language between the two formalisms;

2. If a notion is missing in the metamodel of f_s, we construct an *extended source formalism* $f_{s'}$ with these concepts. In this way, we can easily add operational semantics to a formalism which has none. We then define the transformation language from $f_{s'}$ to f_t. Note that defining a new formalism can lead to redefining the grammar of the language of f_s; defining the transformation language is also defining morphisms between models at each level of system specification as given by [Zeigler et al.2000];

3. We redefine the system model M_s to transform f_s into $f_{s'}$ if needed. Let us call $M_{s'}$ the system model in $f_{s'}$.

4. We apply transformation to the model $M_{s'}$ and get the model M_t. Simulation-based verification and validation is then performed on M_t. If transformation is correct, M_t must respect requirements checked by M_s with the formal method.

Using the modularity properties of the M&S theory, this framework allows us

modular specifications and enables a kind of interoperability. By this, we mean that a system can be splitted into small components which can be verified and validated alone; each component is then composed with the others in order to model the entire system; this one is then validated by simulation. Modularity also allows replacement of each subcomponent without breaking the global behaviour of the system. Components can consequently modeled at different level of abstraction.

First contributions of this framework were validated through the transformation of specifications from PROMELA to FDDEVS and PROMELA to TSM [Yacoub et al.2014a] [Yacoub et al.2014b]. Recall that the given framework was successfully applied to verify and validate, by formal verification and simulation, the specifications of a commercial soccer video game.

3.2 DEv-PROMELA, a Formalism for Discrete-Event Modeling and Verification

As a part of a demonstration of our framework, we are currently developing a new formalism based on the well-known Protocol Meta Language (PROMELA) and its model checker SPIN [Holzmann2003], initally designed for the verification of concurrent protocols. Discrete-Event PROMELA (DEv-PROMELA) allows modeling of discrete-event algorithms by introducing discrete-event concepts into PROMELA. The resulting specifications are more accurate for the representation of discrete-event algorithms and combine the advantages of both DEVS and PROMELA formalisms. DEv-PROMELA indeed provides a clear operationnal semantics and a clear syntax that allows simulation and formal verification. A DEv-PROMELA model is then translated into a PROMELA specification, which preserve structural properties, for formal verification

purposes using the SPIN model checker. And, the DEv-PROMELA specifications are also translated into a DEVS conceptual model, that preserves behavioural properties, and which can be simulated for verification and validation of behaviour the initial model.

4 Conclusion and Future Work

V&V approach combining Formal Methods and Simulation is a promising approach increasing confidence put into models and systems. This approach fills the weakness of formal verification on the one hand, and simulation one the other hand, by relying on the strengths of both domains. Also, modularity proposed by the M&S Framework allows incremental development of models.

Future work is about the adaptation of formal verification algorithms to DEv-PROMELA and making an integrated environment for modeling using the DEv-PROMELA formalism.

Acknowledgements

The authors want to gratefully thank the Process Control System Team in Rousset. This research is part of the RD project "MAGE", from French Investing for the Future national program.

References

[Abdulhameed et al.2014] ABDULHAMEED, A., HAMMAD, A., MOUNTASSIR, H., AND TATIBOUËT, B. 2014. An approach combining simulation and verification for SysML using SystemC and Uppaal. In *CAL 2014, 8ème conférence francophone sur les architectures logicielles*. Paris, France, 9 pages.

[Baier and Katoen2008] BAIER, C. AND KATOEN, J. 2008. *Principles of Model Checking*. MIT Press.

[Bowen and Hinchey1995] BOWEN, J. P. AND HINCHEY, M. G. 1995. Applications of formal methods.

[Clarke2008] CLARKE, E. 2008. The birth of model checking. In *25 Years of Model Checking*, O. Grumberg and H. Veith, Eds. Lecture Notes in Computer Science Series, vol. 5000. Springer Berlin Heidelberg, 1–26.

[Dacharry and Giambiasi2007] DACHARRY, H. P. AND GIAMBIASI, N. 2007. A formal verification approach for devs. In *Proceedings of the 2007 Summer Computer Simulation Conference*. SCSC '07. Society for Computer Simulation International, San Diego, CA, USA, 312–319.

[Heitmeyer1998] HEITMEYER, C. 1998. On the need for practical formal methods. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*. Springer, 18–26.

[Holzmann2003] HOLZMANN, G. 2003. *Spin Model Checker, the: Primer and Reference Manual* First Ed. Addison-Wesley Professional.

[Holzmann and Smith1999] HOLZMANN, G. AND SMITH, M. 1999. Software model checking. In *Formal Methods for Protocol Engineering and Distributed Systems*, J. Wu, S. Chanson, and Q. Gao, Eds. IFIP Advances in Information and Communication Technology Series, vol. 28. Springer US, 481–497.

[Huth and Ryan2005] HUTH, M. AND RYAN, M. 2005. *Logic in computer science: modelling and reasoning about systems*. Cambridge University Press.

[Kuhn et al.2003] KUHN, D. R., CRAIGEN, D., AND SAALTINK, M. 2003. Practical application of formal methods in modeling and simulation. In *Proceedings of SCSC'03, Summer Simulation Multiconference*. Citeseer.

[Li et al.2005] LI, L., SZYGENDA, S. A., AND THORNTON, M. A. 2005. Combining simulation and formal verification for integrated circuit design validation. In *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*. 92–97.

[Missile-Defense-Agency2008] MISSILE-DEFENSE-AGENCY. 2008. Department of defense documentation of verification, validation & accreditation (vv&a) for models and simulations.

[Project-Management-Institute2011] PROJECT-MANAGEMENT-INSTITUTE. 2011. Ieee guide–adoption of the project management institute (pmi(r)) standard a guide to the project management body of knowledge (pmbok(r) guide)–fourth edition. *IEEE Std 1490-2011*, 1–508.

[Saadawi and Wainer2009] SAADAWI, H. AND WAINER, G. 2009. Verification of real-time devs models. In *Proceedings of the 2009 Spring Simulation Multiconference*. Society for Computer Simulation International, 143.

[Sargent2001] SARGENT, R. 2001. Some approaches and paradigms for verifying and validating simulation models. In *Simulation Conference, 2001. Proceedings of the Winter*. Vol. 1. 106–114 vol.1.

[Sargent2004] SARGENT, R. 2004. Validation and verification of simulation models. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*. Vol. 1. –28.

[Sargent1998] SARGENT, R. G. 1998. Verification and validation of simulation models. In *Proceedings of the 30th conference on Winter simulation*. IEEE Computer Society Press, 121–130.

[Trojet2010] TROJET, M. 2010. Approche de vérification formelle des modèles devs à base du langage z. Ph.D. thesis.

[Trojet et al.2009] TROJET, M. W., FRYDMAN, C., AND HAMRI, M. E.-A. 2009. Practical application of lightweight z in devs framework. In *Proceedings of the 2009 Spring Simulation Multiconference*. Society for Computer Simulation International, 154.

[Yacoub et al.2014a] YACOUB, A., HAMRI, M., AND FRYDMAN, C. 2014a. Complementarity between simulation and formal verification - transformation of promela models into fddevs models: Application to a case study. In *4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2014*. 421 – 426.

[Yacoub et al.2014b] YACOUB, A., HAMRI, M., AND FRYDMAN, C. 2014b. A method for improving the verification and validation of systems by the combined use of simulation and formal methods. In *IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2014*. 155–162.

[Zeigler1976] ZEIGLER, B. P. 1976. *Theory of Modeling and Simulation*. John Wiley.

[Zeigler et al.2000] ZEIGLER, B. P., KIM, T. G., AND PRAEHOFER, H. 2000. *Theory of Modeling and Simulation* 2nd Ed. Academic Press, Inc., Orlando, FL, USA.