



# Simulation Fidelity Distance: A Game-Theoretic Framework

Sangeeth Saagar Ponnusamy, Vincent Albert, Patrice Thebault

► **To cite this version:**

Sangeeth Saagar Ponnusamy, Vincent Albert, Patrice Thebault. Simulation Fidelity Distance: A Game-Theoretic Framework. SpringSim Spring Simulation Conference, Apr 2016, Pasadena, United States. 10.23919/TMS.2016.7918815 . hal-01912556

**HAL Id: hal-01912556**

**<https://hal.laas.fr/hal-01912556>**

Submitted on 5 Nov 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Simulation Fidelity Distance: A Game-Theoretic Framework

**Sangeeth saagar Ponnusamy**

Airbus Operations SAS,  
316 Route de Bayonne,  
Toulouse-31060, France  
sangeeth-saagar.ponnusamy@airbus.com

**Vincent Albert**

CNRS, LAAS,  
7 Avenue du Colonel Roche,  
31400, Toulouse, France  
valbert@laas.fr

**Patrice Thebault**

Airbus Operations SAS,  
316 Route de Bayonne,  
Toulouse-31060, France  
patrice.thebault@airbus.com

## ABSTRACT

The paper addresses one of the fundamental questions in using simulation as a means for system verification and validation, namely, how far the simulation model represents the real system according to the given test objective. Extending studies on quantitative approaches in system refinement based on two player games to the field of system simulation, distance notions for simulation fidelity are proposed. This fidelity distance could be quantified through alternating simulation games between the system model and the simulation model. This coverage metric initially proposed in literature, is still an absolute distance measure and inadequate for the purpose of determining the fitness of a model for an intended use, since a game relation is established only between the simulation model i.e. implementation and system model i.e. specification. In addition, all strategies of the players must be explored to quantify this error not only globally but also with respect to the given test scenario. In this paper, these games between two untimed transition systems are modeled as DEVS in the ProDEVS tool and a quantitative reachability graph is generated using the TINA tool to explore all such player strategies in the game. Further, a relative-weighted fidelity distance metric to account for given test objectives is proposed which penalize cheats on expected simulation model behavior more than the other behavior. This quantitative reachability graph is analyzed for consistency conditions, player strategies, counter examples, distribution and evolution of cheats to gain further insight into the simulation model behavior with respect to the system model and test objectives. These game notions are discussed in the context of derivability of experimental frame formalism, where a global test scenario is proposed and then consistently refined with respect to the model development cycle. Extending this quantitative approach to timed systems and interface automata is briefly discussed.

## Author Keywords

Simulation, Verification & Validation, Alternating Games, Quantitative Reachability.

## ACM Classification Keywords

I.6.1 Simulation Theory, I.6.4 Model Validation and Analysis, I.6.5 Model Development.

## 1. INTRODUCTION

In the Modeling & Simulation (M&S) of complex systems, especially for the purpose of Verification and Validation (V&V), one of the fundamental questions in using models to represent a dynamic system is how closely does the model simulate i.e. ‘mimic’ the system behavior?. Simulation or Model Fidelity, also called representativity or faithfulness, is this ability of a model to do whatever the system it intends to represent does. In other words, under similar environmental assumptions i.e. inputs, a model with higher fidelity reproduces as many behaviors as the system. This fidelity could be interpreted as a distance to reality and this paper proposes a mechanism to quantify this distance with respect to all possible or a subset of behaviors of the system based on the notions of game theory and formal verification principles.

The paper is structured as follows, in section 1, M&S for system V&V is briefly outlined through experimental frame formalism. In section 2, state of the art quantitative formal verification notions based on game theory are presented and discussed in the context of simulation fidelity metric. This fidelity distance quantification is then presented according to the given V&V objectives in section 3. A tool to model an untimed transition system and generate the quantitative reachability graph is presented in section 4 followed by an application case in section 6. The paper concludes with a brief overview of ongoing and future work.

## 2. M&S IN SYSTEM V&V

M&S has been widely used as a means to perform V&V activities on the systems. In a classical industrial environment, a system and its representation i.e. simulation model are often developed by different stakeholders with different objectives. These System(s) Under Test (SUT), along with other systems or their models, are then tested at different V&V scenarios by a test team. System designers, who design and develop systems, are often domain experts but do not necessarily have a multi-system end user perspective. On the other hand, testers or the simulation users are not domain experts but know the context under which such SUT will be used. Then, the challenge for the

model developer, who is usually in between these two stakeholders, is how to develop models of the systems called *simulation model* in the context of system V&V. It may be noted that system designers may develop their own model of the system called design model or simply a *system model* and iteratively refine them before developing a real system. Our study does not concern the fidelity of this design model but the simulation model i.e. a subset of design model for the purpose of V&V. In multi system V&V such design models may not be used directly even when available due to practical limitations such as different modeling formalisms, model complexity, challenges in integration, test platform requirements etc. This necessitates a component based design approach for developing a simpler representation of the constituent systems wherein each component must be adequately representative enough to perform V&V on the SUT. However, quantifying fidelity is often a challenging task since it requires real system behavior to compare against the model behavior. This post-priori measurement of fidelity happens often at detrimental cost due to over or under specification of models. Instead, this fidelity needs to be measured a priori both globally and locally i.e. with respect to V&V objectives. In the next section, Experimental Frame (EF), a lucid formalism to represent these V&V scenarios is presented and the problem of quantifying the fidelity of the EF components is discussed.

### 2.1. EXPERIMENTAL FRAME

In the context of studying a system through simulation, an EF describes scenarios under which a model is stimulated, observed and validated according to the requirements. Initially proposed by Ziegler [11], an EF is composed of stimulant (Generator, G), observer (Transducer, T) and assessor (Acceptor, A) components in addition to the components simulating the environment (ENV). The EF is denoted by  $EF = \{M_G \cup M_T \cup M_{env} \cup M_A\}$ , where M refers to the model of EF component. An EF is shown below with some generic interconnection.

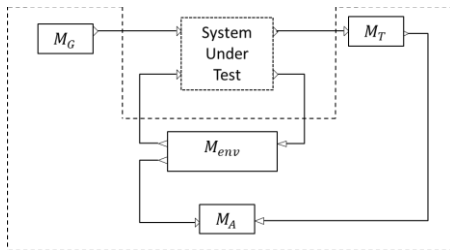


Figure 1: Experimental Frame

In [9], consistency conditions between EF components were discussed. An EF could be composed of real systems or models or both. In V&V through simulation, a model replaces the environmental system (G, T) and is composed with the SUT besides other systems. Then fidelity, informally, is nothing but the fact that the SUT cannot differentiate when models replaces the system.

### 3. SIMULATION FIDELITY DISTANCE

Fidelity, as a distance measure to the reality, could be either an absolute or a relative metric. An absolute fidelity metric is the (set of) distance measure(s) over the simulation model for all possible scenarios of the system. By contrast, a relative measure is scenario driven i.e. it focusses more on the trajectories related to a given scenario than the others. However, prior to quantifying this fidelity vis à vis its test scenarios, the global measure i.e. for all possible scenarios, needs to be addressed. This intuitively means, how far the model ‘mimics’ *all* the possible transitions of the system?. The next section explains the theoretical basis of this measurement based on simulation relations and the two player game theory.

#### 3.1. SIMULATION RELATIONS

In this paper, we use the classical notion of simulation preorders and simulation relations [6] which essentially states two models are (bi)similar if every transition of one model is matched by the other (and vice versa). In addition, notions of alternating simulation relations were proposed by Alur et al in the context of reactive systems [3]. However, these relations are boolean in nature i.e. either the model is exactly similar to the other or not and such boolean notions are too restrictive for practical purposes. It is not possible to distinguish between a more similar model and less similar model among the set of non-similar models. Approximate versions of these similarity relations were proposed in [8] which quantify this degree of similarity between two models. This approach has been applied to the design of safety controllers, formal verification, model reduction etc. for continuous, discrete and hybrid systems. These approximate bisimulation relations essentially give a global error bound i.e. maximum degree of dissimilarity between two models at a given time instant and this can formally be verified by geometric over approximation of the reachability set through zonotopes, ellipsoids etc. In the field of (discrete) simulation, this global bound is over-conservative since according to a scenario a model might still be valid locally despite its global error. This necessitates finding bounds on each trajectory i.e. a quantitative reachability through a two player game between the two models. On the other hand, distance notion based on this two player game proposed for automata [6] and interface automata [7] gives a transition-wise or path-wise distance in the context of implementation, coverage and robustness. However, to the best of our knowledge, a mechanism to quantify this distance for all possible inputs i.e. a superset of test scenarios has not been implemented. This requires generation of a quantitative reachability graph where every possible transition of system is evaluated over a positive real valued distance function.

Let us briefly define some preliminaries before describing the game. It may be recalled that a Finite State Automata (FSA) is defined by a tuple,  $T = \langle \Sigma, X, x^0, \delta, R \rangle$ , where  $\Sigma$  is a finite non-empty set of alphabets or labels,  $X$  is the finite non-empty set of states,  $x^0 \in X$  is the initial non-empty state

set,  $\delta: X \times \Sigma \rightarrow 2^X$  is the (nondeterministic) transition function and  $R \subseteq X$  is the set of accepting states. An accepting run of  $T$  over a finite word  $\omega = w_0 w_1 \dots w_n \in \Sigma$  is the sequence of states  $x_0 x_1 \dots x_n \in X$  such that  $x_n \in R$ . Then the language of  $T$ ,  $\mathcal{L}(T)$  is the set of words accepted by  $T$ .

Let us consider two transition systems,  $T_1 = \langle \Sigma_1, X_1, x_1^0, \delta_1 \rangle$  and  $T_2 = \langle \Sigma_2, X_2, x_2^0, \delta_2 \rangle$ , with  $\tau_1 \in \delta_1, \tau_2 \in \delta_2$ , then  $T_1$  simulates  $T_2$  is denoted by  $T_1 \preceq_S T_2$  and it holds if there exists a binary relation  $f \subseteq X_1 \times X_2$  such that if  $(x_1, x_2) \in f$  then

$$\forall (x_1, \tau_1, x'_1) \exists (x_2, \tau_2, x'_2) \text{ such that } (x'_1, x'_2) \in f \quad (1)$$

and it becomes bisimulation,  $T_1 \approx_{BS} T_2$  when

$$\forall (x_2, \tau_2, x'_2) \exists (x_1, \tau_1, x'_1) \text{ such that } (x'_1, x'_2) \in f \quad (2)$$

These simulation relations, and in addition alternating simulation relations were extended to quantitative game graphs and this notion is used in the next section to quantify the distance between system and simulation model behavior.

### 3.2. SYSTEM vs SIMULATION GAME

Game theoretic notions have been used in verification as well as synthesis perspectives in the formal modeling and analysis of systems [3],[7]. In this section, a two player game is briefly introduced followed by the game between the system and simulation model in the context of quantifying its degree of similarity i.e. fidelity.

A game graph is a tuple,  $g = \langle X, X^1, X^2, E, x_0 \rangle$  where  $X$  a finite set of states is partitioned as  $X^1$  and  $X^2$  for the first and second player respectively,  $E \subseteq X \times X$  is the set of edges,  $x_0$  is the initial state of the play [6]. The dynamics of the transition system described by its states and transitions are interpreted as nodes i.e. states and edges of this game. Then, formally, in the game between system,  $M_{sys}$  and simulation model,  $M_{sim}$  denoted by  $g(M_{sys}, M_{sim})$  with state space  $X_{sys} \times X_{sim}$

$$\text{Player 1 move: } (x_{sys}, \tau_{sys}, x_{sim}) \rightarrow (x'_{sys}, \tau_{sys}, x_{sim}) \quad (3)$$

$$\text{Player 2 move: } (x_{sys}, \tau_{sim}, x_{sim}) \rightarrow (x_{sys}, \tau_{sys}, x'_{sim})$$

The game starts with a move from one state of the system model to the next by the first player followed by the second player on simulation model and this continues until one wins. In particular, simulation relation exists if player 2 always has the winning strategy. The strategy of the player to choose each move may or may not depend on the history of previous moves and in this paper we employ the memory-less strategy. The set of visited states in the game is called a play which is denoted by  $\rho = \rho_1 \rho_2 \dots$  and this is akin to the path of a transition system or trace if there is a propositional evaluation at each such state. Then the degree of accuracy can be measured by a weighted error function,  $\varepsilon$  such as limited average for number of play,  $n_p$ . It is defined as follows

$$\varepsilon(\rho) = \liminf_{n_p \rightarrow \infty} \frac{1}{n_p} \sum_{i=0}^{n_p-1} e(\rho_i, \rho_{i+1}) \quad (4)$$

For example an error of 0.3 means 30% of transitions are ‘cheated’ or alternatively the model is 70% representative. The error function satisfies the reflexivity and triangular inequality i.e. for all  $\delta^{1,2,3} \in \delta$ ,  $\varepsilon(\delta^1, \delta^1) = 0$  and  $\varepsilon(\delta^1, \delta^3) \leq \varepsilon(\delta^1, \delta^2) + \varepsilon(\delta^2, \delta^3)$  respectively [2]. The games discussed henceforth, are only in the context of *untimed* transition systems modeled as FSA and section 6 briefly outlines the current work on timed quantitative games.

In this game, player 1, also called as attacker, plays on the system model and player 2, also called as defender, plays on the simulation model. For every move of the attacker, the defender matches the move or cheats over the move and incurs a penalty. This game is repeatedly played until one player wins. The attacker wins if the defender is not able to match his move and the defender wins if it matches every move of the attacker or attacker has no more moves. In the context of fidelity where the game is played between the system model and the simulation model, the latter is deemed representative if the defender wins. However this necessitates all the moves i.e. transitions of the attacker must be matched. This is too restrictive and infeasible at times and hence the notion of ‘cheating’ similar to the one introduced in [6] is used. It is easy to see that lesser the propensity of the simulation model to cheat, the higher the fidelity will be i.e. cheating is opposite of fidelity. Informally, the game is played as follows,

1. Player 1 plays on system model and hands back the token to player 2.
2. Player 2 plays on the simulation model, matches if the same label exists or cheats with the existing label and hands back the token to player 2
3. The play is over and the error is calculated, for example using Eq.4.

The next play begins and this continues until any one player wins or the play itself is terminated. The game is a perfect information game i.e. the defender has full visibility on the attacker’s move. This game is played in such a way that, the defender plays only the attacker’s label if it is available and if not, it plays all the possible choices. This helps in not only avoiding spurious plays by the defender but also in exhaustively estimating the error.

#### 3.2.1. Quantitative Reachability

In playing the game, the players are often confronted with different choices and hence there exists different strategies at each play. One of the challenges in playing this game is the choice of the strategy. Though different types of strategies have been discussed in literature [6],[7], they are mostly discussed in the context of playing a game on the system vs environment to reach a specific objective such as safety. However, in our case, the objective is to capture how close the game between simulation and system is, such that, they both allow same conclusion to be drawn for an evaluation against a specific V&V objective. In other words, a system may or may not satisfy a particular V&V objective, but, the objective of the simulation model is to

faithfully reproduce whatever the system may choose to do. Hence it is important to evaluate all *possible* strategies i.e. a reachability graph. In addition, such an exhaustive exploration needs to quantify the degree of fidelity in every possible path i.e. a quantitative reachability graph,  $\mathcal{R}_\varepsilon(M_{sys}, M_{sim})$ . This would not only give a path-wise fidelity measure for all possible paths but also help in analyzing the global fidelity as well. This global fidelity could be interpreted as a mean fidelity measure. From Eq.1 which gives a path or trajectory wise fidelity measure, the mean fidelity for all possible such trajectories whose size is  $N_b$ , at the end of a play,  $n_p$  is given by

$$\varepsilon_{n_p}^{avg} = \frac{1}{n_p} \sum_{j=0}^{N_b-1} \varepsilon(\rho_j) \quad (5)$$

Let us illustrate this game by a simple example. Consider a system,  $M_{sys}$  and some (legacy) simulation models,  $M_{sim}^{1,2}$  as shown in Fig 2.a and 2.b,c respectively.

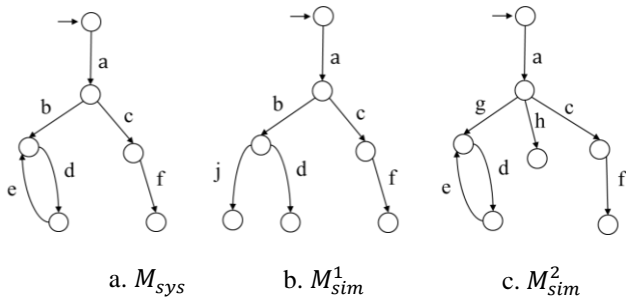


Figure 2: System and Simulation Models

In general, problems of behavioral fidelity come from two sources, namely, *unmodeled* dynamics and *incorrectly* modeled dynamics of the system. The former refers to the missing transitions whereas the latter refers to the incorrect transitions. For example, the transition  $e$  is unmodeled in simulation model (Fig 2.b) and the transition  $b$  is incorrect in model (Fig 2.c) i.e. it is modeled as label  $g$ . Such information can be quantified via these games. Now, let us play this game informally with player 1 choosing label  $a$  in the system model. This label is matched by player 2 playing on the simulation model and the error is 0 for both  $M_{sim}^1$  and  $M_{sim}^2$ . Now the player 1 chooses  $b$ , then player 2 does not cheat in the case of the first simulation model,  $M_{sim}^1$ . But in the case of the second, it cheats by playing on transitions  $g$ ,  $h$  or  $c$  and the error is 1 (or  $\frac{1}{2}$  in case total transition weighted as in Eq. 3). This continues and in fourth play, when player 1 chooses  $e$ , player 2 playing on first simulation model has no more moves and the game is lost. On the other hand, the other path i.e.  $\{a,c,f\}$  of player 1 can be matched exactly by both the simulation models. For the sake of illustration, errors associated in the quantitative reachability is given in the table below for the first four plays. Thus it can be seen clearly that exploring all the paths of models in this turn base game gives significant

insight into the fidelity characteristics of the simulation model.

Table 1: Quantitative Reachability Graph

Play	$\varepsilon(M_{sys}, M_{sim}^1)$	$\varepsilon(M_{sys}, M_{sim}^2)$
1	0	0
2	{0,0}	{0.5,0.5,0.5,0}
3	{0,0}	{0.67, $\infty$ ,0.67,0}
4	{ $\infty$ }	{0.67, $\infty$ }

In generating such a quantitative reachability graph, how the error is measured could be different depending on the user requirement. As remarked in [6], an error could be measured transition wise or moving average etc. and in this paper the error is calculated as weighted sum with respect to transition. In the future, it is intended to measure such different types of error with respect to the test scenario domain.

### 3.2.2. IMPLEMENTATION

The game semantics described in previous sections has been implemented in ProDEVS, a DEVS simulation platform [10]. ProDEVS is a Discrete Event Simulation (DEVS) platform and amongst other features such as FMI co-simulation, it can also be used to do formal verification with TINA toolbox [4]. DEVS is a more general case of the FSA formalism with embedded time and differentiation between input and output labels i.e. akin to interface automata. Since we intend to extend the current quantitative approach to timed automata, and then further to DEVS formalism, we construct our models in ProDEVS. On the other hand, the game semantics are modeled in petri-net formalism. These games are modeled in petri-net using the graphical editor of the TINA toolbox. Petri-nets, with their token based formalism, is amenable to modeling such turn based games between two FSA. In addition, using the TINA reachability generator along with the data encoding in guards and actions of the underlying petri-net transitions the quantitative reachability graph could be generated. This graph is then parsed to perform some analytics for better understanding of the model fidelity.

The DEVS definitions can be found in [1] [10] [11] and in this section we define only the petri-nets. Though petri-nets per se is a richer formalism to model transition systems due to its ability to model parallel processes, we will restrict our petri-net models to FSA where the states are finite and have no parallelism. Formally, a petri-net is a tuple

$$M = \langle P, \tau, A, w, p_0 \rangle \quad (6)$$

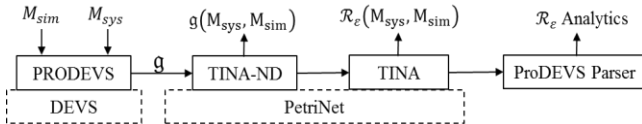
where

- $P$  is a finite set of symbols called places
- $\tau$  is a finite set of symbols called transitions with  $P \cap \tau = \emptyset$
- $A \subseteq (\tau \times P) \cup (P \times \tau)$  is the set of arcs defining the flow relation

-  $w: A \rightarrow N$  is the function defining the respective weights of the arcs

-  $p_0: P \rightarrow N$  is the initial marking

It can be easily seen that a petri-net with neither weights nor parallelism is a classical automaton and with time, it becomes a timed automaton. For the sake of brevity, let us denote places as states and markings denote the current state. In [1], Albert et al discusses the mapping between classic and parallel DEVS semantics to petri-net formalism using which the models are converted and the game is constructed in a single petri-net file and could be run directly from the ProDEVS. Since petri-net simulator per se does not handle data, these are encoded as guards and actions on the transitions through associated c files to generate dll files. The generated reachability graph is in text form and the data needs to be parsed for better understanding and visualization. The parser, written in JAVA and integrated in ProDEVS has many functions such as plotting the evolution of cheats along the play, distribution of cheats etc. In particular, it constructs a reachability tree which can then be visualized. The replay feature allows to choose a particular cheat from the cheat distribution plot to see the associated path to better understand when and where the simulation model behavior differs with respect to the system. The methodology is briefly given in the following figure.



**Figure 3: Implementation**

It can be seen that the modeling and parsing are done in ProDEVS with rest being in TINA. Alternatively, the modeling and game can be done in TINA-ND graphical editor tool as well and the reachability is generated by TINA later. It may be seen that, given a system design model and a simulation model, the game is constructed automatically and the resulting output is exhaustive error quantification over all possible transitions. The simulation user or the developer may then decide to improve the simulation model or relax the V&V requirements. This approach, apart from quantifying the global fidelity independent of V&V objectives, is also useful in iteratively refining the design with respect to V&V scenarios especially in the early system development when the design is not frozen.

#### 4. RELATIVE SIMULATION FIDELITY DISTANCE

Similar to quantifying the global fidelity, games can be extended to quantify the relative fidelity as well. Let us consider the example in Fig 2 and consider a V&V scenario informally (or formally via some temporal logic) stating whenever the user gives the label  $a$  and then  $c, f$  should always be the output with no error. This scenario is satisfied by both the simulation models. On the other hand, consider another scenario, stating whenever a user gives the label  $a$

and then  $b, d$  should always be the immediate output. In this case  $M_{sim}^1$  does better with error 0 than  $M_{sim}^2$  with error 0.5. If the scenario is, given the label  $a$  and then  $b$ , eventually the user must observe  $e$ , then  $M_{sim}^2$  is better than  $M_{sim}^1$  where the game has been lost. Thus, such a local notion helps in replacing system models with simulation models locally or ‘relative’ to the objectives. In other words, globally a simulation model could be far from representing the system but it may be adequate to represent the system for a particular V&V scenario. This relativeness vis à vis scenarios could be taken into account through relative weighting i.e. penalizing more the cheats on labels associated to the scenarios and less the cheats on other labels.

Let us denote the actions of interest on system model,  $M_{sys}$  by  $\tau^\phi \subseteq \tau_{sys} \in \delta_{sys}$  and whenever the defender cheats on these actions it incurs higher penalty than when it does not. The error weighting function is given by  $e: \delta \times N \rightarrow \mathbb{R}_0^+$  where  $\delta = \delta_1 \cup \delta_2$  and  $N$  refers to number of transition. In the two player game with turns  $m=1,2$  the distance is calculated at the end of every defender move i.e.  $\forall 2n$  where  $n \in N$  is the number of transitions. The two different weights are denoted by  $w_1$  and  $w_2$  respectively which could either be a simple positive number or a function of transition  $w_{1,2}(n)$ . Let the label and state of a transition  $\tau$  be  $\omega$  and  $x$  such that  $\omega \in X, x \in \Sigma$ , then

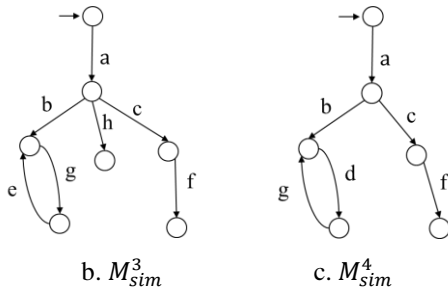
$$\forall \tau_{sim} \in \delta_{sim}, \begin{cases} \{\omega_{sim} \neq \omega_{sys} \wedge \omega_{sys} \in \tau^\phi\} \Rightarrow \varepsilon^\phi = w_1(\mathcal{E}) \\ \{\omega_{sim} \neq \omega_{sys} \wedge \omega_{sys} \notin \tau^\phi\} \Rightarrow \varepsilon^\phi = w_2(\mathcal{E}) \\ \text{else} \Rightarrow \varepsilon^\phi = 0 \end{cases} \quad (7)$$

In assigning weights to the ‘cheating’ transition, more weight  $w_1$  is given to transitions related to V&V requirements called ‘primary’ transitions and less weight,  $w_2$  is given to other transitions called ‘secondary’ transitions. This relies on the discounting principle that models cheating on primary transitions are penalized more and the earlier the cheat, more will be the penalty. In contrary, secondary cheats are penalized more with increasing time. Intuitively, models erring earlier on primary transition are viewed pessimistically whereas models erring earlier on secondary transitions are viewed optimistically on the assumption that they will eventually correct themselves.

For the sake of illustration, consider for every cheating move,  $n_c \leq n_p$ , let the weight varies in steps of -0.1 for primary weight i.e.  $w_1 = (1 - 0.01n_c)$  and +0.1 for secondary weight with each transition i.e.  $w_2 = 0.1n_c$  such that  $|w_1 + w_2| < 1$ . Consider the figure below whose system model is  $M_{sys}$  shown in the figure 4. Let the scenario be, whenever  $a$  (or  $a$  and then  $b$ ) is given there is possibility to get at least four  $e$ ’s at the end of ten transitions.

In general, a scenario independent error quantification will yield a global value of 0.4 at the end of tenth play for both the models. Instead, the labels in scenario ‘a’, ‘e’ are given more weightage during cheating and intuitively one can see

that,  $M_{sim}^3$  is better than  $M_{sim}^4$ . At the end of tenth play,  $n_c$  is 4 for both the models and the relative error becomes  $\varepsilon(M_{sys}, M_{sim}^3) = 0.16$  and  $\varepsilon(M_{sys}, M_{sim}^4) = 0.384$ . Such relative weighting can be integrated in the quantitative reachability graph generation similar to the one shown in section 3.2.



**Figure 4 : Simulation Models Relative Fidelity**

However, the weighting needs to be chosen carefully, a too stringent weighting may not show much difference with absolute error calculation and a too lenient weighting leads to spurious results. Further work is needed in this direction and it is also worth noting that alternatively this relative error perspective needs to be studied as compatibility between the EF components such as generator and acceptor. In the next section, a brief discussion on this distance approach in the context of morphism relations and derivability of EF is presented.

## 5. EF HOMOMORPHISM & DERIVABILITY

The fidelity distance described in section 3 and 4 could be discussed through EF homomorphism [11]. It is known that a morphism relation establishes correspondence between a concrete model i.e. system specification and an abstract model i.e. simulation model. This relation between two models becomes a homomorphism when the transition and output function has been preserved i.e. behavioral equivalence. This fidelity distance notion is then an approximate behavioral equivalence similar to the approximate language equivalence discussed in [8]. The relation between EF abstraction, applicability and derivability through hierarchy and orthogonality is discussed by Ponnusamy et al in [9]. These distance notions are not only useful in hierarchically ordering of details i.e. abstractions but also useful in ordering of scenarios through the concept of derivability. Derivability is the ability to form i.e. derive an EF from another EF. In general, for a given set of V&V scenarios different EF can be formed and it would be useful to build a more general i.e. more capable EF from which other EF can be derived. In other words, it relates two V&V scenarios and naturally, for the purpose of optimizing the test bench usage, a scenario having the widest possible coverage is chosen first and refined further based on the need or criticality. In addition, this helps in optimal development and reuse of models for different scenarios. Formally, derivability,  $\beta$  is defined as

$$\beta_k : EF_k^j \rightarrow EF_{k+1}^j \quad (8)$$

where  $j, k$  are the abstraction and derivability i.e. scenario ordering. Intuitively,  $EF_{k+1}^j$  derivable from  $EF_k^j$  means the former scenario is a subset of the latter i.e. scenario inclusion, at the same level of EF abstraction denoted by  $j$ . For example, in fig 2.a, the scenario  $a$  followed by  $b$  gives  $d$  could be derived from  $a$  gives  $d$  i.e.  $\{a \rightarrow b \rightarrow \dots \rightarrow d\} \subseteq \{a \rightarrow \dots \rightarrow d\}$ . Thus in practice, model having closest distance with respect to the lowest order EF is bound to be closest to the higher order EF derived from this EF as well. This leads to EF components i.e. models satisfying an EF property,  $\varphi_p$ , also satisfies the same property of other EF derived from it. Let a component model of EF be  $M_i$ , with error  $\mathcal{E}$ , then,

$$M_i^\mathcal{E} \models \varphi_1(EF_k) \Rightarrow M_i^\mathcal{E} \models \varphi_1(EF_{k+1}) \mid EF_{k+1} = \beta_k(EF_k) \quad (9)$$

This inclusion relation is due to the transitivity of the derivability as follows,

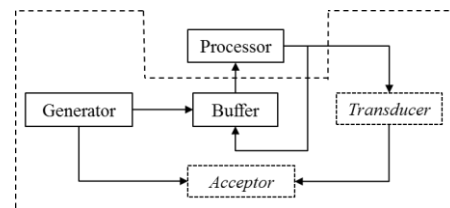
$$(EF_{k+1}^j \subseteq EF_k^j) \wedge (EF_{k+2}^j \subseteq EF_{k+1}^j) \Rightarrow (EF_{k+2}^j \subseteq EF_k^j) \quad (10)$$

where  $k \in \{1..K\}$  gives the limit of such operation. Besides building a ordered database of scenarios, it helps in identifying a scenario not present in the V&V plan that could not be derived from the defined scenarios.

It may be noted that the game theoretic approach assumes formalization of the knowledge about the labels i.e. transitions of the system and simulation model. In other words, the homomorphism relation is established between the labels i.e. equivalence of labels. This assumption is reasonable since the two models being developed by different stakeholders needs to have coherency in labels (ex: labels  $job$  and  $j$  refers to the same input event i.e. an incoming job) before establishing the simulation relation and quantifying the error between them.

## 6. APPLICATION CASE

The buffer [1] is a simple FIFO which receives jobs from the job generator and sends them to the processor whenever the processor is free. Whenever the job is received the queue,  $q$  is incremented and decremented when the job is sent to the processor. The received and sent jobs are denoted by label  $e0$  and  $s0$  respectively, processor status by  $e1$ . Let us imagine the processor to be the SUT and the requirement is to model the buffer with sufficient fidelity such that some scenarios,  $\varphi_{i=1}^N$  on the processor can be tested. This experimentation is illustrated as an EF below



**Figure 5: Processor Experimental Frame**

It can be seen that in addition to the generator and buffer, the experimentation may involve a *Transducer* to interpret the processed and generated jobs and an *Acceptor* which compares the jobs generated vs processed to ascertain the validity of the processor.

The system specification of buffer,  $Buf_{sys}$ , is supplied by the designer and the scenarios by the test team. The model developer who intends to build an abstraction i.e. a model of this buffer,  $Buf_{sim}$ , needs to quantify his model with respect to the system both globally and with respect to  $\varphi_{i=1}^N$ .

Consider an un-timed automaton modeling buffer behavior in ProDEVS and let us ignore the input/output actions and consider them simply as labels i.e.  $?e0$  is simply  $e0$  and  $!s0$  as simple  $s0$ . The system model,  $Buf_{sys}$ , is shown below,

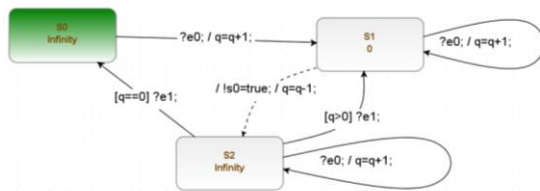


Figure 6: Buffer System Model

Let us specify four candidate models,  $Buf_{sim}^{1..4}$  as below,

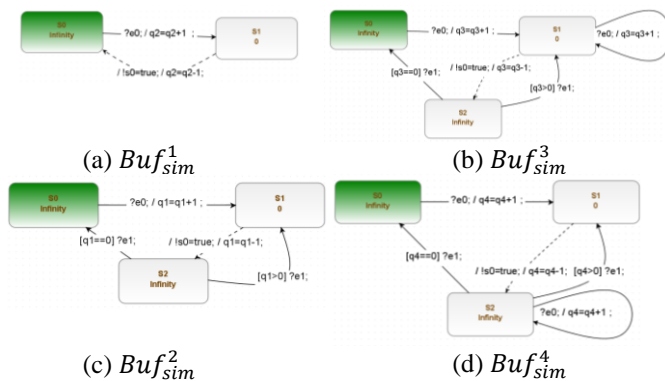


Figure 7: Buffer Simulation Models

For example, the game between the two models, system model and simulation model 1 (c) is informally described as follows,

$n_p = 1$ : From initial state,  $S_0$ , player 1 chooses transition  $e0$  and moves to  $S_1$ . Player 2 does the same.

$n_p = 2$ : From  $S_1$  the player 1 chooses  $e0$ . The player 2 cannot match and thus cheats with  $s0$ .

This continues forever and one can see the error is simply  $(n_p - 1)/n_p$ . Similar games can be played between other models. The following figure illustrates the distribution of trajectories based on the fidelity i.e.  $(1 - \epsilon)$  and it can be seen that higher the number of trajectories close to 100% or required fidelity, the higher the simulation model fidelity.

For the sake of illustration only trajectories up to the third play from a total of  $10^3$  plays are shown.

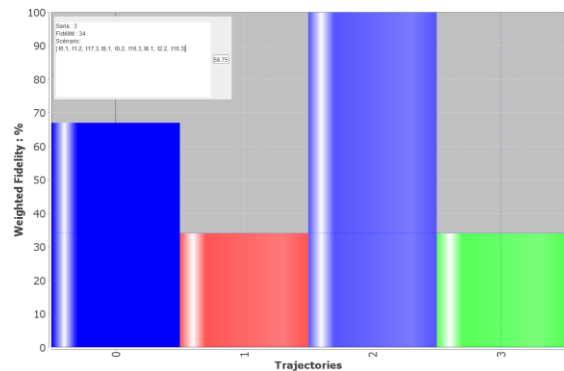


Figure 8: Trajectories Fidelity Distribution

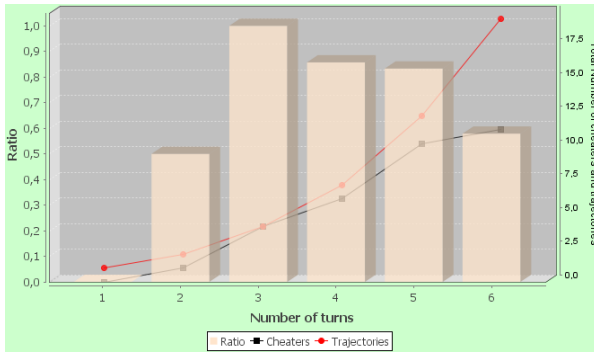
From this graph, one can see that out of four trajectories generated at the end of the third play by the system, only one is matched by the simulation model exactly and the second trajectory (in rose and green) cheats twice out of three transition i.e. 33% representative, whereas the fidelity of the other (in blue) is 67%. A particular trajectory can be picked up for visualization by clicking on the graph as shown in the upper left box inside the figure and in addition, the mean fidelity, in this case  $\sim 67\%$  is also shown. This reachability can also be analyzed as a measure of total number of cheats per turn with respect to the total number of trajectories at that turn. For example in the above example out of four trajectories two are cheating at the third play and in general, lower this ratio, the worse will be the fidelity. This is illustrated for all the four models in Fig 9. In addition, the number of trajectories cheated (in black) and the total number of trajectories (in red) at each play is also given.

In the case of relative cheating, let us consider a scenario,  $\varphi_1$  stating the processor must process all the jobs generated or in other words, no job is lost by the buffer. For this particular scenario, weighting is more on  $e0$  and less on other labels similar to example in section 4. Similar such weightings can be done for other scenarios and analysis is done as in Figs 8,9. In addition, sensitivity of weights to the error for a given scenario can be studied as well to ascertain a viable tradeoff between model abstraction i.e. complexity and fidelity.

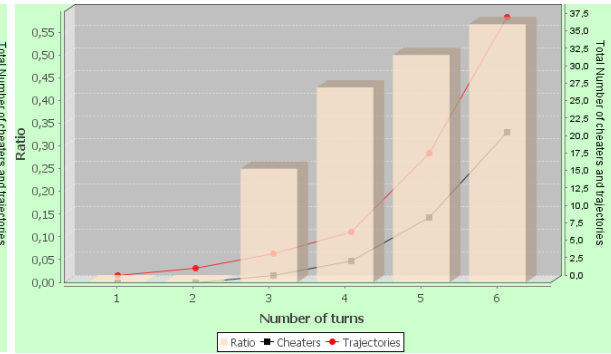
## 7. OUTLOOK & CONCLUSION

In this paper, only untimed games are presented and this is being extended to timed games [8]. Informally, in these timed games between system and simulation model, the defender plays a non-blocking game where the simulation model can cheat on the timing of transitions for the same labels. In addition, this game based quantitative approach is also being extended to open systems modeled as untimed interface automata based on the work of Cerny et al [5].

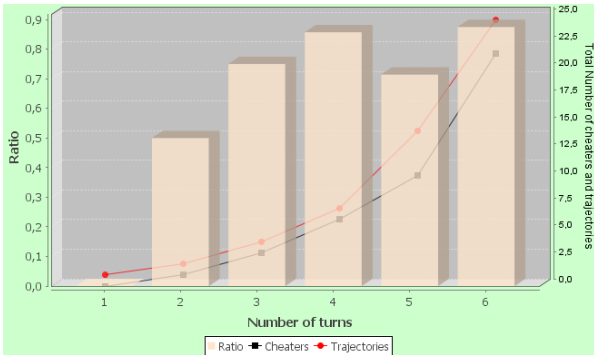




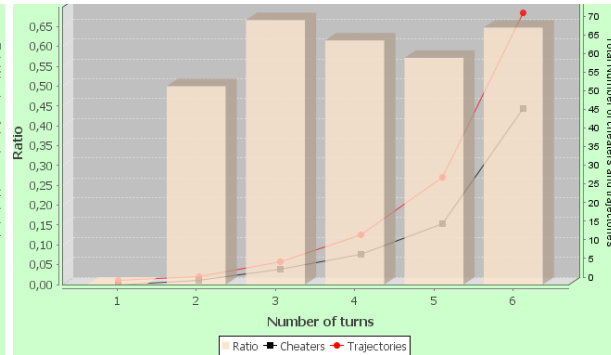
(a)  $Buf_{sys}$  vs  $Buf_{sim}^1$



(c)  $Buf_{sys}$  vs  $Buf_{sim}^3$



(b)  $Buf_{sys}$  vs  $Buf_{sim}^2$



(d)  $Buf_{sys}$  vs  $Buf_{sim}^4$

**Figure 9: Cheating Trajectories Distribution**

This approach needs to be studied in the context of some efficient reachability representations such as Binary Decision Diagrams (BDD) and symbolic approximations for systems to mitigate the problem of state space explosion in higher dimensional systems. In addition, a key point in fidelity quantification is the origin of this distance i.e. how a model is built since there exist different ways of modeling [9]. In the current study, the abstractions are structural i.e. omission of a particular transition [8]. Future works include the extension of our approach to other classes of widely used abstractions such as state aggregation.

## REFERENCES

1. Albert, V., Ponnusamy, S.S., Encoding CDEVS and PDEVs into Timed Petri Net: theory and application, (2016), Journées DEVS Francophones, Submitted.
2. Alfaro, L., Faella, M., Stoelinga, M., Linear and Branching System Metrics, *IEEE Trans. Software Eng* (2009). Vol 35(2), 258–273.
3. Alur, R., Henzinger, T., Kupferman, O., Vardi; M. Alternating refinement relations. *Lecture Notes in Computer Science* (1998), Vol 1466, 163-178.
4. Berthomieu, B., Diaz, M., Modeling and verification of time dependent systems using time Petri nets, *IEEE Trans, on Software Engineering* (1991), Vol. 17, no. 3, pp. 259-273.
5. Cerny, P., Henzinger, T. & Radhakrishna, A., Interface simulation distances. *Theoretical Computer Science* (2014), Vol 560: 348-363.
6. Cerny, P., Henzinger, T. & Radhakrishna, A., Simulation Distances. *Lecture Notes in Computer Science* (2010), Vol 6269, 253–268.
7. Chatterjee, K., Prabhu, V.S., (2015), Quantitative Temporal Simulation and Refinement Distances for Timed Systems, *IEEE Transactions on Automatic Control*, Vol 60, Issue 9, 2291-2306.
8. Girard, A., Pappas, G.J., Approximation Metrics for Discrete and Continuous Systems. *IEEE Transactions on Automatic Control* (2007), Vol 52, Issue 5, 782-798.
9. Ponnusamy, S.S., Thebault P., Albert V., , Consistent Behavioral Abstractions of Experimental Frame (2015), *AIAA Modeling and Simulation Technologies Conference*, US, Accepted.
10. Vu, L.H., Foures, D., Albert, V., (2015), ProDEVs: an event-driven modeling and simulation tool for hybrid systems using state diagrams, *Proceedings of the 8th International Conference on Simulation Tools and Techniques*, 29-37, Greece.
11. Zeigler, B.P., Praehofer, H., Tag, G.K. *Theory of modeling and simulation*, San Diego, California, USA Academic Press, 2000.