Efficient non-anonymous composition operator for modeling complex dependable systems

Silvano Chiaradonna, Felicita Di Giandomenico and Giulio Masetti
ISTI-CNR, Pisa, Italy
Email: silvano.chiaradonna@isti.cnr.it, felicita.digiandomenico@isti.cnr.it, giulio.masetti@isti.cnr.it

***Abstract*—A new model composer is proposed to automatically generate non-anonymous model replicas in the context of performability and dependability evaluation. It is a state-sharing composer that extends the standard anonymous replication composer in order to share the state of a replica among a set of other specific replicas or among the replica and another external model. This new composition operator aims to improve expressiveness and performance with respect to the standard anonymous replicator, namely the one adopted by the Möbius modeling framework.**

I. Introduction

Stochastic model-based evaluation is widely used to analyze the performability, dependability and performance of complex systems. A convenient technique to make model construction easier, and to promote an efficient solution process, is based on the hierarchical submodels composition [1], [2]. To overcome the heavy and error prone manual definition of many submodels, automated procedures are currently available to assist the modeller in generating the hierarchical composed model representing the specific instance of the system. A general approach is based on the definition of primitives, such as the *Rep* operator proposed in the Möbius framework [3], which can be used to represent different components, behaviors or characteristics of a system, automatically replicating a single template model. Each replica is indistinguishable by other replicas and then it is anonymous. Although satisfactory in general terms, such *Rep* operator shows limitations from the efficiency point of view when employed in the modeling of systems composed of a (large) population of similar, but non anonymous, components.

In this paper, a new composition operator, called *NARep*, is introduced to deal with efficient non-anonymous replication inside the Möbius framework. The new operator characteristics are identified and presented in the context of performability and dependability evaluation.

II. The context

The Möbius modeling framework [3] is a powerful modular environment that supports multiple modeling formalisms and solution techniques. Among the others, it supports the Stochastic Activity Network (SAN) formalism, composition operators *Rep* and *Join* [1], performance variables based on rate and impulse rewards (used also to define performability and dependability measures), analytical and simulative solvers.

Join operator is a state-sharing composer that brings together two or more submodels. *Rep* operator is a special case of *Join* that constructs a model consisting of identical copies (replicas) of a submodel. *Rep* and *Join* operators are defined at level of Abstract Functional Interface (AFI) [3], [4], a common interface between model formalisms and solvers that allows formalism-to-formalism and formalism-to-solver interactions. At AFI level, places and activities (transitions) of SAN correspond to state variables and actions. In the following, for sake of simplicity, we will use SAN models and SAN notation (places and activities) also to describe *Rep* and *Join* operators. The state of a model is represented by the current value of all the places defined in the model. A place of a replicated SAN model can be local (assume, at the same time, different values in different replicas) or shared (same place, with only one value, that can be read or updated by each replica). The *Rep* operator can be used to represent multiple components, behaviors or characteristics of a system having identical definition and parameters.

A. Non-anonymous replication in Möbius

Non-anonymous replicas of a single SAN model can be automatically generated through *Rep* operator via a sophisticated use of local places and immediate activities as shown in [5]. Local places, representing replica indexes, are set up so that each replica becomes self-aware of its index. The specific state, number of "tokens" or "marks" in Petri nets dialects, of the i-th non-anonymous replica is represented by the i-th entry $P\text{–>}Index(i)\text{–>}Mark()$ of a shared array-type extended place P, i.e., an extended place defined as an array of states, that is shared among *all* the replicas. Thus, each replica has read and write access to the state of all the other replicas of the same SAN. In addition, a different submodel that shares the place P with a *Rep* has read and write access to the state of each specific replica.

This approach allows to cope with dependability analysis of complex critical systems, with a variety of interconnected components organized according to sophisticated topologies, such as electrical or transportation systems. In [6], which focuses on Smart Grids, a stochastic modeling framework has been proposed, which uses the *Rep* operator as described above to quantitatively assess representative indicators of the resilience and quality of service.

B. Motivations for an enhanced replication operator

Using an anonymous replicator to produce non-anonymous replicas sounds unnatural, while on the other hand the *Join* operator is impractical for large models. Moreover, dependencies introduced by the n-dimensional array-type extended places shared among all the n replicas of a SAN model can lead to a large time overhead when, during the initialisation phase of the simulator, connectivity lists [7] are generated. Implementing a non-anonymous replication with the *Rep* operator and with n-dimensional array-type extended places, shared among n replicas, slows down the connectivity lists construction. In fact, n^2 checks are needed because each replica can access to the state of *all* the other replicas. With simple toy models, varying n from 10 to 50 the connectivity lists generation time overhead remains acceptable, but varying n from 100 to 500 the time overhead can increase of about 100 times. Following these considerations, we propose the *NARep* (*non-anonymous replication*) new operator.

III. *NARep* DEFINITION

NARep automatically constructs a model consisting of non-anonymous indexed replicas of a given submodel. The index of a specific replica can be accessed from the generic SAN submodel using a new reserved keyword $repindex()$, corresponding to a method of the C++ SAN class. Each place P included in a non-anonymous model $aSAN$, replicated n times, corresponds to n different place replicas via the *NARep* operator, and place replicas are accessed with $aSAN{-}{>}Index(0){-}{>}P$,..., $aSAN{-}{>}Index(n-1){-}{>}P$. The *NARep* operator can be used to define whether the replicas of each place P are: (i)

1) local (default): read and write access to replica place $aSAN{-}{>}Index(i){-}{>}P$ limited only to the corresponding i-th replica of the SAN submodel. The generic model $aSAN$, representing the generic replica $repindex()$, can access to the corresponding replica $repindex()$ of the place P, using: $aSAN{-}{>}P{-}{>}Mark()$, or equivalently $aSAN{-}{>}Index(repindex()){-}{>}P{-}{>}Mark()$.
2) place-shared: several place replicas correspond to a single place. For example, if $aSAN{-}{>}Index(0){-}{>}P{-}{>}Mark()$ is equal to $aSAN{-}{>}Index(1){-}{>}P{-}{>}Mark()$, then the command $aSAN{-}{>}Index(repindex()){-}{>}P$ is referred to the same place replica each time that $repindex()$ is equal to 0 or 1.
3) rep-shared: the same place replica is shared among several replicas of the SAN submodel. For example, the place replica $aSAN{-}{>}Index(i){-}{>}P$ can be rep-shared among the $(i-1)$-th and $(i+1)$-th replicas of $aSAN$. In this case, the generic model $aSAN$ can access to the rep-share places using $aSAN{-}{>}Index((repindex()-1)\%n){-}{>}P$, $aSAN{-}{>}Index(repindex()){-}{>}P$ and $aSAN{-}{>}Index((repindex()+1)\%n){-}{>}P$. The list of places that are rep-shared with a replica is obtained using the command $P{-}{>}repshared()$. Using this command, a generic SAN model can represent the access of the i-th replica only to those components that are connected to the i-th replica.
4) up-shared: a local, place-shared or rep-shared place replica can be shared among other submodels (not $aSAN$) at the level on top of $NARep$. For example, $aSAN{-}{>}Index(0){-}{>}P$ and $aSAN{-}{>}Index(1){-}{>}P$ can be local and up-shared with the place $otherSAN{-}{>}Q$, where the model $otherSAN$ is composed with *NARep*. In this case, the Q place included in $otherSAN$ is an extended place defined as a 2 dimensional array, each entry having the same type of the place P, such that $aSAN{-}{>}Index(0){-}{>}P$ is equal to $Q{-}{>}Index(0)$ and $aSAN{-}{>}Index(1){-}{>}P$ is equal to $Q{-}{>}Index(1)$.

IV. DISCUSSION

The proposed *NARep* operator has the following advantages:

- Automatic generation of non-anonymous indexed place and SAN replicas transparent to the modeler; this feature enlarges the framework expressive power.
- A significant reduction of the connectivity lists generation time overhead can be obtained exploiting only actually existing dependencies instead of the assumed complete graph of dependencies. Therefore, the n^2 checks currently performed by the *Rep* operator constitute the worst case for *NARep*, when full connection is considered.
- Symmetries exploitation via the state-space lumping [2] remains possible.

Future work will include:

- *NARep* formalism definition,
- integrate *NARep* into Möbius framework,
- evaluate the actual impact of *NARep* on the connectivity lists generation time overhead; in particular, a significant improvement is expected for the Smart Grid model presented in [6], being the electrical and communications networks sparse.

REFERENCES

[1] W. H. Sanders and J. F. Meyer, "A unified approach for specifying measures of performance, dependability and performability," in *Dependable Computing for Critical Applications, Vol. 4 of Dependable Computing and Fault-Tolerant Systems*, A. Avizienis and J. Laprie, Eds. Springer Verlag, 1991, pp. 215–237.

[2] S. Derisavi, P. Kemper, and W. H. Sanders, "Symbolic state-space exploration and numerical analysis of state-sharing composed models," *Linear Algebra and its Applications, Special Issue on the Conference on the Numerical Solution of Markov Chains 2003*, vol. 386, pp. 137–166, 2004.

[3] D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster, "The Möbius framework and its implementation," *IEEE Trans. on Software Engineering*, vol. 28, no. 10, pp. 956–969, 2002.

[4] J. M. Doyle, "Abstract model specification using the Möbius modeling tool," Master's thesis, University of Illinois at Urbana-Champaign, 2000.

[5] S. Chiaradonna, F. Di Giandomenico, and N. Murru, "On a modeling approach to analyze resilience of a smart grid infrastructure," in *Tenth European Dependable Comput. Conf. (EDCC 2014)*, Newcastle upon Tyne, UK, May 2014, pp. 166–177.

[6] S. Chiaradonna, F. Di Giandomenico, and G. Masetti, "A stochastic modelling framework to analyze smart grids control strategies," in *Accepted to Fourth IEEE Int. Conf. on Smart Energy Grid Eng. (SEGE 2016)*, Oshawa, Canada, August 2016.

[7] A. L. Williamson, "Discrete event simulation in the Möbius modeling framework," Master's thesis, University of Illinois at Urbana-Champaign, 1998.