Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

SIZE MEASUREMENT OF DEVS MODELS FOR SBA EFFECTIVENESS EVALUATION

Hae Young Lee and Hyung-Jong Kim

Department of Information Security
Seoul Women's University
Seoul, 139-774, REPUBLIC OF KOREA

ABSTRACT

Due to the characteristic differences between simulation models and software systems, software development effort estimation approaches, such as function point analyses, might not be appropriate for the estimation of system modeling efforts. In order to enable system modeling efforts to be quantitatively estimated, in this paper, we propose possible approaches to measure the size of formalism-based simulation models, which include 4 ones for atomic models and 4 ones for digraph models. Their merits and demerits are briefly discussed.

1 INTRODUCTION

In order to evaluate the effectiveness of simulation-based acquisition (SBA), system modeling efforts needs to be carefully considered. The characteristics of simulation models, however, differ from those of software systems, so that software development effort estimation approaches, such as function point analyses (ISO/IEC 2011), might not be appropriate for them. Therefore, in this paper, we propose *possible approaches* to measure the size of discrete event system specification (DEVS) (Zeigler, Praehofer, and Kim 2000) models, in order to enable the efforts for building them to be quantitatively estimated based on their size. Once the size has been measured, the cost of the modeling could be calculated from past projects. The merits and demerits of the proposed approaches are discussed in brief.

2 SIZE MEASUREMENT FOR ATOMIC DEVS MODELS

An atomic DEVS model is defined as a 7-tuple: $M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$.

I/O-mapping measurement – One of the easiest approaches to measure the size of an atomic DEVS model could be to consider just its I/O mapping. The size can be obtained by $size(M) = |X| \times |Y|$, where M is an atomic DEVS model, X is M's input event set, and Y is M's output event set. Since $|X|$ and $|Y|$ could be infinite (e.g., $X = \mathbb{R}$) at the formalism level, X and Y must be based on the model's implementation (e.g., X is a 32-bit floating-point type). One of its problems is that the size of a complex model could be underestimated. Another problem is that the size of a model could be overestimated due to *unused elements* in its I/O sets (e.g., while $X = \mathbb{N}$, many of the elements would not be fed into the model).

State-space measurement – The size of an atomic model can be obtained by considering the size of its state space. That is, $size(M) = |S|$, where S is M's state set. Due to possible presence of an infinite state set, the set must be also based on the model's implementation. The remaining time at a state, σ, may be ignored since a continue operation can be viewed as a transition without schedule update, as described in Hwang and Zeigler (2009). Generally, the state set of a model explicitly or implicitly reflects its I/O sets, so that its results would be more precise than that of the I/O mapping measurement. However, the size of a model could be still overestimated due to unused elements in the state set. Also, the size of a model having complicated transitions could be underestimated.

Transition-based measurement – A more precise approach would be to count the number of transitions in a model. In the implementation of a model, transitions are usually represented as a *finite number of relations* which are such that: (external transition relations) $T_{ext} \subseteq S \times E \times X \times S$, where E is the set of constraints over e, usually expressed in a form of $e \in i$, where i is an interval over $\mathbb{R}_{[0,\infty]}$, and (internal transition relations) $T_{int} \subseteq S \times S$. Thus, $size\ (M) = |T_{ext}| + |T_{int}|$. Although complicated transitions can be considered in the approach, the problem of the overestimation caused by unused elements still remains.

Reachability analysis – The size of a model can be obtained through exploring *all reachable state regions* of the model. A state regions can be expressed in $r = (s, i)$, where s $\in S$ and i is an interval over $\mathbb{R}_{[0,\infty]}$. For a model, by exhaustively searching all reachable states from the initial state, a reachability graph for the model can be constructed. The size of the model can be obtained by counting the number of state regions in the graph. The approach, while more accurate, still has the overestimation problem due to unused elements. Moreover, it would be difficult to apply the approach unless models have been already implemented.

3 SIZE MEASUREMENT FOR DIGRAPH DEVS MODELS

A digraph DEVS model is defined as an 8-tuple: $N = \langle X, Y, D, \{M_i\}, C_{EI}, C_I, C_{EO}, Select \rangle$.

Model counting – One of the easiest approach to measure the size of a digraph DEVS model would be to recursively count the number of atomic models in the model. It is similar to source lines of code in software development estimation, so that modelers can *intentionally* increase or decrease the size with ease. Also, the complexity of each atomic model cannot be considered.

Atomic-model-based measurement – This approach uses the sizes of all atomic models in a digraph model to measure the size of the model. Once the sizes of atomic models have been obtained based on an approach in Section 2, the size of a digraph model can be obtained by the sum of them (or the multiplication of them in some cases). The characteristics of the approach are inherited from those of the approach used to measure the sizes of atomic models.

Input set refinement through model-chain analysis – A closed digraph model (i.e., $X = Y = \varnothing$) has a form of 'model chains.' For example, events would be generated by an atomic model (e.g., a generator) at the very beginning, and then be fed into other models. Such events would be finally collected up by an atomic model, such as an acceptor. Thus, the input set of a model can be *refined* by analyzing the output sets of the preceding models in such model chains. The approach could make the overestimation problem due to unused elements alleviated.

Reachability analysis – The size of a digraph model can be precisely obtained by exhaustively exploring all reachable states of the model. Especially in case of closed models, very precise results could be obtained without the overestimation. However, the reachability analysis for a large-scale digraph model would take a very long time, or even be impossible.

ACKNOWLEDGEMENT

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract UD110006MD, Korea.

REFERENCES

ISO/IEC. 2011. *Software Engineering – COSMIC: A Functional Size Measurement Method*. ISO/IEC 19761:2011.

Hwang, M. H., and B. P. Zeigler. 2009. "Reachability Graph of Finite and Deterministic DEVS Networks." *IEEE Transactions on Automation Science and Engineering* 6(3):454-467.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. Theory of Modeling and Simulation. 2nd ed. Academic Press.