A conceptual framework for verification of complex system simulations

Sandrina Dimitrijevic
Faculty of Economics, Belgrade, Serbia
anirdnas@gmail.com

***Abstract*— While process of verification of traditional simulation models is well established, fresh approaches are needed for new types of simulation. That is especially the case for simulation of complex systems which have unique properties (emergent behavior, high sensitivity to changes) that can mask faults in code and wrong design. In this paper we deal with such challenge by proposing a conceptual model for verification of complex system simulation. Namely, we suggest that each verification attempt needs to include usage of individual elements checking along with top-down approach to system testing.**

I. Introduction

Verification is an important aspect of every simulation. It can be defined as "The process of determining that a model or simulation implementation accurately represents the developer's conceptual description and specification. Verification also evaluates the extent to which the model or simulation has been developed using sound and established software engineering techniques." [1]. By verifying the simulation model, we confirm that it is built correctly, without inconsistencies, conflicts and redundancies. Otherwise, the analysis based on it can provide fault results and misleading conclusions which is a situation every researcher wants to avoid.

While verification techniques have been extensively studied and developed for discrete-event system simulations, recently popular complex systems simulations lack such formalism and are in need for a more organized approach [2]. We tend to contribute to the endeavor of better simulation of complex systems by providing a conceptual framework for their verification.

While there are papers which focus on complex system validation, so far there hasn't been an attempt to create a complete framework which can be used for verification [3]. The need for standardized simulation building approach stems from the requirement to build high quality models and establish a common language for scientist to discuss and compare models [4].

In the first part of the paper we explain the reasons why the simulation of complex systems might provide unique research challenges and put the emphasis on the process of software verification. Afterwards, based on the literature review of software verification process we examine usefulness of existing methods and adjust them to the need of complex systems.

II. Simulation of Complex Systems

Complex systems are defined as systems which contain multiple different parts that interact with each other and with the environment. Science of complex systems present a rising research field which has flourished with the rise of computational power and new possibilities in modeling and simulation [5]. They are being used in various fields from economics, biology, physics and more.

However, their usage comes with certain challenges, one of them being the methodology for verification and validation [6,7] Especially verification has been recognized as a critical issue in their quality assurance [8], where one has to be certain that emerging patterns on the macro level are result of system's inherent characteristics not errors in the code or design specification.

New methodological standards for their verification have still not been established while existing ones, which refer to traditional simulation models, might not be suitable for complex system simulation as such systems are unique in many aspects [9].

Some of those aspects are [10]:

1) non-linearity which can even lead to chaotic behavior, a small change can have a big effect on final outcome.
2) emergent behavior which is not easily predictable and can come a surprise to the modeler.
3) heterogeneous agents that adapt and learn through mutual interaction.

For these reasons it is hard to identify faults in code or programming logic. A researcher might be under impression that that he has discovered a new type of emergent behavior but beneath it might lie a coding mistake. That is, one has to treat verification very carefully.

Some traditional verification techniques might not be appropriate. For example, complex systems are very prone to random effects. A correctly built model might behave differently due to different random conditions. That is why technique of internal validity [2] might not work. Additionally, traditional methods usually imply that system components are homogenous, but in complex systems that is opposite [2]. Also, in traditional simulation methods scientists often know what the expected behavior of the whole system is while that assumptions are not valid for complex systems due to their nature [11].

Researchers simply don't know what to expect and that is why errors can be easily overlooked. Model-to-model comparison is also hard to implement is it can happen that similar models with same parameters can provide different results [11]. Usage of boundary conditions is also one of the traditionally and most widely used techniques [7]. However, with complex systems there is an additional challenge which is that boundaries are uncertain [7].

III. A Verification Framework

A. *Traditional Techniques*

Some of the traditional technique are still valid when it comes to verification of complex systems simulation [12]. Those are audit, code inspection and reviews. Depending on the approach they are performed by a group of qualified individuals which follow formalized procedures in order to perform analysis of the simulation code. They have to confirm that all standards have been implemented and that programming code is correct [13].

Usually, the procedure of performing review or inspections has couple of phases which need to be respected. Some of those are overview, preparation, inspection, rework and follow-up [14] Team members inspect the logic of the code, related documents and specification. Even though this technique is prone to standard human errors they are still effective and can provide good results. However, a single researcher is not always in position to get a team of experts to review the simulation code and its correctness.

B. *Top-down approach*

Top-down approach includes splitting the model into smaller parts which can be studied in isolation [14]. Each part presents a subsystem created according to the specification hierarchy. By using the input / output approach each system is tested to see if it behaves according to assumptions[15].

One more top-down approach is that one iteration cycle is also analyzed to see if it produces correct results. Instead of analyzing the whole simulation, a researcher focuses on a single iteration [17].

The downside of using this method is that it can be costly and complex as well, because parts of complex systems can also present complex systems [7].

C. *Tracing*

As constitution parts of a model are heterogeneous and can interact with each other it can be useful to trace their behavior. Individual behavior is followed throughout the whole cycle to ensure that it never deviates from predefined assumptions [16]. That way it can be confirmed that logic of the code is accurate. However, if there are many such heterogenous elements such analysis can take time.

IV. Conclusion

Verification of complex systems is not an easy process but together with traditional techniques of expert code checking there are two more approaches that can be used. Those are splitting the system into smaller parts and testing their behavior in an isolated black-box approach along with tracing the behavior of individual components during the whole simulation cycle.

References

[1] Stevenson, D. E. (2002). Verification and validation of complex systems. *Intelligent Engineering Systems through Artificial Neural Networks*, 159-164.

[2] Xiang, X., Kennedy, R., Madey, G., & Cabaniss, S. (2005, April). Verification and validation of agent-based scientific simulation models. *In Agent-directed simulation conference* (pp. 47-55).

[3] Klügl, F. (2008, March). A validation methodology for agent-based simulations. *In Proceedings of the 2008 ACM symposium on Applied computing* (pp. 39-43). ACM.

[4] Fujimoto, R., Bock, C., Chen, W., Page, E., & Panchal, J. (2016). *Research Challenges in Modeling and Simulation for Engineering Complex Systems*. NSF Report, Arlington, VA

[5] Wolfram, S. (2002). A new kind of science (Vol. 5). Champaign, IL: *Wolfram Media*

[6] Aumann, C. A. (2007). A methodology for developing simulation models of complex systems. *Ecological Modelling*, 202(3), 385-396.

[7] Petty, M. D. (2010, February). Modeling and Validation Challenges for Complex Systems. *In Proceedings of the Spring 2012 Simulation Interoperability Worksho*p (pp. 178-187).

[8] Kundert, K., & Chang, H. (2005). Top-down design and verification of mixed-signal circuits. *Planet Analog*.

[9] Midgley, D., Marks, R., & Kunchamwar, D. (2007). Building and assurance of agent-based models: An example and challenge to the field. *Journal of Business Research*, 60(8), 884-893.

[10] Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, 4(5), 41-60.

[11] Polack, F. A., Andrews, P. S., Ghetiu, T., Read, M., Stepney, S., Timmis, J., & Sampson, A. T. (2010, March). Reflections on the simulation of complex systems for science. *In Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference o*n (pp. 276-285). IEEE.

[12] Whitner, R. B., & Balci, O. (1989, October). Guidelines for selecting and using simulation model verification techniques. *In Proceedings of the 21st conference on Winter simulation* (pp. 559-568). ACM.

[13] Rand, W., & Rust, R. T. (2011). Agent-based modeling in marketing: Guidelines for rigor. *International Journal of Research in Marketing*, 28(3), 181-193.

[14] Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of [17] Esko, O., Jaaskelainen, P., Huerta, P., Carlos, S., Takala, J., & Martinez, J. I. (2010, August). Customized exposed datapath soft-core design flow with compiler support. In Field Programmable Logic and Applications (FPL), 2010 International Conference on (pp. 217-222). IEEE.*

[15] *operations research*, 53(1), 121-173.

[16] Esko, O., Jaaskelainen, P., Huerta, P., Carlos, S., Takala, J., & Martinez, J. I. (2010, August). Customized exposed datapath soft-core design flow with compiler support. *In Field Programmable Logic and Applications (FPL), 2010 International Conference on* (pp. 217-222). IEEE.

[17] Kennedy, R. C., Xiang, X., Cosimano, T. F., Arthurs, L. A., Maurice, P. A., Madey, G. R., & Cabaniss, S. E. (2006).