Systematic data analysis-based validation of simulation models with heterogeneous data sources

Daniel J. Foguelman†, Matias Bonaventura† and Rodrigo Castro†
† Departamento de Computación, FCEyN, UBA and ICC, CONICET, Pab. 1, C1428EGA, Buenos Aires, Argentina.
Email:{dfoguelman,mbonaventura,rcastro}@dc.uba.ar

***Abstract*—Complex networked computer systems are subjected to upgrades on a continuous basis. Modeling and simulation (M&S) of such systems helps with guiding their engineering processes when testing design options on the real system is not an option. Too often many system's operational conditions need to be assumed in order to focus on the questions at hand, a typical case being the exogenous workload. Meanwhile, soaring amounts of monitoring information is logged to analyze the system's performance in search for improvement opportunities. Concurrently, research questions mutate as operational conditions vary throughout its lifetime. This context poses many challenges to assess the validity of simulation models. As the empirical knowledge base of the system grows, the question arises whether a simulation model that was once deemed valid could be invalidated in the context of unprecedented operation conditions.**

This work presents a conceptual framework and a practical prototype that helps with answering this question in a systematic, automated way. MASADA parses recorded operation intervals and automatically parameterizes, launches, and validates simulation experiments. MASADA has been tested in the data acquisition network of the ATLAS particle physics experiment at CERN. The result is an efficient framework for validating our models on a continuous basis as new particle collisions impose unpredictable network workloads.

***Keywords*—Simulation, Data Analysis, Continuous Validation, Process Automation, DEVS, Integration.**

I. INTRODUCTION

Simulating complex computer systems can be a vital requirement to gain insights into a system's behavior. Simulations are often used to study possible effects of alternative operation conditions in a real system, and to drive courses of action in search for improvements. Many key activities need to be orchestrated during a simulation study, e.g. the development of efficient models, acquisition of reliable measurements from the system, verification of correctness of the models, and validation of simulation results.

In many cases simulations are used in engineering projects to drive the design of new features for an existing system. Once changes are introduced, new experiments are planned on the upgraded system to validate the expected results forecasted during the simulation phase.

Simulation frameworks are required to provide solid means for guaranteeing both correctness and reproducibility. As projects grow and evolve in time, the ability to trace back design decisions and relate them with the simulations studies they were based on becomes an increasingly important requirement.

Several well-known simulation frameworks and methodologies exist offering a systematic description of stages to derive reproducible simulation results [1], [2].

Yet, there is a lack of practical tools to help ensuring reproducibility. Tools can be designed to control that required processes are exhaustively followed, preconditions are continuously verified, and steps in a validation cycle are explicitly defined.

Important tasks need to be neatly orchestrated such as model parameterization, definition of simulation metrics and comparison of results against the real system's behavior. There is also a need to verify that decisions made during certain tasks do not fall in contradiction with those made in others. An inconsistent usage of units of measurement is just one illustrative example. The situation can become increasingly problematic in large and complex systems where tasks usually require interaction with heterogeneous information repositories and massive measurement databases.

Process automation is a robust technique to orchestrate tasks in software development projects and is a established tool in many manufacturing industries, but is barely used to assist simulation-based projects.

Routine simulation tasks are often performed manually, following workflows that are rarely explicitly defined. This increases the efforts demanded from experts and scientists to take care of consistency issues, e.g. while parameterizing models or while validating simulation results on a continuous basis.

In this work we introduce a conceptual framework and prototype tool that improves simulation reproducibility and consistency by means of controlled automated model parameterization and simulation validation. The framework helps to structure the process of transforming values from the real system into parameters of the simulation models, and to systematically reuse those values during verification and validation tasks.

A practical prototype tool is implemented and tested for a case study in the ATLAS experiment [3] at CERN [4] where the Trigger and Data Acquisition (TDAQ) farm and communication network [5] plays the role of the real *system under study*.

Our proposed scheme diminishes the chances of introducing errors during model parameterization and enables new validation processes to be integrated with measurement databases that are populated on a continuous basis with the TDAQ system daily operation.

II. THE DEVS MODELING AND SIMULATION FRAMEWORK

The Discrete Event System Specification (DEVS [1]) is a mathematical formal specification based on general systems theory for modeling and simulation of discrete, continuous and hybrid systems [2], [6]. Since its first specification in 1976 [7] DEVS-based tools have been implemented in several programming languages and applied to a wide range of areas in nature, physics, engineering, computing, etc. The formal specification allows for analytic manipulation, offering hierarchical composition of structural (coupled) and behavioral (atomic) models defined by compact tuples of mathematical sets and functions.

A DEVS-based simulation platform [8] was developed to reproduce the TDAQ network behavior under different conditions, evaluate candidate changes for the network control algorithms before their commissioning, and analyze simulation data to detect potential unanticipated behaviors.

III. MODELING AND SIMULATION METHODOLOGY IN THE ATLAS EXPERIMENT

The TDAQ system is in charge of reading out, collecting, and processing in real time vast amounts of physics data produced by the ATLAS detector at CERN [3]. The flow of incoming data is slotted in smaller data structures called physics "Events". ATLAS generates Events at 40 MHz, yielding a raw throughput of approximately 60 Terabyte/s, which is filtered at TDAQ system to store permanently only a fraction of relevant Events (at a rate of 1 kHz, approximately 1 Gigabyte/s).

The TDAQ system is composed of several parallel applications, which collect data and run physics algorithms to reconstruct the Events from smaller data fragments. Applications are hosted across roughly 2000 multicore servers that communicate over a 10 Gbps Ethernet-based network [9] structured with approximately 100 switches. The applications, data control algorithms and network design are in constant evolution. The effect of candidate changes is hard to anticipate, requiring thorough engineering processes.

An iterative and incremental simulation methodology (coherent with the DEVS formal framework) is used to focus each iteration on specific goals and to enable flexibility for choosing the degree of accuracy required for each evaluation.

This methodology strictly separates the entities System, Model, and Simulation and relates them formally by means of the DEVS formalism: the System is first experimented with and then a DEVS model is built, meanwhile Model and System properties can be formally verified. The Model is afterwards read by the Simulator and, according to the DEVS specification (formally verified [1]) a simulated output trajectory is generated that can be validated against the initial experiments with System. Experimental frameworks and parameters are defined for each of the three entities in order for this cycle to be formally correct.

IV. Conceptual Framework

We define an architecture that relies on a conceptual framework to transform values from the system under study into values of the simulation model and vice versa. We categorize these values as *parameters* (values used to configure either the system or the simulation) and *metrics* (logged values for dynamic variables, either monitored on the system or produced by each simulation).

Figure 1 shows several *relationships* between the real system's values and the simulation values, both for *parameters* and *metrics*.

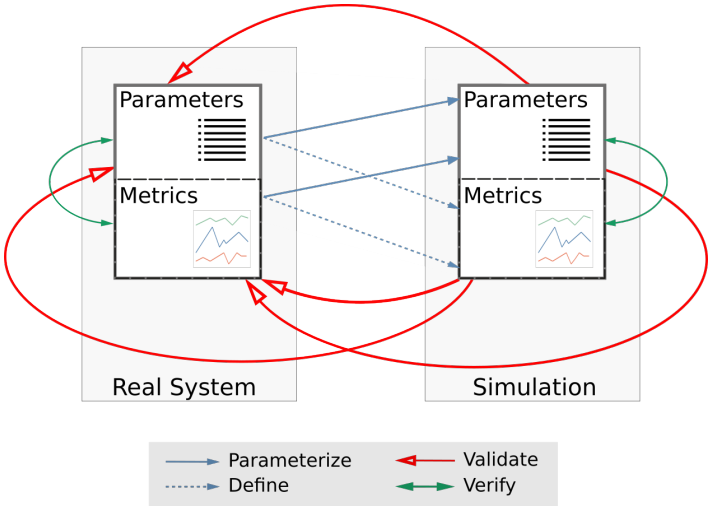


Figure 1: MASADA Conceptual Framework.
Relationships between different data values (Parameters or Metrics) according to their domains (Real System or Simulation)

We define the relationships in terms of the transformations needed to make values of one domain suitable for the other domain. Relationships are in turn categorized according to three *aspects*, depending on the nature of each given bond:

Type We define type as either *parameterization*, *validation* or *internal verification*. The type of the relationship depends on the domain and range of the relationship. It also defines the kind of tasks to be performed during the simulation phase, and whether it is carried before, after or independently from the simulation execution.

Cardinality We define *one-to-one*, *one-to-many*, *many-to-one* and *many-to-many* relationships between domains. This depends on the type of variable that comes into play and it has an important role in the domain and range of the transformation function applied.

Transformation A transformation function could be a statistical operation (e.g. a many-to-one averaging of multiple time-series into one simulation parameter), or simply the identity for the one-to-one or one-to-many cases.

Aspects describing the relationships are interdependent. Each implementation of an instance of this framework will depend on the given data sources, the simulated system and the goals of the simulation experiments. Results are generated from raw data via a relationship that involves a *Transformation* and a *Type*. It is an association of values. On the one hand we have the unprocessed data, and in the other hand the exact process required to transform it into a suitable format for its *validation, verification, parameterization* or *definition* of new data.

For the *parameterization and definition types*, transformations are needed to extract values from the configuration of the real system and translate them into configurations of the simulation models. Example transformation functions for one-to-one relationships are scaling procedures. An example for many-to-one relationships is the lumping of several metrics down to a single simulation parameter by means of aggregation procedures. These values are refilled into raw values by the relationship entity. The distinction between them is their targets: in one case *parameters* are generated, in the other case *metrics* are defined.

As for the *validation type*, transformations enable the comparison of many-to-many relationships like system metrics against simulation metrics, many-to-one like metrics to parameters, one-to-one for simulation parameters against systems parameters, and one-to-many for simulation parameters against systems metrics.

These transformations rely on data analysis techniques supporting the validation analyses that will ultimately explain the degree of accuracy with which simulations approximates reality. Descriptive and summary statistics are good examples of transformations.

In the *internal verification type*, metrics and parameters are checked for internal consistency within a given domain (system or simulation). As for the system domain, it requires knowledge about the system constraints, e.g. metrics that should not exceed certain parameterized value. As for the simulation domain, model consistency checks detect whether e.g. a metric produced by the simulator is consistent with a parameter that specifies statistical properties on said metric.

This conceptual framework allows to store the parameters, metrics and relationships used for a given experiment, constituting part of the evidence needed to replicate it in the future. By so doing we enhance reproducibility, foster the reuse of sound data management techniques, and reduce the complexity of the simulation execution tasks.

V. The MASADA Tool

To test the conceptual framework we implemented the *Modeling and Simulation Automated Data Analysis* (MASADA) tool, a python2.7 command line application using SQLite3 for data handling support.

The tool implements an Extract, Transform, Load architectural pattern that connects with ATLAS experiment data sources. In particular, we connect to PBeast [10] via its REST APIs. The application transforms the extracted information into Python Objects that are persisted by the ObjectRelationMapper (ORM) layer. This allows to extend the datasources and the data types easily upgrading our simulation parameters and metrics objects to increase precision.

In fig. 2 we present an architectural and procedural view of *MASADA*. The left to right path corresponds to the parameterization process discussed in fig. 1. This is how information flows from the real system

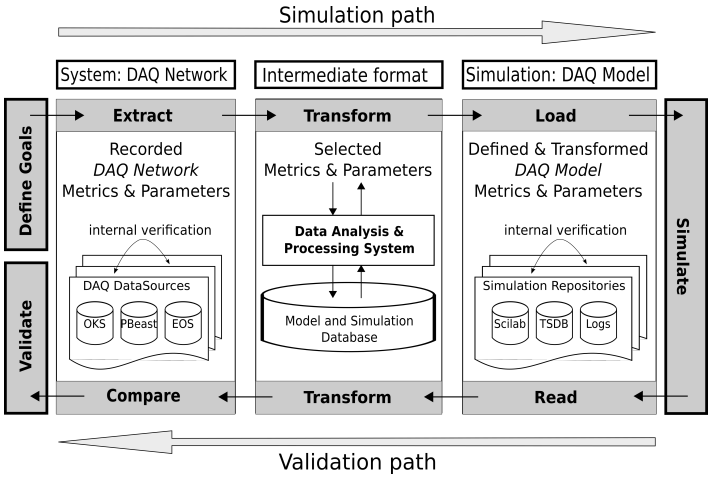


Figure 2: MASADA Architectural View

into the simulated one. The right to left direction explains the process of validation and how information is transformed from the simulation repositories (Scilab numerical software, Open Timeseries Database or log files) until it is suitable for comparison.

VI. Validation Results

We ran multipoint simulations, where each point represents a simulation parameterized to reproduce a single physics Run, with values taken from the TDAQ network. In this section we show how historical validation interacts with parameters variability and multipoint validation techniques: we simulate imitating conditions from the past subjected to subtle changes. Then, results are compared against metrics from the real system. In fig. 3 we plot the average amount of accepted and rejected events aggregated by racks of servers. The plot compares one second of TDAQ system operation. We observe that the average amount of processed events in the real system are consistently 10% higher than in the simulation. Meanwhile, fig. 4 shows that data presents an acceptable degree of similarity in terms of skewness. This graph was obtained by creating a normal probability plot to compare each measurement with the quantiles of the normal distribution.

The validation of these data sets was very conclusive in order to guide future modeling efforts to match the real system with more precision. The symptoms we diagnosed with this procedure are scaling mismatches related to the Events' throughput. The introduction of new metrics helps to determine which dimensions of the simulation model require further refinements.

Moreover, MASADA can keep running the same validation process automatically on a continuous basis as new physics collisions take place in the ATLAS detector, enriching the system's evidence databases.

VII. Conclusions and Future Work

We introduced MASADA, a framework for continuous simulation validation that tackles important aspects of validity, reproducibility and maintainability by automating error prone data analysis and transformation tasks. The tool enables continuous simulation validation in a very particular context, the ATLAS experiment at CERN. Having automated the parameters extraction and metrics transformation considerably diminished configuration time, allowing for quicker responses in the face of changing operation scenarios. The latter allows for more complex validation techniques like parameter variability, providing useful extra insights about the quality of the simulation model.

MASADA also enables the reuse of best practices in data comparison allowing for two way validations: from the real system to the simulation (e.g.: how do we validate a relevant system metric against simulation outcomes?) and from the simulation to the system measurements (e.g.: how do we validate an interesting, unexpected simulation outcome against evidences in the real system?)

There are many validation techniques available, while selecting those that provide the best information requires careful, often crafty treatments. This is an usual scenario in simulation projects in general. MASADA offers a platform within which the best validation strategies can be encoded building up a consistent and reusable repository.

The extension of the tool is currently planned to add integration with big data-specific back-ends, such as distributed file systems and databases, and with existing well-known scientific workflow systems.

References

[1] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[2] G. A. Wainer and P. J. Mosterman, *Discrete-event modeling and simulation: theory and applications*. CRC Press, 2010.

[3] ATLAS Collaboration, "The ATLAS experiment at the CERN large hadron collider," *Journal of Instrumentation*, vol. 3, no. 8, 2008.

[4] D. Pestre, "L'organisation européenne pour la recherche nucléaire (CERN): Un succès politique et scientifique," *Vingtieme siecle. Revue d'histoire*, pp. 65–76, 1984.

[5] ATLAS Collaboration, "The ATLAS high-level trigger, data acquisition and controls technical design report," 2003.

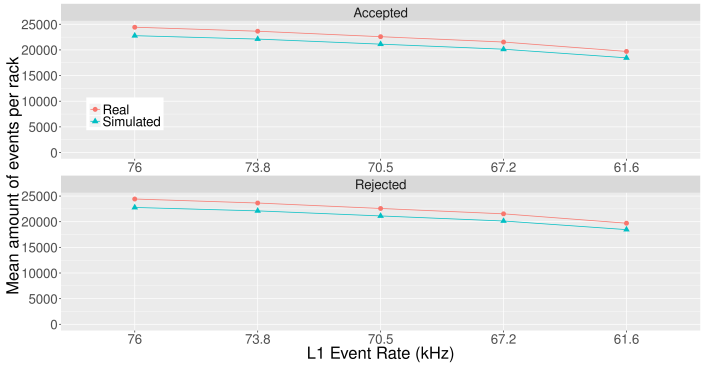
[6] F. E. Cellier and E. Kofman, *Continuous system simulation*. Springer Science & Business Media, 2006.

[7] B. Zeigler, "Theory of modeling and simulation. John Wiley & Sons," *Inc., New York, NY*, 1976.

[8] M. Bonaventura, D. Foguelman, and R. Castro, "Discrete event modeling and simulation-driven engineering for the ATLAS data acquisition network," *Computing in Science & Engineering*, vol. 18, no. 3, pp. 70–83, 2016.

[9] M. Astigarraga Pozo, "Evolution of the ATLAS trigger and data acquisition system," in *Journal of Physics: Conference Series*, vol. 608, no. 1. IOP Publishing, 2015, p. 012006.

[10] A. D. Sicoe, G. L. Miotto, L. Magnoni, S. Kolos, and I. Soloviev, "A persistent back-end for the ATLAS TDAQ online information service (p-beast)," in *Journal of Physics: Conference Series*, vol. 368, no. 1. IOP Publishing, 2012, p. 012002.



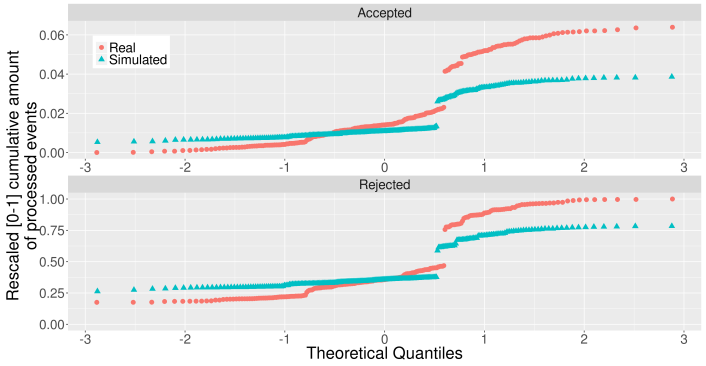


Figure 3: Real vs Simulated Amount of Events Processed. **Figure 4:** Real vs Simulated Amount of Events Processed. Probability Plot