MASADA: A MODELING AND SIMULATION AUTOMATED DATA ANALYSIS FRAMEWORK FOR CONTINUOUS DATA-INTENSIVE VALIDATION OF SIMULATION MODELS

Daniel Foguelman, Matías Bonaventura and Rodrigo Castro
CERN, CH-1211 Geneva 23, Switzerland and Universidad de Buenos Aires, Buenos Aires, Argentina
{dfoguelman,mbonaventura,rcastro}@dc.uba.ar

July 30, 2016

KEYWORDS

Simulation, Data Analysis, Continuous Validation, Process Automation, DEVS, Integration.

ABSTRACT

Complex networked computer systems are subjected to upgrades on a continuous basis. Modeling and simulation of such systems helps with guiding their engineering processes when testing design options on the real system is not an option. Too often many systems' 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.

In this work we present 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. We tested MASADA 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.

ATL-DAQ-PROC-2016-014
24 September 2016

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 (Zeigler et al., 2000; Wainer and Mosterman, 2010).

Nevertheless, 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 (Collaboration, 2008) at CERN (Pestre, 1984) where a data acquisition farm and communication network called TDAQ (Collaboration et al., 2003) 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 daily operation.

BACKGROUND AND RELATED WORK

The DEVS Modeling and Simulation Framework

The Discrete Event System Specification (DEVS (Zeigler et al., 2000)) is a mathematical formal specification based on general systems theory for modeling and simulation of discrete, continuous and hybrid systems (Wainer and Mosterman, 2010; Cellier and Kofman, 2006). Since its first specification in 1976 (Zeigler, 1976) 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 (Zeigler et al., 2000; Wainer and Mosterman, 2010).

We developed a DEVS-based simulation platform (Bonaventura et al., 2016) 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.

Modeling and Simulation Methodology in the ATLAS Experiment

The Trigger and Data Acquisition (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 (Collaboration, 2008). The constant flow of incoming data is slotted in logical 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 to store permanently only a fraction of relevant Events (at a rate of 1 kHz, approximately 1 Gigabyte/s).

TDAQ 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 an 10 Gbps Ethernet-based network (Astigarraga, 2015) 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 (Zeigler et al., 2000)) 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.

Scientific Workflows, Data Management, and Simulation Reproducibility

Workflow Management Systems are software platforms that execute well-defined sequences of steps and tasks. They provide an infrastructure for the creation, maintenance, and orchestration of such tasks (e.g. executing a script, performing data analysis, running a simulation, launching long-run intensive jobs, etc.)

In many simulation projects repetitive tasks can run on a continuous basis.

Parameterization of the simulation might require to query, gather together and analyze information coming from different systems. When simulations are run distributed on clusters of nodes, the launching of the simulation itself requires detailed extra configuration. Data produced by the simulation usually requires extra processing to correlate variables, search for certain desired/undesired patterns, produce plots, etc. Later, for validation purposes the simulated data needs to be contrasted against real measurements, which again might need collecting extra information from the real systems

and manipulating these data-sets to produce meaningful results.

The adoption of workflow systems in science is a natural way of organizing these steps, providing means for preserving the reproducibility of scientific methods by encoding explicitly their processes (Zhao et al., 2012). Nevertheless, most workflow systems demand important maintenance efforts (Ribault and Wainer, 2012)

Data Management in Science

In most scientific areas there is an impact of the big-data era bringing new needs to analyze, process, and understand massive amounts of information in a reliable way. Bigger, higher dimensional data-sets are constantly made available to scientists. Computer simulations can generate huge data repositories, raising the need for appropriate methods to manage such information efficiently. As data quality gets enhanced, data analysis methods need to evolve to detect subtle effects in the more and more extensive data-sets. Investments on enhancing management and analysis techniques of big-data scientific repositories can open up the possibility of finding new, previously overlooked scientific evidences (Kargín et al., 2013; Gray et al., 2005). Thus, a focus in data exploration and interactive data analysis and integration can become key to move M&S-based science forward.

Reproducibility and Evidence-Based Simulation

Reproducibility in M&S is the ability to recreate a simulation run obtaining the exact same results across several runs when parameterizing the model with a same set of initial values (regardless of interpretations on the results). In turn, reproducibility is key for peer review-based model verification.

Furthermore, there is a need for testing the validity of a model regarding its ability to reproduce faithfully system's behavior under changing conditions of operation.

Evidence-based simulation is the procedure by which a model is configured so as to mimic as close as possible the evidence that can be collected from measurements of the real system, both in terms of static parameters and of dynamic variables.

Hence, there is also a need for consistently share and trace the data used to configure and validate simulation models to support both simulation reproducibility and evidence-based simulation studies.

PROPOSED SOLUTION

In the current section we propose a solution to tackle systematically the previously introduced issues. We aim at coping with an increasing magnitude of the parameters space, the error prone process of simulation configuration, the potential lack of a systematic validation process, the need for better validation methods, and the inefficiency of manually executed workflows.

We propose a framework to validate simulations in a coherent and comprehensive way, less error prone, robust and scalable. It is based on dynamically enlarged evidence databases tightly interconnected with our system under study, i.e. the TDAQ network.

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 (bottom) shows several *relationships* between the real system's values and the simulation values, both for *parameters* and *metrics*.

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.

In Figure 1 (top) we depict an UML diagram of the main MASADA entities. The system under study and the simulation model generates raw data. This *raw data* is a composition of either *Metric* or *Parameter* values. 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. Another example for many-to-one relationships is the

lumping of several metrics down to a single simulation parameter by means of aggregation procedures. This values are refilled into raw values by the relationship entity. The distinction between them are 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.

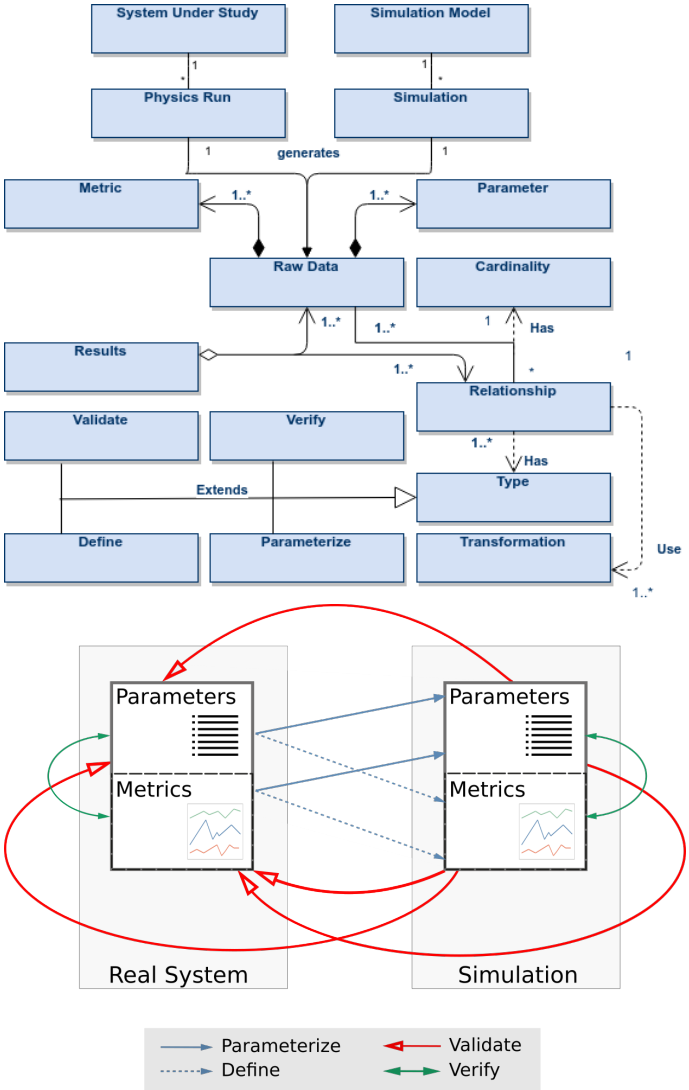

Figure 1: MASADA Conceptual Framework. Top: Main entities. Bottom: Relationships between different data values (Parameters or Metrics) according to their domains (Real System or Simulation)

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. As it will be detailed in The Process of Simulation Validation, some of the analyses used are predictive validation, historical data validation, parameter variability, hypothesis testing and confidence intervals.

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.

The Process of Simulation Validation

The process to determine whether the simulation model and its associated data-sets are a good representation of the real world is called validation.

Validation techniques can be split into quantitative and qualitative analysis.

Qualitative validation takes place when we compare simulated data against real measurements using graphical techniques like histograms, scatter plots, behavior graphs. These are used to cross compare multiple experimental conditions (Ni et al., 2004; Sargent, 1996). Quantitative techniques are used in order to measure error like the family of mean error analysis (squared, absolute squared or rooted error) or to understand the distribution of numerical data. Examples of numerical techniques are the use of confidence intervals (Sargent, 2010) or hypotheses testing (Balci and Sargent, 1981).

CASE STUDY FOR THE TRIGGER AND DATA ACQUISITION (TDAQ) NETWORK

In this section we describe the tools and techniques developed for a real world scenario, demonstrating an application of the MASADA conceptual framework. The case study builds upon a simulation model developed before for the ATLAS TDAQ network at CERN (Bonaventura et al., 2016).

We first describe overall details of the simulation model to understand the parameter space of the system, and then introduce the need for better parameterization techniques, systematic validation, and reproducibility. Afterwards, the tool is presented along with specific parameters, transformations, and validations techniques we integrated into the MASADA framework (including predictive validation, historical data validation, parameter variability, hypothesis testing and confidence intervals).

The case study is representative of common simulation

contexts where the size and complexity of the data-sets scale up becoming hard to manipulate manually.

The Modeled Data-Flow

The simulation model focuses on predicting the flow of data traversing the TDAQ network when incoming physics Events are analyzed and filtered in real-time. Each experiment in ATLAS is defined as a physics *Run*. Several applications are involved in this process, whose algorithms and parameters define the data-flow as follows:

- The first-level trigger (L1) subsystem filters Events using low-latency custom hardware, and temporarily stores accepted Events in the read-out system (ROS) nodes.
- For each accepted Event, the L1 sends a signal to the high-level trigger supervisor (HLTSV) node, indicating that new Events are ready to be further analyzed by a second round of software algorithms. The rate at which the L1 sends Events to the HLTSV is a key parameter to the simulation model and is defined as **L1EventRate**.
- The HLTSV assigns each Event to be processed to one of the available processing units (PU) that, in order to analyze an Event, request data to the ROS nodes through iterative steps (Event data is not requested all at once, but only most relevant regions first).
- Physics-specific software algorithms define how much data is requested and to which nodes at the ROS. This varies from Event to Event so for the simulation a distribution parameter is used defined as **ROSRequestRate**.
- Also the units of data stored in each ROS node (called fragments) is different for each Event and is parameterized in the simulation as a distribution called **FragmentSizes**.
- Each server of the TDAQ farm can host many PU applications (usually one per core) so the data collection manager (DCM) application is in charge of handling the communication of each server with the rest of the applications. In particular, the DCM uses a credit-based traffic shaping system to workaround a TCP Incast (Kulkarni and Agrawal, 2014) undesired effect that leads to increased delays.
- The amount of credits available in the DCM limits the in-flight requests on the network, and is a parameter called **DCMCredits** both for the real and the simulated system.

More than 100 other parameters are used in the simulation that affect the network's performance, such as the amount of accepted and rejected Events (**Accept/RejectRate**), and the time physic algorithms take to process each Event (**PUProcessingTime**).

Different metrics are used to validate the simulation, such as percentage of utilization of network links and processing nodes, buffer occupancies, total number of Events processed (either accepted or rejected), etc. Metrics are retrieved from the simulation output (several millions of values). In order to study what-if scenarios simulation parameters are swept yielding new sets of metrics for each simulation.

Challenges of the TDAQ Simulation and Validation Process

The parameter space in the TDAQ system is complex, challenging the safe and sound parameterization and validation of simulations.

The order of magnitude is near to 10^7 metric values tracked during each single physics Run, gathered by multiple monitoring systems and persisted into distributed databases. Each system retrieves metrics using different technologies, sampling periods, and time granularity. After each physics Run, about 8 hours of recorded information is available to be parsed and compared against new simulations.

Once relevant metrics are identified, multiple data sources and values need to be consolidated in order to produce new parameters and metrics that are suitable for the simulation model. These transformations are not trivial as they differ for each metric/parameter, possibly involving multiple stages of calculations.

Finally, each simulation execution produces huge amounts of information from different sub components that need to be post processed in order to obtain meaningful results. Simulation must ultimately be contrasted against real measurements (which in turn require consolidation from different sources) to assess the degree of accuracy of the simulation.

The option of performing manually these data manipulation operations is time-consuming and error-prone, and leaves no explicit documentation on how and why simulation parameters and metrics were chosen. In daily practice it can lead to ill-configured simulations, weaken simulation-based conclusions, and hamper scientific reproducibility.

Also, the real system behavior changes within the same Run. For example, along a Run different triggering algorithms are executed, modifying TDAQ's rates, processing times, Events filtered, etc. This arises the need of simulating multiple points for each Run, but each point requires its own set of correlated parameters to be carefully defined following the real system behavior. Moreover, several Runs are performed within the same day pursuing different goals (parameters can drastically change). If the simulation model aims at reproducing all ATLAS Runs the parameterization and validation process should not be time-consuming and allow for a continuous analysis of reproducibility accuracy.

These requirements lead us into the development of a new tool that extracts the information from the various real monitoring systems, and parameterizes simulations

based on statistically summarized data from each Run. Also, the tool allows the analysis of simulation results for the validation process by contrasting with the real system's measurements. This way, the new tool allows for continuously checking that the simulations reproduce the real system's behavior by automatizing the parameterization and validation process.

Selected Metrics, Parameters and Transformations

For the implementation of the framework we elicited some of the most interesting variables. The parameterization process is critical for the configuration of the simulation. The selection criteria was based on values which varied during real Runs like *L1EventRate* and *Accept and Reject rate*. *ROSRequestRate* and *FragmentSize* are values that represent part of the stochastic nature of the physics Runs.

Finally, the *DCMCredits* is a parameter that changes the system overall behavior, and it is manually configured in real Runs. For validation we used *Amount of processed events* to evaluate workload and performance. For internal verification we check that the *Amount of events processed* per second are less than the *L1EventRate*.

In Table 1 we detail the relationships between variables in the MASADA implementation. We detail the type of the relationship and the transformation function.

Table 1: Variables Transformations in the MASADA Scope. SuS=System under Study.

In SuS	Relationship	In simulation
Variable: DCM Credits		
Parameter	Parameterization	Parameter
Identity transformation		
Variable: L1 Event Rate		
Metric	Definition	Parameter
Average transformation		
Variable: Accept / Reject Rate		
Metric	Definition	Parameter
Count per Rack and then avg by time bin		
Variable: ROS Request Rate		
Metric	Definition	Parameter
Frequency analysis over the requested ROS nodes		
Variable: Fragment sizes		
Metric	Definition	Parameter
Normal estimators from sampled values		
Variable: Amount of Events processed		
Metric	Validate	Metric
Summation of events per rack and then avg		

Validation techniques

As discussed in The Process of Simulation Validation, the following subset of validation techniques were chosen from the literature:

Behavior plots To understand relationship between real and simulated values, like Scatter plots.

Boxplots To understand mean, median, extreme values and outlier distributions.

Probability plots For analyzing the distributions differences in shape. All previous fall into graphical techniques.

Non-parametric tests Like the KS Test, Pearson correlation coefficients.

Multipoint simulation is a precise example of Parameter Variability validation. It is carried by creating a single conceptual unit with the parameters and metrics logged during simulation. These techniques can act in combination with e.g. Historical Data Validation, Predictive Validation and Parameter Variability.

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 (Sicoe et al., 2012) 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.

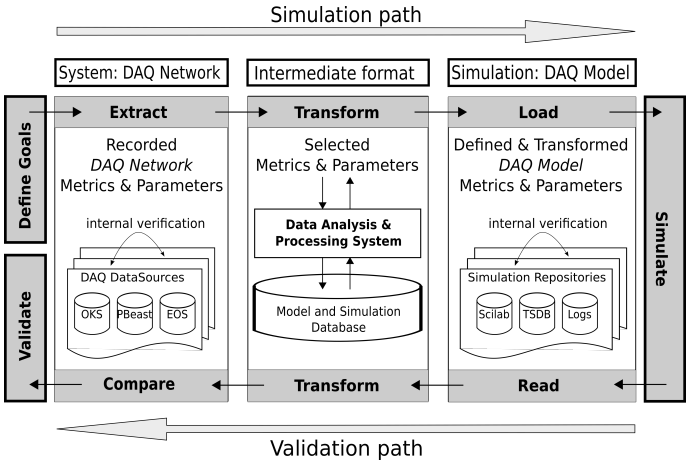

Figure 2: MASADA Architectural View

The use cases implemented in the tool are defined in the following tasks:

1. Initialization of the back-end (creation of common files, data bases, etc).
2. Configuration of a Physics Run.
3. Slicing of the experimental Run into smaller, configurable samples aggregated by time bin. This step is optional but very useful as it allows to retrieve multiple parameters and metrics for the same Run with a single command.

4. Retrieval of one or multiple data-sets for parameters and metrics.
5. Summarization of the values extracted from a Run into formats suitable for the simulation to consume.

In the 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 into the simulated one. In this case we extract the parameters **L1EventRate, ROSRequestrate, FragmentSizes, DCMCredits** and **Accept/Reject Rate** from the ATLAS TDAQ monitor systems: **OKS** (Alexandrov et al., 2001), **EOS** (Peters and Janyst, 2011) and **PBeast** (Sicoe et al., 2012).

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.

We prototype and test candidate validation techniques prior to integrating them into *MASADA* using the R statistical language. We then implement these techniques into the MASADA tool using the Pandas (McKinney, 2010) data analysis toolkit.

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 rack. The plot compares one second of TDAQ operation. We observe that the average amount of processed events in the real system are 10% higher than in the simulation.

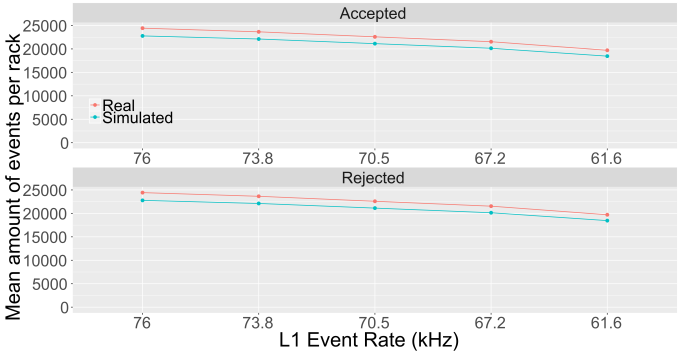

Figure 3: Real vs Simulated Amount of Events Processed. Behavior Plot

We analyze the statistic values summarized in the boxplots on fig. 4 for the same variable as in as fig. 3. Each boxplot corresponds to one different *L1EventRate* extracted from the same real experiment. We observe that real measurements show bigger dispersion between the first and third percentile while simulated values are much more compact, suggesting the need to further enhance models.

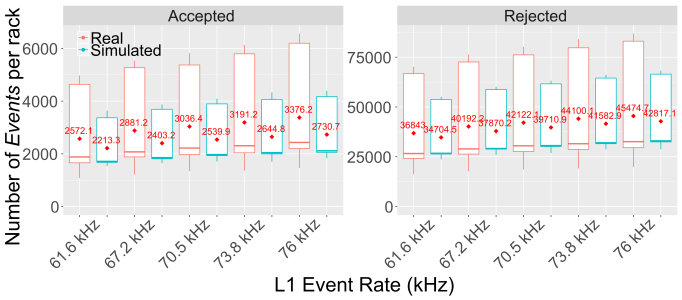

Figure 4: Real vs Simulated Amount of Events Processed. Boxplot

Meanwhile, fig. 5 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.

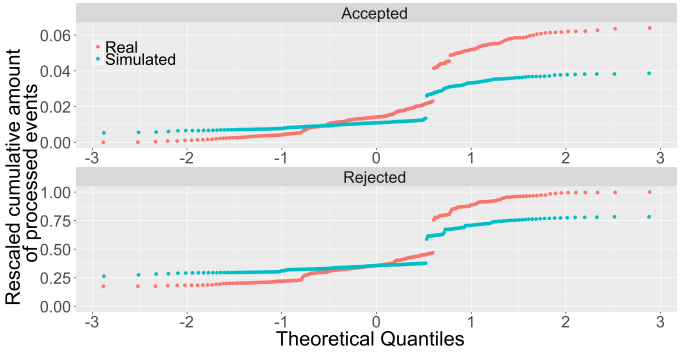

Figure 5: Real vs Simulated Amount of Events Processed. Probability Plot

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.

After the next iteration of the modeling tasks, MASADA will automatically repeat all the tasks depicted in the workflow of Figure 2, producing the next version of the validation plots presented in this section.

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.

CONCLUSIONS AND FUTURE WORK

In this 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.

Our tool enables continuous simulation validation in a very particular context, the ATLAS experiment at CERN.

Having automated the parameters extraction and metrics transformation considerably diminishes 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 and selecting the ones that give the best information needs careful selection and crafting. This is not only an ATLAS TDAQ simulation problem, but an usual scenario in simulation projects.

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 backends such as distributed file systems and databases, and with existing well-known scientific workflow systems.

References

I Alexandrov, A Amorim, E Badescu, D Burckhart-Chromek, M Caprini, M Dobson, R Hart, R Jones, A Kazarov, S Kolos, et al. ATLAS DAQ configuration databases. In *Proceedings of CHEP2001 Conference, Beijing, China*, pages 608–611, 2001.

ME Pozo (on behalf of the ATLAS Collaboration) Astigarraga. Evolution of the ATLAS trigger and data acquisition system. In *Journal of Physics: Conference Series*, volume 608, page 012006. IOP Publishing, 2015.

Osman Balci and Robert G Sargent. A methodology for cost-risk analysis in the statistical validation of simulation models. *Communications of the ACM*, 24(4):190–197, 1981.

Matías Bonaventura, Daniel Foguelman, and Rodrigo Castro. Discrete event modeling and simulation-driven engineering for the ATLAS data acquisition network. *Computing in Science & Engineering*, 18(3): 70–83, 2016. doi: http://dx.doi.org/10.1109/MCSE.2016.58. URL http://scitation.aip.org/content/aip/journal/cise/18/3/10.1109/MCSE.2016.58.

François E Cellier and Ernesto Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.

ATLAS Collaboration. The ATLAS experiment at the CERN large hadron collider. *Journal of Instrumentation*, 3(8), 2008.

ATLAS Collaboration et al. The ATLAS high-level trigger, data acquisition and controls technical design report. 2003.

Jim Gray, David T Liu, Maria Nieto-Santisteban, Alex Szalay, David J DeWitt, and Gerd Heber. Scientific data management in the coming decade. *ACM SIGMOD Record*, 34(4): 34–41, 2005.

Yağíz Kargín, Milena Ivanova, Ying Zhang, Stefan Manegold, and Martin Kersten. Lazy etl in action: Etl technology dates scientific data. *Proceedings of the VLDB Endowment*, 6(12):1286–1289, 2013.

Santosh Kulkarni and Prathima Agrawal. *Analysis of TCP Performance in Data Center Networks*. Springer, 2014.

Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

Daiheng Ni, John Leonard, Angshuman Guin, and Billy Williams. Systematic approach for validating traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board*, (1876):20–31, 2004.

Dominique Pestre. L'organisation européenne pour la recherche nucléaire (CERN): Un succès politique et scientifique. *Vingtieme siecle. Revue d'histoire*, pages 65–76, 1984.

Andreas J Peters and Lukasz Janyst. Exabyte scale storage at CERN. In *Journal of Physics: Conference Series*, volume 331, page 052015. IOP Publishing, 2011.

Judicaël Ribault and Gabriel Wainer. Using workflows and web services to manage simulation studies (wip). In *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium*, page 50. Society for Computer Simulation International, 2012.

Robert G Sargent. Verifying and validating simulation models. In *Proceedings of the 28th conference on Winter simulation*, pages 55–64. IEEE Computer Society, 1996.

Robert G Sargent. A new statistical procedure for validation of simulation and stochastic models. 2010.

Alexandru D Sicoe, Giovanna Lehmann Miotto, Luca Magnoni, Serguei Kolos, and Igor Soloviev. A persistent back-end for the ATLAS TDAQ online information service (p-beast). In *Journal of Physics: Conference Series*, volume 368, page 012002. IOP Publishing, 2012.

Gabriel A Wainer and Pieter J Mosterman. *Discrete-event modeling and simulation: theory and applications*. CRC Press, 2010.

Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

BP Zeigler. Theory of modeling and simulation. Jhon Wiley & Sons. *Inc., New York, NY*, 1976.

Jun Zhao, José Manuél Gómez-Pérez, Khalid Belhajjame, Graham Klyne, Esteban Garcia-Cuesta, Austin Garrido, Kristina Hettne, Maree Roos, David De Roure, and Carole Goble. Why workflows breakunderstanding and combating decay in taverna workflows. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–9. IEEE, 2012.