SIERRA: A Modular Framework for Research Automation

Demonstration Track

John Harwell
University of Minnesota
United States
harwe006@umn.edu

London Lowmanstone
University of Minnesota
United States
lowma016@umn.edu

Maria Gini
University of Minnesota
United States
gini@umn.edu

ABSTRACT

Modern intelligent systems researchers form hypotheses about system behavior and then run experiments using one or more independent variables to test their hypotheses. We present SIERRA, a novel framework structured around that idea for accelerating research developments and improving reproducibility of results. SIERRA makes it easy to quickly specify the independent variable(s) for an experiment, generate experimental inputs, automatically run the experiment, and process the results to generate deliverables such as graphs and videos. SIERRA provides reproducible automation independent of the execution environment (HPC hardware, real robots, etc.) and targeted platform (arbitrary simulator or real robots), enabling exact experiment replication (up to the limit of the execution environment and platform). It employs a deeply modular approach that allows easy customization and extension of automation for the needs of individual researchers, thereby eliminating manual experiment configuration and result processing via throw-away scripts.

KEYWORDS

Scientific Method, Research Automation, Simulation, Real Robots

ACM Reference Format:
John Harwell, London Lowmanstone, and Maria Gini. 2022. SIERRA: A Modular Framework for Research Automation : Demonstration Track. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

In modern intelligent agents research, reproducibility is crucial. Configuring experiments for testing theories can be time consuming, and further compounded by the need to deal with different configurations for specific platforms (e.g., ROS [11]) and execution environments (e.g., SLURM [12] clusters). Ad hoc tools and scripts are frequently developed to meet these needs on a per-project basis, and reused or modified as needed between projects. Scripts for processing and visualizing experimental results are developed similarly. In this paper, we present SIERRA[1], a framework designed to automate the process of hypothesis testing and results processing. SIERRA is complementary to other reproducibility-enhancing tools for AI research such as Robotarium [9] and IEEE Code Ocean.

SIERRA is *modular*, and can be extensively customized to meet a diverse range of researcher needs. Originally developed for robotics research, its modular design makes it capable of supporting almost any platform and execution environment in AI research.

SIERRA experiments are fully *reproducible*, regardless of the platform targeted and the execution environment on which the experiments are run. SIERRA currently supports the ARGoS [10], Gazebo [7], and ROS [11] platforms. To the best of our knowledge no automation exists for these platforms for hypothesis testing and results processing—crucial parts of research. Some partial automation of Webots [8] was done in [2]; such automation is a subset of SIERRA's capabilities. Independent from the targeted platform, SIERRA currently supports several execution environments for running experiments: PBS/SLURM clusters, real robots, and the local machine. New execution environment or platform support can be easily added as python modules. Through automation, SIERRA streamlines the process of publishing reproducible research: all automation provided by SIERRA is idempotent.

SIERRA is a *cohesive framework* for end-to-end automation of the scientific method in intelligent systems research. SIERRA automates the pipeline described in Section 2 and Fig. 1, eliminating the need for throw-away scripts, greatly reduces menial re-configuration of experimental inputs across platforms and execution environments, and provides a uniform interface for generating camera-ready deliverables such as graphs and videos for inclusion in academic papers. Essentially, SIERRA handles the "backend" parts of research, allowing researchers to focus on the "research" aspects: developing theories and testing hypotheses via experimental evaluation.

2 SIERRA OVERVIEW

SIERRA supports *research queries*: a hypothesis expressed as a query of an independent variable over some range, expressed in a user-defined command line syntax (e.g., `population_size.Log8` for representing varying the number of agents across {1,2,4,8}). The expressed query is used to define a *batch experiment*: a set of *experiments* each with a different value of the independent variable (number of agents in the above example). Each experiment consists of one or more *experimental runs* which can be simulation runs or real robot trials. For some types of research, multiple experimental runs are required due to randomness (e.g., imperfect sensors/actuators on real robots); SIERRA provides configurations for managing this complexity and for generating and plotting statistical distributions over results, and handles all necessary platform and execution environment configuration. However, it does not attempt to homogenize configurations such that the results of a research query are the same, regardless of platform and execution environment (which is not possible in general).

SIERRA has been used for several publications in top conferences [3, 5] using ARGoS and PBS/SLURM HPC clusters. In this

[1] https://github.com/swarm-robotics/sierra.git

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

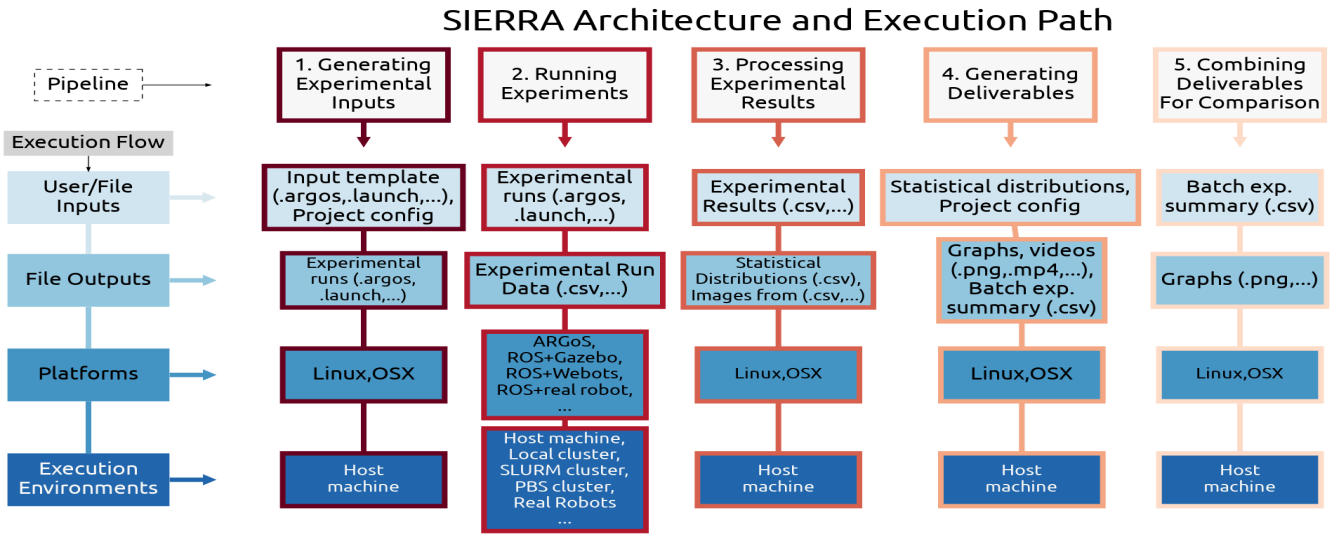

Figure 1: Architecture of SIERRA, organized by pipeline stage. Pipeline stages are listed left to right, with an approximate joint architectural/functional stack from top to bottom for each stage. "…" indicates areas where SIERRA is designed via python plugins to be easily extensible. "Host machine" indicates the machine SIERRA was invoked on.

work, we further demonstrate its utility by generating experiments for ROS and real robots [2]. We also demonstrate how SIERRA can be used to explore independent variables in a short demonstration [3]. SIERRA addresses the need for reproducibility and research acceleration in intelligent systems, and we therefore strongly argue for its inclusion in any researcher's toolbox.

2.1 Experimental Input Generation

To generate the batch experiment, a template XML file is modified according to the research query, with one experiment generated for each "value" of the independent variable(s), so that by comparing results across experiments in the batch, changes in behavior in response to the different "values" can be observed. Each "value" may correspond to a single change to the template via (e.g., `population_size.Log8` for changing the number of agents), or it can correspond to multiple changes (e.g., testing across a set of environments which require setting multiple XML parameters for each one). SIERRA also supports changing additional parts of the template input file uniquely for each experiment in the batch, or uniformly for all experiments, providing unparalleled expressiveness to support research automation through experiment generation.

2.2 Running Experiments

Experimental run inputs are executed on some simulator or real-robot *platform* in some *execution environment*. For example a researcher may want to run simulations targeting ARGoS [10] on a SLURM managed cluster, or to run experiments generated from the *same* research query on the ROS [11] platform with a real robot (e.g., Turtlebot3 [1]). Switching between platforms and/or execution environments naturally requires some code changes (e.g., cross-compiling). However, a lot of additional researcher time is spent figuring out how to get an experiment to run within a given environment, which slows down the actual research. SIERRA handles this complexity, reducing the burden on researchers.

2.3 Processing Experimental Results

Outputs from executed experiments are processed. This includes statistical distribution generation across experimental runs for each experiment in a batch, as well as across experiments in a batch, and converting output .csv files into images which can be stitched together into videos during stage 4.

2.4 Generating Deliverables

Processed experimental results are used to generate deliverables to accompany or be part of published research. This can include graphs or videos showing different aspects of the system's response to the research query. SIERRA's automation in this stage makes it easy to modify a specific graph or video, if, for example, one needs to be modified at a reviewer's request, eliminating the tedious process of locating previously written throw-away scripts to regenerate it, again saving time. Furthermore, this automation also improves reproducibility: with a single SIERRA command on a properly configured environment researcher B could reproduce the exact results of researcher A if both were using SIERRA and they had access to researcher A's agent code.

2.5 Deliverable Comparison

After deliverable generation, multiple deliverables are combined to provide side-by-side comparisons. For example, comparing the performance of a new reinforcement learning method against the current state-of-the-art method, each of which was independently generated during the previous stage may be desirable.

[2] https://www-users.cse.umn.edu/~harwe006/showcase/aamas-2022-demo
[3] https://www-users.cse.umn.edu/~harwe006/showcase/aamas-2022-demo/#demonstration-experimental-results-visualization-stage-4

REFERENCES

[1] Robin Amsters and Peter Slaets. 2020. Turtlebot 3 as a Robotics Education Platform. In *Robotics in Education*, Munir Merdan, Wilfried Lepuschitz, Gottfried Koppensteiner, Richard Balogh, and David Obdržálek (Eds.). Springer International Publishing, Cham, 170–181.

[2] Matt Franchi. 2021. Webots.HPC: A Parallel Robotics Simulation Pipeline for Autonomous Vehicles on High Performance Computing. arXiv:2108.00485

[3] John Harwell and Maria Gini. 2019. Swarm Engineering Through Quantitative Measurement of Swarm Robotic Principles in a 10,000 Robot Swarm. In *Proc. Twenty-Eighth Int'l Joint Conference on Artificial Intelligence, IJCAI-19*. 336–342.

[4] John Harwell and Maria Gini. 2021. Improved Swarm Engineering: Aligning Intuition and Analysis. *IEEE Transactions on Robotics* (2021).

[5] John Harwell, London Lowmanstone, and Maria Gini. 2020. Demystifying Emergent Intelligence And Its Effect On Performance In Large Robot Swarms. In *Proc. Autonomous Agents and Multi-agent Systems (AAMAS)*. 474–482.

[6] John Harwell, Angel Sylvester, and Maria Gini. 2021. Characterizing The Limits of Linear Modeling of Non-Linear Swarm Behaviors. arXiv:2110.12307 [cs.RO]

[7] Nathan Koenig and Andrew Howard. 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3. IEEE, 2149–2154.

[8] O. Michel. 2004. Webots: Professional Mobile Robot Simulation. *Journal of Advanced Robotics Systems* 1, 1 (2004), 39–42.

[9] Daniel Pickem, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt. 2016. The Robotarium: A remotely accessible swarm robotics research testbed. arXiv:1609.04730 [cs.RO]

[10] Carlo Pinciroli et al. 2012. ARGoS: a Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems. *Swarm Intelligence* 6 (12 2012), 271–295.

[11] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. 2009. ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software.

[12] Andy B. Yoo, Morris A. Jette, and Mark Grondona. 2003. SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing*, Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 44–60.