RAPID: A Rapid Prototyping Platform for Industrial Automation

Sunghoon Hong, Junseok Park, Whiyoung Jung, Deunsol Yoon, Woohyung Lim, Soonyoung Lee, Kanghoon Lee

LG AI Research
Seoul, South Korea
{sunghoon.hong, frank.park227, whiyoung.jung, dsyoon, w.lim, soonyoung.lee, kanghoon.lee}@lgresearch.ai

Abstract

Industrial automation in smart logistics and factories requires simulation platforms that support rapid environment building before costly physical deployment. Yet existing tools often require substantial expertise, complex setup, and long configuration times, hindering agile prototyping. We present RAPID, a simulation platform with two components: layout design, which enables intuitive visual configuration of factory layouts, and behavior simulation and validation, which allows users to attach behavior models and evaluate system performance. RAPID lowers the entry barrier to industrial simulation, letting users apply existing behavior models or trained reinforcement learning (RL) agents to new layouts with minimal effort. This approach lets practitioners prototype facilities in minutes rather than weeks and gives researchers a standardized environment for benchmarking multi-agent RL and coordination algorithms. By combining rapid design with simulation-based validation, RAPID accelerates automation development from concept to implementation.

Introduction

Industrial automation leverages fleets of autonomous agents in smart logistics and factories to achieve unprecedented operational efficiency (Wurman, D'Andrea, and Mountz 2008; Krnjaic et al. 2024; Morais et al. 2025). Success in these applications requires careful optimization of two interdependent dimensions: facility layouts—the spatial configuration of infrastructure elements such as racks, conveyors, and robots—and control strategies that govern agent behaviors. Given the complexity of these systems and the substantial investment required for physical deployment, simulation has become essential for industrial automation. It enables practitioners to model various configurations, test control algorithms, and predict system performance without the costs, making it the cornerstone of automation development.

Existing simulation platforms prioritize modeling fidelity and flexibility over accessibility. AnyLogic (Borshchev 2013) offers powerful behavior modeling through finite automata (Hopcroft, Motwani, and Ullman 2006), enabling detailed specification of predefined agent behaviors. NVIDIA Omniverse (NVIDIA 2025) delivers photorealistic digital twins with advanced physics simulation and ray-traced graphics. However, this sophistication requires substantial computational resources, extensive technical expertise, and prolonged setup times. Consider a factory manager who needs to quickly evaluate the impact of adding new robots or reconfiguring conveyor systems—what should be a straightforward assessment becomes a time-consuming technical project. This reveals a critical gap: practitioners need rapid prototyping tools, but existing platforms require them to sacrifice speed for the sophisticated capabilities.

To bridge this gap, we developed RAPID (**RA**pid **P**rototyping platform for **InD**ustrial automation), a unified platform that combines powerful simulation capabilities with an intuitive, visual, and interactive workflow. RAPID serves two distinct user communities. *Industry practitioners* can prototype and validate facility designs in minutes rather than weeks, modifying layouts, testing operational strategies, and applying existing control policies or trained reinforcement learning (RL) agents to new configurations without extensive reconfiguration. *Researchers* can develop and evaluate multi-agent RL algorithms in a versatile environment. They can train RL policies and test their approaches across diverse industrial scenarios, all with reproducible results. By addressing both practical deployment and rigorous evaluation needs, RAPID accelerates the entire automation development cycle from concept to implementation.

The RAPID Platform

RAPID's client-server architecture is engineered for a responsive user experience, featuring a JavaScript front-end (PixiJS, Plotly.js) for fluid, interactive visualization and a high-performance Python back-end (FastAPI, Gradio, custom simulation core) for robust simulation execution. The platform's core philosophy is to separate design from analysis, which is realized through two dedicated views: the *environment builder* for rapid layout design and the *simulation dashboard* for validation as illustrated in Figure 1.

Rapid Layout Design

A core design principle of RAPID is the rapid modeling of complex layouts from scratch. This is enabled by the platform's interactive, grid-based visual editor.

Environment Builder Interface For **layout configuration**, users first define the workspace by setting an appro-

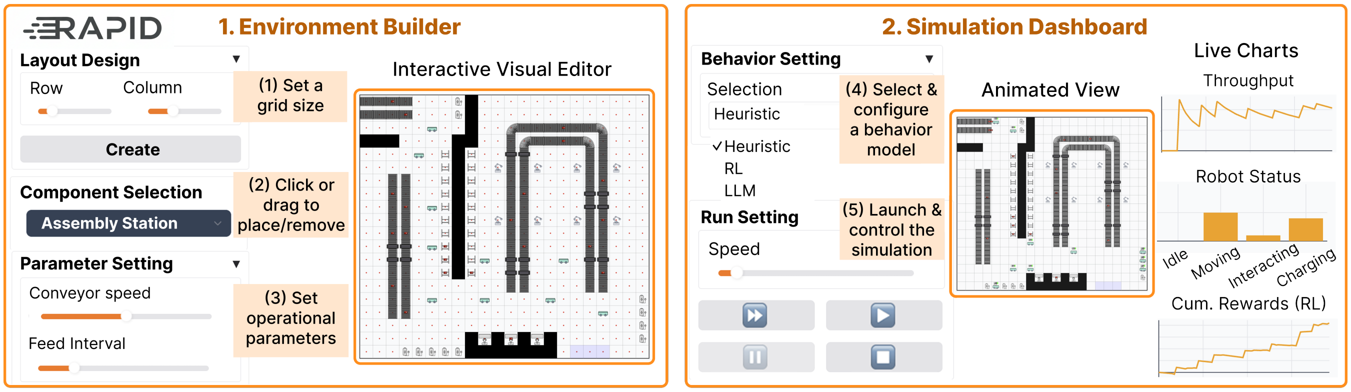

Figure 1: RAPID user interface. The Environment Builder (left) facilitates intuitive layout design, and the Simulation Dashboard (right) visualizes real-time agent behavior and key performance metrics.

priate grid size. They then populate this grid by selecting components—such as robots, charging stations, and storage shelves—from a dropdown menu and placing them via clicking or dragging. For **parameter configuration**, users specify the operational conditions by setting key parameters like the order queue size. For reproducibility and advanced customization, all designs can be exported and imported as YAML files.

Sample Scenarios Although the environment builder allows the design of diverse scenarios, we provide the following four scenarios as representative examples: *order picking* and *stowing* as core inbound/outbound logistics tasks; an integrated *fulfillment center* as a smart logistics model; and an *automated production line* as a scenario demonstrating smart factory applicability.

- *Order Picking:* Robots deliver inventory shelves to a stationary operator, who picks items to fulfill orders. The robot then returns the shelf to storage, aiming to minimize overall order completion times.
- *Stowing:* Robots bring shelves to operators to be replenished with new inventory before returning them to storage, aiming to minimize stowing completion times.
- *Fulfillment Center:* This scenario represents the smart logistics workflow, encompassing both order picking and stowing. The objective is to co-optimize the dual goals of minimizing order completion times and stowing times.
- *Automated Production Line:* This scenario models a smart factory process where robots transport materials between key production points, such as workstations and conveyor belts. The objective is to maximize throughput while maintaining the production flow.

Rapid Simulation and Validation

Another core design principle of RAPID is to accelerate the validation of various behavior models. To achieve this, the platform provides performance visualization and utilizes a modular architecture standardized by the PettingZoo API (Terry et al. 2021). This design effectively decouples the simulation environment from the behavior model, with all interactions managed through standard function calls.

Simulation Dashboard Interface Once an environment is configured, users can validate their design and analyze system performance through the simulation dashboard. The workflow begins by selecting a behavior model. During simulation, the dashboard provides real-time performance visualization through a dual-view display: an **animated view** showing agent behaviors and interactions, synchronized with **live charts** tracking key performance indicators such as throughput and task completion times. This integrated visualization enables users to immediately observe how strategic decisions impact operational metrics.

Behavior Models The platform's standardized interface enables a plug-and-play approach, allowing any model that adheres to the input-output format to be integrated. As a demonstration of this flexibility, we have implemented the following representative models:

- *Heuristic algorithm:* Users can directly implement their existing or new operational logic as custom Python scripts and apply them within the platform.
- *RL policy:* We model the problem using multi-agent RL. The platform's PettingZoo API compliance enables direct integration with multi-agent RL frameworks like EPyMARL (Papoudakis et al. 2021), allowing users to readily apply various algorithms such as MAPPO (Yu et al. 2022) and QMIX (Rashid et al. 2020).
- *LLM planner*: The platform provides a console for LLM response and a highly engineered prompt that effectively gives high-level plans with executable low-level actions. Users can integrate and use various LLMs of interest.

Conclusion and Future Work

RAPID bridges the gap between complex, high-fidelity simulators and the need for rapid prototyping by integrating an intuitive visual editor with a powerful simulation engine for diverse behavior models. It significantly lowers the barrier to experimentation, enabling both practitioners and researchers to validate their concepts in minutes and thereby accelerate the automation development cycle. Future work will expand our library of components and supported behavior models.

References

Borshchev, A. 2013. *The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6*. Lisle, IL: AnyLogic North America. ISBN 9780989573177.

Hopcroft, J. E.; Motwani, R.; and Ullman, J. D. 2006. *Introduction to Automata Theory, Languages, and Computation*. Boston, MA: Pearson Addison Wesley, 3rd edition.

Krnjaic, A.; Steleac, R. D.; Thomas, J. D.; Papoudakis, G.; Schäfer, L.; To, A. W. K.; Lao, K.-H.; Cubuktepe, M.; Haley, M.; Börsting, P.; et al. 2024. Scalable Multi-Agent Reinforcement Learning for Warehouse Logistics with Robotic and Human Co-Workers. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 677–684. IEEE.

Morais, P. H.; Vivaldini, K. C.; Kato, E. R.; and Inoue, R. S. 2025. A Review of Robot Fleet Management. *IEEE Access*.

NVIDIA. 2025. NVIDIA Omniverse. https://www.nvidia.com/omniverse. Accessed: 2025-08-28.

Papoudakis, G.; Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Terry, J. K.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L. S.; Dieffendahl, C.; Horsch, C.; Perez-Vicente, R.; et al. 2021. Pettingzoo: Gym for Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34: 15032–15043.

Wurman, P. R.; D'Andrea, R.; and Mountz, M. 2008. Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. *AI magazine*, 29(1): 9–9.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Advances in Neural Information Processing Systems*, 35: 24611–24624.