

# PANSim: A System for Simulating and Visualizing Planning Against Nature

Erol Medenčević<sup>1,2</sup>, Jakub Med<sup>1,2</sup>, Lukáš Chrpa<sup>2</sup>

<sup>1</sup>Faculty of Electrical Engineering, Czech Technical University in Prague

<sup>2</sup>Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague  
medenero@fel.cvut.cz, jakub.med@cvut.cz, chrpaluk@cvut.cz

## Abstract

Planning against nature provides a formal framework for addressing the inherent nondeterminism found in environments shaped by exogenous events. Although several visualization tools exist for classical planning, they generally lack support for such exogenous events. We introduce PANSim, an interactive simulation and visualization tool specifically designed for planning against nature tasks. PANSim features a frontend developed in Unity for clear visual interpretation of planning in such domains and a backend implemented in Python. Relying on a standardized JSON communication format, the backend can function as a standalone simulator, allowing users to easily connect and evaluate any custom agent with or without the visual interface. We demonstrate the capabilities of the tool using three default agents designed to generate robust plans, showcased in two included environments: the Perestroika domain, inspired by a retro game, and the Autonomous Underwater Vehicle (AUV) domain.

## Code —

<https://gitlab.com/automated-planning/tools/pansim>

## Introduction

*Automated planning* is a key component of artificial intelligence, especially in domains where decision-making is based on a known environmental model, defined actions, and specific constraints. Usually, the environment is formalized using state variables and actions with explicit preconditions and effects. Such a formal structure allows the planner to find a sequence of actions, a plan, to navigate from an initial state to a targeted goal state. Such models assume a predictable, deterministic world, which rarely holds in practice.

In reality, agents often face uncertainty driven by uncontrollable external forces, termed *exogenous events* or *acts of nature*, which can affect the state of the world independently of the agent's actions, making planning more dynamic.

To address these challenges, we introduce *PANSim*, a visualization tool previously presented at AAAI (Medenčević, Med, and Chrpa 2026), which transforms the acting of planning agents into a clear and accessible graphical form. It helps researchers analyze and debug agent behavior while effectively communicating planning concepts to broader, non-expert audiences. PANSim can also be used for public relations (PR) purposes to promote *planning and acting* research for students or potential industry partners. The tool is

built with reusability in mind, allowing simulations to be executed repeatedly, compared across different problems, and presented in a consistent and visually coherent way.

## Related Work

The concept of planning with exogenous events has been studied for some time (Dean and Wellman 1990; Musliner, Durfee, and Shin 1993; Iocchi, Nardi, and Rosati 2000). Addressing planning tasks with external influences usually requires reasoning over a large state space. Methods that reason with Markov Decision Process (MDP) models (Mausam and Kolobov 2012) are a good example. These methods generate policies, assigning the most promising action in each reachable state. Policies, however, might not be easy to interpret and explain to human operators who supervise the acting agent. This problem can be mitigated by an introduction of the concepts studied in *planning against nature* (Chrpa, Gemrot, and Pilát 2020; Chrpa, Pilát, and Med 2021). Robust plans, the solution concept of planning against nature, are related to weak and strong policies in Fully Observable Non-Deterministic (FOND) planning (Cimatti et al. 2003; Ghallab, Nau, and Traverso 2016); while planning problems with exogenous events can be compiled into FOND tasks (Chrpa, Pilát, and Gemrot 2019), FOND approaches do not scale well in this setting (Chrpa, Pilát, and Med 2021).

Regarding visualization, several tools have been developed for classical planning. Earlier tools such as *itSimple* (Vaquero et al. 2007) and *GIPO* (Simpson, Kitchin, and McCluskey 2007) focus on domain modeling, helping users design domains and translate them into *PDDL* (Ghallab et al. 1998).

More recent tools have shifted the focus to visualizing the execution of plans. *Planimation*, for example, generates animations using declarative animation profiles (Chen et al. 2019). Instead of hardcoding visual logic, users define mapping rules that associate *PDDL* predicates with changes in visual object properties (e.g. position, appearance), where objects themselves are represented as 2D sprites. Another approach is *vPlanSim*, an interactive graphical interface that allows users to go through a generated plan action by action, inspecting how each step alters the underlying state of the world (Roberts et al. 2021). Although these tools make plan execution easier to follow, they are primarily designed for classical planning settings and do not explicitly address

nondeterministic environments or external influences.

PANSim was developed mainly using the *Unity* engine (Unity Technologies 2023). The creation of this tool was inspired by *PDSim* (Pellegrin and Petrick 2024), which similarly visualizes planning domains within Unity. However, *PDSim* proved unsuitable for our purposes because it lacks support for exogenous events, which are a crucial component of our target domains. Therefore, PANSim builds on the core logic of an existing textual simulator (Chrupa, Pilát, and Med 2021) to properly handle and visualize the execution of the plan specifically for planning against nature tasks. Unlike existing tools that animate a pre-computed plan trace, PANSim runs the simulation live. Exogenous events are generated on-the-fly each run, meaning the visualization is inherently non-deterministic and each execution may differ.

## Planning Against Nature

The concept of *planning against nature* formalizes planning tasks for an acting agent in the presence of exogenous events that are randomly triggered by nature (Chrupa and Karpas 2024). In this setting, the state of the environment is not solely determined by the actions of the agent, adding a significant layer of nondeterminism. Importantly, nature is not an adversarial entity with its goals; it does not intentionally hinder the agent but triggers exogenous events at random.

The assumption of behavior of nature used in PANSim follows the work of Chrupa, Pilát, and Med (2021), where nature can execute any set of independent events (that do not interfere with each other) between the actions of the agent. To navigate these environments, the tool implements three agents used for the experimental analysis in a work of Chrupa, Pilát, and Med (2021) that focus on generating plans which are guaranteed to avoid dead-ends while executing.

Currently, PANSim supports two domains that feature such exogenous events (Chrupa, Pilát, and Med 2021).

**Perestroika** Inspired by the classic retro game *Toppler*<sup>1</sup>, the acting agent navigates a grid to collect resources on two types of platforms: permanently accessible *solid platforms* and *shrinking platforms* that nature unpredictably changes, causing them to decrease in size, disappear, or reappear. If a platform vanishes while the agent is standing on it, the agent falls into a dead-end.

**AUV** The Autonomous Underwater Vehicle (AUV) agent navigates a grid to sample scattered resources and reach an escape cell, while avoiding monitoring ships whose non-deterministic movements are controlled by nature: *wandering ships* that can unpredictably wait or move in either direction, and *cruising ships* that move exclusively in one direction, disappearing at exits and reappearing at entry points. Collision with a ship results in a dead-end.

## Technical Solution

The architecture of PANSim is divided into front-end and back-end components, with the goal of separating the planning logic and simulation from the graphical part. The front-

end is a Unity application, while the back-end is implemented in Python (Python Core Team 2026) using the Unified Planning Library (Micheli et al. 2025). During simulation, these components communicate over the network. The user interacts exclusively with the front-end, without any need to interact directly with the simulation part.

**Front-end** It allows users to select domains and problems, control simulation playback, and adjust the view. In addition, users can provide custom PDDL problem files. The front-end is able to interpret these files to create the necessary game objects and position them within the scene. Specifically, it parses the PDDL problem file and uses the names of objects and predicates as string identifiers to instantiate the corresponding domain-specific game objects and trigger their animations. Upon receiving a state from the back-end, the front-end displays it to the user by triggering the corresponding animations of the objects. This happens twice in each turn: once to display the state after applying the agent’s action, and once to display the state after exogenous events are applied. Note that the front-end visualization is currently domain-dependent; while the back-end can simulate any valid planning against nature domain, the visual layer requires domain-specific assets and string mappings.

**Back-end** It deals with the execution of the plan and implements the behavior of nature (Chrupa, Pilát, and Med 2021). In the domain’s PDDL, exogenous events are represented similarly to standard actions but distinguished by the `:event` keyword. Unlike the previous implementation, the back-end processes actions and events directly in a lifted representation. Each turn proceeds in two phases: the back-end applies the agent’s chosen action and sends the resulting state to the front-end; once visualized, it generates and applies exogenous events and sends the updated state back to begin the next turn.

All communication relies on channel-specific *JSON* (Ecma International 2017) message formats, making it straightforward for any external agent to connect to the system. Importantly, the back-end can function as a standalone simulator to evaluate custom agents even without the Unity front-end.

**Limitations** The main limitation of PANSim is that the visualization is domain-dependent, currently restricted to the two included domains, although the back-end can simulate any valid planning against nature domain.

## Future Work

The future development of PANSim will focus on incorporating a wider variety of planning domains and improving user interactivity. As part of this effort, we plan to introduce a playable mode designed for educational purposes and the general public. This feature will allow users to manually solve tasks and compare their performance against automated agents. Furthermore, to ensure smooth visualization even during the simulation of very large problem instances, we plan to implement adaptive graphical detail. Last but not least, an addition of novel state-of-the-art agents is planned in the near future.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Perestroika\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Perestroika_(video_game))

## References

- Chen, G.; Ding, Y.; Edwards, H.; Chau, C. H.; Hou, S.; Johnson, G.; Sharukh Syed, M.; Tang, H.; Wu, Y.; Yan, Y.; Tidhar, G.; and Lipovetzky, N. 2019. Planimation. In *ICAPS 2019 System Demonstrations*. Berkeley, CA, USA.
- Chrpa, L.; Gemrot, J.; and Pilát, M. 2020. Planning and Acting with Non-Deterministic Events: Navigating between Safe States. In *AAAI*, 9802–9809. AAAI Press.
- Chrpa, L.; and Karpas, E. 2024. On Verifying Linear Execution Strategies in Planning Against Nature. In Bernardini, S.; and Muise, C., eds., *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2024, Banff, Alberta, Canada, June 1-6, 2024*, 86–94. AAAI Press.
- Chrpa, L.; Pilát, M.; and Gemrot, J. 2019. Compiling Planning Problems with Non-deterministic Events into FOND Planning. In *Proceedings of the RCRA International Workshop*.
- Chrpa, L.; Pilát, M.; and Med, J. 2021. On Eventual Applicability of Plans in Dynamic Environments with Cyclic Phenomena. In *KR*, 184–193.
- Cimatti, A.; Pistore, M.; Roveri, M.; and Traverso, P. 2003. Weak, strong, and strong cyclic planning via symbolic model checking. *Artif. Intell.*, 147(1-2): 35–84.
- Dean, T.; and Wellman, M. 1990. *Planning and Control*. Morgan Kaufmann Publishers.
- Ecma International. 2017. The JSON Data Interchange Syntax. Standard ECMA-404, Ecma International.
- Ghallab, M.; Knoblock, C.; Wilkins, D.; Barrett, A.; Christianson, D.; Friedman, M.; Kwok, C.; Golden, K.; Penberthy, S.; Smith, D.; Sun, Y.; and Weld, D. 1998. PDDL – The Planning Domain Definition Language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.
- Ghallab, M.; Nau, D. S.; and Traverso, P. 2016. *Automated Planning and Acting*. Cambridge University Press. ISBN 978-1-107-03727-4.
- Iocchi, L.; Nardi, D.; and Rosati, R. 2000. Planning with sensing, concurrency, and exogenous events: logical framework and implementation. In *KR*, 678–689.
- Mausam; and Kolobov, A. 2012. *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.
- Medenčević, E.; Med, J.; and Chrpa, L. 2026. PANSim: Visualization Tool for Planning and Acting against Nature. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(48): 41640–41642.
- Micheli, A.; Bit-Monnot, A.; Röger, G.; Scala, E.; Valentini, A.; Framba, L.; Rovetta, A.; Trapasso, A.; Bonassi, L.; Gerevini, A. E.; Iocchi, L.; Ingrand, F.; Köckemann, U.; Patrizi, F.; Saetti, A.; Serina, I.; and Stock, S. 2025. Unified Planning: Modeling, manipulating and solving AI planning problems in Python. *SoftwareX*, 29: 102012.
- Musliner, D. J.; Durfee, E. H.; and Shin, K. G. 1993. CIRCA: a cooperative intelligent real-time control architecture. *IEEE Trans. Systems, Man, and Cybernetics*, 23(6): 1561–1574.
- Pellegrin, E. D.; and Petrick, R. P. A. 2024. Planning Domain Simulation: An Interactive System for Plan Visualisation. In Bernardini, S.; and Muise, C., eds., *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2024, Banff, Alberta, Canada, June 1-6, 2024*, 133–141. AAAI Press.
- Python Core Team. 2026. *Python: A dynamic, open source programming language*. Python Software Foundation.
- Roberts, J. O.; Mastorakis, G.; Lazaruk, B.; Franco, S.; Stokes, A. A.; and Bernardini, S. 2021. vPlanSim: An Open Source Graphical Interface for the Visualisation and Simulation of AI Systems. In *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, 486–490.
- Simpson, R. M.; Kitchin, D. E.; and McCluskey, T. L. 2007. Planning domain definition using GIPO. *The Knowledge Engineering Review*, 22(2): 117–134.
- Unity Technologies. 2023. Unity. Game development platform.
- Vaquero, T. S.; Romero, V.; Tonidandel, F.; and Silva, J. R. 2007. itSIMPLE2.0: An Integrated Tool for Designing Planning Domains. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS)*, 336–343. AAAI Press.