# JEDAI: A System for Skill-Aligned Explainable Robot Planning

## Demonstration Track

Naman Shah*
Arizona State University, USA
shah.naman@asu.edu

Pulkit Verma*
Arizona State University, USA
verma.pulkit@asu.edu

Trevor Angle
Arizona State University, USA
taangle@asu.edu

Siddharth Srivastava
Arizona State University, USA
siddharths@asu.edu

## ABSTRACT

This paper presents JEDAI, an AI system designed for outreach and educational efforts aimed at non-AI experts. JEDAI features a novel synthesis of research ideas from integrated task and motion planning and explainable AI. JEDAI helps users create high-level, intuitive plans while ensuring that they will be executable by the robot. It also provides users customized explanations about errors and helps improve their understanding of AI planning as well as the limits and capabilities of the underlying robot system.

## KEYWORDS

AI in Education; Explanations; Task and Motion Planning; Robotics

**ACM Reference Format:**
Naman Shah, Pulkit Verma, Trevor Angle, and Siddharth Srivastava. 2022. JEDAI: A System for Skill-Aligned Explainable Robot Planning: 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

AI systems are increasingly common in everyday life, where they can be used by laypersons who may not understand how these autonomous systems work or what they can and cannot do. This problem is particularly salient in cases of taskable AI systems whose functionality can change based on the tasks they are performing. In this work, we present an AI system JEDAI (JEDAI Explains Decision-Making AI) that can be used in outreach and educational efforts to help laypersons learn how to provide AI systems with new tasks, debug such systems, and understand their capabilities.

The research ideas brought together in JEDAI address three key technical challenges: (i) abstracting a robot's functionalities into high-level actions (capabilities) that the user can more easily understand; (ii) converting the user-understandable capabilities into low-level motion plans that a robot can execute; and (iii) explaining errors in a manner sensitive to the user's current level of knowledge so as to make the robot's capabilities and limitations clear.

JEDAI utilizes recent work in explainable AI and integrated task and motion planning to address these challenges and provides a simple interface to support accessibility. Users select a domain and

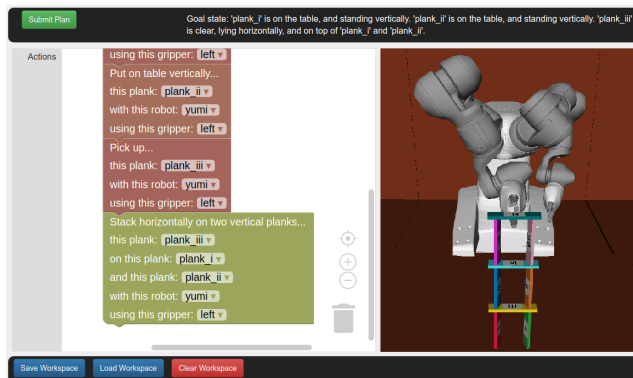---

*Equal Contribution. Alphabetical Order.

**Figure 1: JEDAI system with a Blockly-based plan creator on the left and a simulator window on the right.**

an associated task, after which they create a plan consisting of high-level actions (Fig. 1 left) to complete the task. The user puts together a plan in a drag-and-drop workspace, built with the Blockly visual programming library [9]. JEDAI validates this plan using the Hierarchical Expertise Level Modeling algorithm (HELM) [22, 23]. If the plan contains any errors, HELM computes a user-specific explanation of why the plan would fail. JEDAI converts such explanations to natural language, thus helping to identify and fix any gaps in the user's understanding. Whereas, if the plan given by the user is a correct solution to the current task, JEDAI uses a task and motion planner ATM-MDP [19, 20] to convert the high-level plan, that the user understands, to a low-level motion plan that the robot can execute. The user is shown the execution of this low-level motion plan by the robot in a simulated environment (Fig. 1 right).

Prior work on the topic includes approaches that solve the three technical challenges mentioned earlier *in isolation*. This includes tools for: providing visualizations or animations of standard planning domains [2, 5, 6, 8, 15, 18]; making it easier for non-expert users to program robots with low-level actions [11, 13, 27, 28]; and generating explanations for plans provided by the users [4, 10, 12, 14]. In addition, none of these works make the instructions easier for the user, have the ability to automatically compute user-aligned explanations, and work with real robots (or their simulators) at the same time. JEDAI addresses all three challenges in tandem by using 3D simulations for domains with real robots and their actual constraints and providing personalized explanations that inform a user of any mistake they make while using the system.
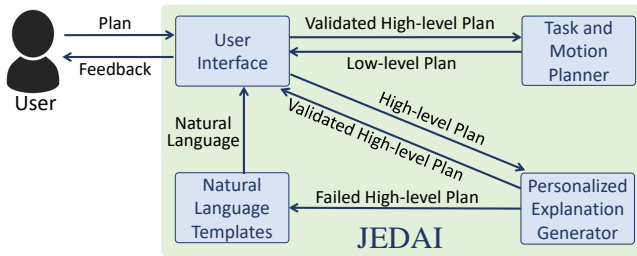
**Figure 2: Architecture of JEDAI showing interaction between the four core components.**

## 2 ARCHITECTURE

Fig. 2 shows the four core components of the JEDAI framework: (i) user interface, (ii) task and motion planner, (iii) personalized explanation generator, and (iv) natural language templates. We now describe each component in detail.

**User interface** JEDAI's user interface (Fig. 1) is made to be unintimidating and easy to use. The Blockly visual programming interface is used to facilitate this. JEDAI generates a separate interconnecting block for each high-level action, and action parameters are picked from drop-down selection fields that display type-consistent options for each parameter. Users can drag-and-drop these actions and select different arguments to create a high-level plan.

**Personalized explanation generator** Users will sometimes make mistakes when planning, either failing to achieve goal conditions or applying actions before the necessary preconditions are satisfied. For inexperienced users in particular, these mistakes may stem from an incomplete understanding of the task's requirements or the robot's capabilities. JEDAI assists users in apprehending these details by providing explanations personalized to each user.

Explanations in the context of this work are of two types: (i) non-achieved goal conditions, and (ii) violation of a precondition of an action. JEDAI validates the plan submitted by the user to check if it achieves all goal conditions. If it fails to achieve any goal condition, the user is informed about it. JEDAI uses HELM to compute user-specific contrastive explanations in order to explain any unmet precondition in an action used in the user's plan. HELM does this by using the plan submitted by the user to estimate the user's understanding of the robot's model and then uses the estimated model to compute the personalized explanations. In case of multiple errors in the user's plan, HELM generates explanation for one of the errors. This is because explaining the reason for more than one errors might be unnecessary and in the worst case might leave the user feeling overwhelmed [16]. An error is selected for explanation by HELM based on optimizing a cost function that indicates the relative difficulty of concept understandability which can be changed to reflect different users' background knowledge.

**Natural language templates** Even with a user-friendly interface and personalized explanations for errors in abstract plans, domain model syntax used for interaction with ATM-MDP presents a significant barrier to a non-expert trying to understand the state of an environment and the capabilities of a robot. To alleviate this, JEDAI uses language templates that use the structure of the planning formalism for generating natural language descriptions for goals,

actions, and explanations. E.g., the action *"pickup (plank_i gripper_left)"* can be described in natural language as "pick up *plank_i* with *the left gripper*". Currently, we use hand-written templates for these translations, but an automated approach can also be used.

**Task and motion planner** JEDAI uses ATM-MDP to convert the high-level plan submitted by the user into sequences of low-level primitive actions that a robot can execute.

ATM-MDP uses sampling-based motion planners to provide a probabilistically complete approach to hierarchical planning. High-level plans are refined by computing feasible motion plans for each high-level action. If an action does not accept any valid refinement due to discrepancies between the symbolic state and the low-level environment, it reports the failure back to JEDAI. If all actions in the high-level plan are refined successfully, the plan's execution is shown using the OpenRAVE simulator [7].

**Implementation** Any custom domain can be set up with JEDAI. We provide five built-in domains, each with one of YuMi [1] or Fetch [29] robots. Each domain contains a set of problems that the users can attempt to solve and low-level environments corresponding to these problems. Source code for the framework, an already setup virtual machine, and the documentation are available at: https://github.com/aair-lab/AAIR-JEDAI. A video demonstrating JEDAI's working is available at: https://youtu.be/MQdoikcnhbY.

## 3 CONCLUSIONS AND FUTURE WORK

We demonstrated a novel AI tool JEDAI for helping people understand the capabilities of an arbitrary AI system and enabling them to work with such systems. JEDAI converts the user's input plans to low level motion plans executable by the robot if it is correct, or explains to the user any error in the plan if it is incorrect. JEDAI works with off-the-shelf task and motion planners and explanation generators. This structure allows it to scale automatically with improvements in either of these active research areas. JEDAI's vizualization-based interface could also be used to foster trust in AI systems [3].

JEDAI uses predefined abstractions to verify plans provided by the user. In the future, we plan on extending it to learn abstractions automatically [21]. JEDAI could also be extended as an interface for assessing an agent's functionalities and capabilities by interrogating the agent [17, 25, 26] as well as to work as an interface that makes AI systems compliant with Level II assistive AI – systems that makes it easy for operators to learn how to use them safely [24]. Extending this tool for working in non-stationary settings, and generating natural language descriptions of predicates and actions autonomously are a few other promising directions of future work.

# REFERENCES

[1] ABB. 2015. ABB YuMi - IRB 14000. *https://new.abb.com/products/robotics/collaborative-robots/irb-14000-yumi* (2015).

[2] Angeline Aguinaldo and William Regli. 2021. A Graphical Model-Based Representation for Classical AI Plans using Category Theory. In *ICAPS 2021 Workshop on Explainable AI Planning*.

[3] Emma Beauxis-Aussalet, Michael Behrisch, Rita Borgo, Duen Horng Chau, Christopher Collins, David Ebert, Mennatallah El-Assady, Alex Endert, Daniel A. Keim, Jörn Kohlhammer, Daniela Oelke, Jaakko Peltonen, Maria Riveiro, Tobias Schreck, Hendrik Strobelt, and Jarke J. van Wijk. 2021. The Role of Interactive Visualization in Fostering Trust in AI. *IEEE Computer Graphics and Applications* 41, 6 (2021), 7–12.

[4] Martim Brandao, Gerard Canal, Senka Krivić, and Daniele Magazzeni. 2021. Towards Providing Explanations for Robot Motion Planning. In *Proc. ICRA*.

[5] Gan Chen, Yi Ding, Hugo Edwards, Chong Hin Chau, Sai Hou, Grace Johnson, Mohammed Sharukh Syed, Haoyuan Tang, Yue Wu, Ye Yan, Tidhar Gil, and Lipovetzky Nir. 2019. Planimation. In *ICAPS 2019 System Demonstrations*.

[6] Emanuele De Pellegrin and Ronald P. A. Petrick. 2021. PDSim: Simulating Classical Planning Domains with the Unity Game Engine. In *ICAPS 2021 System Demonstrations*.

[7] Rosen Diankov and James Kuffner. 2008. *OpenRAVE: A Planning Architecture for Autonomous Robotics*. Technical Report CMU-RI-TR-08-34. Carnegie Mellon University, Pittsburgh, PA, USA.

[8] Filip Dvorak, Anil Agarwal, and Nikolay Baklanov. 2021. Visual Planning Domain Design for PDDL using Blockly. In *ICAPS 2021 System Demonstrations*.

[9] Google. 2017. Blockly. *https://github.com/google/blockly* (2017).

[10] Sachin Grover, Sailik Sengupta, Tathagata Chakraborti, Aditya Prasad Mishra, and Subbarao Kambhampati. 2020. RADAR: Automated Task Planning for Proactive Decision Support. *Human–Computer Interaction* 35, 5-6 (2020), 387–412.

[11] Gaoping Huang, Pawan S. Rao, Meng-Han Wu, Xun Qian, Shimon Y. Nof, Karthik Ramani, and Alexander J. Quinn. 2020. Vipo: Spatial-Visual Programming with Functions for Robot-IoT Workflows. In *Proc. CHI*.

[12] Valmeekam Karthik, Sarath Sreedharan, Sailik Sengupta, and Subbarao Kambhampati. 2022. RADAR-X: An Interactive Mixed Initiative Planning Interface Pairing Contrastive Explanations and Revised Plan Suggestions. In *Proc. ICAPS*.

[13] Sai Prasanth Krishnamoorthy and Vikram Kapila. 2016. Using A Visual Programming Environment and Custom Robots to Learn C Programming and K-12 STEM Concepts. In *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education*.

[14] Ashwin Kumar, Stylianos Loukas Vasileiou, Melanie Bancilhon, Alvitta Ottley, and William Yeoh. 2022. VizXP: A Visualization Framework for Conveying Explanations to Users in Model Reconciliation Problems. In *Proc. ICAPS*.

[15] Maurício Cecílio Magnaguagno, Ramon Fraga Pereira, Martin Duarte Móre, and Felipe Rech Meneguzzi. 2017. WEB PLANNER: A Tool to Develop Classical Planning Domains and Visualize Heuristic State-Space Search. In *ICAPS 2017 Workshop on User Interfaces and Scheduling and Planning*.

[16] Tim Miller. 2019. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial Intelligence* 267 (2019), 1–38.

[17] Rashmeet Kaur Nayyar, Pulkit Verma, and Siddharth Srivastava. 2022. Differential Assessment of Black-Box AI Agents. In *Proc. AAAI*.

[18] Jamie O. Roberts, Georgios Mastorakis, Brad Lazaruk, Santiago Franco, Adam A. Stokes, and Sara Bernardini. 2021. vPlanSim: An Open Source Graphical Interface for the Visualisation and Simulation of AI Systems. In *Proc. ICAPS*.

[19] Naman Shah, Deepak Kala Vasudevan, Kislay Kumar, Pranav Kamojjhala, and Siddharth Srivastava. 2020. Anytime Integrated Task and Motion Policies for Stochastic Environments. In *Proc. ICRA*.

[20] Naman Shah and Siddharth Srivastava. 2021. Anytime Stochastic Task and Motion Policies. *arXiv preprint arXiv:2108.12537* (2021).

[21] Naman Shah and Siddharth Srivastava. 2022. Using Deep Learning to Bootstrap Abstractions for Hierarchical Robot Planning. In *Proc. AAMAS*.

[22] Sarath Sreedharan, Siddharth Srivastava, and Subbarao Kambhampati. 2018. Hierarchical Expertise Level Modeling for User-Specific Contrastive Explanations. In *Proc. IJCAI*.

[23] Sarath Sreedharan, Siddharth Srivastava, and Subbarao Kambhampati. 2021. Using State Abstractions to Compute Personalized Contrastive Explanations for AI Agent Behavior. *Artificial Intelligence* 301 (2021), 103570.

[24] Siddharth Srivastava. 2021. Unifying Principles and Metrics for Safe and Assistive AI. In *Proc. AAAI*.

[25] Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. 2021. Asking the Right Questions: Learning Interpretable Action Models Through Query Answering. In *Proc. AAAI*.

[26] Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. 2022. Discovering User-Interpretable Capabilities of Black-Box Planning Agents. In *AAAI 2022 Workshop on Explainable Agency in Artificial Intelligence*.

[27] David Weintrop, Afsoon Afzal, Jean Salac, Patrick Francis, Boyang Li, David C. Shepherd, and Diana Franklin. 2018. Evaluating CoBlox: A Comparative Study of Robotics Programming Environments for Adult Novices. In *Proc. CHI*.

[28] Mario Winterer, Christian Salomon, Jessica Köberle, Rudolf Ramler, and Markus Schittengruber. 2020. An Expert Review on the Applicability of Blockly for Industrial Robot Programming. In *Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*.

[29] Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. 2016. Fetch and Freight: Standard Platforms for Service Robot Applications. In *IJCAI 2016 Workshop on Autonomous Mobile Service Robots*.