

LEARNS: Learning Enhancement and Recognition Navigation System

Stewart Anderson¹, Felipe Meneguzzi¹

¹King's College, University of Aberdeen
School of Natural and Computer Sciences
Department of Computer Science
{first.last}@abdn.ac.uk

Abstract

Introduction

1. **Introduction to the LEARNS project and its significance.**
2. **The current challenges in secondary education and the need for automated planning solutions.**
3. **Overview of the goals and objectives.**
4. **Brief outline of the structure of the paper and what readers can expect.**

Background

1. **Overview of related work in the field of using AI and planning for educational assistance** (Castillo et al. 2009) summary.
2. **The specific challenges faced by secondary education that necessitate innovative solutions**, diversity of learning styles and technological integration of learning solutions **TODO: Find Material.**

Adaptive Learning and Intelligent Tutoring Systems

Adaptive learning systems provide personalised learning experiences. They adapt educational content and activities to the specific needs and abilities of each individual learner using AI algorithms. Key features of adaptive learning include:

- **Personalisation:** Adaptive learning systems create a unique learning path for each student by adjusting content and pacing based on real-time performance data, as discussed in El-Sabagh (2021).
- **Immediate Feedback:** These systems offer instant feedback, helping students identify and address mistakes as they occur, leading to a deeper understanding of the material. [Citation Required]
- **Data-Driven Insights:** Adaptive learning systems collect and analyse data on student interactions, providing educators with valuable insights into student progress and areas requiring attention. [Citation Required]

Limitations:

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

- **Content Quality:** The effectiveness of adaptive learning heavily depends on the quality of the learning materials. Inaccurate or outdated content may hinder personalisation efforts. [Citation Required]
- **Overemphasis on Data:** Overreliance on data and algorithms can result in a mechanistic approach to education, potentially overlooking the importance of the human element. Rojas et al. (2022) covers collaborative problem-solving and providing feedback in real-time, but does not necessarily provide an application to a classroom.
- **Resource Requirements:** Implementing adaptive learning systems requires significant investments in technology and content development, which can be a barrier for institutions with limited resources. [Citation Required]

Intelligent Tutoring Systems simulate one-on-one tutoring experiences for students, using AI planning techniques to provide real-time guidance and adapt instructional approaches. Key features of ITS include:

- **Real-time Guidance:** ITS offers students immediate feedback, clarifications, and guidance as they work through problems or assignments. [Citation Required]
- **Step-by-Step Assistance:** In subjects requiring sequential problem-solving, ITS provides step-by-step assistance, helping students grasp complex concepts. [Citation Required]
- **Performance Assessment:** ITS continuously assesses a student's knowledge and skills, offering targeted practice exercises and content based on their performance. [Citation Required]

Limitations of Intelligent Tutoring Systems

- **Lack of Contextual Understanding:** ITS may lack a deep understanding of the broader context of learning, potentially leading to recommendations that are technically correct but not pedagogically appropriate. Garrido, Onaindia, and Sapena (2009) cannot monitor learning routes.
- **Complex Development:** Developing effective ITS can be complex and costly, requiring significant resources and expertise in algorithm development and content creation (Corbett, Koedinger, and Anderson 1997).
- **Technology Access:** Like adaptive learning, the effectiveness of ITS relies on technology access, which may

not be equally available to all students. [Citation Required]

Theoretical Framework In automated planning, the use of predicates and objects allow us to represent and reason about the world, what tasks may need to be carried out in that world, and what goals are achievable in the context of that world. A predicate $p \in P$ is denoted by an n -ary predicate symbol, applied to a sequence of zero or more terms τ_0, \dots, τ_n . Terms are either constants or variables. We can refer to ground predicates, which represent logical values according to some interpretation as facts, which are divided into two types: positive and negated; as well as having constants for truth and false (\top and \perp respectively). Now that we have defined a formal element for individual facts of the environment, we aggregate those which are relevant into a state. A state S is a finite set of positive facts f which follow the closed-world assumption. It follows that f is true in S if $f \in S$. We are also able to assume a simple inference relation; such that $S \models f$ iff $f \in S$, $S \not\models f$ iff $f \notin S$, and $S \models \bigwedge_{n=0}^{\Sigma} f_n$ iff $\{f_0, \dots, f_n\} \supseteq S$. Planning domains describe the dynamics of an environment by employing operators characterised by a restricted first-order logical representation, creating schemata for actions aimed at altering the state. A schema is a template that offers a broad overview of task completion. It acts as a blueprint for a series of actions in different scenarios, allowing for the capture of common strategies for achieving goals. These schema templates promote knowledge reuse and adaptation, making problem-solving more efficient by tailoring abstract plans to the current context. An operator a is represented through a tuple $\langle name(a), pre(a), eff(a) \rangle$; where $name(a)$ represents the signature or description of a ; $pre(a)$ describes the preconditions of a and $eff(a)$ represents the effects of a . $eff(a)^+$ and $eff(a)^-$ divides $eff(a)$ into an add-list of positive predicates and a delete-list of negated predicates respectively. An action is a ground operator instantiated over its free variables. An action a is applicable to S if and only if $S \models pre(a)$, and generates a new state S' , such that $S' := (S \cup eff(a)^+ / eff(a)^-)$. A pair $\langle \Sigma, \mathcal{A} \rangle$ represents a planning domain definition Ξ , specifying the knowledge of the domain model, consisting of a finite set of facts and a finite set of actions (represented by Σ and \mathcal{A} respectively). Planning instances comprise both the planning domain and the planning problem, describing a finite set of objects of the environment. These are the initial state, and the goal state, which represent where the agent(s) would start and finish. A planning instance Π is represented by a tuple $\langle \Xi, \mathcal{I}, \mathcal{G} \rangle$, whereby $\Xi = \langle \Sigma, \mathcal{A} \rangle$ represents the domain definition; $\mathcal{I} \supseteq \Sigma$ is the initial state specification, defined through specifying the value for all facts in the initial state; and $\mathcal{G} \supseteq \Sigma$ is the goal state specification, representing a singular goal, or a set of goals (that may be extremely large) satisfying the goal specification of the desired state to be achieved. Classical planning representations often separate the definition of the initial state (\mathcal{I}) and the goal state (\mathcal{G}) as part of a planning problem (Ξ) to be utilised with a domain, such as STRIPS (Fikes and Nilsson 1971) or PDDL (Ghallab et al. 1998). For a planning instance $\Pi = \langle \Xi, \mathcal{I}, \mathcal{G} \rangle$, a

plan π is a sequence of actions $\langle a_0, \dots, a_n \rangle$ modifying the initial state \mathcal{I} into a state $S \models \mathcal{G}$, whereby the goal state \mathcal{G} holds by the successive execution of actions in a plan π . A plan is optimal (π^*) with a length $|\pi^*|$ if there exists no other plan π' for $\Pi \mid \pi' < \pi$. Actions may have an associated cost applied to them. If an action does not have an associated cost, it is assumed to have the unit cost (1) assigned to it. A plan (π) is optimal if its cost and respective length meets the evaluation criterion — whether that's minimal, maximal, or otherwise. A goal recognition task is a tuple $T_{GR} = \langle \Xi, \mathcal{I}, \mathcal{G}, O \rangle$; such that $\Xi = \langle \Sigma, \mathcal{A} \rangle$ is a planning domain definition; \mathcal{I} is the initial state; \mathcal{G} is the set of possible goals, also assuming the inclusion of some correct hidden goal G^* ($G^* \in \mathcal{G}$); and $O = \langle o_0, \dots, o_n \rangle$ is an observation sequence of executed actions. The observation sequence O is satisfied by a valid plan π , following that with each observation $o_i = name(a)$, $a \in \mathcal{A}$, with a being part of π , and that plan π transitioning \mathcal{I} into G^* through the sequential execution of actions in π . The optimal solution to a goal recognition task may consist of a singleton set that exclusively contains the correct hidden goal ($G^* \in \mathcal{G}$) achieved by the observation sequence O of a plan execution. In this context, we consider a solution suboptimal if it includes more than one goal, as all such goals are equally probable. The solution to a goal recognition task $T_{GR} = \langle \Xi, \mathcal{I}, \mathcal{G}, O \rangle$ is a non-empty subset of the set of possible goals $\mathbf{G} \supseteq G \mid \forall G \in \mathbf{G}$, there exists a plan π_G generated from a planning instance, and the observation sequence O is satisfied by π_G .

Modelling Secondary Education in PDDL

1. Technical description of the LEARNS architecture.
2. Application of PDDL for modelling various scenarios.
3. Presentation of structures used in LEARNS for learning style recognition.

Experimentation and Results

1. Detailed account of the experiments conducted to evaluate the LEARNS system.
2. Presentation of data and results from the automated planning experiments.
3. Analysis of the performance of LEARNS in real-world educational scenarios.

Goal Recognition Classical planning, as discussed in Fox and Long (2003), treats time as relative and considers only causal dependencies among actions. However, real-world problems often entail complexities, including temporal aspects, numerical values, stochastic effects, and dynamic environments. Numeric planning extends classical planning by incorporating numeric state variables and utilising languages such as PDDL 2.1 (Fox and Long 2003) and PDDL+ (Fox and Long 2006). These formalisms enable the representation of time-dependent changes, either as discrete time-dependent effects of durative actions or continuous process-dependent alterations. PDDL+ serves as an extension of PDDL designed specifically to model hybrid systems by integrating continuous processes and events, as outlined by Haslum et al. (2019). Its primary purpose is to

facilitate the representation of planning domains that combine discrete and continuous elements.

Introduced in Scala, Haslum, and Thiébaux (2016) and Scala et al. (2016), the Expressive Numeric Heuristic Search Planner (ENHSP) is compatible with both PDDL 2.1 and PDDL+. ENHSP operates as a forward heuristic search planner, converting PDDL into an asymptotic relaxed planning graph. In this graph, nodes correspond to states explored by the planner, and a heuristic function guides the search process. This function directs the exploration towards nodes whose associated states are reachable from the initial state and bring the system closer to the desired goals.

Conclusions and Future Work

1. **Summary of the technical implementation and its contributions to adaptive learning.**
2. **Evaluation of the framework's success in supporting diverse learning styles.**
3. **Discussion of the practical implications of LEARNS for educational institutions.**
4. **Identification of areas for future development and improvement.**
5. **Final remarks on the potential impact of LEARNS in the field of educational technology.**

References

- Castillo, L.; Morales, L.; González-Ferrer, A.; Fdez-Olivares, J.; Borrajo, D.; and Onaindía, E. 2009. Automatic generation of temporal planning domains for e-learning problems. *Journal of Scheduling*, 13(4): 347–362.
- Corbett, A. T.; Koedinger, K. R.; and Anderson, J. R. 1997. Intelligent Tutoring Systems. In *Handbook of Human-Computer Interaction*, 849–874. Elsevier.
- El-Sabagh, H. A. 2021. Adaptive e-learning environment based on learning styles and its impact on development students' engagement. *International Journal of Educational Technology in Higher Education*, 18(1).
- Fikes, R. E.; and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4): 189–208.
- Fox, M.; and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*, 20: 61–124.
- Fox, M.; and Long, D. 2006. Modelling Mixed Discrete-Continuous Domains for Planning. *Journal of Artificial Intelligence Research*, 27: 235–297.
- Garrido, A.; Onaindía, E.; and Sapena, O. 2009. Automated Planning for Personalised Course Composition. In *2009 Ninth IEEE International Conference on Advanced Learning Technologies*, 178–182.
- 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.
- Haslum, P.; Lipovetzky, N.; Magazzeni, D.; and Muise, C. 2019. *An Introduction to the Planning Domain Definition Language*. Springer International Publishing.
- Rojas, M.; Sáez, C.; Baier, J.; Nussbaum, M.; Guerrero, O.; and Rodríguez, M. F. 2022. Using Automated Planning to Provide Feedback during Collaborative Problem-Solving. *International Journal of Artificial Intelligence in Education*.
- Scala, E.; Haslum, P.; and Thiébaux, S. 2016. Heuristics for Numeric Planning via Subgoalting. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, 3228–3234. AAAI Press. ISBN 9781577357704.
- Scala, E.; Haslum, P.; Thiebaux, S.; and Ramirez, M. 2016. Interval-Based Relaxation for General Numeric Planning. In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence, ECAI'16*, 655–663. NLD: IOS Press. ISBN 9781614996712.