ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/317661739

A Review of Automated Planning and its Application to Cloud e-Learning

Conference Paper · May 2017

CITATIONS
4

READS
250

3 authors, including:

Krenare Pireva Nuci
The University of Sheffield
25 PUBLICATIONS **46** CITATIONS

SEE PROFILE

Petros Kefalas
The University of Sheffield
124 PUBLICATIONS **786** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Erasmus KA2 View project

Project The impact of using the online learning resources in the quality of learning process - The teacher perspective View project

All content following this page was uploaded by Krenare Pireva Nuci on 19 June 2017.

The user has requested enhancement of the downloaded file.

A Review of Automated Planning and its Application to Cloud e-Learning

Krenare Pireva[1], Petros Kefalas[2], and Anthony J. Cowling[3]

[1] South-East European Research Center, 24 P. Koromila, 54622, Thessaloniki, Greece
[2] The University of Sheffield International Faculty, City College, 3 L. Sofou, 54624, Thessaloniki, Greece
[3] Computer Science Department, The University of Sheffield, 211 Portobello, Sheffield, UK

Abstract. Automated planning is being used in various domains for generating processes that require to bridge a current and a desired state of affairs. Learning can be seen as a process that guides a learner to bridge her current knowledge and skills to some desired ones. The main issue is to select the most appropriate learning resources to include in a personalised learning path. This becomes even more challenging in Cloud e-Learning, where the resources can be anything that is stored in the Cloud. This paper gives an overview of the fundamental concepts of planning as a key area of artificial intelligence and furthermore it explores existing planners and algorithms used for different purposes. Automated planning is introduced as the final process of Cloud e-Learning. A practical example is presented to demonstrate suitability of planning to the generation of personalised learning paths.

1 Introduction

Given an initial and a desired state of a world, **planning** is the process of generating a sequence of actions in partial or complete order so that, if these actions are performed one can reach the desired goal. In Artificial Intelligence the planning process can be fully automated in a variety of ways depending on the nature of the problem as well as the constraints imposed for the final solution (plan).

Learning can be viewed as a planning process. The learner is at some initial state with skills and knowledge already acquired through previous experience and would like to change (learn) to a new desired state which will contain more skills and knowledge. The process of assembling learning material to form a, so called, learning path is equivalent to a planning.

Cloud e-Learning (CeL) is a new paradigm for e-learning[1, 2] in which learners are presented with an automatically generated learning path that utilize any suitable sources from the cloud. CeL is considered as an advancement of e-Learning and aims to provide personalised services that will increase interaction between users by sharing a pool of experiences and knowledge available in cloud and suggest structured courses that match learners preferences and knowledge level. The knowledge available on the cloud comprise different sources for CeL. In CeL, we consider that everything stored in the cloud can be potentially used for learning. The goal is to automatically put together such learning objects in a sequence (CeL path) that reasonably meets the profile and desires of the learner.

The aim of this paper is to review automated planning, formulate the generation of a learning path in CeL as a planning process and propose what type of process is the most appropriate to generate a personalised learning path in CeL.

The remainder of this paper is structured as follows: Section 1 introduce the paper. Section 2 convers the planning and the terminology used throughout the paper. Section 3, treats the Learning as a planning process, whereas section 4, gives a concrete examples of planning in CeL. And finally, concluding the paper.

2 Planning

In our everyday life, the usual tasks are accomplished intuitively as an automatic reaction without having to plan in advance anything. With the increasing complexity of tasks, there is a need to plan, and even in some complex cases, there is a need to plan different alternatives in order to achieve certain goals. **Planning** is an important component of rational behaviour[3] and could be defined as the task to design the behaviour of entities that act individually either on their own or as part of a group of activities [4]. The purpose of Planning as a subfield of AI is to cover the computational aspect of intelligence rather than just performing a plan as a set of activities for providing a solution to particular problems. A **Plan** is defined as a sequence or parallelization of activities or actions, which aim is to achieve specified goals and satisfies the domain constraints based on some initial state given a priori. Often, the problems are described using conceptual models, which are used to describe the elements of problems, through explanation of basic concepts, analysis of the requirements and representation of them.

2.1 Planning Formal definition

A planning domain and problem is usually modeled through representation languages, such as STRIPS, ADL, PDDL. In principle, in order to generate a plan using classical planner, three components must be defined: the description of the system, the initial state and the objectives (the goals). Formally, a **planning problem** is a tuple:

$$P = (S, A, E, \gamma, s_0, g) \quad (1)$$

where

- S is defined as the **set of states**;
- A is the **set of actions** which are going to be performed in order to achieve the stated goal;
- E is a **set of events**;

Table 1. Taxonomy of Domains for Planning

Planning Domain	Description
Path and Motional	Commonly used to find a path for a robot or agent, from the initial state to the defined goal. The algorithms are used in different fields, starting from bioinformatics, animation of characters, industrial automation, robot navigation etc.
Perception	Concerned to process the current state of environment, by gathering the information through sensors. It relies in decision theory of problem, when, which and how the information are needed. For example, the perception planner is required when modeling a complex environment from set of images.
Information gathering	A form of perception is assembled while querying the system instead of sensing
Communication	Outflow in dialog between various agents in order to justify when and how to query required information and which feedback to provide in the meantime
Navigation	Combines the path and perception planning in order to explore the environment. For example following a particular road by processing and avoiding the obstacles as component of the particular road

- γ is the **state transition function** denoted as γ: S x A x E $\rightarrow 2^S$;
- s_0 is the **initial state**;
- g is the **set of goal states**.

2.2 Types of Planners

Planners involve the representation of actions executed by intelligent agents. Since there are various types of actions, we have different types of planners which are applied for various tasks, such as: path and motional planning, process planning, perception planning, navigation planning, etc., each of them described in table 1.

In the other side, there are different approaches on planning, there could be domain **specific/depended** planning or **domain independent** planning, **on-line** or **offline** planning, **classical** or **temporal** planning, **linear** or **non-linear** planning respectively. The domain specific planners are specified precisely for particular problems and their drawback is that each planning problem is tightly connected with the domain problem. Whereas, domain independent relies in an abstract model, starting from the simplest model of action which allows a limited reasonable action to those advanced models with more complex capabilities [3]. Meanwhile, a partial-order plan or non-linear planner starts the initial state with a partial plan and continues to refine the plan until the goal state is achieved. The actions within partial-order plan are unordered, except those necessary, whereas, the total-ordered plan or linear planner generates a sequence of totally ordered

Table 2. Taxonomy of Techniques for Planning

Planning Technique	Description	Planners
Total order	The total order technique or linear planning specify the exact ordering of the actions within the plan. Example in state-space planning, a totally ordered plan is refined.	SHOP[5], HATP [6]
Partial ordered	The partial ordering technique or non-linear planning specifies the ordering of the actions only when necessary. Example in plan-space planning, a partial-ordered plan is refined continually until the desired plan could satisfy the state goals.	UC-POP[7], NOAH[8], PL-PLAN[9]
Heuristic Task Network	HTN Planning approach provides a plan by decomposing the tasks into smaller subtasks by selecting heuristically the best decomposition among the possible ones until reaching the primitive tasks that can be performed directly by planning operators	AltAlt[10], FF[11], GRT[12], LPG[13], VHPOP[14], H2O[15]
SAT-based and Contingency	SAT as a logic-base approach converts the planning problem into Satisfiability problem and the plan is generated based in the efficient solution of the resulting satisfiability problem. In both techniques the actions are not deterministic, and their effects may or may not be observable.	SATPLAN[16], Madagascar[17], ZANDER[18], BlackBox[19]
Temporal	The temporal planning differs from the classical planning, cause the action have durations and some of them might be executed concurrently.	LPG-td[20], TALplanner[21], OPTIC[22], CRIKEY[23]
Case-based	The case based planning approach, adapts (reusing previous plans or partial of plans) previous cases with similar initial and goal state by recalling them from the library and modifying the retrieved solution for new upcoming problems.	CHEF[24], CaPER[25], Prodigy/Analogy[26]

actions, even when steps do not need to be ordered. Based on the algorithms used, each of the planning technique is described in table 2, considering some of the planners used in each of the specified techniques.

2.3 Techniques for Planning

The scenario of classical planning could be defined as a static planning for one scenario, with a known initial state, deterministic actions performed one at a time, and the algorithms used are usually categorized into: state-space planning, plan-space planning [27]. The **Plan-Space (PSP)** planner differs from the **State-Space (SSP)** planner not only in search space but also how the problem is solved. For example PSP uses a partial planning with infinite actions that will be refined continually until the final goals are satisfied whereas SSP uses a finite sequence of actions that is proposed from initial state to final goal. For example, using SSP the node is the initial state and the arc is the transition, whereas using PSP planner, a node is defined as a partially specified plan, and the arc is the refinement operations to further complete the partial plan.

The scenario of neoclassical planning encounters the parallelized activities through graph-based planning and satisfiability algorithms, through AI planning techniques. The neoclassical planners provide an open planning approach while taking in consideration several extension to classical planning, such as time, resources and information gathering action.

The automated planning conceptualized as automated reasoning relies in domain independent and in order to solve a problem, the planners take as input the problem specification and the knowledge about its domain. Based on the forms of reasoning as planning capabilities there are identified: (i) Project planning, (ii) Scheduling and resource allocation and (iii) plan synthesis. Among all, the scheduling and resource allocation include temporal, precedence and resource constraints to be used from each action. A scheduling application takes the action together with resource constraints and optimization criteria as input and returns the temporal organized plan with resource allocation which aims to achieve the defined input criteria. Generally, in automated planning, the Planning and Scheduling are related problems, where the planning deals mainly how to generate a set of actions (the plan) in order to achieve the specified goal, whereas the scheduling is concerned on time and resource allocation for the set of actions defined previously.

During the last decades, there has been done a lot of research toward planning in different domains, by proposing new methods and techniques for improving the planning systems either by introducing new definition languages or by developing algorithms with improvement performances in known and unknown environments. For example, in [28] [29] [30], are developed flexible and distributed planning of multi-agent systems in dynamic environment.

2.4 Representation language PDDL

Planning Domain Definition Language representation language (PDDL) is a standard notation used to encode planning domains. There are different versions of PDDL, mainly supporting different syntactic features such as [31]: conditional effects, basic strips style actions, specification of hierarchical actions etc. The PDDL modeling language is inspired by STRIPS and ADL a previously specification languages for describing the system[32]. PDDL, as a domain definition language is supported by various planners, through which it could define the properties of the domain, the precondition and the actions. Using the defined properties the planner is aiming to generate a plan for achieving the desired goal. PDDL contains requirement clause, such as: typing, strips, fluent etc which could be used further in the function and actions only if they are primarily declared.

3 Learning as a Planning Process

Learning can be defined as a change of state in the learner's cognitive, psychomotor and affective domains [33]. Learning is based on learning outcomes from the levels **Bloom Taxonomy** and ways (teaching and assessment methods) to accomplish them. Therefore, the learners are confronted with a series of learning materials, which we call **Learning Objects (LOs)**, such as texts, videos, assignments, exams etc. that they have to achieve in order to meet the learning outcomes. These form a **learning path** which can be seen as a solution to a planning problem. One could define learning as a planning process as follows:

$$Learning = (S_l, A_l, \gamma_l, s_{0l}, g_l) \tag{2}$$

where:

- S_l is the set of all possible states that characterise a learner;
- A_l is the set of all LOs;
- γ_l a set of transitions which change the state of a learner;
- s_{0l} is the initial state of the learner
- g_l is the set of learning outcomes to achieve

Lately, the automated planning has been also proposed to be integrated in learning domain through learning activities, for being able to develop various learning designs. Garrido et al. (2014) proposed a three level approached procedure to generate learning designs using domain independent planners. The learning activities represented by XML schema are translated through metadata in automated planning, where (i) the course definition is presented as panning domain, (ii) the students learning information as a planning problem of that domain and (iii) the learning design as a plan generated by a domain independent planner. Each of LOs within the planning domain is presented as one or more planning actions, its dependencies relations as preconditions and its outcomes as effects[34]. R-Moreno et al.[35] presented CAMOU as a tool to facilitate the learning and acquire knowledge through interaction between students and teachers and also to help the latter to design courses through IPSS, an integrated automated reasoning system in CAMOU which uses planning and scheduling modules as main reasoning module. In [36], a way how to personalise an e-learning path is presented, based on case-based planning (CBP). Case-based planning is used for definition, memorization, retrieval and adaptation of learning routes. In order to provide solutions to a particular planning problem with respect to CBP, these steps are followed: (i) to retrieve plan that is stored in memory, (ii) to repair the actual plan if any discrepancies are faced, (iii) to test and revise the tested plan, and finally (iv) to store as a new case in the library of case bases. The previous CPB generated plans are stored as cases and can be reused to solve similar planning problems in the future. The best stored learning routes for each students profile and course objective could be reused further, so the system does not have to create a plan from scratch. When discrepancies are detected, the learning route is readapted and improved to meet new objectives, and finally a new learning route is stored further. This proposal as explained contributes on translating the e-learning template into PDDL (Planning description definition language) durative actions and CBP repository contained personalised learning information based on case-based planner. This LOs repository is modified by teachers, and the final approach is tested as an added value in open elearning platforms, such as Moodle and ILIAS. In [37], a system called PASER (Planner for Automatic Synthesis of Educational Resources) is proposed which deals with a larger problem such as synthesizing curricula using planning and machine learning techniques rather than dealing only with courses. The system is very general and it aims to use an automated planner, given the initial state, the available actions and the goals, which then resulted in producing an entire curriculum.

4 Planning in CeL

4.1 A brief Overview of CeL

CeL as a new paradigm of e-Learning, aims to provide personalised learning paths by sharing a pool of knowledge resources available in the cloud[**? ?**].

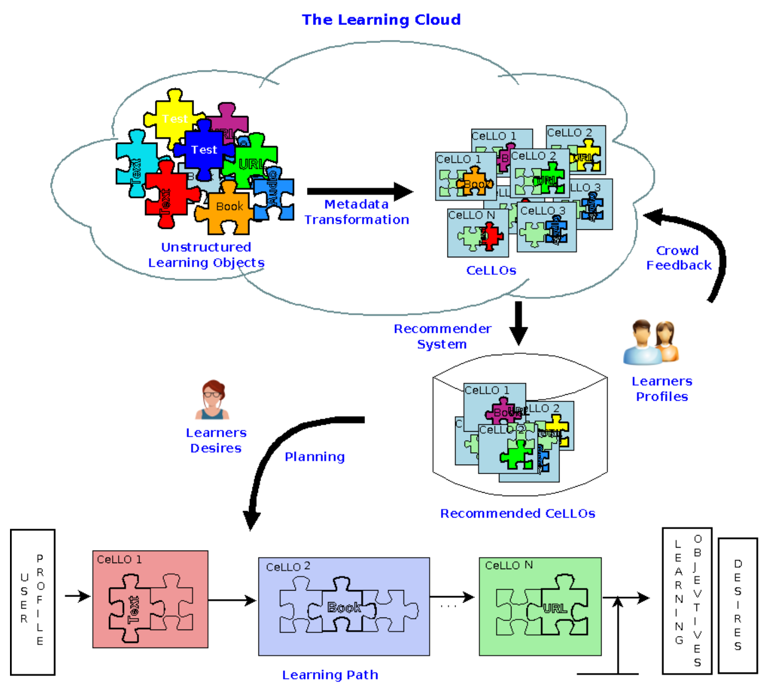

Fig. 1. The overall view of CeL

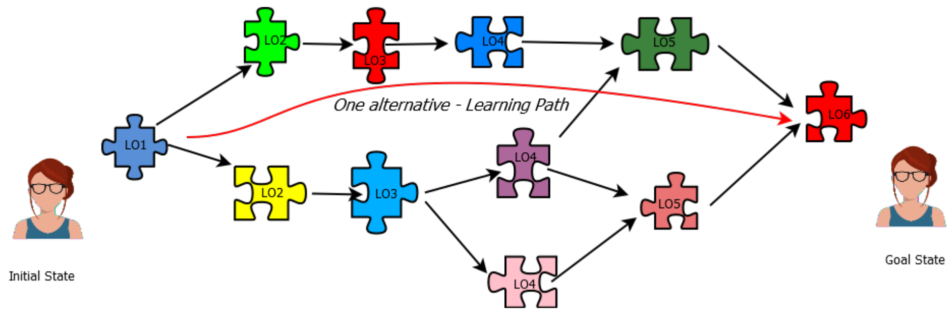

Fig. 2. The proposed learning path from CeL planner

Figure 1 presents the big picture of the CeL, including all the processes and technologies used. To provide limited number of learning objects to the planner that matches learners profile and desire, the CeL Recommender System filters the most relevant ones. Details of the various processes involved are presented elsewhere, such as the representation of learning objects [38] and the recommender system [39].

The automated planning as the final process shown in Figure 2 generates a personalised learning path, considering the background of the learner together with their desire as initial state, and the achieved learning outcomes as the goal state. In a nutshell, the plan defines a sequence of CeLLOs having learning outcomes (LeOs) that correspond to what the student knows and what the student achieves respectively. Planning offers alternative learning paths in case that a learner needs to backtrack to a previous point due to failing to meet the LeOs.

4.2 CeL as a Planning Problem

Therefore, with the process described above we end up with a pool of suitable CeLLOs that would take part in the planning process. Formally, the Planning in CeL is a tuple:

$$PCeL = (S_{cel}, A_{cel}, \gamma_{cel}, s_{0cel}, g_{cel}) \tag{3}$$

where:

- S_{cel} is the set of all possible propositions that describe the user profile, knowledge, skills and desires
- A_{cel} is the set of all CeLLOs
- γ_{cel} is the state transition function which given a state of a learner and a CeLLO returns a new state which includes new knowledge and skills that the learner has acquired through this CeLLO
- s_{0cel} is the initial state of the learner
- g_{cel} is the set of goal states that include the desires in terms of skills and knowledge by the learner

In the context of CeL, defined in previous papers [**? ?**], the planning approach as the final phase, where all recommended CeLLOs, are offered as part of the planning problem and the CeL planner, will try to synthesize the right CeLLOs in the personalised sequence based on learners background and learners interest (Algorithm 1).

Algorithm 1: Invoking Automated Planning in order to generate a personalised learning path

Input : Recommended CeLLOs from the CeL and profile constraints of learner
Output: personalised learning path for the learner

```
if !isEmpty(recommendedCeLLOs) then
    Action 1: Select the potentially relevant existing CeLLOs;
    Action 2: Insert the selected CeLLO from CeL to the plan;
    Action 3: Propose the personalised plan to the learner;
else
    reInitiate the CeLRS;
end
```

4.3 Planning in CeL: An example

Here we present an example, in which a learner (learner 1) is interested to learn java so that she can be able to acquire skills at level 4 of the bloom taxonomy, i.e. analysis. The learner profile is listed among other profiles in Table 3.

Table 3. Sample Learner Profiles

Name	Knows	Type of Learner	Desires to Learn
learner1	maths at level(1) and algorithms at level (1)	visual	java at level(4)
learner2	maths at level(3)	audio	ai at level(4)

Learner1 expresses her desire through an unstructured query. The CeL recommender system filters the number of available CeLLOs which might be relevant to the desire of the learner. Relevance is determined also by the ontology related to the desire, in this case, java is related to variables, control statements of programming languages through the ACM ontology [40]. Some of them are videos, audios, podcast and others texts format types, while some others are self-evaluation tests to assess learner's progress (Table 4). The CeLLOs that are potentially relevant contain materials about algorithms, java, object oriented programming and maths. In each of the CeLLOs the cognitive level of the contained material is defined (Bloom level), as well as the pre-requisites required

Table 4. Sample CeLLOs in some abstract format

Type of Learner	Available Format	Cello ID	Bloom level	Topic	Prerequisites
visual	video	c1	4	java syntax	none
visual	video	c2	3	oop	none
visual	video	c3	3	algorithms	control statements at level(3) and variables at level(3)
visual	text	c4	1	maths	none
visual	text	c5	3	control statements	none
visual	text	c6	3	variables	none
audio	podcast	c7	3	control statements	none
audio	podcast	c8	3	variables	none
any	test	t1	4	java syntax	none
any	test	t2	3	oop	none
any	test	t3	3	algorithms	none
any	test	t4	1	maths	none
any	test	t5	3	control statements	none
any	test	t6	3	variables	none

in order to be able to deal the material. For example, in order to deal with algorithms one must deal with control statements and variables (CeLLO c3).

A simple linear Planner will create a goal state start out of the desires of the learner. The learner's profile forms the initial state. The plan generated is the learning path which consists of the most appropriate CeLLOs.

In our example the personalised learning path for learner1 based on her profile and her desires is as follows:

1. Watch *c2*, a video on *oop*;
2. Take the test *t2*;
3. Study text *c5* on *control statements*;
4. Take the test *t5*;
5. Study text *c6* on *variables*;
6. Take the test *t6*;
7. Watch the video *c3* on *algorithms*;
8. Take the test *t3*;
9. Watch the video *c1* on *Java syntax*;
10. Take the test *t1*.

4.4 Discussion

In addition to the previous examples, there might be a need to define the duration of each action (watch, study, take test etc.) that the learner should do. In such case, we should specify the time frames as constraints for the action, precondition and effects. If we consider the same actions with planning and scheduling

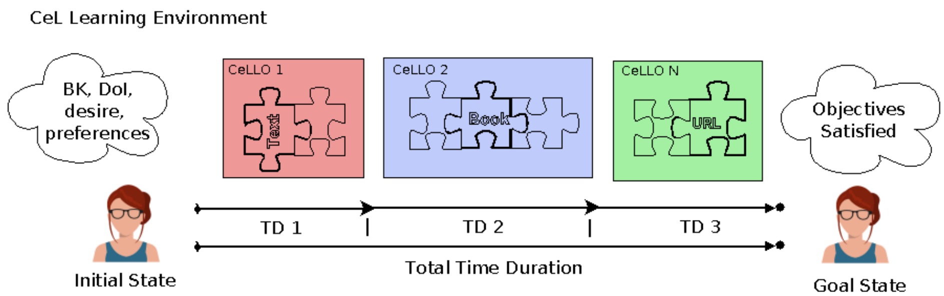

Fig. 3. The CeL Planning Domain

techniques, beside the constrainsts, the action is specified with its resource requirements as well (which might be consumable or reusable resources) and three variables (starting time, ending time and duration).

In CeL, the CeLLOs are treated as reusable resources, which have fixed duration, as shown in figure 3. During learning, the learner may face problems, that is, fail to follow the personalised path for some reason, e.g. fail the assessment test. In such case, the planner should be able to define alternatives learning paths or to re-plan from that point of failure.

5 Conclusion

We have formally defined Cloud e-Learning as a Planning problem with the goal to find a personalised learning path for any learner with a specific profile and particular desires to acquire new knowledge and skills. The validity of the approach was demonstrated through an example. So far, we have managed to implement the problem using linear planning, i.e. STRIPS notation, through PDDL. Future work will include to consider the temporal planning techniques and to investigate more the benefits of Planning and Scheduling techniques, particularly the case of 'job-shop' problem, as a new technology which besides the time constraints deals also with resource constraints, as consumable or borrowable resources.

Bibliography

[1] Krenare Pireva and Petros Kefalas. The use of multi agent systems in cloud e-learning. In *Doctoral Student Conference on ICT*, pages 324–336, 2015.

[2] Krenare Pireva, Petros Kefalas, Dimitris Dranidis, Thanos Hatziapostolou, and Anthony Cowling. Cloud e-learning: A new challenge for multi-agent systems. In *Agent and Multi-Agent Systems: Technologies and Applications*, pages 277–287. Springer, 2014.

[3] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004.

[4] Drew McDermott and James Hendler. Planning: what it is, what it could be, an introduction to the special issue on planning and scheduling. *Artificial Intelligence*, 76(1-2):1–16, 1995.

[5] Dana Nau, T-C Au, Okhtay Ilghami, Ugur Kuter, Dan Wu, Fusun Yaman, Héctor Munoz-Avila, and J William Murdock. Applications of shop and shop2. *IEEE Intelligent Systems*, 20(2):34–41, 2005.

[6] Raphaël Lallement, Lavindra De Silva, and Rachid Alami. Hatp: An htn planner for robotics. *arXiv preprint arXiv:1405.5345*, 2014.

[7] J Scott Penberthy, Daniel S Weld, et al. Ucpop: A sound, complete, partial order planner for adl. *Kr*, 92:103–114, 1992.

[8] Earl D Sacerdoti. A structure for plans and behavior. Technical report, DTIC Document, 1975.

[9] P Fournier-Viger and L Lebel. Pl-plan, java open-source ai planner. http://www.philippe-fournier-viger.com/plplan/, 2009. Online; accessed 10 May 2017.

[10] Romeo Sanchez Nigenda, XuanLong Nguyen, and Subbarao Kambhampati. Altalt: Combining the advantages of graphplan and heuristic state search. *Knowledge Based Computer Systems*, pages 409–421, 2000.

[11] Emil Keyder and Héctor Geffner. The ff (ha) planner for planning with action costs. *Proceedings of the International Planning Competition (IPC)*, 2008.

[12] Ioannis Refanidis and Ioannis Vlahavas. Grt: a domain independent heuristic for strips worlds based on greedy regression tables. In *European Conference on Planning*, pages 347–359. Springer, 1999.

[13] Alfonso Gerevini and Ivan Serina. Lpg: A planner based on local search for planning graphs with action costs. In *AIPS*, volume 2, pages 281–290, 2002.

[14] Håkan LS Younes and Reid G Simmons. Vhpop: Versatile heuristic partial order planner. *Journal of Artificial Intelligence Research*, 20:405–430, 2003.

[15] Nathaniel Waisbrot, Ugur Kuter, and Tolga Könik. Combining heuristic search with hierarchical task-network planning: A preliminary report. In *FLAIRS Conference*, pages 577–578, 2008.

[16] Henry Kautz and Bart Selman. Satplan04: Planning as satisfiability. *Working Notes on the Fifth International Planning Competition (IPC-2006)*, pages 45–46, 2006.

[17] Jussi Rintanen. Madagascar: Scalable planning with sat. *Proceedings of the 8th International Planning Competition (IPC-2014)*, 2014.

[18] Stephen M Majercik and Michael L Littman. Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence*, 147(1-2):119–162, 2003.

[19] Henry Kautz and Bart Selman. Unifying sat-based and graph-based planning. In *IJCAI*, volume 99, pages 318–325, 1999.

[20] Alfonso Gerevini, Alessandro Saetti, Ivan Serina, and Paolo Toninelli. Lpg-td: a fully automated planner for pddl2. 2 domains. In *In Proc. of the 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04) International Planning Competition abstracts*, 2004.

[21] Jonas Kvarnström and Patrick Doherty. Talplanner: A temporal logic based forward chaining planner. *Annals of mathematics and Artificial Intelligence*, 30(1):119–169, 2000.

[22] J Benton, Amanda Jane Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. In *ICAPS*, volume 77, page 78, 2012.

[23] Keith Halsey, Derek Long, and Maria Fox. Crikey-a temporal planner looking at the integration of scheduling and planning. In *Workshop on Integrating Planning into Scheduling, ICAPS*, pages 46–52, 2004.

[24] Kristian J Hammond. Chef: A model of case-based planning. In *AAAI-86 Proceedings*, pages 267–271, 1986.

[25] Brian P Kettler, James A Hendler, William A Andersen, and Matthew P Evett. Massively parallel support for case-based planning. *IEEE Expert*, 9(1):8–14, 1994.

[26] Manuela M Veloso. *Planning and learning by analogical reasoning*, volume 886. Springer Science & Business Media, 1994.

[27] I Vlahavas and I Refanidis. Planning and scheduling. http://www.eetn.gr/index.php/about-eetn/eetn-publications/ai-research-in-greece/planning-and-scheduling. Online; accessed 10 May 2017.

[28] Rune M Jensen and Manuela M Veloso. Aset: A multi-agent planning language with nondeterministic durative tasks for bdd-based fault tolerant planning. In *Proc. 15th International Conference on Automated Planning and Scheduling (ICAPS-05).2005*, 2005.

[29] Antonín Komenda, Jiří Vokřínek, Michal Pěchouček, Gerhard Wickler, Jeff Dalton, and Austin Tate. Distributed planning and coordination in non-deterministic environments. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1401–1402. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[30] Hisashi Hayashi. Stratified multi-agent htn planning in dynamic environments. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 189–198. Springer, 2007.

[31] Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins SRI, Anthony Barrett, Dave Christianson, et al. Pddlthe planning domain definition language version 1.2. 1998.

[32] M Kalisch and S König. Comparison of strips, adl and pddl, introduction to ai planning. *Lecture Notes, Bamberg University.[4] Baker, K*, 1998.

[33] David R Krathwohl. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218, 2002.

[34] Antonio Garrido, Eva Onaindia, Lluvia Morales, Luis Castillo, Susana Fernández, and Daniel Borrajo. Modeling e-learning activities in automated planning. In *Proceedings of International Competition on Knowledge Engineering for Planning and Scheduling*, 2009.

[35] Marıa D R-Moreno, David Camacho, and Unai Obieta. A plan-based tool for automatic elearning courses redesign. *International Journal of Computer Science Applications*, 2008.

[36] Antonio Garrido, Eva Onaindia, and Oscar Sapena. Automated planning for personalised course composition. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pages 178–182. IEEE, 2009.

[37] Dimitris Vrakas, Fotis Kokkoras, Nick Bassiliades, and Ioannis Vlahavas. Towards automatic synthesis of educational resources through automated planning. In *Hellenic Conference on Artificial Intelligence*, pages 421–431. Springer, 2006.

[38] K Pireva, P Kefalas, and I Stamatopoulou. Representation of learning objects in cloud e-learning. Work in progress, Paper Submitted 2017.

[39] K Pireva and P Kefalas. A recommendation system based on hierarchical clustering for cloud e-learning. Work in progress, Paper Submitted 2017.

[40] The 2012 acm computing classification system. https://www.acm.org/publications/class-2012, 2012. Online; accessed 10 May 2017.

View publication stats