Unified Planning: A Python Library Making Planning Technology Accessible

Andrea Micheli,[1] Alexandre Arnold,[2] Arthur Bit-Monnot,[3] Luigi Bonassi,[6] Luca Framba,[1] Alfonso Emilio Gerevini,[6] Selvakumar Hastham Sathiya Satchi,[3] Malte Helmert,[8] Félix Ingrand,[3] Luca Iocchi,[7] Uwe Köckemann,[5] Oscar Lima,[4] Fabio Patrizi,[7] Federico Pecora,[5] Guillaume Poveda,[2] Gabriele Röger,[8] Alessandro Saetti,[6] Alessandro Saffiotti,[5] Enrico Scala,[6] Ivan Serina,[6] Sebastian Stock,[4] Florent Teichteil-Koenigsbuch,[2] Alessandro Trapasso,[7] Paolo Traverso,[1] Alessandro Valentini[1]

[1]Fondazione Bruno Kessler, [2]Airbus AI Research, [3]LAAS-CNRS, [4]DFKI, [5]Örebro University,
[6]Università degli Studi di Brescia, [7]Università degli Studi di Roma La Sapienza, [8]University of Basel

amicheli@fbk.eu, alexandre.arnold@airbus.com, abitmonnot@laas.fr, framba@fbk.eu, alfonso.gerevini@unibs.it, shasthamsa@laas.fr, malte.helmert@unibas.ch, felix@laas.fr, iocchi@diag.uniroma1.it, uwe.kockemann@oru.se, oscar.lima@dfki.de, patrizi@diag.uniroma1.it, federico.pecora@oru.se, guillaume.poveda@airbus.com, gabriele.roeger@unibas.ch, alessandro.saetti@unibs.it, asaffio@aass.oru.se, l.bonassi005@unibs.it, enrico.scala@unibs.it, ivan.serina@unibs.it, sebastian.stock@dfki.de, florent.teichteil-koenigsbuch@airbus.com trapasso@diag.uniroma1.it, traverso@fbk.eu, alvalentini@fbk.eu

Abstract

The aim of the AIPlan4EU project is to make state-of-the-art planning technology easily accessible for real-world applications and to integrate planning as a service on the European AI-on-demand platform. At the very heart of the project is the development of a unified planning framework that permits to programmatically specify planning tasks and to interact with different planning engines through a common interface.

The demo[1] presents the AIPlan4EU vision for the unified planning framework and the current state of the development.

Introduction

Planning is a relevant technology for many application areas such as agile manufacturing, agrifood or logistics. Although there are many techniques that are mature in terms of science and systems, several obstacles hinder their adoption to practise. For example, it can be hard to find the right techniques for a given planning problem, there are no shared standards to use them, and there is no easy access to expertise on how to encode domain knowledge into a planner.

The AIPlan4EU[2] project aims to address these obstacles in several ways: 1) by developing a uniform, user-centered framework to access the existing planning technology, 2) by implementing bridges that connect the framework with established industry standards used in different application domains, 3) by devising concrete guidelines for innovators and practitioners on how to use this technology, and 4) by making planning available as a service on the European AI-on-Demand platform[3]. All efforts of the project are driven by the needs of actual real-world use cases, both from within the project consortium and recruited by means of cascade funding. In the ICAPS 22 demo we focus on the unified planning framework and the current state of its development.

[1]https://youtu.be/DhZpdTRnfyU

[2]https://www.aiplan4eu-project.eu/

[3]https://www.ai4europe.eu/

Vision

Figure 1 depicts the role of the unified planning framework within the overall AIPlan4EU vision. The framework provides an API that allows the users to specify their planning task programmatically and to interact with the planning engines. A planning engine can be any tool useful for planning, for example a standard one-shot solver, a provider of task transformations, or a plan validator.

Planning engines are integrated via a plugin system in which they declare what planning modes they support and whether a certain task is within their scope. Based on this information, the framework can select or suggest suitable engines and their interaction is coordinated by means of the unified planning framework. Expert users have the opportunity to fully take control and to precisely request specific systems and configurations.

For using planning technology for a specific use-case, one can either directly access the API provided by the unified planning framework or build on a *technology-specific bridge* (also developed within the project), that connects the framework with established technology from the application domain (such as common warehouse management systems).

The Unified Planning Framework

The Unified Planning framework[4] is a Python3 library (indicated as UP library) giving uniform access to multiple planning approaches and engines. The library is released under the permissive Apache 2.0 open-source license and can be used as a technological enabler for quickly prototyping planning-based applications. The library is designed around two concepts: advanced modeling primitives to express planning problems mixing declarative and procedural paradigms; and "*operation modes*", which define the classes of interaction with planning engines.

[4]https://github.com/aiplan4eu/unified-planning

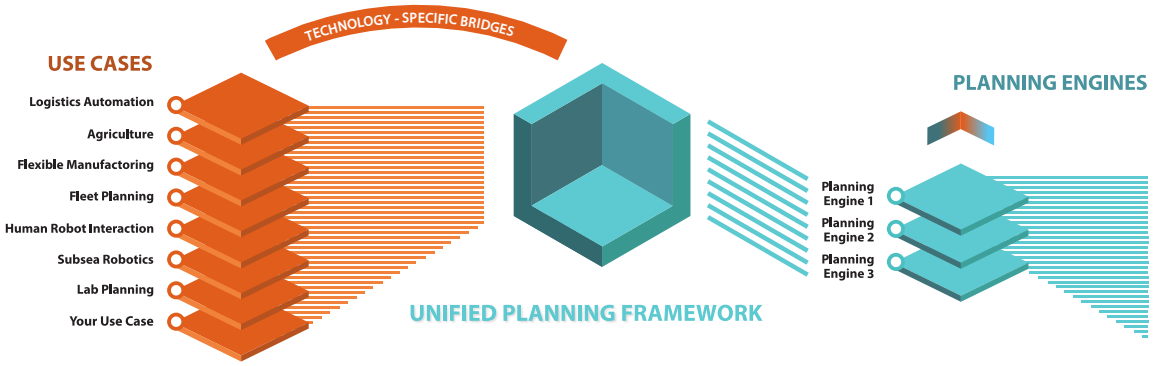

Figure 1: The AIPlan4EU vision

The UP allows the creation and manipulation of planning problems using features from classical, numeric and temporal planning. The core of the modeling is the `Problem` class that contains all the aspects of the planning specification. Such a class can be manually populated using the UP API, or can be generated by parsing a formal language (e.g., PDDL), or can be created by the interoperability interfaces with other frameworks (e.g., *Tarksi*), or can be obtained by mixing the previous approaches. For example, it is possible to parse a partial PDDL specification (e.g., the definition of the types and actions for a navigation problem) and programmatically complete the planning problem definition using any data available and arbitrary Python logic (e.g., creating the set of locations from a GUI or fixing the topology using online mapping services). The UP library also proposes the idea of "transformers", which are model-to-model rewritings that can be used for simplifying or compiling away modeling features such as conditional effects and disjunctive preconditions.

Once a problem is defined in the UP library, we can pass it to planning engines via "operation modes". An operation mode is an interface abstracting a possible interaction with a class of planning engines giving access to the engine functionalities under different circumstances (e.g., open-loop, closed-loop, anytime, validation). The simplest operation mode is the `OneshotPlanner`, which defines the interface (as Python methods signatures) that an engine must offer in order to be used as a one-time plan generator (e.g., how planners are used in the IPC). Differently, a `PlanValidator` operation mode defines the interface for checking if a given plan is valid for a problem. Operation modes serve two purposes: first, they ease the extension of new planning engines by simply creating an implementation of the operation mode; second, they abstract the peculiarities of every planning engine, allowing the creation of pipelines of heterogeneous tools. For example, one could use planning engine A to ground a problem, create a plan using another planning engine B and repair a plan at execution time through a third planning engine C. This has the potential of extending the reach of automated planning substantially as one can take the best of all worlds. The library automatically detects the modeling features used in a problem and asks each engine to declare which problem kind the engine does support. The UP library can thus automatically filter the planning engines and select the appropriate ones.

The design of operation modes is a continuous process over the entire project: it considers both requirements from use cases, and actual capabilities of planning engines.

Similar Initiatives

Tarksi (Francès, Ramírez, and Collaborators 2018) is a framework for the modeling of planning tasks, and the UP already exploits its grounding mechanism. The problem of setting up and using different planning systems in a uniform way is also tackled by *Planutils* (Muise et al. 2022). *Scikit-decide* (Arnold et al. 2019)[5] is an open-source library initiated by Airbus AI Research which provides a common interface for solvers and domains in Planning, Scheduling and Reinforcement Learning. It allows practitioners to search for solvers from different research communities which are compatible with a given domain's characteristics and to easily compare them (e.g. solving scheduling problems with Constraint Programming or Reinforcement Learning). UP provides a bridge to *scikit-decide* which can support procedurally defined action effects. The UP main novelty is to provide in a unified way both the abstraction facilities for calling different planning systems homogeneously (similar to *planutils*) and programmatic access to the planning problem definition (similar to *Tarski*, but with support for several planning paradigms).

Ongoing and Future Work

At the moment the UP supports three different operation modes: `OneShotPlanner`, `PlanValidator` and `Grounding`, and is ongoing the design of new operation modes to support the use of planning in an online setting, and in contexts where solution quality matters. Also, we are working on extending the modeling features to support multi-agent planning and hierarchical structures.

Acknowledgements

The AIPlan4EU project has received funding from the European Union's Horizon 2020 research and innovation programme under GA no. 101016442.

[5] https://github.com/airbus/scikit-decide

References

Arnold, A.; Dupont, G.; Geisser, F.; Gretton, C.; Poveda, G.; Régnier-Coudert, O.; Teichteil-Königsbuch, F.; Thiébaux, S.; and Trevizan, F. W. 2019. AIRLAPS: an AI toolbox for Reinforcement Learning, Planning and Scheduling. In *System Demonstrations Track of the 29th International Conference on Automated Planning and Scheduling (ICAPS 2019)*.

Francès, G.; Ramírez, M.; and Collaborators. 2018. Tarski: An AI Planning Modeling Framework. https://github.com/aig-upf/tarski.

Muise, C.; Pommerening, F.; Seipp, J.; and Katz, M. 2022. Planutils: Bringing Planning to the Masses. In *System Demonstrations Track of the 32nd International Conference on Automated Planning and Scheduling (ICAPS 2022)*.