Filuta: Operationalized Composite AI Platform

Filip Dvorak, Slavomír Švancár, Tomáš Balyo, Josef Ryzí, Michal Ficek, Martin Doušek

Filuta AI
Pernerova 51, Prague 186 00, Czechia
{filip,slavo,tomas,josef,michal,martin}@filuta.ai

Abstract

This paper introduces the Filuta platform, which aims to address the limitations of current AI and ML tooling by providing a versatile environment for composing and integrating multiple modeling techniques, such as AI Planning, machine learning, scheduling, graph theory, constraint programming, linear programming, and large language models. Recognizing the value of the composition approach in solving real-world problems, the platform facilitates learning individual models from data subsets, enabling user augmentation, and generating stand-alone intelligent automation services. The paper further discusses the various stages of platform usage, from data ingestion and modeling to composition and deployment.

Introduction

Over the last two decades, the artificial intelligence (AI) landscape has been largely shaped by machine learning (ML) techniques, which have excelled in learning continuous functions from big datasets. These data-driven approaches have brought forth remarkable advancements in various domains, including image recognition, natural language processing, and recommendation systems. While we have seen very fast growth in tools supporting big-data-driven approaches to machine learning such as (LeDell and Poirier 2020), the tooling for causal and symbolic models often applied in AI Planning has been lagging behind, with a few notable exceptions such as (Dolejsi et al. 2019) and (Muise 2016).

There are many techniques for modeling real-world problems. Yet, we believe there is not and will not be a single technique that would strongly dominate as the most efficient modeling technique across the majority of real-world problems. Instead, modeling of real-world problems can be seen as a composition of sequenced, interleaved and nested techniques, using the most efficient technique for each of the sub-problems that the real-world problem decomposes into. We can see a significant industrial validation of the composition approach to solving industrial problems in the studies performed by (Gartner 2022), where Composite AI has shifted into one of the most promising techniques after deep learning and machine learning have been underdelivering in the industrial projects.

The goal of the Filuta platform is to provide a common ground for the composition of a multitude of modeling techniques, including AI Planning, machine learning, scheduling, graph theory, constraint programming, linear programming, and large language models such as GPT (Floridi and Chiriatti 2020). The platform already provides capabilities to learn individual models from subsets of provided datasets. It also allows the human user to augment the models, compose the models into an operational schema with well-defined inputs, outputs, and goals, and generate a stand-alone intelligent automation service. The service then acts towards user-defined goals based on the model composition and observation of the world.

It the paper we will often refer to our demonstration video (Filuta 2023), where we use the classic toy example of Hanoi towers, and the real-world example of Cloud Kitchens[1] to showcase the capabilities of our platform.

In the following sections, we describe the different stages of platform usage, from data ingestion to modeling, composition, and deployment.

Online and Offline Data Connectors

When building real-time intelligent services, we need to be capable of processing offline data, which are usually used for learning the models. They can be large, and their processing time is not significantly constrained. Online data are expected to be coming in real-time in the production environment when the service is deployed, and the results of processing the real-time data are also going to be produced and published by the service.

Filuta builds upon an existing framework of connectors implemented in (Airbyte 2023) supporting several hundreds of online and offline data sources from CSV to (databricks 2023). Currently we support tabular data for ML models and time-series for synthesizing planning models. We will be expanding to further modalities in the future. The simplest experiment can be performed by dropping a file into the platform canvas in a browser, which uploads the file and creates a file-connector. A good example of time-series data is a log from the execution of a plan as shown in (Filuta 2023).

[1] where the Filuta platform has been deployed to optimize the operations from food production to order delivery.

Modeling

The modeling phase encapsulates multiple model design and learning techniques. Applying different filters and transformations upon a dataset ingested from a data connector, the user can define the state space, which is then used for learning the model. The same dataset can be used to define state spaces for multiple different modeling approaches. For example, we can imagine a predictive task, that is modeled as a dependency of the temperature of an engine based on speed, road incline, and the total weight, and at the same time we can model a vehicle logistics using classical planning (Ghallab, Nau, and Traverso 2004).

Filuta has been supporting modeling with classical planning using (Dvorak, Agarwal, and Baklanov 2021) machine learning using (LeDell and Poirier 2020) and constraint modeling using CP-SAT in (Google 2022).

Large Language Models

Large language models (LLM) recently gained strong traction and the access to them has improved significantly. We demonstrate stable capabilities of LLM for labeling and describing planning actions in the Filuta platform and in the demo (Filuta 2023) building upon the text-davinci-003 LLM (open.ai 2023).

Our further experiments will be focused on building simulators, compositions and models from customized LLMs

Composition

Following on (Dvorak 2021) we are encapsulating reasoning and action execution into the nodes of a behavior tree providing the grounds to define a composed automated planning and execution system in terms of tree composition. The tree composition can be seen as an operationalization of multiple models together with the corresponding solvers, actors, goals, and APIs into a service. Any two models in a composition are related either by

- nesting, when one model is fully nested within another, and the solver operating with the outer model can invoke the solver operating the inner model,
- precedence, when solver for the earlier model can be fully executed before running the solver for the later model,
- interleaving, when solvers for both models can run concurrently and interfere with each other for example by exchanging their best solutions.

These three possible model composition approaches are available to the solution architect and it is upon the architect to design a composition appropriate for the given real-world problem. While the Filuta platform offers blueprint-compositions for different industries, we can often observe variations that fundamentally change the composition - e.g. having a customer who operates distribution centers both in cities and rural areas. They may want to use a composition where a bin-packing problem precedes vehicle routing in the cities and vice-versa in the rural areas, where the vehicles are rarely full.

The models share a single observation space, picking only the data relevant for them - e.g. in our Cloud Kitchen example (Filuta 2023), vehicle routing with time windows precedes running a planner. We first produce optimal assignments of orders to drivers, which is equivalent to adding control knowledge to the more general planning problem, which is solved consequently. We consider each model to be tied with a solver, which keeps solving the problem instances generated from the observations every time it is triggered.

The concept of model composition in the Filuta platform is a tool we give to the solution architect. If desired, the architect can build and compose models to guarantee optimal solutions. There can also be cases when optimality is infeasible and the goal is to find solutions quickly with a reasonable distance from optimality.

Deployment

Deployment of the services produced by Filuta is a one-click action from the user's point of view and a cloud-agnostic deployment of a Kubernetes cluster (Kubernetes 2023) at the back end. The user can then monitor the status of the deployment and interact with it at the defined IP through the API captured in the composition.

Acknowledgment

The application of the Filuta platform for the Cloud Kitchen use-case has been partially supported by AIPlan4EU within the framework "H2020 Programme for Research and Innovation" and the source code of the components has been released under the Apache 2.0 license.

Summary

We have introduced Filuta as an intelligent automation platform that allows us to model real-world problems both in hand and through learning techniques, compose the models together with goals and APIs into an intelligent automation service, and deploy the service in a one-click user experience. This paper is tightly connected and references our online demonstration video (Filuta 2023).

The work on the Filuta platform is ongoing with a large pipeline of experiments especially in the area of LLM applications in design, modeling, and automated domain synthesis. At the same time, we are growing the number of industrial applications of our platform in various domains ranging from manufacturing and logistics to compliance, banking, insurance and game-testing.

Finally, the whole platform is expected to be modeled in itself in near future.

References

Airbyte. 2023. Airbyte.com.

databricks. 2023. www.databricks.com.

Dolejsi, J.; Long, D.; Fox, M.; and Muise, C. 2019. From a Classroom to an Industry From PDDL "Hello World" to Debugging a Planning Problem. In *System Demonstrations at the Twenty-Ninth International Conference on Automated Planning and Scheduling (ICAPS)*.

Dvorak, F. 2021. Real-time Planning and Execution for Industrial Operations. In *Proceedings of 9th ICAPS Workshop on Planning and Robotics*.

Dvorak, F.; Agarwal, A.; and Baklanov, N. 2021. Visual Planning Domain Design for PDDL using Blockly.

Filuta. 2023. youtu.be/5npMi6V9Te8.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30: 681–694.

Gartner. 2022. What's new in Artificial Intelligence from the 2022 Gartner Hype cycle.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory and Practice*. The Morgan Kaufmann Series in Artificial Intelligence. Amsterdam: Morgan Kaufmann. ISBN 978-1-55860-856-6.

Google. 2022. Google OR-Tools.

Kubernetes. 2023. Kubernetes.io.

LeDell, E.; and Poirier, S. 2020. H2o automl: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML*, volume 2020.

Muise, C. 2016. Planning.Domains. In *The 26th International Conference on Automated Planning and Scheduling - Demonstrations*.

open.ai. 2023. open.ai.