eProPlan: a tool to model automatic generation of data mining workflows

Kietz, J U ; Serban, F ; Bernstein, A

eProPlan: A Tool to Model Automatic Generation of Data Mining Workflows

— Extended Abstract —

Jörg-Uwe Kietz and **Floarea Serban** and **Abraham Bernstein**[1]

Abstract. This paper introduces the first ontological modeling environment for planning Knowledge Discovery (KDD) workflows. We use ontological reasoning combined with AI planning techniques to automatically generate workflows for solving Data Mining (DM) problems. The KDD researchers can easily model not only their DM and preprocessing operators but also their DM tasks, that are used to guide the workflow generation.

1 Introduction

Current DM-Suites support the user only with the manual creation of KDD workflows, but this is time consuming and requires experience and knowledge about the DM operators as well as the DM-Suites. Over the course of time several people have tried to build systems that can automate this process[1, 2, 11]. These approaches have shown that AI planning is a way of supporting the user with automatically generated workflows. But modeling for planning is difficult as well since one has to describe the DM domain in terms of operations that can be applied. To our knowledge none of these systems contain a model of the operators of a state-of-the-art DM-Suites nor are they publicly available.

In this paper we present eProPlan[2], the first ontology-based Integrated Development Environment (IDE) for planning applications. Together with the DMWF-DMO (Data Mining Work Flow-Ontology) and over 100 modeled RapidMiner operators it is the first publicly available planner for planning DM workflows. We also defined a programming interface (IDA-API), such that generated workflows can be delivered directly to DM-Suites like RapidMiner or general Service-Workflow engines like Taverna [6].

In Section 2 we present the architecture of our system, Section 3 describes the steps and audience of the demonstration, we conclude with a short discussion of innovations of eProPlan over related work in Section 4.

2 eProPlan Architecture

The system comprises several Protégé 4 [9] plugins, that allow the users to describe and solve DM problems. By using the public-domain ontology-editor Protégé 4 as the base environment we exploit the advantages of an ontology as a formal model for the domain knowledge. But instead of over-using the ontological inferences for planning (as done in [3, 12]) we decided to extend the ontological formalism with the main components of a plan, namely operator conditions & effects for classic planning and tasks & methods for Hierarchical Task Network (HTN) planning [10].

DMO is our DM-Ontology. It consists of two main parts: The DMWF [7] which contains IO-objects, operators, goals, tasks and methods as well as the decomposition of tasks into methods and operators. The DMWF is used by the AI-planner. The DMOP [4] focuses on the operators' features and is used by the probabilistic-planner to optimize the plans generated by the AI-planner.

eProPlanI is our reasoner plugin, i.e. the interface of our reasoner & planner for Protégé. It combines ontological reasoning and HTN-planning. Other plugins (like eProPlanP) rely on it since they retrieve and display useful information about the DM process (operators and their applicability). **eProPlanO** is an editor for operators. Operators have conditions and effects expressed in an extended SWRL language [5] as shown in [7]. Our customized tab displays for each operator its conditions and effects (both current and inherited), only current conditions and effects can be edited. We extended the SWRL editor from Protégé 4 and built our own editor that has a syntax checker and a variable binding checker.

eProPlanM provides the support for HTN modeling as a task/method editor, users can define their own task/method decompositions. Methods and tasks' modeling – consists of three steps and is managed by a customized tab with three class views. The first view presents a tree with the decomposition of tasks into methods, methods into tasks and operators classes. The user can easily add/ delete new tasks and methods as well as specify their decomposition into substeps. The next view displays the conditions and contributions for methods. The third view shows the method bindings for a selected method.

eProPlanG represents a DM-task specification editor. It enables the user to specify the input data as well as describe the DM-task she wants to solve in terms of main goals and optional subgoals the planner has to reach.

eProPlanP is the support for workflow execution and visualization. It has mainly two functionalities: provides the user a step-by-step planner – it displays applicable operators for a data set and can apply them, and shows all the plans for a specific DM task.

IDA-API is the programming interface to the reasoner & planner used to build Intelligent Discovery Assistants (IDA) based on the services of the planner. Within the e-Lico project Rapid-I is currently using this API to integrate the planner into the leading open-source DM-Suite RapidMiner, such that RapidMiner users can get automatically generated DM workflows to execute them there.

[1] University of Zurich, Department of Informatics, Dynamic and Distributed Information Systems Group, Binzmühlestrasse 14, CH-8050 Zurich, Switzerland {kietz|serban|bernstein}@ifi.uzh.ch

[2] eProPlan, DMWF-DMO and RapidMiner-operator ontology can be downloaded from http://www.e-lico.eu/eProPlan. The IDA-API and its integration into RapidMiner will be available soon. For a first impression this site also contains screen-shoots and preliminary demo videos.

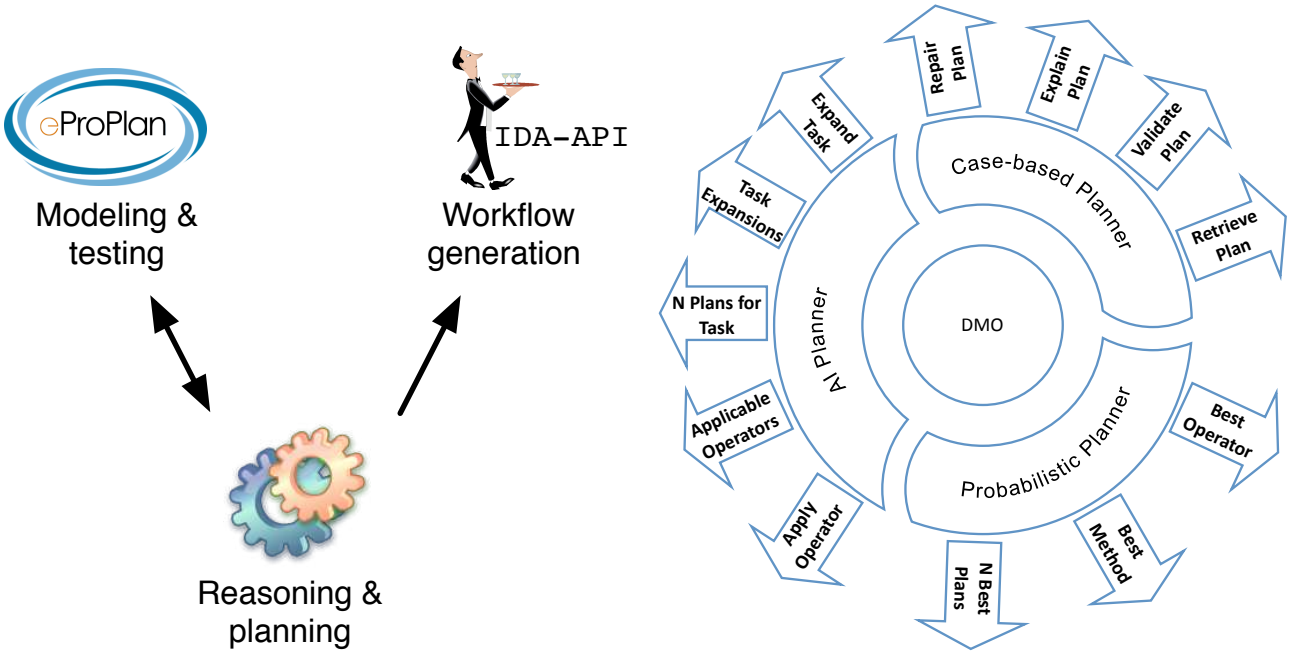

Figure 1: (a) eProPlan architecture (b) The services of the planner

The complete system will have the services displayed in Figure 1b which are developed as part of the e-Lico project. Currently the AI planner's services are available, the other two parts are under development.

3 Demonstration

The demonstration will cover the whole life-cycle of DM-workflow planning from modeling data sets, preprocessing- and DM-operators, DM-goals and task/method decompositions, via testing the model in eProPlan by entering specific goals and getting the DMWF-metadata description of concrete data sets from a data analysis service, to the generation of complete DM-workflows within eProPlan and within RapidMiner, where the generated workflows can be executed.

The demo is of interest for the KDD researchers as well as for the KDD practitioners. Researchers may want to model their own operators to be used in generated workflows. An advanced usage scenario is that researchers model their own task/method decompositions, e.g. to let the planner generate systematic experiments to be executed by a DM-Suite. DM-Suite developers may be interested in the services the system can add to their suite. Practitioners can get an impression of how future systems can support them in their daily work of doing DM projects. An extended description of the system and how it is used for auto-experimentation is available in [8].

4 Conclusions

There are many attempts at automating the generation of KDD workflows: Žáková et. al [12] automatically generate workflows using a knowledge ontology and a planning algorithm based on the Fast-Forward system. However, they only return the shortest workflow with the smallest number of processing steps – no alternatives are generated. IDEA [1], the Intelligent Discovery Assistant (IDA), provides users with systematic enumerations of valid DM processes. It is based on an ontology of DM operators that guides the workflow composition and contains heuristics for the ranking of different alternatives. CITRUS [11] consists of an IDA which offers user-guidance through mostly a manual process of building the workflows. It uses planning for plan decomposition and refinement.

Similar to IDEA our system also provides many plans. We believe that this is a necessary condition as the system may not know all the tradeoffs between operators and/or the user's desiderata. Also, as far as we know, ePropPlan is the first and only KDD workflow planner that combines this ability with an integrated Planning Development Environment, can be easily integrated into existing DM suites through its IDA-API, as it is publicly available.

Acknowledgements: This work is supported by the European Community 7^{th} framework ICT-2007.4.4 (No 231519) "e-Lico: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science".

REFERENCES

[1] Abraham Bernstein, Foster Provost, and Shawndra Hill, 'Towards Intelligent Assistance for a Data Mining Process: An Ontology-based Approach for Cost-sensitive Classification', *IEEE Transactions on Knowledge and Data Engineering*, **17**(4), 503–518, (2005).

[2] M. Charest, S. Delisle, O. Cervantes, and Y. Shen, 'Bridging the gap between data mining and decision support: A case-based reasoning and ontology approach', *Intelligent Data Analysis*, **12**(2), 211–236, (2008).

[3] Claudia Diamantini, Domenico Potena, and Emanuele Storti, 'KD-DONTO: An Ontology for Discovery and Composition of KDD Algorithms', in *Service-oriented Knowledge Discovery (SoKD-09) Workshop at ECML/PKDD09*, (2009).

[4] Melanie Hilario, Alexandros Kalousis, Phong Nguyen, and Adam Woznica, 'A data mining ontology for algorithm selection and meta-learning', in *Service-oriented Knowledge Discovery (SoKD-09) Workshop at ECML/PKDD09*, (2009).

[5] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, http://www.w3.org/Submission/SWRL/, 2004.

[6] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M.R. Pocock, P. Li,

and T. Oinn, 'Taverna: a tool for building and running workflows of services', *Nucleic acids research*, **34**(Web Server issue), W729, (2006).
[7] J.-U. Kietz, F. Serban, A. Bernstein, and S. Fischer, 'Towards cooperative planning of data mining workflows', in *Service-oriented Knowledge Discovery (SoKD-09) Workshop at ECML/PKDD09*, (2009).
[8] J.-U. Kietz, F. Serban, A. Bernstein, and S. Fischer, 'Data mining workflow templates for intelligent discovery assistance and autoexperimentation', Technical report, University of Zürich, (06 2010). Available at `http://www.e-lico.eu/public/SoKD10.pdf`.
[9] H. Knublauch, R.W. Fergerson, N.F. Noy, and M.A. Musen, 'The Protégé OWL plugin: An open development environment for semantic web applications', *Lecture notes in computer science*, 229–243, (2004).
[10] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, W. Murdock, D. Wu, and F.Yaman., 'SHOP2: An HTN planning system.', *JAIR*, **20**, 379–404, (2003).
[11] R. Wirth, C. Shearer, U. Grimmer, T. Reinartz, J. Schlösser, C. Breitner, R. Engels, and G. Lindner, 'Towards process-oriented tool support for knowledge discovery in databases', *Principles of Data Mining and Knowledge Discovery*, 243–253, (1997).
[12] M. Žáková, V. Podpečan, F. Železný, and N. Lavrač, 'Advancing data mining workflow construction: A framework and cases using the orange toolkit', in *Service-oriented Knowledge Discovery (SoKD-09) Workshop at ECML/PKDD09*, (2009).