Analyzing Plans and Comparing Planners in itSIMPLE$_{3.1}$

Tiago Stegun Vaquero[1,2] and **José Reinaldo Silva**[1] and **J. Christopher Beck**[2]
[1]Department of Mechatronic Engineering, University of São Paulo, Brazil
[2]Department of Mechanical & Industrial Engineering, University of Toronto, Canada
tiago.vaquero@poli.usp.br, reinaldo@usp.br, jcb@mie.utoronto.ca

Introduction

Real planning problems arise from real application domains. One significant challenge to achieving satisfactory planner performance, in terms of both plan quality and planning speed, is the development of a clear understanding and accurate model of the application domain. Lack of knowledge or ill-defined requirements typically propagate to poor project specifications, then to an erroneous planning model and finally to unsatisfactory planner performance. Ideally, assumptions and models leading to incorrect and poor quality plans should be spotted and fixed in the design process. One useful approach to finding such problems is the analysis of plans generated by different planning techniques. Hence the need to include a re-modeling cycle in any real application design process.

Following this idea, itSIMPLE$_{3.1}$ not only allows users to test the PDDL model that is generated from a UML specification with a set of modern planners, it also provides a set of tools for plan analysis. Besides the existing capabilities of plan visualization in UML and the definition of plan quality metrics, itSIMPLE$_{3.1}$ brings extended features aimed to help users to (1) perform experiments with different planners, (2) evaluate plan quality, (3) compare planner performance, and (4) compare different model refinements. In all these new features, users observe and analyze automatically generated reports that contain useful information for investigating plans, the performance of planners, and the impact of certain modifications.

In this short paper, we briefly describe these new features in itSIMPLE$_{3.1}$. We start by introducing how users define and represent the quality metrics that guide the plan evaluation. We then show how user can set-up planning experiments to study and analyze planners and domain models. Finally, we describe the reports generated by itSIMPLE$_{3.1}$.

Defining Quality Metrics

Some of the research effort in the itSIMPLE project has been directed to plan quality analysis. One of the extended functionalities available in itSIMPLE$_{3.1}$ supports the definition of plan quality metrics and criteria acquisition. The main objective of this new functionality is to capture domain metrics and criteria from users and to use them to evaluate and compare plans. This feature aims to help the designer identify and explore their own metrics and their preferences on the metric values.

In order to capture metrics and criteria, itSIMPLE provides an interface in which users specify and select the variables that correspond to key parameters for measuring the quality of the plan. Metrics can be, for instance, a variable of the domain (e.g., *travel-distance* or *total-fuel-use*), an action counter that can involve specific characteristics (e.g., how many times action *move* appears in the plan with the first parameter being *loc1*), or an linear function involving several domain variables. These metrics can be maximized or minimized by planners or just observed by users. Each one of these metrics can have a preference function that maps variable values to scores in the interval [0,1] (where 0 is unsatisfactory and 1 is satisfactory). The definition of metrics in itSIMPLE was inspired by the work of (Rabideau, Engelhardt, and Chien 2000).

These metrics and their preference functions are used to evaluate the plans produced by planners. These plan evaluations can be used while analyzing models and planners or when performing planning experiments. The evaluations (metric values and plan scores) can support and lead designers to modified their models accordingly to their expectations. Such modification process is performed manually, but automatic refinement procedures have been investigated.

Performing Experiments with Planners

A number of planners can be used within itSIMPLE$_{3.1}$'s graphical interface: Metric-FF, FF, SGPlan, MIPS-xxl, LPG-TD, LPG, hspsp, SATPlan, Plan-A (IPC-6), blackbox (version 4.3), MaxPlan (IPC-5), LPRPG (beta version 1), and Marvin (IPC-4). Since itSIMPLE$_{3.0}$, the requirements tags of the (automatically generated) PDDL can be used to select the planners that can handle a given domain.

Generally, during research experiments in planning, we might want to do the following: test a specific domain model with different planners; test a particular planner with different planning domains; compare the performance of a set of planners in a given domain model; or compare the performance of a set of planners in a set of planning domains (what is generally done during the International Planning Competition). itSIMPLE$_{3.1}$ allows the user to perform all these kinds of experiments.

In itSIMPLE$_{3.1}$, experiments are normally done as follows. Users first select which planners and domain models will be used in an experiment. The tool lets designers specify time-outs for all planners or a specific time-out for a particular planner. The tool, then, handles the experiment automatically, while showing to the user the progress and the status of the process. During the execution of every planner, itSIMPLE$_{3.1}$ records essential information and data: not only the speed (runtime) and solvability of the planner for a given problem instance, but also the quality of the resulting plans based on the defined metrics. All information and data from the experiments are recorded in a XML file which is used to display the results to the user in the form of a report.

Generating Reports

In this section we describe the reports that itSIMPLE$_{3.1}$ can generate from the data record of the experiments.

Plan Report

When a user wants to analyzed a particular plan, itSIMPLE$_{3.1}$ can generate a HTML report that shows basic information about the planner, the evolution of all metrics using charts (so user can identify peaks, maximum and minimum), the individual score for each metric, and, finally, the overall score of the plan.

Plan and Planner Comparison Report

When considering experiments with multiple planners and/or multiple domains, itSIMPLE$_{3.1}$ generates a comparison report that shows how planners perform for each problem instance concerning speed, solvability, number of actions, plan cost, quality of metrics, and plan quality. The report contains tables that list all these data. The report also contains two charts for every domain in the experiment: the first one correlates "number of actions" and "planners" considering every problem instance in the domain; the second correlates "speed" (time) and "planners" also considering every problem instance in the domain. These charts are very similar to those presented in the IPC results. At the end of the comparison report, we provide a summary of the best planners concerning the categories speed, quality and plan length. This summary is made by counting how many times each planner dominates on problem instances in each category. This report, also in HTML format, can be very useful to identify better planners as well as critical domains and problem instances. In fact, it can simulate the evaluation process generally done in IPC.

Since every plan is stored in the experiments data file, users can quickly simulate or visualize a chosen plan using itSIMPLE's interface to do a deeper investigation. Comments can be added to the plan which is stored in the XML file for further analysis and reuse.

Comparing Refined Domains

Plan analysis can help validation of the model and can also guide model modification and refinement. Recent research work with itSIMPLE (Vaquero, Silva, and Beck 2010) has shown that several observations, hidden requirements, and potential modifications to the model can be discovered while simulating the plan in a virtual environment. These modifications produce new (refined) models and a lead to subsequent plan analysis. The cycle of re-modeling and analysis naturally raises the need to compare the impact of inserted modifications, i.e., comparing different planners on different versions of the model. In order to help on such comparison tasks, itSIMPLE$_{3.1}$ produces a special report that combines several experiments on a particular domain. With such a combination of data, the tool can show (1) the model in which the planners produced the the best quality plans, (2) the model in which the planners had the fastest response and (3) the model in which the planners had the best plan length. The resulting report contains tables that show the performance of the planners in each problem instance of every (refined) model and also column charts illustrating the best models for each problem instance and for all experiment. Figure 1 illustrates an overall evaluation of four models where each criteria has been mapped so that higher means better. This figure shows that, in this case, Model AB is the best model for all criteria.

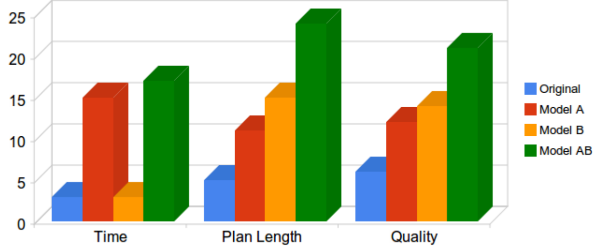

Figure 1: Overall comparison of models in itSIMPLE$_{3.1}$

Conclusion

The itSIMPLE project is in ongoing development. We have recently put some efforts on integrating itSIMPLE with other tools (such as virtual prototyping environments, model checking, and the Automatic Validation tool (VAL) for PDDL), as well as improving the modeling features. itSIMPLE$_{3.1}$ can be found in our website http://dlab.poli.usp.br.

References

Rabideau, G.; Engelhardt, B.; and Chien, S. 2000. Using generic preferences to incrementally improve plan quality. In *Proceedings of the Fifth AIPS, Breckenridge, CO.*

Vaquero, T. S.; Silva, J. R.; and Beck, J. C. 2010. Improving planning performance through post-design analysis. In *Proceedings of ICAPS 2010 workshop on Scheduling and Knowledge Engineering for Planning and Scheduling (KEPS). Toronto, Canada. To be published.*