# LoLA as Abstract Planning Engine of PlanICS*

Artur Niewiadomski[1] and Karsten Wolf[2]

[1] ICS, Siedlce University, 3-Maja 54, 08-110 Siedlce, Poland
artur.niewiadomski@uph.edu.pl
[2] ICS, University of Rostock, 18051 Rostock, Germany
karsten.wolf@uni-rostock.de

**Abstract.** An abstract planning (AP) is the first phase of the web service composition in the PlanICS framework. We propose an automatic translation of AP to reachability problem in high-level Petri nets, and exploiting the LoLA tool to solve it. We present our approach together with a prototype implementation, and preliminary experimental results.

**Keywords:** Web Service Composition, Abstract Planning, PlanICS, LoLA

## 1 Introduction

PlanICS [1] is a system that solves the Web service composition problem by dividing it into several stages. The first phase, called the *abstract planning*, deals with an *ontology* which contains a hierarchy of *classes* describing sets of real-world services and processed object types.

This paper reports the use of LoLA [3] tool as an abstract planning engine. To this aim, we developed a translator which takes an ontology and a user query as the input, and builds a high-level Petri net augmented with a state-predicate formula $P$. The translation is performed in such a way that some marking satisfying $P$ is reachable only if the planning problem has a solution. Moreover, the comparison of very first results with those obtained with SMT-based abstract planner [2] shows that our new approach is very promising.

## 2 Translation

At this early stage of our work we assume that service descriptions do not contain alternatives, and we do not take the type inheritance into account. According to the restrictions above, the main ideas of our translation are as follows. The object types of PlanICS become LoLA sorts, i.e., domains for tokens on places. These sorts are records consisting of components corresponding to object attributes. Moreover, for each object type we put a single place in a high-level output net,

* Partially supported by the European Union from resources of the European Social Fund. Project PO KL “Information technologies: Research and their interdisciplinary applications”, Agreement UDA-POKL.04.01.01-00-051/10-00.

**Table 1.** Experimental results

| | | **avg** | | | | LoLA | | PlanICS | |
|---|---|---|---|---|---|---|---|---|---|
| **types** | **attrs** | **objs** | **conds** | **services** | **length** | **time[s]** | **mem[MB]** | **time[s]** | **mem[MB]** |
| 78 | 58 | 3,14 | 1,58 | 64 | 6 | **0,05** | **6,71** | 3,88 | 12,17 |
| 92 | 62 | 3,98 | 2,06 | | 12 | **0,07** | **7,26** | 155 | 34,76 |
| 106 | 74 | 3,88 | 2,17 | | 15 | **0,08** | **7,59** | 593 | 63,19 |
| 137 | 136 | 3,53 | 1,77 | 128 | 6 | **0,33** | 30,32 | 16,57 | **16,67** |
| 143 | 113 | 3,28 | 1,55 | | 12 | **0,06** | **9,92** | 242 | 44,16 |
| 142 | 112 | 3,34 | 1,71 | | 15 | **0,06** | **9,85** | 1007 | 69,8 |
| 151 | 209 | 3,74 | 1,89 | 256 | 6 | **4,95** | 434 | 37,26 | **21,11** |
| 151 | 173 | 3,6 | 1,76 | | 12 | **12,1** | 934 | 659 | **69,01** |
| 151 | 166 | 3,51 | 1,75 | | 15 | **22,5** | 1639 | 1038 | **85,85** |
| 101 | 141 | 5,16 | 5,06 | 50 | 5 | 382 | 6643 | **13,18** | **17,12** |
| 121 | 145 | 6,92 | 10,4 | | | - | >7000 | **15,43** | **19,28** |
| 127 | 132 | 7 | 7,2 | | | 67,3 | 1045 | **12,63** | **17,97** |
| 127 | 181 | 6,1 | 4,62 | | 10 | - | >7000 | **92,66** | **90,92** |
| 119 | 193 | 5,96 | 4,9 | | | - | >7000 | **155** | **230,7** |

and thus *worlds* (sets of objects in some states) are represented by markings. The PlanICS services, which transform worlds by producing new objects and changing states of existing ones, become transitions consuming and producing tokens from/to world places w.r.t. service descriptions. Finally, the part of user query specifying the initial worlds is translated to a set of (initially marked) places and transitions which put appropriate tokens to world places. Similarly, a set of special places and transitions, together with a state-predicate formula, correspond to the query fragment concerning the goal of the plan. Moreover, some reductions of an input ontology have been implemented. Their aim is to remove all object types and attributes which are neither used by services, nor query, what leads to a significant improvement of LoLA performance.

## 3 Experiments and Conclusion

In order to evaluate the efficiency of our approach, we generated a number of parametrized random benchmarks using PlanICS Ontology Generator. Then, we tried to solve them exploiting a prototype PlanICS2LoLA implementation and LoLA tool, as well as an SMT-based planner. The obtained results are summarized in Table 1. We marked in bold the better ones. The overall conclusion is that due to ontology reductions applied during the translation, LoLA in many cases outperforms the SMT-based abstract planner.

## References

1. D. Doliwa et al. PlanICS - a Web Service Compositon Toolset. *Fundam. Inform.*, 112(1):47–71, 2011.
2. A. Niewiadomski and W. Penczek. Towards SMT-based Abstract Planning in PlanICS Ontology. In *Proc. of KEOD 2013*, pages 123–131, 2013.
3. K. Schmidt. LoLA: A Low Level Analyser. In *Application and Theory of Petri Nets 2000*, volume 1825 of *LNCS*, pages 465–474. Springer Berlin Heidelberg, 2000.