On the Use of UML.P for Modeling a Real Application as a Planning Problem

Tiago Stegun Vaquero[1] Flavio Tonidandel[2] Leliane Nunes de Barros[3] José Reinaldo Silva[1]

[1]Escola Politécnica – Universidade de São Paulo - Design Lab. – São Paulo, Brazil
[2]Centro Universitário da FEI
IAAA – Artificial Intelligence Applied in Automation Lab - São Bernardo do Campo, Brazil
[3]Instituto de Matemática e Estatística – Universidade de São Paulo - São Paulo, Brazil
Email: tiago.vaquero@poli.usp.br; flaviot@fei.edu.br; leliane@ime.usp.br; reinaldo@usp.br

Abstract

There is a great interest in the planning community to apply all developments already achieved in the area to real applications. Such scenario makes the community focus on Knowledge Engineering (KE) applied in modeling of planning problems and domains. In this paper, we propose the use of UML for Planning Approach, denominated UML.P, during planning domain modeling process. We also discuss the exposure of UML.P to a real application, e.g., the sequencing car problems in an assembly line. This modeling experience, using a classical manufacturing problem, provides some insights and considerations that can contribute to a general KE process for planning.

Introduction

The recent efficiency improvement and the rising demand for planning systems have become a great motivation to apply all developments already achieved, in real and complex applications. In this scenario, Knowledge Engineering methodologies become more important since modeling actions are considered to be the bottleneck of practical planning systems development. This has been addressed in several initiatives, such as the first International Competition on Knowledge Engineering for Planning and Scheduling - ICKEPS 2005. This competition brought extremely important modeling issues and showed powerful tools, such as itSIMPLE (Vaquero et al 2005) and GIPO (McCluskey et al 2003), that can help designers to better understand, specify, verify and validate their planning models.

itSIMPLE proposes a special use of UML – Unified Modeling Language (OMG, 2003) - in a planning approach, named UML.P, to be used during the planning domain specification and modeling process. We believe that such an approach can contribute to finding better modeling solutions as well as to identify relevant domain issues and features that otherwise could not be recognized by a totally action-driven specification.

This paper, based on (Vaquero et al, 2006), describes an experience on the exposure of UML.P to a real application such as the Car Sequencing problem in an assembly line (Nguyen 2003). This is a classical manufacturing problem which encompasses many challenging features such as planning with resources, sequencing, scheduling, optimizing, and others.

The Car Sequencing problem is based on the daily job of factories where a production day to each ordered vehicle must be assigned according to production lines capabilities and delivery dates. This interesting domain was extracted from the fourth ROADEF Challenge 2005 (Nguyen 2003). In this paper, we show that, by using UML.P, we can recognize some interesting features during the modeling process which gives important information for the final planning domain model.

In order to verify the obtained model we have chosen the Metric-FF planner (Hoffmann 2003), to generate plans to be further analyzed by the ROADEF solution-checking tool. Since there was no planner capable of reading a domain specification in UML, the UML.P was translated to the PDDL 2.1 (Fox and Long, 2003) format which was given as an input for Metric-FF.

Modeling with UML.P

The UML – Unified Modeling Language is one of the most used languages to model a great variety of applications and it was first defined in the OMG Unified Modeling Language Specification between 1996 and 1997 (OMG, 2003). Besides, the UML has flexibility to assist many kinds of models in an object-oriented fashion.

A well-known method and a modeling process that drives the modeler through the entire process (like those intrinsic in UML) can help experts and non-experts developers and planning modelers to design their models.

Since UML is a general purpose modeling language, some specification features are intrinsically related to planning domains. For that reason, the UML.P (UML in a Planning Approach) was first defined at (Vaquero *et al.* 2005), where the planning concepts are specified in UML.

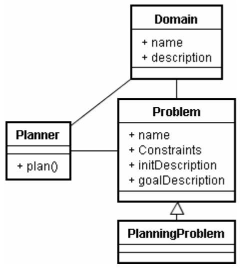

Figure 1: General schema for planning environment

This approach first considers relationships between planners, domains and planning problems. In a planning context, the modeling process follows the principles that: domains have their own description and specification; problems are associated to domains and they have their own constraints, initial conditions and goal descriptions; planners plan over associated problems and domain descriptions. These principles distinguish the description of a domain from the description of a problem, based on the knowledge of the role of a planner. Figure 1 shows a schema where these concepts are summarized.

In UML, Use Case specifications are usually described in natural language, but UML.P makes it different. Since natural language specifications can create redundancies, a proposal of using a structured Use Case specification contributes to minimize these problems.

In UML.P, the Class, Object and StateChart Diagrams are used. In the Class Diagram, the classes' attributes and associations give a visual notion of the semantic.

There are three kinds of associations: Simple Association, Aggregation and Composition. The simple associations just connect classes identifying some appropriate meanings. This kind of association has a name, a semantic direction, role names and a multiplicity definition on both sides of the connection. The multiplicity can be seen as constraints in the class diagram. It shows how the classes associate with each other. The Aggregation and the Composition Associations are a special kind of association which can be read as a "have", "belong" or "made of" association.

In order to specify the dynamic behavior of actions, the *StateChart diagram* is necessary. The constraints on the Class diagram and pre and post conditions on the StateChart diagram are specified using the language OCL (OMG 2003).

Any class in a Class diagram has its own StateChart diagram especially those that perform actions. Each diagram does not intend to specify all changes caused by an action, instead, it shows only the changes that it causes in an object of the StateChart diagram's class.

A problem statement in a planning domain is characterized by a situation where only two points are known: the initial and goal state. The diagram used to describe these states is called Object Diagram or can be called a snapshot of the class diagram.

Car Sequencing as a Planning Problem

The Car Sequencing problem requirements, which will be used as the running example throughout the paper, were extracted from an important system competition called ROADEF 2005 provided by RENAULT Corporation. In the Car Sequencing planning process, customer orders are sent to car factories in real-time. The factory must assign the production to each ordered car according to the delivery dates, constraints and production line capabilities. The car sequence established daily for a production line has a direct effect on the factory and on the quality of the car. More details can be found in (Nguyen 2003).

This challenging manufacturing planning problem encompasses interesting features such as planning with resources, sequencing, optimization, flexibility and others that make the problem even more complex when combined. All these aspects make this planning domain an excellent challenge for a planning driven modeling process such as the proposed UML.P.

The requirements focus basically on paint shop, assembly line and body shop. On one hand, it must be considered the minimization of the paint solvent which is used to wash the spray guns each time the color is changed. On the other hand, the smoothness of the workload must be considered. In fact, it is important to avoid an overload of the assembly line by some special features of a car. Therefore, a ratio N/P is considered, where N is the number of vehicles in a consecutive sequence and P is vehicles with special features.

The violation of each special feature ratio N/P must be minimized by the planner. The domain and problem becomes more interesting when the ordered vehicles require multiple special features in a production making the sequencing problem even harder.

UML.P Model Representation

The planning domain modeling process followed three main phases: (i) Problem Requirements, (ii) Design and (iii) Model Test and Result Analysis.

The initial phase encompassed the familiarization with the problem and the complete understanding of the domain and the definition of a sound set of planning domain's features. All the requirements were gathered and discussed towards a unified view of the domain and problem. From the analysis and discussion of the car sequencing problem requirements described previously, it was possible to define the Domain scope using three Use Case diagrams in a high level of abstraction. For instance, in order to better clarify what really happens during the vehicle painting, the Activity Diagram was used for a visual explanation. Figure 3 shows such a diagram.

The flow in Figure 3 starts at the left black circle and it ends at the right black circle. This activity diagram summarizes the role and capabilities of the *SprayGun* in our defined domain scope.

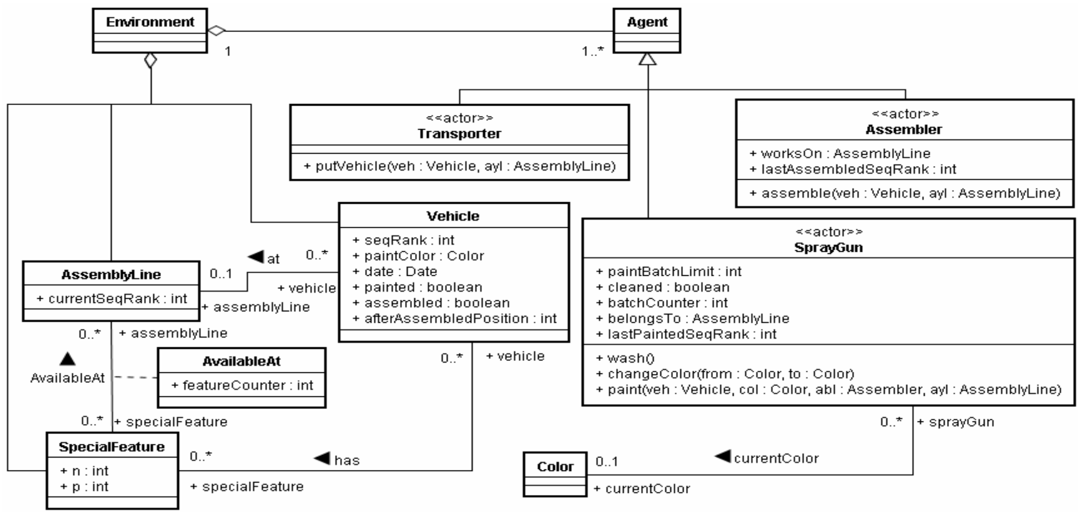

Figure 2: Class diagram for the Car Sequencing Domain

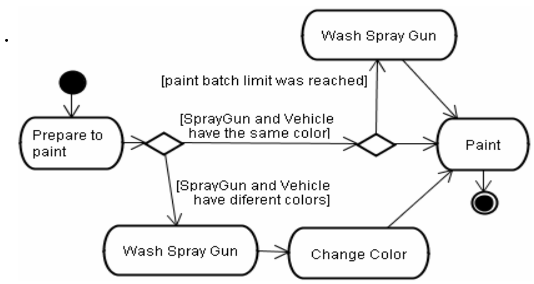

Figure 3 – Activity diagram for painting a vehicle

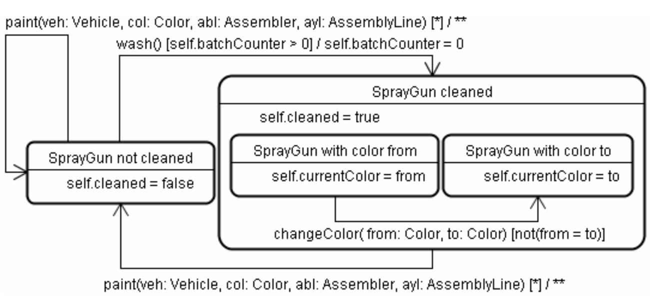

Figure 4: StateChart diagram – SprayGun

In order to structure all the static concepts of the domain with an object-oriented approach, the Car Sequencing Class diagram was built. Figure 2 shows the Class diagram resulting from several iterations between Class diagram and StateChart diagram specification

In order to model the dynamic behavior of the Car Sequencing domain it was necessary to use the StateChart diagram. Following the UML.P, the classes *Vehicle*, *Tranporter*, *SprayGun* and *Assembler* require a StateChart diagram. An example of the StateChart Diagram is described in figure 4.

Optimization aspects are important in the domain of car sequencing. Since optimization aspects concern the problem in form of solution constraints, it was preferable to model optimizations after having the entire model. We had only to decide which are the critical actions that minimize or maximize some resource or value. In our current example the critical actions for optimization are the *changeColor* (context *SprayGun*) and *assemble* (context *Assembler*) since we need to reduce the paint solvent and also penalize the sequence that does not satisfy the N/P pre-defined relation as explained below in this paper.

Semantic Translation: UML.P to PDDL

Since the Sequence Car domain was already modeled in UML.P, it was easier to specify the PDDL 2.1 model (Fox and Long, 2003) than if we tried to model this domain in PDDL from scratch. During the translation process, which was done by hand, it was clear that PDDL has some limitations such as: it is not possible to use subsequent conditional effects inside a universal quantification. This fact forced PDDL code to a repetition of constraints.

However, most constraints expressions described in OCL were translated to expressions in PDDL that use universal and existential quantifications. During the translation process, some constraints such as multiplicities among classes were lost.

To verify the model described in PDDL, generated by our modeling approach, we chose the Metric-FF (Hoffman 2003). Since we are concerned with quality, time was not taken into account.

In order to make this verification, two main scenarios in PDDL were generated. The first scenario (s1) represents the set of problems that focus only on the optimization of the paint solvent. The second scenario (s2) focuses on the optimization of the workload in an assembly line, i.e. the minimization of sequence violations. The Metric-FF result is schematically showed in Table 1 where the given sequence was v4, v1, v6, v5, v3, v8, v7 and v2.

D-1 (last vehicles)					Production day D (Metric-FF solution)							
A5	A4	A3	A2	A1	v4	v1	v6	v5	v3	v8	v7	v2
			X			X			X			X
				Y		Y		Y		Y		Y
		Z					Z					Z

Table1: Metric-FF solution for a given car sequencing problem. The columns represents which special features the vehicles have (ex: v1 has X and Y).

There are some solutions to ROADEF managed by Simulated Annealing or Constraint Satisfaction Programming (CSP) (IME, 2005). The Metric-FF results for these scenarios were compared with some of these systems by using a solution-checking tool to check the validity and quality of the solutions. This solution-checking tool, provided by the ROADEF competition organizers, penalizes each time there is a constraint violation . Table 2 shows the some results.

Scenario	Problem	Metric-FF	A1	A2	C1
s1	6 cars 2 colors	10000	10000	10000	10000
	8 cars 2 colors	30000	10000	40000	10000
	8 cars 3 colors	50000	20000	30000	20000
s2	6 cars 1 spf	0	0	0	0
	8 cars 1 spf	0	0	10000	0
	6 cars 2 spf	0	0	0	0
	8 cars 2 spf	2000	0	60000	0
	8 cars 3 spf	2000	0	90000	0

Table2: Results of the solution quality comparing Metric-FF, two Simulated Annealing (A1 and A2) and one CSP (C1). Score 0 means excellent solution. Spf stands for special features.

The Metric-FF solutions (output) have similar qualities when compared to the others. By analyzing the results we can say that planning domain model, created by UML.P methodology, is correctly leading the planner to find high quality solutions, very similar to those generated by dedicated scheduling techniques. The analysis also encourages the use of planning system to real domains as well as its improvements.

Modeling Considerations

Many approaches to create the ROADEF directly in PDDL 2.1 by our group have failed since none of them could come with a complete and executable set of actions for the problem. Using a modeling method that drives the whole modeling process was important to reach a good domain model. The top-down design phase (from more abstract to more specific) also contributed to a unified view of the domain and facilitates the contributions of different domain experts, stakeholders and planning experts. The idea is that a top-down design forces designers to make a deeper analysis of the domain and the problem requirements in all abstraction levels. An important point was the use of diagrams (visualization) during modeling process where all domain aspects were discussed in a unified viewpoint.

Some difficulties were faced during UML.P modeling process, mainly at StateChart diagrams. This diagram marks the transition between semi-formal to formal specification and this transition is sometimes not natural. Some modeling solutions were described in OCL specification which refines the diagrams. This iteration is necessary but it may confuse planning non-experts.

Future Works and Conclusions

This paper has shown a simplified real application, the Car Sequence problem extracted from the ROADEF Challenge 2005, modeled in UML.P for planning purposes. The model, initially described in UML.P, was translated to PDDL and verified by using the Metric-FF planner (Hoffmann 2003).

Many difficulties that a designer can find when modeling a domain from scratch using PDDL can be reduced or totally overcome by using UML.P and OCL.

The exposure of UML.P to this real application had shown that there is still a lot of work to be done. For the future, we intend to implement the new features incorporated by the ROADEF domain, like OCL and constraints, into the itSIMPLE tool.

References

Fox, M. and Long, D. 2003. PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research* 20:61-124.

Hoffmann, J. 2003. The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *JAIR magazine*, special issue of 3rd planning competition.

IME 2005. ROADEF Challenges. Available at http://grenoble.ime.usp.br/~gold/cursos/2005/mac5758/roadef.html

Nguyen, A. 2003. *Challenge ROADEF'2005: Car Sequencing Problem.* Renault - Information System and Technologies - Advanced Studies. France.

OMG - Object Management Group, 2001. *Unified modeling language specification: version 1.4.* URL http://www.omg.org/uml.

T. L. McCluskey, D Liu, R. Simpson. 2003. GIPOII: HTN Planning in a Tool-supported Knowledge Engineering Environment *Proceedings of ICAPS03 AAAI press.*

Vaquero, T. S., Tonidandel, F. and Silva, J.R. 2005. *The itSIMPLE tool for Modeling Planning Domains*. ICAPS 2005 Competition ICKEP, Monterey, California, USA.

Vaquero, T. S., Tonidandel, F. and Silva, J.R. 2006. *The itSIMPLE tool for Modeling Planning Domains*. Technical Report. Brazil. Available at: http://www.pmr.poli.usp.br/d-lab/site