Abstract

The planning problem is based on a production specification that is composed of real world tasks which can be described by functions and constraints. This work deals with a description of a reactive workshop guide tool system, which takes unknown factors into account. This system is based on task definition and the feasibility of the tasks can be verified by analyzing the coherence among their constraints by using constraints propagation.

Introduction

The AI planning problem is basically described as given an initial world description, a goal world description and a set of tasks/operations/actions, to find a sequence of actions, called plan, which leads from the initial world description to the goal world description. The approach presented here is the base of LIA’s research subject which deals with formal specification of operation tasks and reactive workshop guide tool. Figure 1 illustrates the structure of a reactive workshop guide tool system. The dot arrow represents the workshop feedback in a possible failure, brought about by unknown factors.

A formal task specification aims to describe what will be done but not how it will be done. Then, from a production specification, expressed in a specific technique vocabulary, the formal specification phase produces, based on task definition (Ouriachi 1991), a formal and general description of a task. Next, by way of an "off-line" planning (or predictive planning), a set of plans allowing to realize this task can be described (Ouriachi 1991; Cunha 1999). From this predictive plan, one basic plan is chosen, according to some criteria, to be carried out; orders (actions) are giving to the workshop to be realized as they advanced (on-line planning). The others plans can be used, entirely or partially, as an alternative solution if some problems happen at the time realization of the task. In the next sections, each one of these phases is presented.

Formal Specification

A formal task specification aims to describe the function of a task T without considering the manner of it will be realized (Ouriachi 1991). It is noted by a triplet $SF(T) = (\theta_I, \mathcal{H}, \theta_F)$ where: i) $\theta_I$ indicates the initial world descriptions and/or some preconditions needfull to carry out the task T; ii) $\theta_F$ indicates the goal world descriptions and/or some post-conditions which must be verified after performing the task T; finally iii) $\mathcal{H}$ is a function whose application allows a transformation of the initial world description into the goal world description.

A task specification can be refined by describing the various abstraction levels of its functionality. In a formal task specification, these abstraction levels can be expressed by using intermediated tasks description. In other words, in an abstraction level $i$, the formal specification of a task T can be expressed through an intermediated tasks set $\{T_{i1}, T_{i2}, \ldots, T_{in}\}$ with respective formal specification $SF(T_{ik}) = (\theta_{Ik}, \mathcal{H}_k, \theta_{fk})$ $1 \leq k \leq n$.

The $SF(T_{ik})$ expressions are named sub-specifications of T and the intermediated specification of T, at the level $i$, is done by the n-uple below :

$SF(T) = \langle (\theta_{i1}, \mathcal{H}_{i1}, \theta_{f1}), (\theta_{i2}, \mathcal{H}_{i2}, \theta_{f2}), \ldots, (\theta_{in}, \mathcal{H}_{in}, \theta_{fn}) \rangle_{op}$

where $op$ represents the relationship among the tasks $T_{ik}$, $1 \leq k \leq n$. In other words, $op$ indicates, for example, if the tasks $T_{in}$ (also called subtasks of T) must be realized in sequence ($op = s$) or if one of them can be chosen ($op = c$) to achieve the T functionality. A choice among subtasks of a task characterizes either a similarity between them or an option according to the case (preconditions and/or post-conditions).

The level the less abstract is represented by a named elementary specification which is expressed exclusively by predefined functions, which specify elementary tasks.
The operations set which compose $\Pi$ is chosen consisting in tasks $T_r, T_{r0}$ and $T_{r4}$. Thus, based on some criteria (e.g., time, costs), a basic plan $P_i$ can be carried out. The partial replacement means finding an alternative task that can replace only the failed subtask. Then, the execution of $P_i$ goes on according to the original basic plan. In the previous example, suppose $P_i = \langle SF(T_4), SF(T_2), SF(T_5), SF(T_{19}), SF(T_{20}), SF(T_{13}), SF(T_{18}), SF(T_3), SF(T_{17}) \rangle$. If the task $T_{19}$ fails, the on-line planning system can propose its replacement by $T_{20}$. After the execution of $T_{20}, P_i$ goes on with $T_{15}$. By contrast, if a failure happens at the time execution of $T_{17}$, only the predictive plan system can propose $T_{18}$ to stand in for $T_{17}$.

**Conclusion and Perspectives**

This paper presented an approach, based on task definition, which aims to supply a reactive workshop guide tool system. The various abstraction level of a task functionality on the one hand allows the reuse of task specifications and on the other hand can offer choices between tasks with similar functionality. These choices can facilitate a reactivity not only in the on-line planning level (scheduling revision [E. Bensana & Sicard 1993]) but also in the off-line planning level (planning revision). My personal further research topics include the introduction of constraints techniques in tasks feasibility analyze in predictive planning system, by analyzing, inspired by [Nareyek 1999], the E-task-graph constraint satisfaction. The goal is do not include in the predictive plan set a plan relative to tasks not well described, e.g., tasks whose the pre and/or post-conditions are no coherent.

**References**


