LAFORIA Technical Report 92/27, October 1992.

A Theoretical Analysis of the "Uselessness" of White Knights

Éric Jacopin
LAFORIA,
4, Place Jussieu, Boîte 169,
75252 Paris Cedex 05, France
jacopin@laforia.ibp.fr

Claude Le Pape, Jean-François Puget
ILOG,
2 Avenue Galliéni, BP 85,
94253 Gentilly Cedex, France
lepape@ilog.fr, puget@ilog.fr

ABSTRACT

The formal analysis of domain-independent planning has been largely influenced by David Chapman's system Tweak, built upon a modal truth criterion (MTC), which is proved correct and complete. However, D. Chapman ends his work arguing without any formal proof that the white knight technique used in his MTC is "pretty useless" (a fact informally accepted in the planning community). Here, it is formally shown that white knights are useless for the completeness of Tweak: although some partially ordered plans cannot be reached without white knights, it turns out that for every totally ordered plan, some partial order of some "smaller" plan can be reached without white knights. A new modal truth criterion derived from that of Tweak is presented, which (i) keeps the expressiveness of Tweak, (ii) entails correctness and completeness and (iii) does not contain the white knight strategy.

RÉSUMÉ

L'analyse formelle d'une algorithmique indépendante d'un domaine d'application a été très influencée par le travail de David Chapman qui introduit la notion de Critère Modal de Vérité (CMV) pour permettre à son planificateur, TWEAK, d'être correct et semi-décidable. Cependant, Chapman termine son analyse en argumentant sans preuve qu'une partie du CMV est plutôt inutile. Nous montrons ici formellement que cette inutilité est réelle en ce qui concerne la complétude de TWEAK: bien que la stratégie du white knight s'avère nécessaire pour construire certains plans non linéaires (un tel plan est donné en exemple), il apparaît que la complétude ne s'attache qu'au nombre des opérateurs d'un plan et pas à l'ordre entre ces opérateurs. Ceci est dû à la pauvreté de l'expressivité des opérateurs utilisés. Un nouveau critère modal de vérité est présenté ; celui-ci (i) conserve l'expressivité de TWEAK, (ii) la correction et la complétude et (iii) ne contient pas de white knight.

1. Introduction

In this paper, we present a reduction of the formal framework of conjunctive planning defined in Chapman's article [1], where he presents theoretical results along with a planner, Tweak. Tweak is a correct and complete planning procedure.

Chapman [1, page 361] argues that a certain part of his formal framework is "useless", but does not modify the planner so as to take into account his argument. We propose to take the formal framework of Tweak as a basis and see how the “uselessness” can be taken into account. It appears that the so-called white knight strategy is redundant, and therefore effectively useless to reach correctness and completeness, defined as follows in the framework: a partially ordered plan is fully correct if and only if each of its possible executions (i.e. a totally ordered plan) is correct; a planning procedure is complete if and only if it constructs a correct plan whenever one exits. Note that the number of operators in the partially ordered plan and in its possible executions is the same. Consequently, in finding a solution, the number of operators is crucial, not the partial order: there might be several different partial orders for the same total order. A new modal truth criterion without the white knight strategy is presented, which keeps correctness and completeness.

We start with the usual definitions; let L be a language whose alphabet only consists of predicate symbols (0-ary predicate symbols shall be called propositional symbols), constant symbols, variables, the connective $\neg$ (for "not") and the usual punctuation symbols "(", ")" and ",". Terms are only constructed from constant symbols and variables. The syntax for formulas is as follows:

(1) True and False are formulas.

(2) If t_1 and t_2 are terms then $t_1 = t_2$ is a formula.

(3) If $t_1,...t_n$ are terms and P is an n-ary predicate then $P(t_1,...,t_n)$ is a formula.

(4) If f_1 is a formula then $\neg f_1$ is a formula; the formula $(\neg\neg f_1)$ reduces to f_1.

Definition 1.1 A *state* or a *situation* is a set of formulas of L.

Definition 1.2 An *operator* O is a couple (pre,pos) where pre(O), the preconditions of the operator, and pos(O), the postconditions of the operator, are states such that:

(i) the operator is *applicable* in a situation s iff $pre(O) \subseteq s$.

(ii) let $pos_\neg(O)$ the set made of the negation of all formulas of pos(O) (abusively, $pos_\neg(O)=\neg pos(O)$); let \ and $\cup$ stand for the set difference and the set union, respectively. The resulting situation r of the *application* of the operator O in the situation s is:

$$r = (s \setminus pos_\neg(O)) \cup pos(O).$$

(iii) O *denies* p if $\neg p \in pos(O)$.

(ii) does not mean that the set difference shall be computed before the set union: every formula of pos(O) will be effective at the same time. Consequently, if f_L is a formula of L, then f_L and $\neg f_L$ cannot both belong to pos(O); otherwise, there is no way to determine the truth value of f_L. This entails that $(s \setminus pos_\neg(O)) \cup pos(O) = (s \cup pos(O)) \setminus pos_\neg(O)$.

Definition 1.3 A *planning problem* is a triple (I,G,Op) where I is the initial state of the problem, G the final state of the problem and Op is the set of available operators for the problem.

Definition 1.4 If (I,G,Op) is a planning problem, then (∅,I) is *the initial operator* and (G,∅) is the *final operator* of the planning problem.

Definition 1.5 A *partially ordered plan* is a couple $(\pi,<_\pi)$ where π is a set of operators and $<_\pi$ is a partial order on π.

Definition 1.6 A *totally ordered plan* is a couple $(\pi,<)$ where π is a set of operators and < is a total order on π.

Strictly speaking, plans include, as in [11], a third component, i.e. the set of unification constraints built from the constraint posting technique integrated in Tweak. These constraints are not our main concern here; therefore, for simplicity reasons, we chose not to include this component.

As a consequence of the above definitions, the couple ({(∅,I),(G,∅)},{((∅,I),(G,∅))}) is called the *initial plan* for the planning problem (I,G,Op); in any plan, the initial operator is such that it is before every other operator and the final operator is such that every other operator is before it.

There always exists a total order compatible with a partial order. Assuming a partially ordered plan $(P,<_P)$ is ultimately executed by applying, one at a time, every operator of P in respect with the partial order $<_P$, the execution of the plan transforms the partial order into a total order. A correct plan is such that its execution transforms the initial situation into the final one:

Definition 1.7 A *correct totally ordered plan* to a planning problem (I,G,Op) is a totally ordered plan $(\pi,<)$ such that:

- (∅,I) and (G,∅) belong to π.
- (∅,I) precedes every other operator in π.
- (G,∅) is preceded by every other operator in π.
- every operator is applicable in the situation resulting from the application of the preceding operators.

Definition 1.8 A *correct partially ordered plan* $(\Pi,<_\Pi)$ to a planning problem (I,G,Op) is a partially ordered plan such that every totally ordered plan $(\Pi,<)$ with < compatible with $<_\Pi$ is a correct totally ordered plan.

2. Tweak: A modal truth criterion for non linear planning

2.1. A modal truth criterion to ensure completeness

In his paper, Chapman presents a modal truth criterion (MTC) which, when treated as a procedure to incrementally build plans, allows only correct plans to be constructed [1, page 340]:

> *A proposition p is necessarily true in a situation s iff two conditions hold: there is a situation t equal or necessarily previous to s in which p is necessarily asserted; and for every step C possibly before s and every proposition codesignating with p which C denies, there is a step W necessarily between C and s which asserts r, a proposition such that r and p codesignate whenever p and q codesignate. The criterion*

for possible truth is exactly analogous, with all the modalities switched (read "necessary" for "possible" and vice versa).

With this MTC, Chapman introduces a vocabulary: situation t is the “establisher” of proposition p, every step C a “clobberer” and every step W a “white knight”; finding or introducing an establisher is the process of establishment; processing clobberers is declobbering; promotion is the process of constraining s before C; and demotion is the process of declobbering with a white knight when the white knight is the establisher itself (i.e. constraining C before t). There are two possible uses of the MTC: (i) to verify that a given plan is correct and (ii) to incrementally build a correct plan if there exists one. Within the Tweak framework, the verification problem is decidable and the plan construction problem is semi-decidable: as a verification procedure, the MTC allows to determine whether a partially ordered is correct or not (in time polynomial in the number of operators in the plan); interpreted as an incremental planning procedure, the MTC allows to find a plan if one exists (and it may loop when there is no correct solution to the planning problem).

The MTC is built into Chapman's planner, Tweak, as a goal achievement procedure (GAP) which ensures that partial plans built by the planner are correct: the GAP is "derived by interpreting the necessary truth criterion as a non-deterministic procedure" [1, page 341]. As in Chapman's paper, let $\aleph$ stand for "necessary", = and ≠ stand for codesignation and noncodesignation respectively, < and ≤ stand for time order and the temporal precedence or equality respectively; then, in logical notation, the necessary truth criterion reads:

$$
\begin{array}{l}
\exists t,\ \aleph\ (t \leq s) \wedge \aleph\ \text{asserted-in}(p,t) \wedge \\
\quad \forall C,\ \aleph\ (s \leq C) \vee \\
\quad\quad \forall q\ \aleph\sim\text{denies}(C,q) \vee \\
\quad\quad\quad \aleph\ (p \neq q) \vee \\
\quad\quad\quad \exists W\ \ \aleph\ (C < W) \wedge \\
\quad\quad\quad\quad \aleph\ (W < s) \wedge \\
\quad\quad\quad\quad \exists r\ \text{asserts}(W,r) \wedge (p{=}q \Rightarrow p{=}r)
\end{array}
$$

Notice that to interpret the above formula, it is necessary to extend the notion of temporal order < so that < applies to both situations and operators. Chapman does not clearly state his conventions about this. The above formula holds under the following interpretation: if s_1 is the situation prior to the application of S and s_2 the situation following the application of S, we have $s_1 < S < s_2$. A logical reconstruction of the MTC using only plan steps and no modal operator can be found in [2,7]. A statement of the MTC with situations only can be found in [6] and a statement of the MTC with operators only in [11].

2.2. Semi-decidability of MTC; looping on finite problems

The outcome lemma [1, page 371] presents the three possible outcomes of the algorithm built in Tweak: (1) success, (2) failure and (3) looping. Success and failure are unavoidable: if there exists only one solution, Tweak must find it; if there exists none, Tweak may say so. But if looping might be unavoidable in the general case where a Turing machine can be encoded into the Tweak representation with an infinite set of constants describing the planning problem — first undecidability theorem [1, page 370], it should be obvious that if there is a finite set of possible terms in the context of the problem and there is no solution then a planner should stop and report failure. Unfortunately, the outcome lemma contains a finite problem which causes Tweak to loop. This problem consists of two operators that clobber themselves mutually; if one operator is in the plan, the second cannot appear in the plan, otherwise the plan is incorrect. Consequently, there is no solution (i.e. no correct plan) which can contain these two operators together. The problem that makes Tweak loop is an unsolvable problem where the goal state needs the two operators to be established:

$$(\{\neg g,\neg h\},\ \{g,h\},\ \{(\{\neg h\},\{\neg h,g\}),(\{\neg g\},\{h,\neg g\})\})$$

is the triple representing this problem where g and h are propositional symbols.

The question is: "Why does Tweak loop on a finite problem with no solution?". Tweak is built upon the MTC. If Tweak loops then it is because the MTC keeps being possibly true; should it become necessarily false at some moment, the construction of a plan would stop. Then, the previous question becomes: "What might be kept possible (but should not) in the MTC?".

Lemma 2.1. Declobbering using a white knight is always possible.

Proof. A new instance of the clobbered establisher can always be used as a new white knight; since the white knight operator is newly inserted, it is possible to add the precedence constraints (i.e. $W<A$ or $C<W$) to the partial order and the implication constraint is immediately checked since r *is* p. ■

Henceforth the looping is visible: adding a white knight to a plan is adding an operator whose preconditions must be achieved. But if this white knight is itself a clobberer, it must be processed as such: the GAP may well insert another clobberer to declobber this white knight, and so forth. This is what happens in the looping problem of Tweak [1, page 371]: "Tweak loops on this problem [because it is] building plans that are longer and longer chains of steps that alternately assert" the alternately clobbered propositions.

3. Reducing Tweak: dropping white knights

3.1. A reduced modal truth criterion

Since adding a new operator as a white knight to a plan is a source for looping, the very first idea is to constrain the white knight strategy. A simple restriction is to disallow the insertion of new operators as white knights in the plan (or, in other terms, to search white knights among those operators which already belong to the plan, i.e. which are establishers of some propositions in the plan). Another—more extreme—restriction is to choose the establisher itself (and not a new instance of it) as a white knight. In choosing t as a white knight for itself, we here choose to substitute plan step W by situation t, meaning that situation t is constructed from the effects of W; the MTC becomes:

$$\begin{array}{l}
\exists t,\ \aleph\ (t \le s) \wedge \aleph\ \text{asserted-in}(p,t) \wedge \\
\quad \forall C,\ \aleph\ (s \le C) \vee \\
\quad\quad \forall q\ \aleph\sim\text{denies}(C,q) \vee \\
\quad\quad\quad \aleph\ (p \ne q) \vee \\
\quad\quad\quad\quad \aleph\ (C<t) \wedge \\
\quad\quad\quad\quad \aleph\ (t \le s) \wedge \\
\quad\quad\quad\quad \exists r\ \text{asserts}(t,r) \wedge (p=q \Rightarrow p=r)
\end{array}$$

where (W<s) has been transformed in (t≤s) with the previous explanations. It is always possible to find an r that is asserted in t and such that p=q⇒p=r: one just has to choose r=p. Then, the existential clause on r is always true. Then, noting the repetition of ℵ (t≤s), and that ℵ (C<t) does not depend on q, the MTC becomes:

$$\exists t,\ \aleph\ (t \le s) \wedge \aleph\ \text{asserted-in}(p,t) \wedge$$
$$\forall C,\ \aleph\ (s \le C) \vee \aleph\ (C < t) \vee$$
$$\forall q\ \aleph \sim\text{denies}(C,q) \vee \aleph\ (p \ne q)$$

We here see that demotion (C<t) is a symmetric technique to promotion (s≤C). This last version of the MTC does not contain the white knight strategy but still mixes situations and plan steps; but it now does not loop on Tweak's looping problem: the only possible hope for declobbering in this problem relies on inserting a new operator. As no operators are now inserted within declobbering and neither the two possible orderings (i.e. s≤C or C<t), nor separation are applicable, the construction of the plan is stopped and failure is reported. The last step in this reduction can be to avoid mixing situations and operators. This can be done in remarking that situation t is constructed from the effects of an operator, say T, and situation s from the preconditions of another operator, say S; henceforth, the asserted-in(p,t) predicate just means that p is an effect of T. Then, t≤s is equivalent to T<S, s≤C to S≤C, and C<t to C≤T; one finally obtain:

$$\exists T,\ \aleph\ (T < S) \wedge \aleph\ p \in pos(T) \wedge \qquad \text{(establishment)}$$
$$\forall C,\ \ \aleph\ (S \le C) \vee \qquad \text{(promotion)}$$
$$\aleph\ (C \le T) \vee \qquad \text{(demotion)}$$
$$\forall q\ \aleph\ \neg q \notin pos(C) \vee \aleph\ (p \ne q) \qquad \text{(separation)}$$

Note that a similar reduction can be found in [4], but with no analysis of the consequences of the reduction and no comment on the resulting non-looping behaviour.

Let us call R-MTC this final reduced version of the MTC for the rest of this paper. R-MTC defines four plan modification operators: establishment, promotion, demotion and separation. We will refer to these operators as planning operators. The english reading of R-MTC is:

> *A proposition p is necessarily true in a situation s iff two conditions hold: there is a situation t equal or necessarily previous to s in which p is necessarily asserted; and for every step C and every proposition codesignating with p that C denies, step C is necessarily not between t and s. The criterion for possible truth is exactly analogous, with all the necessary modalities switched (read "possible" for "necessary").*

Note that without its modalities, R-MTC is equivalent to Chapman's truth criterion for complete (totally ordered) plans [1, page 366].

3.2. Completeness and correctness of the reduced modal truth criterion

Since R-MTC is a reduced truth criterion, the space of constructible non linear plans is not the same as the space generated with MTC. This comes from the elimination of the implication constraint in the MTC: it rejects the construction of some non linear plans. Let us consider the non linear partial plan of Figure 1 where nodes are operators, T1 asserts p and r, T2 asserts p and q, and C1 and C2 clobber p. We want preconditions p, q and r of operator G to be asserted.

According to MTC, this plan is correct: we need an establisher for p; let us take T1. There are two clobberers for T1: C1 and C2. C1 is such that C1<T1<G. So The white knight is in the plan and is the establisher T1 itself. There are no precedence constraints between C2 and T1, but C2 is such that there exists another white knight, T2, which prevents p from being denied. MTC then accepts the plan as correct. One can note that the plan of Figure 1 is symmetric in indices, and taking T2 as an establisher for p would not have change the correctness of the plan. There are no clobberers for q and r: T1 correctly establishes r and so does T2 for q.

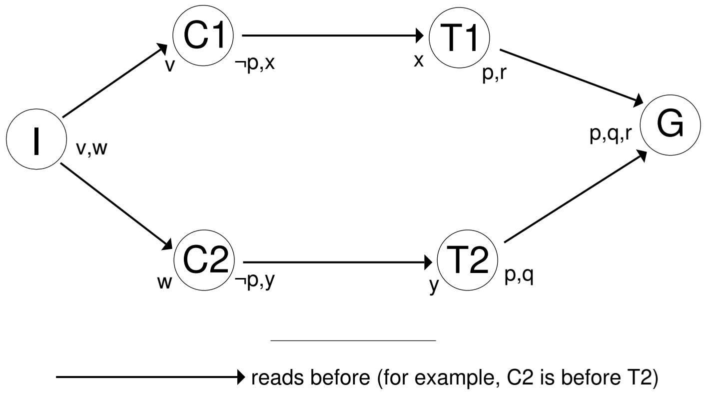

Figure 1: A partially ordered plan which is correct for MTC and incorrect for R-MTC.

Tweak (with the MTC) may generate the plan of Figure 1. On the other hand, the plan does not pass the R-MTC test. If T1 is taken as the establisher for p, C1 is effectively declobbered by T1, but C2<T1 must be added to the plan so as to be accepted by R-MTC. If T2 is taken as an establisher for p, then C1<T2 must be added to the plan.

From this example, we can conclude: (i) that some correct partially ordered plans cannot be reached using R-MTC while MTC allows their construction and (ii) that R-MTC cannot be used as a criterion for checking the correctness of any partially ordered plan. However, we have the following theorems.

Theorem 3.1. Any partially ordered plan which passes the R-MTC test passes the MTC test.

Proof. Immediate (R-MTC is a particularization of MTC). ■

Corollary 3.2 (Correctness of R-MTC as the basis of a planning procedure). Any planner using R-MTC builds correct plans.

Proof. Let (I,G,Op) be a planning problem and $(P,<_P)$ a plan built by R-MTC. If the planner accepts $(P,<_P)$ then $(P,<_P)$ passes the R-MTC test. From theorem 3.1 $(P,<_P)$ passes the MTC test. Hence $(P,<_P)$ is a correct plan since MTC is a correct truth criterion. ■

Definition 3.3. A correct partially ordered plan $(Q,<_Q)$ is *minimal* if and only if for every correct partially ordered plan $(P,<_P)$, $P \subseteq Q$ implies $P = Q$.

Theorem 3.4. $\mathcal{S}_{\mathcal{R}\text{-}\mathcal{T}weak}$, the set of all possible linearizations derived from minimal non linear plans built with R-MTC as solutions to a planning problem, is equal to $\mathcal{S}_{\mathcal{T}weak}$, the set of all possible linearizations derived from minimal non linear plans built with MTC as solutions to the same planning problem.

Proof. The proof is in two steps: we show (1) that $\mathcal{S}_{\mathcal{R}\text{-}Tweak}$ is a subset of $\mathcal{S}_{Tweak}$ and (2) that $\mathcal{S}_{Tweak}$ is a subset of $\mathcal{S}_{\mathcal{R}\text{-}Tweak}$; the consequence is the equality of the two sets.

Step (1), $\mathcal{S}_{\mathcal{R}\text{-}Tweak}$ is a subset of $\mathcal{S}_{Tweak}$: Immediate (R-MTC is a particularization of MTC).

Step (2), $\mathcal{S}_{Tweak}$ is a subset of $\mathcal{S}_{\mathcal{R}\text{-}Tweak}$: Let us consider a linearization L of a minimal partially ordered plan built with MTC as a solution to a planning problem. For every operator A in the plan and every precondition p of A, there exists an operator T(p,A) in the plan such that (i) T asserts p and (ii) no operator C such that T<C<A in the linearization denies p. In addition, the minimality of the plan implies that every operator O in the plan is either the initial operator, the final operator or T(p,A) for some A and some p of A (otherwise O can be removed and the plan is not minimal). Hence the set of operators composing the plan can be generated as a set of establishers using R-MTC. The generation of these establishers, followed by the assertion of ordering constraints as long as such constraints are necessary, results into a partially ordered plan which accepts L as a linearization. Hence $\mathcal{S}_{Tweak}$ is a subset of $\mathcal{S}_{\mathcal{R}\text{-}Tweak}$.

Conclusion: $\mathcal{S}_{Tweak} = \mathcal{S}_{\mathcal{R}\text{-}Tweak}$. ■

Corollary 3.5 (Completeness of R-MTC as the basis of a planning procedure). Let (I,G,Op) be a planning problem. If there exists a correct plan $(Q,<_Q)$ for the (I,G,Op) problem, then there exists a sequence of application of the planning operators of R-MTC (i.e. establishment, promotion, demotion and separation) that achieves a correct partially ordered plan $(P,<_P)$ with $P \subseteq Q$.

Proof. Given $(Q,<_Q)$, let us consider a minimal plan $(R,<_R)$ with $R \subseteq Q$ (the existence of such a minimal plan follows from definition 3.3). Since Tweak is complete, $\mathcal{S}_{Tweak}$ necessarily contains all the possible linearizations of $(R,<_R)$. Since $\mathcal{S}_{Tweak} = \mathcal{S}_{\mathcal{R}\text{-}Tweak}$, $\mathcal{S}_{\mathcal{R}\text{-}Tweak}$ contains all the possible linearizations of $(R,<_R)$. As $(R,<_R)$ accepts at least one linearization, this implies the existence of a sequence of application of the planning operators of R-MTC to achieve a correct partially ordered plan $(P,<_P)$ with P=R. ■

These theorems bring up some remarks. First, it is confirmed here that plans should not be built according to any truth criterion, but according to the fact that they are solutions to a planning problem: each of their possible executions, a correct linear plan, shall transform the initial state into the final state. The theorems clearly show that there may be several distinct partial orders for the same correct execution but to reach such correct executions, the white knight is not needed: establishment, demotion, promotion and separation suffice. Second, $\mathcal{S}_{Tweak} = \mathcal{S}_{\mathcal{R}\text{-}Tweak}$, but R-MTC may not be able to build a plan with as few ordering constraints as necessary. There are some plans that MTC can build but that R-MTC cannot (the plan of Figure 1 is one of them): the space of partial plans constructible by R-MTC is simpler that the one of MTC, which can practically entail more efficiency in the construction of solutions [5].

4. Conclusions

Consistently with Chapman's remarks, we have proved that white knights are not necessary in the process of constructing plans to reach goal situations from initial situations. We have also shown that the non-consideration of white knights does not prevent from finding a plan with a minimal number of operator instances. This has been shown by reducing the MTC into a new one and by proving that the same minimal linearizations were constructible by both MTCs. Henceforth, the white knight strategy is not necessary for completeness in Tweak.

It must be clear that a R-Tweak planner does more commitment than Tweak; some partial orders and thus some non linear plans are not directly reachable by R-Tweak. However, given a correct partially ordered plan $(P,<_P)$ (for example generated by R-Tweak), it is easy to determine whether an ordering constraint in $<_P$ is removable or not: for example, one just has to remove the constraint from $<_P$ and use Chapman's MTC as a verification procedure; if the new plan passes the test, then the constraint *is* removable. Since Chapman's MTC, interpreted as a verification procedure, runs in polynomial time, we can, starting from $(P,<_P)$, derive in polynomial time a plan $(P,<<_P)$ with $<<_P$ a minimal set of ordering constraints. In many cases, this polynomial time complexity will be negligible compared to the search time for plan generation [5, 10]. Moreover, efficient algorithms are indeed available to remove unnecessary ordering constraints from totally or partially ordered plans [3, 8, 9, 10].

One should note that although Chapman's implementation of Tweak "never uses [the white knight]" [1, page 360], it was a correct and complete procedure.

Minton, Bresina and Drummond [5] have suggested that a Tweak-like planner can be much less efficient that a R-Tweak-like planner on certain problems; the main idea being that, on certain problems, a Tweak-like planner can generate more partial plans sharing common linearizations that in the case of a R-Tweak-like planner. This practical viewpoint is formally corroborated in this paper. Planning techniques such as hierarchical planning or meta-planning, taking advantage of the [white knight]less formal framework presented here can be more efficient—and still correct and complete.

ACKNOWLEDGMENT

We thank Michel De Glas, Jacques Ferber and his team at LAFORIA, Paris 6, Les Gasser, Joachim Hertzberg, Subbarao Kambhampati, Philippe Morignot, Pierre Regnier and Sam Steel for various form of support in drafting this paper. The authors are listed in alphabetical order.

REFERENCES

1. Chapman, D., Planning for conjunctive goals, *Artificial Intelligence* **32** (1987) 333-377.

2. Jacopin, E. and Puget, J-F., From Tweak to Pweak: Reducing the Non Linear Planning Framework, LAFORIA Tech. Rept. 90/29, University of Paris 6, Paris, France (December 1990).

3. Kambhampati, S. and Kedar, S., A Unified Framework for Explanation-Based Generalization of Partially Ordered and Partially Instantiated Plans, ASU CSE Tech. Rept. 92-008, Arizona State University, Tempe, Arizona (April 1992).

4. McCluskey, T.L., Experience-driven heuristic acquisition in general problem solvers, Ph.D. thesis, Computer Science Department, The City University, London, England (December 1988).

5. Minton, S., Bresina, J. and Drummond, M., Commitment strategies in planning: a comparative analysis, in: *Proceedings of 11th IJCAI*, Sydney, Australia (1991) 259-265.

6. Morignot, P., Critères de vérité en planification, Ph.D. thesis, École Nationale Supérieure des Télécommunications, Paris, France (May 1991). (In French)

7. Puget, J-F., Apprentissage par explication d'échecs: théorie et applications, Ph.D. thesis, University of Paris 11, Orsay, France (January 1990). (In French)

8. Regnier, P., Recherche et exploitation du parallèlisme en planification, Ph.D. thesis, Paul Sabatier University, Toulouse, France (May 1992). (In French)

9. Regnier, P. and Fade, B., Complete Determination of Parallel Actions and Temporal Optimization in Linear Plans of Actions, in: *Proceedings of the European Workshop on Planning*, J. Hertzberg (Ed.), Lecture Notes in Artificial Intelligence 522, Springer Verlag (1991) 100-111.

10. Veloso, M., Pérez, A. and Carbonell J., Non-Linear Planning with Parallell Resource Allocation, in: *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, K. Sycara (Ed.), San Diego, CA (1990), 207-212.

11. Yang, Q. and Tenenberg, J., AbTweak: Abstracting a Non-linear Least Commitment Planner, in: *Proceedings of the 8th AAAI*, Boston, MA (1990), 204-209.