Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, August 2001

SiN: Integrating Case-based Reasoning with Task Decomposition

Héctor Muñoz-Avila,[1] David W. Aha,[2] Dana S. Nau,[1] Rosina Weber,[2]
Len Breslow[2] & Fusun Yaman[1]
[1]Department of Computer Science, University of Maryland, College Park, MD 20742-3255
[2]Navy Center for Applied Research in Artificial Intelligence,
Naval Research Laboratory (Code 5515), Washington, DC 20375
[1]lastname@cs.umd.edu [2]lastname@aic.nrl.navy.mil

Abstract

This paper describes SiN, a novel case-based planning algorithm that combines conversational case retrieval with generative planning. SiN is provably correct, and can generate plans given an incomplete domain theory by using cases to extend that domain theory. SiN can also reason with imperfect world-state information by incorporing preferences into the cases. Our empirical validation shows how these preferences affect plan quality.

1 Introduction

Generative planners traditionally require a complete domain theory, which provides a clear semantics for the planner's inferencing mechanism. This allows a planner to be used in different domains. However, in many planning domains, developing a complete domain theory is infeasible.

In this paper we present a case-based planning algorithm called *SiN* (SHOP integrated with NaCoDAE), which integrates the *SHOP* generative planner [Nau *et al.*, 1999] with *NaCoDAE, a* conversational case retriever [Breslow & Aha, 1997]. SiN is a provably correct algorithm that does not require a complete domain theory nor complete information about initial or intermediate world-states.

In addition to describing SiN, which has been implemented in HICAP [Muñoz-Avila *et al.*, 1999], we present sufficient conditions to ensure its correctness, show how SiN represents preferences in cases to generate plans in the context of imperfect world state information, and describe an empirical analysis that demonstrates the impact of the preferences on plan quality.

In the following sections, we introduce some terminology, detail SiN, including theoretical results on its semantics with respect to incomplete domain theory, present SiN's empirical evaluation to show the role of preferences to handle incomplete world state information, and discuss the implications of these results.

2 Motivation

SiN's design was partly motivated by the following characteristics of military planning operations.

- Military operations are strongly hierarchical [Mitchell 1997; Muñoz-Avila *et al.*, 1999]. Thus, we chose to represent plans using Hierarchical Task Networks (HTNs) [Erol *et al.*, 1994].
- There is an incomplete domain theory, in the form of general guidelines (doctrine) and standard operating procedures (SOPs). However, neither doctrine nor SOPs can be used to derive detailed tactical plans, which often require knowledge about previous experiences. Thus, SiN uses SHOP to perform first-principles reasoning and NaCoDAE to employ previous experiences.
- Military planners do not have complete information about the current situation; part of the planning includes dynamic information gathering, typically to assess enemy capabilities and/or deployment. In SiN, NaCODAE is used to plan with an incomplete world state using *preferences*, which we define in Section 4.

3 Notation and definitions

An ***HTN*** is a set of tasks and their ordering relations, denoted as $N=(\{t_1,\ldots,t_m\},<)$ ($m\geq0$), where $<$ is a binary relation expressing temporal constraints between tasks. Decomposable tasks are called ***compound***, while non-decomposable tasks are called ***primitive***.

A ***domain theory*** consists of methods and operators for generating plans. A ***method*** is an expression of the form $M=(h,P,ST)$, where h (the method's ***head***) is a compound task, P is a set of ***preconditions***, and ST is the set of M's (children) ***subtasks***. M is ***applicable*** to a task t, relative to a ***state*** S (a set of ground atoms), iff matches(h,t,S) (i.e., h and t have the same predicate and arity, and a consistent set of bindings _ exists that maps variables to values such that all terms in h match their corresponding ground terms in t) and the preconditions P are ***satisfied*** in S (i.e., there exists a consistent extension of _, named _', such that $\forall\ p \in P\ \{p\_' \in S\}$), in which case $M(t,S)=ST\ \_'$.

An ***operator*** is an expression of the form $O=(h,aL,dL)$, where h (the operator's ***head***) is a primitive task, and aL and dL are the so-called ***add-*** and ***delete-lists***. These lists define how the operator's application transforms the current state S: every element in the add-list is added to S and every element in the delete-list is removed from S. An operator O is ***applicable*** to a task t, relative to a state S, iff matches(h,t,S).

A ***planning problem*** is a triple (T,S,D), where T is a set of tasks, S is a state, and D is a domain theory. A ***plan*** is the collection of primitive tasks obtained by decomposing all compound tasks in a planning problem (T,S,D).

4 Cases in SiN

In many domains it is impossible to assume that a complete domain theory of the world is known. For example, this is true when planning for non-combatant evacuations (NEOs). However, a partial domain theory exists for NEOs, and it can be elicited from doctrine and standard operating procedures [DOD, 1997].

Reasoning about parts of the domain for which no domain theory is available is done through cases. A ***case*** C is an instance of a method, denoted by $C=(h,P,ST,Q)$, where h, P, and ST are defined as for methods and Q is a set of <question,answer> pairs. Q defines ***preferences*** for matching a case to the current state. Preferences are useful for ranking cases in the context of incomplete world states and/or domain theories because, as we will show, they focus users on providing relevant additional state information.

5 SiN mixed-initiative planner

SiN integrates SHOP and NaCoDAE's task decomposition algorithms. A single (current) state S is maintained in SiN that is accessible to and updateable by both SHOP and NaCoDAE. Answers given by the user during an interaction with NaCoDAE are added to S (i.e., each question has a translation into a ground atom). Changes to the state that occur by applying SHOP's operators are also reflected in S.

SHOP generative planner. At any point during the planning process, SHOP is refining a task list T' relative to a state S and a domain theory D. Initially, T' is the set of tasks T in the planning problem (T,S,D). SHOP performs *ordered task decomposition* [Nau *et al.*, 2000], meaning that the tasks must be totally ordered (i.e., the < relation on HTNs is a total order). SHOP also maintains the partial solution plan p being derived (i.e., the primitive tasks in T'). Initially p is empty. SHOP selects the first task t in T' and continues as follows:

- If t is primitive and has an applicable operator O, then O is applied to t, S is updated accordingly, t is removed from T' and added to the end of p.
- Else if t is compound and has an applicable method M (that has not yet been applied to t), then M is applied, which replaces t in T' with M's subtasks.
- Else if T' is not empty, then SHOP backtracks.
- Else SHOP fails.

SHOP terminates when T' is empty, in which case p is the solution, or when SHOP tries to backtrack on a compound task t whose applicable methods have been exhausted.

NaCoDAE mixed-initiative case retriever. Users interact with NaCoDAE in *conversations*, which begin when the user selects a task t. NaCoDAE responds by displaying the top-ranked cases whose pre-conditions are satisfied and whose heads match t. Cases are ranked according to their similarity to the current state S, which is the state that exists at that time during the conversation. Similarity is computed for each case C by comparing the contents of S with Q, C's <q,a> preference pairs. (That is, each pair is represented as a monadic atom in S, and similarity for a given <q,a> preference pair becomes a membership test in S). NaCoDAE also displays questions, whose answers are not known in S, ranked according to their frequency among the top-ranked cases. The user can select and answer (with a) any displayed question q, which inserts $<q,a>$ into S. This state change subsequently modifies the case and question rankings. A conversation ends when the user selects a case C, at which time the task t is decomposed into ST (i.e., C's subtasks).

SiN integrated planning algorithm. SiN receives as input a set of tasks T, a state, S, and a knowledge base $I \cup B$ consisting of an incomplete domain theory I and a collection of cases B. The output is a solution plan p consisting of a sequence of operators in I. Both SHOP and NaCoDAE assist SiN with refining T into a plan. As does SHOP, SiN maintains the set of tasks in T' that have not been decomposed and the partial solution plan p. At any point of time, either SHOP or NaCoDAE is in control and is focusing on a compound task $t \in T'$ to decompose. SiN proceeds as follows:

- **Rule # 1**: If SHOP is in control and can decompose t, it does so and retains control. If SHOP cannot decompose t, but NaCoDAE has cases for decomposing t, then SHOP will cede control to NaCoDAE.
- **Rule # 2**: If NaCoDAE is in control, it has cases for decomposing t whose pre-conditions are satisfied. If the user applies one of them to decompose t, then NaCoDAE retains control. If NaCoDAE has no cases to decompose t or if the user decides not to apply any applicable case, then if t is SHOP-decomposable, NaCoDAE will cede control to SHOP.

If neither of these rules applies, then SiN backtracks, if possible. If backtracking is impossible (e.g., because t is a task in T), this planning process is interrupted and a failure is returned.

By continuing in this way, and assuming that the process is not interrupted with a failure, SiN will eventually yield a plan *p* (i.e., consisting only of primitive tasks).

6 Correctness of SiN

In this section we will assume that SiN performs ordered task decomposition. That is, we assume that all tasks are totally ordered and at each iteration, when refining a set of tasks *T'*, SiN will start by decomposing the first task in *T'*. A relaxation of this condition should be possible once we modify SiN to include the extended version of SHOP that can represent partial-order task relations [Nau *et al*, 2001].

If *I* is an incomplete domain theory and *B* is a case base (i.e., a set of cases), then a domain theory *D* is ***consistent*** with $I \cup B$ iff (1) every method and operator in *I* is an instance of a method or operator in *D* and (2) for every case *C=(h,P,ST,Q)* in *B*, there is a method M=(*h',P',ST'*) in *D* such that *h*, *P*, and *ST* are instances of *h', P'* and *ST'* respectively. Although many different domain theories might be consistent with $I \cup B$, in general we will not know which of these is the one that produced *I* and *B*. However, we can prove that SiN is correct in the sense that, if it succeeds in outputting a plan, then that plan could have been generated by SHOP using any domain theory consistent with $I \cup B$.

Proposition (Correctness of SiN). Let *T* be a collection of tasks, *S* be an initial state, *I* be an incomplete domain theory, and *B* be a case base, and let SiN(*T,S,I,B*) represent the invocation of SiN with those items as inputs. Suppose that SiN performs ordered task decomposition. Then:

(1) If SiN(*T,S,I,B*) returns a plan *p*, then for every domain theory *D* consistent with $I \cup B$, *p* is a solution plan for the planning problem (*T,S,D*).

(2) If SiN(*T,S,I,B*) cannot find a plan, then there is a domain theory *D* consistent with $I \cup B$ such that no solution plan exists for (*T,S,D*).

The proof is done by induction on the number of iterations of the SiN algorithm. The proof shows that each SiN task decomposition in ($T,S,I \cup B$) corresponds to a SHOP task decomposition in (*T,S,D*). This is sufficient to prove correctness because SHOP is known to be correct [Nau *et al.*, 1999]. We omit the details of the proof due to space limitations.

This proposition suggests that cases in SiN supply two kinds of knowledge: first, they provide control knowledge, similar to the knowledge encoded in cases using derivational replay when a complete domain theory is available [Veloso, 1994; Ihrig & Kambhampati, 1994]. Because cases are instances of methods, applying a case is comparable to a replay step in which the method selected to decompose a task is the one in the case's derivational trace. The main difference is that, while cases in replay systems correspond to a complete derivational trace, cases in SiN correspond to a single step in the derivational trace. Second, cases in SiN augment the domain theory and, thus, provide domain knowledge as do cases in many case-based planners (e.g., [Hammond, 1986]).

7 Imperfect World Information

SiN uses NaCoDAE to dynamically elicit the world state, which involves obtaining the user's preferences. Depending on the user's answers, cases will get re-ranked. When solving a task, the user can choose any of the cases, independent of their ranking, provided that all their preconditions are met. The preferences play a pivotal role in determining plan quality due to the absence of a complete domain theory.

Consider the following two simplified cases:

Case 1:

Head: selectTransport(ISB,NEOsite)
Preconditions: HelosAvailable(ISB)
Questions-Answer pairs: Weather conditions? Fine
Subtasks: Transport(ISB,NEOsite,HELOS)

Case 2:

Head: selectTransport(ISB,NEOsite)
Preconditions: groundTransportAvailable(ISB)
Questions-Answer pairs:

- Weather conditions? Rainy
- Imminent danger to evacuees? No

Subtasks: Transport(ISB,NEOsite,GroundTransport)

These cases both concern the selection of transportation means between an intermediate staging base (ISB) and the NEO site (NEOsite). The first case suggests using helicopters provided that they are available at the ISB. The second one suggests using ground transportation provided that the corresponding transportation means are available at the ISB. If the two cases are applicable, because both preconditions are met, the answers given by the user will determine a preference between them. For example if the weather is rainy and there is no immediate danger for the evacuees, NaCoDAE would suggest the second case. The rationale behind this is that flying in rainy conditions is risky. Thus, selecting ground transportation would be a better choice.

8 Evaluation

Our experiments focused on the role of preferences in the plan generation process in the context of incomplete world state information and how they affect the quality of the resulting plans. Towards this goal, we developed two planning domains where the contents of the world state can significantly impact choices during planning.

For these experiments, we encoded an automatic user that dynamically provided preferences when asked by NaCoDAE. The user provided preferences with a pre-defined bias (defined in the following sections) towards certain kinds of solutions. The automatic user will always select the case with the highest similarity. In situations where several candidate cases had the same highest similarity, the automatic user selected one of them randomly. The purpose was to observe whether the resulting plans reflect the user's bias despite the incomplete information about the world state.

8.1 The Personal Travel Domain

The first domain was the *personal travel domain*. Its plans concern traveling from locations in Washington, DC to downtown New York City (NYC). We encoded 7 transportation methods (3 inter- and 4 intra-city). Plans consist of 3-5 planning segments. States indicate different locations, whether connections between locations exists and by which means, weather conditions, etc. The knowledge base consists of 10 methods and 1 operator for SHOP and 40 cases for NaCoDAE.

Our personal travel plan evaluator can generate a different time duration each time it is given a plan and world state because of its non-deterministic execution. For each run, it outputs whether the plan succeeded and, if so, the trip's duration. A plan fails when segment delays cause a late arrival for a segment requiring a fixed time departure (e.g., an airplane flight). For each segment, we applied a delay function that is influenced by world state conditions. For example, a flight segment will incur a longer delay for higher chances of large snow accumulation, especially on holidays (i.e., high travel days). Segments are categorized into short, medium, and long lengths, and delays can range from 0 up to 4.5 times a segment's anticipated duration. Smaller multiples are used for maximum delays for medium (3.5) and long (2.5) duration segments.

We selected ten goals, corresponding to ten pairs of departing and arrival locations in Washington, DC and downtown NYC, respectively. For each goal, we generated 10 random world states, thus yielding 100 total planning problems. SiN was then used to generate a plan for each problem, thus yielding 300 plans. Each was executed 10 times by the plan evaluator, for a total of 3000 runs.

Table 1: Results for the personal travel domain.

Preference	*Duration*	*Price*	*Success*
Bus	676	85	92.2%
Train	466	176	94.3%
Plane	375	338	77.0%

Table 1 summarizes the results. When the user's bias was given towards taking the Bus for the intercity part of the trip. This preference yields maximal (676 minutes) durations, but has the cheapest price. On the other extreme, when the Plane is the preferred means of transportation, the duration is the shortest but the price is the most expensive. The lowest success rate occurs for plane trips; it reflects missing connections due to external factors such as the weather. Although a bias expresses a preference for a certain transportation mode, it does not imply that it was always selected; world state conditions may prevent the use of some travel modes for particular situations.

8.2 The NEO Planning Domain

The second domain was the *Noncombatant Evacuation Operations Domain*. Its plans involve performing a rescue mission where troops are grouped and transported between an initial location (the assembly point) and the NEO site (where the evacuees are located). After the troops arrived at the NEO site, evacuees are re-located to a safe haven. Planning involves selecting possible pre-defined routes, consisting of 4 segments each. The planner must also choose a transportation mode for each segment. In addition, other conditions were determined during planning such as whether communication exists with State Department personnel and the type of evacuee registration process. SiN's knowledge base included 6 operators, 22 methods, and 51 cases.

As with the personal travel domain evaluator, the NEO planning evaluator can generate a different output each time it is given a plan and a world state because of its non-deterministic execution. For each run, it outputs the plan execution duration, the time it took to reach the evacuees, and the evacuee casualties. Similar to the personal travel domain, we applied a delay function that is influenced by world state conditions. The Neo Planning evaluator is more complex than the personal travel evaluator because there are more conditions that can affect the output variables and these conditions may interact. For example, a small-sized force will incur fewer delays because embarking troops in the transportation means will take less time than for large-sized forces. However, smaller force increase the chances of hostile attacks, which if they occur will delay the operation.

We had a single task, to perform a NEO, and generated 100 random world states, thus yielding 100 planning problems. SiN was then used to generate a plan for each problem, thus yielding 200 plans, and each was executed 10 times by the plan evaluator for a total of 2000 runs.

Table 2: Results for the NEO planning domain.

Preference	*Duration*	*Time to Reach Evacuees*	*Evacuee Casualties*
Helicopter	38.5	28.7	11%
Ground Vehicle	48.1	34.8	16%

Table 2 summarizes the results. The helicopter transport preference yields plans that have shorter execution durations (38.5 hours) and require less time to reach the evacuees (28.7 hours). In addition, average casualties among evacuees is less (11%), due mainly to the shorter time to reach them, and land travel is generally riskier than air travel. Still the number of casualties among evacuees is high even with helicopters. This is due to the simple bias encoded in the simulated user. Human users could yield better (and worse) plans by dynamically providing more sophisticated preferences depending on the world state conditions.

8.3 Discussion

The experiments show the capabilities of SiN in allowing the user to guide the planning process towards their preferences while dynamically capturing world-state conditions. Despite our use of simplistic simulated users, the quality of the plans reflect the user's bias.

9 Related Work

Table 3 compares seven different features of SiN to those of other planning systems.

Table 3: Comparisons between different systems. Conventions: Gen=Generative; CBP=Case-Based; M-I=Mixed-initiative; I=Interleaved control structure; DK=Cases are used to supply domain knowledge; CK=Cases are used to supply control knowledge.

System	Gen	CBP	M-I	I	DK	CK
SiN	√	√	√	√	√	√
CHEF		√			√	
MI-CBP	√	√	√			√
NaCoDAE		√	√		√	
Prodigy/ Analogy	√	√				√
SHOP	√					
SIPE II	√		√			

We first discuss the features shown in columns 2–5 of Table 3. SHOP [Nau *et al.*, 1999], as is typical of generative planners, requires a complete domain theory. CHEF [Hammond, 1989] and DIAL [Leake *et al*., 1997] are case-based, but do not have a generative component, and thus need a large case base to perform well across a wide variety of problems. Prodigy/Analogy [Veloso, 1994], DerSNLP [Ihrig & Kambhampati, 1994], and Paris [Bergmann & Wilke, 1995] integrate generative and case-based planning, but require a complete domain theory and are not mixed-initiative. SIPE II [Wilkins, 1998] is a mixed-initiative generative planner, but does not use cases. NaCoDAE [Muñoz-Avila *et al*., 1999] is a mixed-initiative case-based planner, but does not employ generative planning.

At least three other integrated (case-based/generative), mixed-initiative planners exist. MI-CBP [Veloso *et al.*, 1997], which extends Prodigy/Analogy, limits interaction to providing it with user feedback on completed plans. Thus, it must input, or learn thru feedback, a sufficiently complete domain theory to solve problems. In contrast, SiN gathers information it requires from the user through NaCoDAE conversations, but does not learn from user feedback. CAPlan/CbC [Muñoz-Avila *et al.*, 1997] and Mitchell's [1997] system use interaction for plan adaptation rather than to acquire state information.

Among integrated case-based/generative planners, SiN's interleaved control structure is unique in that it allows both subsystems to equally control the task decomposition process. In contrast, other approaches either use heuristics (Prodigy/Analogy; MI-CBP) or order case-based prior to generative planning [DerSNLP; Mitchell, 1997], although Paris does this iteratively through multiple abstraction levels. Distinguishing the relative advantages of these control strategies is an open research issue.

The final two columns of Table 3 refer to the types of contributions made by cases in CBP systems. CHEF and NaCoDAE both use cases to provide domain knowledge, while Prodigy/Analogy uses cases for control knowledge (i.e., determining which planning constructs to apply). In contrast, SiN uses cases to both provide domain knowledge (i.e., instances of methods) and control knowledge (i.e., it allows the user selects which of these instance methods to apply).

CaseAdvisor [Carrick *et al*., 1999], like SiN, integrates conversational case retrieval with planning. While CaseAdvisor applies pre-stored hierarchical plans to gather information to solve diagnosis tasks, SiN instead uses its case retriever to gather information and applies cases to refine hierarchical plans.

Planning with incomplete information has been the subject of frequent research in planning (e.g., [Golden *et al*., 1996]). Typically, a distinction between sensing and planning actions is made, where the former involve queries to external information sources and the latter involves inferencing steps. This is comparable to querying using NaCoDAE and task refinement using SHOP.

10 Final Remarks

We presented the SiN algorithm for case-based HTN planning. SiN was motivated by three requirements for planning military operations: plans are hierarchical, there is no complete domain theory explaining all possible courses of action, and planners do not have complete information about the current situation.

Our work includes the following contributions:

- SIN is a provably correct algorithm for case-based planning with incomplete domain theories.
- SiN can tolerate incomplete world state information by representing preferences in the cases. Our experimental results show that a user can dynamically guide SiN by giving preferences to it as part of the user's normal interaction with SiN during the planning process.
- SiN provides a bridge between two classical approaches for case-based planning, in which cases either provide

control knowledge or domain knowledge. In SiN, cases provide both kinds of knowledge.

- SiN's ability to combine both experiential and generative knowledge sources can be beneficial in real-world domains where some processes are well known and others are obscure but recorded memories exists on how they were performed. Planning for NEOs is a typical example of this type of domain. We have integrated SiN into HICAP [Muñoz-Avila *et al.*, 1999], a system designed to support these kinds of operations.

Our creation of SiN was made possible because of the similarity between NaCoDAE's cases and SHOP's methods. For our future work, we want to further exploit this similarity to ease knowledge acquisition for plan generation. To this end, we have started working to create algorithms for learning HTN methods automatically from cases.

Acknowledgements

The authors would like to thank the reviewers for their helpful comments. This research was supported by funding from DARPA, ONR, AFRL (F306029910013 and F30602-00-2-0505), ARL (DAAL0197K0135), and the University of Maryland General Research Board. Opinions on SiN expressed in this paper are those of the authors and do not necessarily reflect the opinions of the funding agencies.

References

[Bergmann & Wilke, 1995] Bergmann R., & Wilke W. Building and refining abstract planning cases by change of representation language. *Journal of Artificial Intelligence Research*, 3, 53-118, 1995.

[Breslow & Aha, 1997] Breslow, L. A. & Aha, D. W. NaCoDAE: Navy Conversational Decision Aids Environment (Technical Report AIC-97-018). NCARAI, Washington, DC, 1997.

[Carrick *et al.*, 1999] Carrick, C., Yang, Q., Abi-Zeid, I., & Lamontagne, L. Activating CBR systems through autonomous information gathering. *Proceedings of the Third International Conference on Case-Based Reasoning* (pp. 74-88). Seeon, Germany: Springer, 1999.

[DOD, 1997]. *Joint tactics, techniques and procedures for noncombatant evacuation operations* (Joint Report 3-07.51). Washington, DC: Department of Defense, Joint Chiefs of Staff, 1997.

[Erol *et al.*, 1994] Erol, K., Nau, D., & Hendler, J. HTN planning: Complexity and expressivity. *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 123-1128). Seattle, WA: AAAI Press, 1994.

[Hammond, 1989] Hammond, K. *Case-based planning: Viewing planning as a memory task*. Boston, MA: Academic Press, 1989.

[Ihrig & Kambhampati, 1994] Ihrig, L.H., & Kambhampati, S. Derivational replay for partial order planning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence* (992-997). Seattle, WA: AAAI Press, 1994.

[Golden *et al*., 1996] Golden, K., Etzioni, O., and D. Weld. "*Planning with execution and Incomplete Information*," Technical Report, Department of Computer Science, University of Washington, TR96-01-09, February, 1996.

[Leake *et. al*., 1997] Leake, D., Kinley, A., & Wilson, D. A Case Study of Case-Based CBR. *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 371-382). Providence, RI: Springer, 1997.

[Mitchell, 1997] Mitchell, S.W. (1997). A hybrid architecture for real-time mixed-initiative planning and control. *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence* (pp. 1032-1037). Providence, RI: AAAI Press, 1997.

[Muñoz-Avila *et al.*, 1999] Muñoz-Avila, H., McFarlane, D., Aha, D.W., Ballas, J., Breslow, L.A., & Nau, D. Using guidelines to constrain interactive case-based HTN planning. *Proceedings of the Third International Conference on Case-Based Reasoning* (pp. 288-302). Munich: Springer, 1999.

[Muñoz-Avila & Weberskirch, 1996] Muñoz-Avila, H., & Weberskirch, F. Planning for manufacturing workpieces by storing, indexing and replaying planning decisions. *Proceedings of the Third International Conference on AI Planning Systems*. Edinburgh: AAAI Press, 1996.

[Nau *et. al*., 2000] Nau, D., Aha, D.W., & Muñoz-Avila, H. Ordered task decomposition. In Y. Gil & K. Myers (Eds.) *Representational Issues for Real-World Planning Systems: Papers from the AAAI Workshop* (Technical Report WS-00-18). Menlo Park, CA: AAAI Press, 2000.

[Nau *et. al*., 1999] Nau, D., Cao, Y., Lotem, A., & Muñoz-Avila, H. SHOP: Simple hierarchical ordered planner. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (pp. 968-973). Stockholm: AAAI Press, 1999.

[Nau *et. al*., 2001] Nau, D., Muñoz-Avila, H., Lotem, A., Mitchell, S. & Yue, C. Total-order planning with partially ordered subtasks. To appear in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*.

[Veloso, 1994] Veloso, M. *Planning and learning by analogical reasoning.* Berlin: Springer-Verlag, 1994.

[Veloso *et. al*., 1997] Veloso, M., Mulvehill, A.M., & Cox, M.T. Rationale-supported mixed-initiative case-based planning. *Proceedings of the Ninth conference on Innovative Applications of Artificial Intelligence* (pp. 1072-1077). Providence, RI: AAAI Press, 1997.

[Wilkins, 1998] Wilkins, D.E. Using the SIPE-2 planning system: A manual for Version 5.0 (Working Document). Menlo Park, CA: Stanford Research International, Artificial Intelligence Center, 1998.