

Rationale-Supported Mixed-Initiative Case-Based Planning *

Manuela M. Veloso

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
mmv@cs.cmu.edu

Alice M. Mulvehill

BBN
Cambridge, MA 01238
amm@bbn.com

Michael T. Cox

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
mcox@cs.cmu.edu

Abstract

Mixed-initiative planning envisions a framework in which automated and human planners interact to jointly construct plans that satisfy specific objectives. In this paper, we report on our work engineering a robust mixed-initiative planning system. Human planners rely strongly on past planning experience to generate new plans. ForMAT is a case-based system that supports human planning through the accumulation of user-built plans, query-driven browsing of past plans, and several plan functionality analysis primitives. Prodigy/Analogy is an automated AI planner that combines generative and case-based planning. Stored plans are annotated with plan rationale and reuse involves adaptation driven by this rationale. Our system, MI-CBP, integrates ForMAT and Prodigy/Analogy into a real-time message-passing mixed-initiative planning system. The main technical approach consists of allowing the user to specify and link objectives that enable the system to capture and reuse plan rationale. We present MI-CBP and its concrete application to the domain of military force deployment planning. This synergistic system increases the planning efficiency of human planners through automated suggestion of similar past plans and plausible plan modifications.

Introduction

In mixed-initiative planning, automated and human planners need to interact in order to jointly construct a plan that achieves the goals in a specific mission statement. In case-based planning, past plans (cases) that solved problems similar to the current problem are retrieved from memory, adapted to fit the current situation, and reused and stored as a new plan. Challenges exist when engineering either mixed-initiative or case-based planning systems. This paper reports on a successful mixed-initiative application of an AI planning system integrated with a case-based human-driven planning system.

It is well known that human planners rely strongly on past planning experience to generate new plans. ForMAT

is a case-based system that supports real human planning through the accumulation of user-built deployment plans, query-driven browsing of past plans, and a rich variety of plan functional analysis primitives (Mulvehill 1996a; 1996b). Prodigy/Analogy is a fully-automated planner that combines generative and case-based planning. Based on defined planning actions, it creates plans, interprets and stores planning episodes, and retrieves and reuses multiple past plans that are found similar to the new problem. Stored plans are annotated with plan rationale and reuse involves adaptation driven by the plan rationale (Veloso 1994a).

While the ForMAT system has been successfully used to develop deployment plans for military exercises and demonstrations, the developers have observed that better plans are generated by more experienced planners. We believe that this behavior is due to the fact that ForMAT has no way to suggest to less experienced planners how a similar plan should be modified. One of the main objectives of the work reported in this paper was to develop techniques for providing better support to a wider range of ForMAT users, building upon automated modification capabilities available in Prodigy/Analogy,

The contributions of our work on integrating mixed-initiative rationale-supported planning between ForMAT and Prodigy/Analogy include:

- Integration of an AI research system with a concrete real planning application.
- Engineering of capabilities in Prodigy/Analogy and ForMAT for the real-time continued acquisition and generation of planning cases.
- Support of the user with an interactive environment where past cases can be recalled and adapted by the user independently; with suggestions from Prodigy/Analogy; or automatically using the rationale underlying the original plan.

This paper is organized as follows. We briefly overview the ForMAT and Prodigy/Analogy systems. We introduce MI-CBP, our Mixed-Initiative Case-Based Planning system, as the integration of the two systems in its general form. We introduce the real-time communication primitives and the specific implications and developments underlying the mixed-initiative framework. We then present a detailed

*Copyright (c) 1997, American Association for Artificial Intelligence (www.aaai.org). All rights reserved. This research is sponsored as part of the DARPA/RL Knowledge Based Planning and Scheduling Initiative under grant number F30602-95-1-0018. The authors thank Steven Christey for his help in the implementation of the system.

example of MI-CBP's performance in a realistic force deployment task. Finally, we draw conclusions and briefly discuss some of the lessons learned and future work.

Although issues and examples are presented here, a more thorough description of the MI-CBP project is available at our home page, <http://www.cs.cmu.edu/afs/cs/project-prodigy/Web/Mixed-init/>, and we have several video presentations to provide demonstrations of the running system.

From Research to Application

The MI-CBP project began as two fully-implemented and large independent systems, whose integration would achieve the challenging goal of creating an operational mixed-initiative planning system. Both the ForMAT and Prodigy systems follow a different perspective. On one hand, ForMAT is a user-driven case-based planner that handles real cases built by users for force deployment of military troops. ForMAT provides sophisticated case library browsing and retrieval mechanisms as well as facilities for maintaining consistency within plans and for analyzing plans along a multitude of dimensions. However, no automated plan construction or adaptation is provided. Prodigy/Analogy, on the other hand, is integrated into the fully-automated Prodigy4.0 planner. Based on well-structured domain knowledge and planning actions, Prodigy/Analogy creates plans, interprets and stores planning episodes, and retrieves and reuses multiple past plans that are found similar to the new problem. The test domains it can handle are manufactured, although realistic. Hence, Prodigy/Analogy has a disciplined knowledge organization, interpretable planning cases, and full-automated storage, retrieval, and adaptation.

Our development of MI-CBP, as an integration of these two systems, consists of the very interesting task of bringing an AI research planning system to the service of a realistic application. We start with a brief description of the two individual systems and then we present MI-CBP.

ForMAT

ForMAT (Force Management and Analysis Tool) is a knowledge-based prototype developed to support military force deployment planning. ForMAT uses case-based reasoning techniques, along with an ensemble of internal functions and interface functions to support force deployment plan creation, modification and indexing.

ForMAT focuses on force planning, and, within the military transportation planning domain, force planning information is often encapsulated in a data structure called the force module (FM). The FM is typically a grouping of combat, combat support, and combat service support forces. The elements of a FM are linked together or uniquely identified by ULNs (Unit Line Numbers) within a deployment plan or TPFDD (Time Phased Force Deployment Data). The FMs may be extracted from or adjusted as an entity within the TPFDD to enhance the flexibility and usefulness of operational plans. In theory, these FMs form a library which can be drawn upon to quickly build a new plan.

Using ForMAT, a user can create a new FM from parts of existing FMs by cutting ULNs from one FM and past-

ing them into another FM. During a session, indices are employed to retrieve stored FMs; however, a user can also modify and/or create new indices to better describe a given FM. Using ForMAT, the user can develop a hierarchy of lower level FMs which describe the capability and/or the organization of some *parent* FM. These lower level FMs (i.e., children) can consist of existing FMs which have been copied from the FM library or they can be created by the user at any time. To date the system contains 322 FM cases derived from 17 plans (TPFDDs). Ten of the plans were built through the reuse of FMs in the other plans.

Prodigy/Analogy

Prodigy is a planning and learning architecture (Veloso *et al.* 1995) where different machine learning techniques are developed for acquiring planning domain models, and control knowledge for improving efficiency in plan generation and plan quality.

Prodigy/Analogy (Veloso 1994a; 1994b) achieves the integration of analogical reasoning into classical operator-based generative planning as a method of learning control knowledge to improve planning performance. Increase of planning efficiency occurs by the generation, storage, retrieval, and replay of annotated derivational traces of planning episodes. Prodigy/Analogy compiles complete problem solving cases that are used to guide future similar situations. Learned knowledge is flexibly applied to new problem solving situations even if only a partial match exists among problems. Figure 1 illustrates the complete planning cycle of Prodigy/Analogy. Generative planning using the substrate operator-based planner, Prodigy4.0 (Carbonell *et al.* 1992), is used if additional planning is needed when replaying the past planning cases.

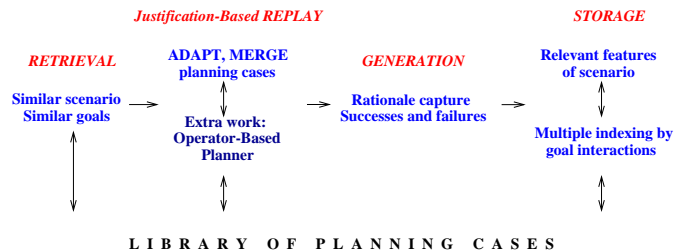


Figure 1: Prodigy/Analogy: Retrieval, replay, generation, and storage of planning cases. An operator-based planner is used as a generative planner when additional planning is needed.

Reasoning by analogy in Prodigy/Analogy consists of the automated flexible replay of derivational traces of previously solved problems to guide the search for solutions to similar new problems, avoiding a completely new search effort. Derivational analogy is a *reconstructive* method by which *lines of reasoning* are transferred and adapted to a new problem (Carbonell 1986) as opposed to only the final solutions. Automatic generation of the derivational episodes that

become the planning cases occurs by extending the base-level planner with the ability to examine its internal decision cycle, recording the justifications for each decision during its search process. There are mainly three different kinds of justifications that are annotated: links among choices capturing the subgoal structure, records of alternatives that failed or remained unexplored, and pointers to any applied guidance. A stored problem solving episode consists of the successful solution trace augmented with these annotations, i.e., the derivational trace.

When a new problem is proposed, Prodigy/Analogy retrieves from the case library one or more problem solving episodes that may partially cover the new problem solving situation. The general replay mechanism involves a complete interpretation of the justification structures annotated in the past cases in the context of the new problem to be solved. Equivalent choices are made when the transformed justifications hold. When that is not the situation, Prodigy/Analogy plans for the new goals using its domain operators adding new steps to the solution or skipping unnecessary steps from the past cases. We have applied the rationale-supported planning framework of Prodigy/Analogy in several domains including a realistic route planning domain using the real map of the Pittsburgh city (Haigh, Shewchuk, & Veloso 1996; Haigh & Veloso 1995).

MI-CBP: Mixed-Initiative Case-Based Planning

ForMAT is designed to address a specific application, namely the construction of deployment plans (TPFDDs) that describe what, when and how military forces will be deployed in support of a mission. For large missions, the process by which these plans are constructed can be very complex and time consuming. Representatives from the various services and commands involved in the overall operation plan (OPLAN) must collectively decide how best to allocate the limited transportation resources (aircraft, ships, trucks and trains) needed to support force deployment and to achieve the many military goals of the mission. Because this process is so time consuming, OPLANS are often built that *anticipate* future needs. These OPLANS, along with their associated TPFDDs, are stored and maintained until their execution is warranted. When selected for execution, the plan will generally have to be modified to fit the particular details of the current situation. Providing advice to the user on how to modify a stored deployment plan or on how to modify a deployment plan that is built from a variety of deployment plans (TPFDDs) has been a major challenge for ForMAT developers.

MI-CBP is the result of a technology integration experiment (TIE) within the DARPA-supported Planning Initiative program (Tate 1996). The TIE was established to explicitly provide mixed-initiative plan development and repair support to a force deployment user, using Prodigy/Analogy's rationale-supported replay and adaptation mechanisms.

Capturing and Using the Problem's Rationale

Our goal in developing the MI-CBP system is to investigate and to experiment with sophisticated methods for providing timely modification and/or repair support to the ForMAT user. However, in order for Prodigy/Analogy to reason about data from ForMAT, information about the plan rationale, (in particular, *objectives* of the force deployment plans) has to be provided. That is the mission's *goals* need to be known. MI-CBP therefore required some engineering in ForMAT, namely:

- The development of a goal editor and a goal consistency checking mechanism to support the ForMAT user in specifying a series of goals/subgoals with associated arguments.
- The development of a ForMAT "task window" in which plan suggestions from Prodigy/Analogy are presented. The user can use the task window to accept, reject or ignore the modification information. This provides feedback to Prodigy for use in providing better modification suggestions over time.

Communication between ForMAT and Prodigy/Analogy

MI-CBP achieves a real-time communication between ForMAT and Prodigy/Analogy through message passing. Figure 2 shows the flow of control from ForMAT to Prodigy/Analogy and back to ForMAT. Several messages are passed between ForMAT and Prodigy/Analogy. Some messages trigger specific actions in the receiving system. In particular, Figure 2 shows the actions resulting from a ForMAT user's goal specification. The input goals are translated into Prodigy goals that can be used to generate a force module plan within Prodigy. In its advising role, Prodigy uses the goals as a retrieval probe to select similar past planning cases, and provides suggestions to the ForMAT user on the necessary modifications to be done to the past plan according to the new goals.

Suggested modifications are presented to the ForMAT user via the task window. The user can *accept*, *ignore*, or *reject* the advice received from Prodigy/Analogy. Figure 3 shows how these actions can be used to help Prodigy/Analogy refine and improve the accuracy of its suggestions.

Our communication approach involves real-time message-passing and is concretely addressed in the two systems by:

- The development of a specialized ForMAT history file that captures user interactions and the sending of this information to Prodigy/Analogy.
- The development of several intra-process communications protocol mechanisms called "watchdogs" that support internet FTP message passing between the two systems.

MI-CBP easily allows the introduction of new messages. It is the responsibility of each end system to generate the appropriate message processing modules.

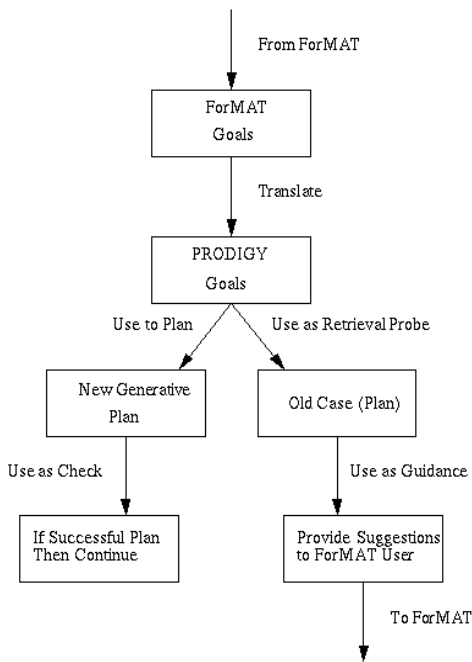


Figure 2: Feedforward flow of control.

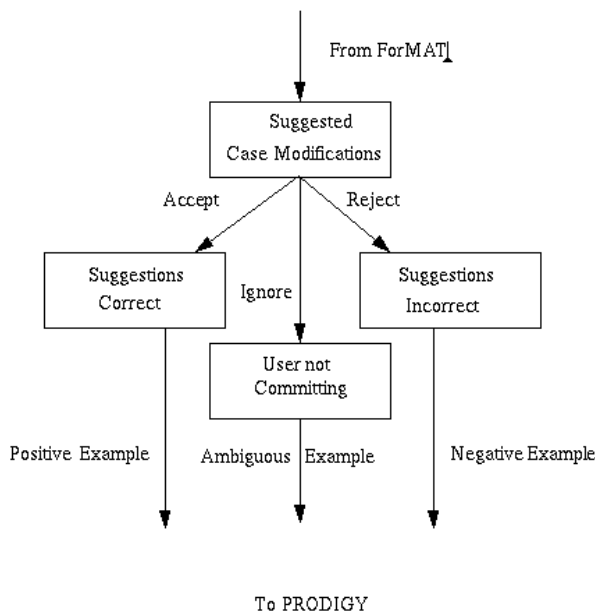


Figure 3: Feedback flow of control.

Research and Operational Issues Addressed

The development of MI-CBP raised many research, engineering and operational issues. From a *research* perspective, the issues include (but see (Cox & Veloso 1997) for further

details of the research issues):

- Interpretation of user objectives and scenario into rationale and action models.
- Under-specified goal statement from users: system plans for existentially quantified goal statements.
- Users specify goals that are subgoals of other goals: planner focuses on higher-level goals and refinement brings up the subgoals.
- Users expect plans of good quality: selection of resources to opportunistically achieve multiple objectives.
- Definition of communication triggers to enable interaction.
- Class organization of planning actions (e.g., all the “send” operators) – some fully subsume others.
- Learning from user’s history and reaction to guidance suggested.

From an *operational* perspective, the integrated system had to meet certain performance criteria to be of any value to actual users. At a minimum, MI-CBP had to provide timely and useful modification suggestions. Because of the scope of the force deployment domain, MI-CBP’s performance was measured along the following performance dimensions:

- Indicate to the ForMAT user the existence of one or more similar existing plans for reuse.
- Remind the ForMAT user to update the values of certain TPFDD fields (e.g., point of debarkation and destination) each time a new FM is created from scratch or via an FM copy.
- If a superfluous FM is copied into the new TPFDD, suggest force removal (e.g, remove the Hawk battalion).
- If there is a goal that has not been supported by the new TPFDD, suggest force addition (e.g., add security-police) or suggest force replacement (e.g., substitute an existing F-15 specification with an F-A10a specification).

In summary, the MI-CBP integration experiment successfully addressed the following issues: real-time inter-process communication using the watchdog approach, user’s objectives and mission rationale capture, goal-based plan retrieval, and plan-rationale modification.

An Illustrative Example

A few plans were initially generated to provide an initial case library. These plans were all concerned with securing an airport and deploying tactical fighters to airports. Each plan was associated with textual guidance. Using ForMAT, several FM queries were executed in order to build the TPFDDs that could support the plans.

Let a new mission statement be entered:

Provide military police to secure the town and the airstrip in Tuzla so that military a/c can land there. Deploy 3 squadrons of A-10s to Aviano to provide CAS to the Bosnia AOR (Area of Responsibility).

The user is asked to read the guidance and to specify the objectives (goals and/or subgoals) that are implied by the guidance. The ForMAT goal editor provides support for this activity (see Figure 4). Once the information is saved, the goals and/or subgoals can be attached as FM indices to the appropriate FM within the plan. The watchdog mechanism then sends the saved goals to Prodigy in the form used by ForMAT:

```

SECURE-TOWN-CENTER-HALL
    GEOGRAPHIC-LOCATION = BOSNIA
SEND-MILITARY-POLICE
    FORCE = MILITARY-POLICE
    GEOGRAPHIC-LOCATION = BOSNIA
    FORCE-QUANTITY = 1
DEPLOY-TFS
    AIRCRAFT-TYPE = A10A
    GEOGRAPHIC-LOCATION = BOSNIA
    AC-QUANTITY = 3
SECURE-AIRPORT
    GEOGRAPHIC-LOCATION = BOSNIA

```

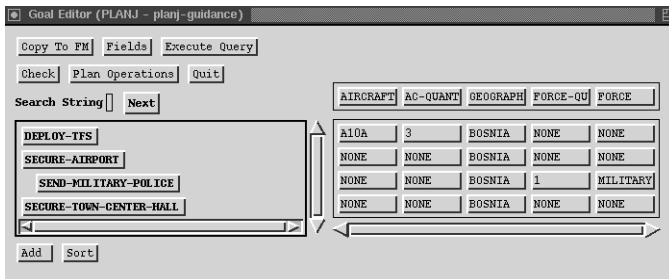


Figure 4: ForMAT's goal editor.

The incoming ForMAT goals trigger Prodigy's own analysis of the mission. Prodigy then translates the goals received into its own internal representation:

```

(TOWN-CENTER-SECURE-AT BOSNIA)
(EXISTS ((<M-P> MILITARY-POLICE))
  (IS-DEPLOYED <M-P> BOSNIA))
(EXISTS ((<A> A10A))
  (IS-DEPLOYED <A> BOSNIA))
(AIRPORT-SECURE-AT BOSNIA)

```

Prodigy can generate an abstract-level plan at the force module level. Prodigy/Analogy uses the goals to retrieve similar plans solved in the past. Based on the difference between objectives and scenarios of the new and stored past missions, Prodigy/Analogy provides the user with suggested modifications. The exact output is as follows:

```

(:ON-ACTION
  ("7/17/1996: 16:00:30" :SAVE-GOALS
    "~mi-cbp/db/planj-guidance"
  (1 PLANJ
    (:DESCRIPTION
      "goal: Provide military police to secure
      the town and the airstrip in Tuzla so that
      military a/c can land there. Deploy 3
      squadrons of A-10s to Aviano to provide
      CAS to the Bosnia Area of Responsibility."))

```

```

(:GOALS
  (38 :SECURE-TOWN-CENTER-HALL PLANJ () () ()
    ((GEOGRAPHIC-LOCATION BOSNIA)))
  (37 :SEND-MILITARY-POLICE PLANJ () () (35)
    ((FORCE MILITARY-POLICE)
     (GEOGRAPHIC-LOCATION BOSNIA)
     (FORCE-QUANTITY |1|)))
  (36 :DEPLOY-TFS PLANJ () () ()
    ((GEOGRAPHIC-LOCATION BOSNIA)))
  (35 :SECURE-AIRPORT PLANJ () (37) ()
    ((GEOGRAPHIC-LOCATION BOSNIA)))
  (:GUIDANCE-FILE "planj-guidance"))
(:MESSAGE 13 "The goals of PLANJ are most
  similar to this plan.")
(:MESSAGE 14 "The following cases are also
  relevant: (PLANE).")

```

Figure 5 displays examples, as shown in ForMAT, of advice from Prodigy/Analogy concerning modifications to be made on the similar retrieved past plans. Advice is accessible in two ways: through the "advice message window" that Prodigy/Analogy sends directly to ForMAT and/or through the ForMAT "task window," as represented in Figure 5.

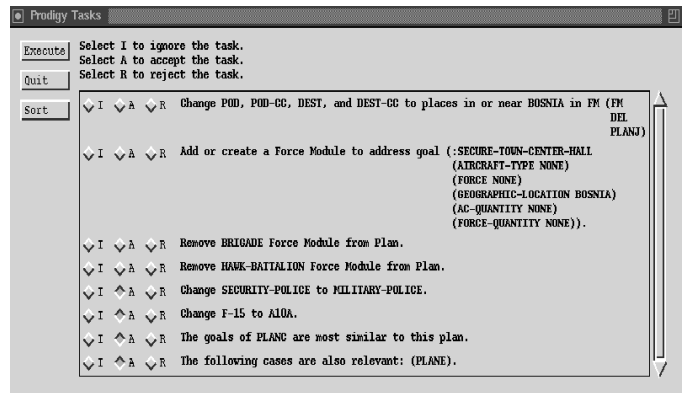


Figure 5: Prodigy advice as seen through ForMAT's task window.

The advice window (not shown) includes the actions that triggered the Prodigy/Analogy suggestion. Prodigy/Analogy will provide advice each time it detects one of the several "triggers" in the ForMAT history. The triggers are currently used to facilitate Prodigy/Analogy's interpretation of ForMAT output (history) and to provide Prodigy/Analogy with sufficient context so as to generate suggestions that are relevant. Each trigger corresponds to some combination of ForMAT operations.

The ForMAT "task window" (Figure 5) displays the advice from Prodigy/Analogy as tasks that the user might or might not want to perform. The "task window" does not display redundant advice. The "task window" is created to enable the ForMAT user to not only view Prodigy/Analogy advice but to be able to accept, reject, or ignore this advice. In turn, this user response is provided to Prodigy/Analogy as information that could enable Prodigy/Analogy to "learn" to make more appropriate modification suggestions over time.

Discussion and Conclusion

MI-CBP is a successful application of an AI automated generative and case-based planning system to a realistic human-driven case-based planning scenario. MI-CBP was provided with three training plan sets, then incrementally tested in a series of seven tests, conducted in real-time over the internet since April 1996. By the third test, MI-CBP was able to satisfy each of the performance objectives described in Section 3.3. By the fifth test, information was available through the ForMAT Task Window (Figure 5) to indicate how "useful" the ForMAT user found MI-CBP suggestions. Figure 5 is a snapshot of a ForMAT Task window that was generated to support the user in building PlanJ. Note that half of the suggestions have been accepted by the user at the time of the window capture. Our data indicates that by the end of the demonstration, most of the suggestions for PlanJ were accepted by the ForMAT user. Ignored or rejected suggestions correlated with suggestions for DEST-CC changes or with suggestions about the usefulness of previous plans.

Our work developing MI-CBP represented a major innovative application of research to a real task. A mixed-initiative system proved to be quite challenging to design and develop even within a well-focused application. For example, although MI-CBP was able to provide useful suggestions to the ForMAT user, comments by ForMAT users indicate that suggestions are not always received when they are needed. As an additional example, an unrestrained user will inevitably violate the tacit assumptions embedded in automated planners, such as the expectation that all input goals are top-level, grounded, state conditions. Prodigy/Analogy required significant control additions to manage such unexpected goal forms (Cox & Veloso 1997). Furthermore, MI-CBP directly addresses several core issues of relevance to the integration of other systems, while other more detailed or application-dependent ones remain to be investigated. MI-CBP follows a disciplined approach, but maintains the user's flexibility and control.

There are a few points worth highlighting that were implementation options that proved to be significantly responsible for the successful implementation of MI-CBP.

- The automated planner in Prodigy/Analogy reasons at the force module level. It was completely unrealistic to reason about the level of detail of the ULN actions.
- ForMAT and Prodigy/Analogy communicate by message passing. This significantly alleviates the representational burden that a common language would impose on the implementation. Messages focus attention and trigger behaviors. Therefore there is no need to fully delve into either system.
- Use of case-based planning technology proved appropriate to bridge the gap between human and machine planners.

MI-CBP emphasizes the importance of providing a planning environment that facilitates synchronicity between the human planner and the plan modification/repair and suggestion system that is running in tandem. Providing better

intra-process control between these environments is critical in order to develop a more facile planning system.

In our future work, a mechanism needs to be built that supports the parsing of the guidance into a set of goals and associated arguments, since manual parsing of the guidance is tedious and subjective. Furthermore, a mechanism must be developed to aid Prodigy/Analogy, when utilizing the ForMAT history files (especially for context and Prodigy/Analogy environmental initialization of the state information); Prodigy/Analogy currently uses the objective-based information from the ForMAT history files.

Finally, since Prodigy/Analogy not only uses case-based reasoning methods, but supports plan generation strategies, considerations should be taken to provide the ForMAT user access to the generated plan. Generated plans may be preferred when there are no plans to reuse and/or when plan adaptation is overly complex.

References

- Carbonell, J. G.; Blythe, J.; Etzioni, O.; Gil, Y.; Joseph, R.; Kahn, D.; Knoblock, C.; Minton, S.; Pérez, A.; Reilly, S.; Veloso, M. M.; and Wang, X. 1992. PRODIGY4.0: The manual and tutorial. Technical Report CMU-CS-92-150, Computer Science Dept., Carnegie Mellon University.
- Carbonell, J. G. 1986. Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In Michalski, R.; Carbonell, J.; and Mitchell, T., eds., *Machine Learning, An Artificial Intelligence Approach, Vol. II*, 371–392. Morgan Kaufmann.
- Cox, M. T., and Veloso, M. M. 1997. Controlling for unexpected goals when planning in a mixed-initiative setting. In *Submitted to EPIA-97*.
- Haigh, K. Z., and Veloso, M. M. 1995. Route planning by analogy. In *Case-Based Reasoning Research and Development, Proceedings of ICCBR-95*, 169–180. Springer.
- Haigh, K. Z.; Shewchuk, J.; and Veloso, M. M. 1996. Exploring geometry in analogical route planning. *To appear in Journal of Experimental and Theoretical AI*.
- Mulvehill, A. M. 1996a. Building, remembering, and revising force deployment plans. In Tate, A., ed., *Advanced Planning Technology*. Menlo Park, CA: AAAI Press.
- Mulvehill, A. M. 1996b. Reusing force deployment plans. In *AAAI Symposium on Adaptation of Knowledge for Reuse*. Menlo Park, CA: AAAI Press.
- Tate, A., ed. 1996. *Advanced Planning Technology*. Menlo Park, CA: AAAI Press.
- Veloso, M. M.; Carbonell, J.; Pérez, M. A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating planning and learning: The PRODIGY architecture. *Journal of Experimental and Theoretical Artificial Intelligence* 7(1):81–120.
- Veloso, M. 1994a. *Planning and Learning by Analogical Reasoning*. Springer Verlag.
- Veloso, M. M. 1994b. Flexible strategy learning: Analogical replay of problem solving episodes. In *Proc. of 12th Nat. Conf. on Artificial Intelligence*, 595–600. AAAI Press.