In Proceedings of the AAAI 2000 workshop on Representational Issues for Real-world Planning Systems.

# PLANET: A Shareable and Reusable Ontology for Representing Plans

**Yolanda Gil and Jim Blythe**
University of Southern California / Information Sciences Institute
Marina del Rey, CA 90292
`gil@isi.edu, blythe@isi.edu`

## Abstract

Ontologies are becoming a recognized vehicle for knowledge reuse, knowledge sharing, and modeling. This paper presents PLANET, an ontology for representing plans. To show that PLANET is a comprehensive ontology that can be *reused* to build new applications, we describe several specializations of it to represent plans in three different real-world domains that were made by adding minor extensions to PLANET's general definitions and classes. In past work, we have developed several plan evaluation systems in these domains, and have integrated them in several occasions with plan editors and plan generation systems. For each of these integrations, and often for each system in the integration, a new format for exchanging plans was devised. PLANET can represent all of the plans that we and others have used in these domains, providing a standard universal format for knowledge *sharing* that can be used as an interlingua in integrated planning applications. Finally, the paper discusses how we have used PLANET as a *knowledge modelling* tool to design representations for courses of action in a military domain, guiding us to make useful distinctions and modelling choices.

## Introduction

As we develop larger and more complex intelligent systems in knowledge-intensive domains, it becomes impractical and even infeasible to develop knowledge bases from scratch. Recent research investigates how to develop intelligent systems by drawing from libraries of reusable components that include both ontologies (Neches *et al.* 1991) and problem-solving methods (Breuker and Van de Velde 1994). This paper introduces PLANET[1], a reusable ontology for representing plans. PLANET complements recent efforts on formalizing, organizing, and unifying AI planning algorithms (Kambhampati et al 1995; Yang 1990; Nunes et al 1997) by focusing on the representation of plans, and adds a practical perspective in that it is designed to accomodate a diverse range of real-world plans (including manually created ones). As more complex planning systems are developed to operate in knowledge-intensive environments, ontologies present an approach to enable richer plan representations (Cohen *et al.* 1998; Valente *et al.* 1999).

We have drawn from our past experience in designing, developing and integrating planning tools, and expect PLANET to ease these tasks in the future in three ways. First, we have already found it useful for *knowledge modelling*. By providing a structure that formalizes useful distinctions for reasoning about states and actions, a knowledge engineer can find the semantics of informal expressions of plans (e.g., textual or domain-specific) through designing mappings to the ontology. Reports on efforts to model plans in various application domains (Nau et al 1995; Knoblock 1996) indicate the difficulties of representing real-world domains, and point out the need for better methodologies for knowledge modelling for planning and for richer representations of planning knowledge. We believe that PLANET takes a step in that direction. Second, a plan ontology can be a central vehicle for *knowledge reuse* across planning applications. PLANET contains general, domain-independent definitions that are common and useful across planning domains. To create a plan representation in a new domain, these general definitions can be used directly and would not need to be redefined for every new domain. Only domain-dependent extensions will need to be added. Third, PLANET should facilitate integration of planning tools through *knowledge sharing*. Currently, practical efforts to integrate planning tools are done by designing separate interchange formats for (almost) each pair of tools, since designing a more universal format is costly and often more difficult than designing the entire set of pairwise formats. These difficulties are in part because these systems include decision-support tools such as plan editors, plan evaluation tools, and plan critiquers (Bienkowski and Hoebel 1998), which represent plans in ways that are different from traditional AI plan generation systems. An ontology like PLANET can provide a shared plan representation for systems to communicate and exhange information about the plan, and can facilitate the creation of a common, overarching knowledge base for future integrations of planning tools. An example of a successful integration of planning tools through a knowledge base is shown in (Valente *et al.* 1999).

PLANET makes the following representational commitments to provide broad coverage. First, planning contexts that refer to domain information and constraints that form the background of a planning problem are represented explicitly. Planning problems, which supplement the context with information about the initial state of the world and the goals, are represented explicitly and are accessible from the

---

[1] PLANET: a PLAN semantic NET

context. Alternative plans themselves are then accessible from each planning problem for which they are relevant. Second, PLANET maintains an explicit distinction between *external constraints*, which are imposed on a context or planning problem externally to a planning agent (including user advice and preferences), and *commitments* which the planning agent elects to add as a partial specification of a plan (for example, a step ordering commitment). The current version of PLANET does not represent aspects related to the execution of plans and actions, adversarial planning, or agent beliefs and intentions.

We present the main definitions in PLANET, including initial planning context, goals, actions and tasks, and choice points. Next, we describe three specializations of PLANET for three real-world domains where plans are of a very different nature. We conclude with a discussion of related work and some anticipated directions for future work.

## PLANET: An Ontology for Representing Plans

This section describes how different aspects of a plan are represented in PLANET. As a convention, we use boldface to highlight terms that are defined in PLANET when they are first introduced and described in the text. Figure 1 shows a diagram of the major concepts and relations in the ontology.

### Planning Problems, Scenarios, and Contexts

A **planning problem context** represents the initial, given assumptions about the planning problem. It describes the background scenario in which plans are designed and must operate on. This context includes the initial state, desired goals, and the external constraints.

A **world state** is a model of the environment for which the plan is intended. When using a rich knowledge representation system, the state may be represented in a context or microtheory. A certain world state description can be chosen as the **initial state** of a given planning problem, and all plans that are solutions of this planning problem must assume this initial state.

The **desired goals** express what is to be accomplished in the process of solving the planning problem. Sometimes the initial planning context may not directly specify the goals to be achieved, instead these are deduced from some initial information about the situation and some abstract guidance provided as constraints on the problem.

We make a distinction between *external constraints* imposed on planning problems and the *commitments* made by the plan. **External constraints** may be specified as part of the planning context to express desirable or undesirable properties or effects of potential solutions to the problem, including user advice and preferences. Examples of external constraints are that the plan accomplishes a mission in a period of seven days, that the plan does not use a certain type of resource, or that transportation is preferrably done in tracked vehicles. Commitments are discussed later.

The initial requirements expressed in the planning problem context need not all be consistent and achievable (for example, initial external constraints and goals may be incompatible), rather its aim is to represent these requirements as given. A plan may satisfy or not satisfy external constraints. PLANET represents these options with multiple **planning problems** for each planning problem context, which may add new constraints and goals, or relax or drop given ones. A planning problem is created by forming specific goals, constraints and assumptions about the initial state. Several plans can be created as alternative solutions for a given planning problem. A planning problem also includes information used to compare alternative candidate plans. Planning problems can have descendant planning problems, which impose (or relax) different constraints on the original problem or may assume variations of the initial state. Typically, AI planning systems assume one given planning problem and do not address this process, which is essential when working with real-world environments.

A planning problem may have a number of **candidate plans** which are potential solutions. A candidate plan can be **untried** (i.e., it is yet to be explored or tested), **rejected** (i.e., for some reason it has been rejected as the preferred plan) or **feasible** (i.e., tried and not rejected). One or more feasible plans may be marked as **selected**. All of these are sub-relations of candidate plan.

### Goals, Objectives, Capabilities, and Effects

A **goal specification** represents anything that gets accomplished by a plan, subplan or task. Both capabilities and effects of actions and tasks are subtypes of goal specification, as well as posted goals and objectives. Goals may be variabilized or instantiated. **State-based goal specifications** are goal specifications that typically represent goals that refer to some predicate used to describe the state of the world, for example ‘achieve (at Jim LAX)’, ‘deny (at Red-Brigade South-Pass)’ or ‘maintain (temperature Room5 30)’. **Objective-based goal specifications** are goal specifications that are typically stated as verb- or action-based expressions, such as ‘transport brigade5 to Ryad’.

Goal specifications also include a **human readable description** used to provide a description of a goal to an end user. This is useful because often times users want to view information in a format that is different from the internal format used to store it. This could be a simple string or a more complex structure.

### Actions, Operators, and Tasks

**Plan task descriptions** are the actions that can be taken in the world state. They include templates and their instantiations, and can be abstract or specific. A plan task description models one or more **actions** in the external world.

A **plan task** is a subclass of **plan task description** and represents an instantiation of a task as it appears in a plan. It can be a partial or full instantiation. A **plan task template** is also a subclass of **plan task description** that denotes an action or set of actions that can be performed in the world state. In some AI planners the two classes correspond to operator instances and operator schemas respectively, and in others they are called tasks and task decomposition patterns.

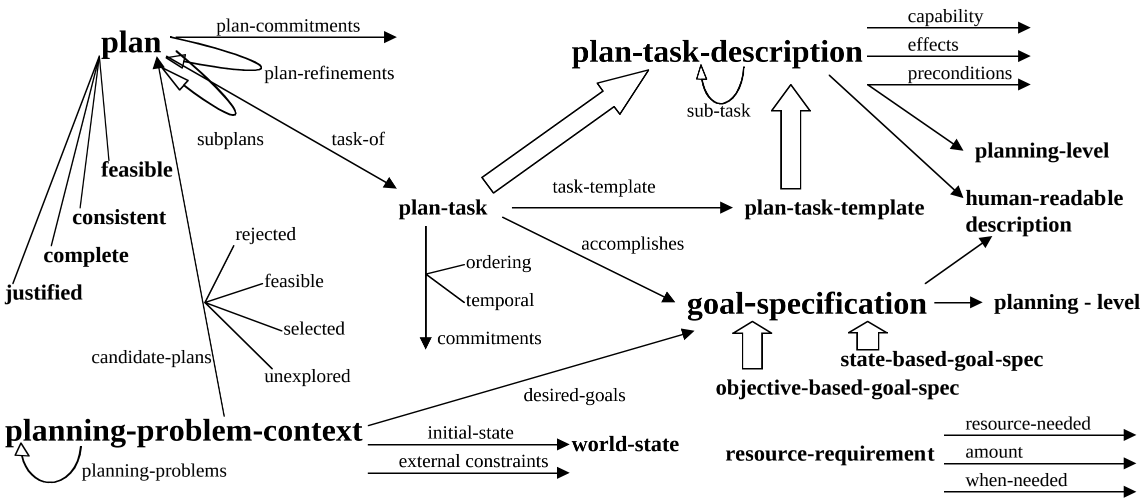

Figure 1: An overview of the PLANET ontology. Arrows pointing into space represent relations whose ranges are not fixed in the ontology.

Plan task descriptions have a set of preconditions, a set of effects, a capability, and can be decomposed into a set of subtasks. Not all these properties need to be specified for a given task description, and typically planners represent tasks differently depending on their approach to reasoning about action. The **capability** of a task or task template describes a goal for which the task can be used. A **precondition** represents a necessary condition for the task. If the task is executed, its **effects** take place in the given world state. Tasks can be decomposed into **subtasks** that are themselves task descriptions. Hierarchical task network planners use task decomposition or operator templates (represented here as plan task templates) and instantiate them to generate a plan. Each template includes a statement of the kind of goal it can achieve (represented as a **capability**), a decomposition network into **subtasks**, each subtask is matched against the task templates down to primitive templates, represented as **primitive plan task descriptions**. Other planners compose plans as an ordered set of primitive **plan steps** (often called operators, as in STRIPS and UCPOP (Weld 1994)). Plan steps are specializations of primitive plan task descriptions that have some set of effects, as they are typically used in means-ends analysis planners.

Like goal specifications, plan task descriptions also include a **human readable description**. Some AI planners specify this information as a set of parameters of the task that are used to determine which subset of arguments will be printed when the plan is displayed.

**Planning levels** can be associated to task descriptions as well as to goal specifications. Some AI planners assign levels to tasks (e.g., SIPE (Wilkins 1988)), others assign levels to particular predicates or goals (e.g., ABSTRIPS). Levels are also used in real-world domains, for example military plans are often described in different levels according to the command structure, echelons, or nature of the tasks.

## Plans

A **plan** represents a set of commitments to actions taken by an agent in order to achieve some specified goals. It can be useful to state that a plan forms a **sub-plan** of another one. For example, military plans often include subplans that represent the movement of assets to the area of operations (i.e., logistics tasks), and subplans that group the operations themselves (i.e., force application tasks).

## Choice Points, Alternatives, Decisions, and Commitments

In searching or designing a plan, a number of choices typically need to be made. At a given choice point, several alternatives may be considered, and one (or more) chosen as selected. Such choices are represented in PLANET as a type of **commitment**. Commitments can be made in both plans and tasks. **Plan commitments** are commitments on the plan as a whole, and may be in the form of actions at variously detailed levels of specification, orderings among actions and other requirements on a plan such as a cost profile. The tasks that will form part of the plan are represented as a subset of the commitments made by the plan. **Task commitments** are commitments that affect individual tasks or pairs of tasks. An **ordering commitment** is a relation between tasks such as (before A B). A **temporal commitment** is a commitment on a task with respect to time, such as (before ?task ?time-stamp). Another kind of commitment is the selection of a plan task description because it **accomplishes** a goal specification. This relation records the intent of the planning agent for the task, and is used in PLANET to represent causal links.

## Discussion

PLANET does not include representations for some entities that are typically associated with planning domains, *e.g.* agents, resources, time, and location. Different systems that reason about plans use different approaches to represent and reason about these entities. Separate ontologies for them can be developed and integrated with PLANET. We use PLANET in combination with an ontology of Allen's time relations and the OZONE resource ontology (Smith, Lassila, & Becker 1996), and in combination with an ontology of plan evaluations and critiques that we have developed. For systems and domains where there is no need for complex representations of agents, resources, time, and location, it is trivial to extend PLANET with simple representations of them and we have done so ourselves for some of the domains described below.

## Using PLANET for Real-World Domains

This section describes how we used PLANET to represent plans in three different domains. Although all three are military domains, the plans are of a radically different nature in each case. In the first two domains, plans were built manually by users and needed to be represented as given, containing potential flaws and often serious errors. In the JFACC domain, plans are hierarchically decomposed and have verb-based objectives. Information about causal links and task decomposition templates is not provided. In the COA domain, plans have a hierarchical flavor that is not always explicitly represented in the plan. In the Workarounds domain, plans were generated automatically by an AI planner. This section describes the domains in more detail.

### PLANET-JFACC

This is a domain of air campaign planning where users follow the strategies-to-tasks methodology (Todd 1994; Thaler 1993). In this approach, users start with high-level objectives and decompose them into subobjectives all the way down to the missions to be flown. Using a plan editing tool, a user defines objectives, decomposes them into subobjectives, and can specify temporal orderings among plan steps. Some subobjectives may be handed to an automated planner to be fleshed out to lower levels. The rest of this discussion will focus on the representation of these manually created plans.

Figure 2 shows an excerpt of an air campaign plan as it would be specified by a domain expert, indicating the hierarchical decomposition through indentation. Options (marked with stars) indicate disjunctive branches of the plan that are explored as alternatives. The bottom of the figure shows how a user can specify an objective.

**Air campaign objectives** are verb-based statements, so we represent them as a subclass of objective-based goal specifications. Some of their clauses are turned into constraints on the goal, including temporal constraints (*within 21 days*), geographical constraints (*in Western Region*), and resource constraints (*using B-52s from airbase XYZ*). Each objective may have several children and several parents (unlike plans generated by hierarchical AI planners where there is only one parent). **Options** indicate alternative ways to decompose an objective, and are represented as a specialization of alternative plans. The decomposition hierarchy is divided into **levels**, including low-level air tasks and other higher-level air objectives.

O1: Eliminate enemy SSM threat to US allies by D+5
  *Option1: Destroy all known enemy SSM launchers and launch facilities by D+5
    O111: Destroy fixed enemy SSM launch sites by D+5
      O1111: Destroy [them] on NW area using precision weapons
        *Option1: Destroy [them] using stealth aircraft
          O111111: Destroy [them] using F117 with GBU-27
        *Option2: Destroy [them] using SEAD aircraft
launch sites by D+5
    O112: Destroy storage facilities for SSM equipment by D+5
  *Option2: Disrupt and disable the enemy C2 infrastructure for SSM
O2: Airlift wounded and civilian non-combatants by D+2

*Objective ID:* O-152 *Level:* AO *Phase:* II *Parents:* O-98, O-61
*Statement:* Maintain air superiority over NW sector
*Sequence restrictions:* Before O-138, Before O-124

Figure 2: An excerpt of an air campaign plan and the specification of an objective.

### PLANET-COA

This is a Course of Action (COA) analysis problem in a military domain of relevance to the DARPA High Performance Knowledge Bases (HPKB) Program (Cohen *et al.* 1998). We developed a critiquing tool that finds flaws in manually developed COAs for Army operations at the division level. A COA is specified by a user as a set of textual statements (*who* does *what*, *when*, *where*, and *why*), together with a sketch drawn over a map. The PLANET-COA ontology allows us to represent the COA that results from joining both text and sketch, which is the input to our critiquing tool. An example of part of a COA textual statement follows:

*On H hour D day, a mechanized division attacks to seize OBJ SLAM to protect the northern flank of the corps main effort. A mechanized brigade attacks in the north, as an economy of force, to fix enemy forces in zone denying them the ability to interfere with the main effort's attack in the south. A tank heavy brigade, the main effort, passes through the southern mechanized brigade and attacks to seize the terrain vicinity of OBJ SLAM denying the enemy access to the terrain southwest of RIVER TOWN. [...]*

A typical COA includes the overall mission and a set of tasks that need to be performed divided into five categories: close, reserve, security, deep and rear (not shown here). The close statements always contains a main effort for the COA and a set of supporting efforts. In our representation, the mission defines two important features of the plan: its top-level goal (e.g., *protect the northern flank of the corps main effort*), and an indication of the top-level task to be used to accomplish that goal (e.g., *attack to seize OBJ SLAM*). We define COA **problem** as a subclass of planning-problem, make its problem goal the top-level goal indicated in the mission statement, and add the rest of the mission statement as a constraint on how to select tasks in the plan. The five task categories are represented as sub-plans (they are not subtasks or subgoals but useful categories to group the unit's activities). Each sentence in the statement is turned into a plan task as follows. There is a specification of *who* is doing the task, e.g., *a mechanized brigade*, which is represented as the agent of the plan-task. There is an indication of *what* is to be done, e.g., *attacks to fix enemy forces*, which is interpreted as a fix plan-task (where fix is a kind of task that is a subclass of the class attack). The *why* (or *purpose*) e.g., *to deny enemy forces the ability to interfere with the* COA*'s main effort* can be a state-based ("enable P", "prevent P") or action-based ("protect another unit from enemy"). Therefore, the ontology defines the **purpose** of a COA task as a goal specification that can be either an effect or a capability of the plan-task. The *where*, e.g., *in the North* is the location of the plan task. The *when* clause (e.g., *H hour D day*) is represented as a temporal commitment or as an ordering commitment if it is specified with respect to another task. Finally, the **main effort** and **supporting efforts** are defined as specializations of the subtask relation.

The PLANET ontology also represents the context, assumptions, and situation in which the plan is supposed to work in this domain. A COA is supposed to accomplish the mission and other guidance provided by the commander, and to work in the context of the given situation as analyzed by the commander's staff, which includes terrain information and enemy characteristics. We define COA **problem context** as a subclass of planning-problem-context, and define its scenario to be composed of **commander products** and **staff products**. All COA problems are attached to this problem context.

## PLANET-Workarounds

We developed a tool to aid in military target analysis by analyzing how an enemy force may react to damage to a geographic feature (e.g., a bridge or a tunnel) (Cohen *et al.* 1998). The possible workarounds include using alternative routes, repairing the damage, or breaching using engineering techniques such as installing a temporary bridge. Because the purpose of damaging the target is typically to delay the movement of some enemy unit or supply, it is important to estimate how long the implementation of the workaround will take. Note that this depends on what actions can be performed in parallel. The system was also designed to show not one possible workaround plan but several options that the enemy may take.

We divided the problem in two. First, we used the AI planner Prodigy (Veloso *et al.* 1995) to generate a workarounds plan. We added information to the operators about the resources used, and which resources are non-shareable. The planner then generated a partial order of workaround steps, in which unordered steps can be completed in parallel. Second we built a plan evaluation system to estimate the time that each step takes to complete and calculate the overall duration based on the partial order. This is a knowledge-based system that used several ontologies of engineering assets, units, and workaround steps and plans.

PLANET did not exist when this workarounds plan ontology was first developed, so we describe a reimplementation using PLANET. Actions are represented as primitive plan steps. The ordering commitments and resources used are straightforward to represent in PLANET. In the planner we subdivided the step parameters into those whose values affected plan correctness and those that were only used to determine the length of the plan after it was created. This distinction had not been captured in the original system.

| Domain | Axioms | Concepts | Rels | Covered concepts | Coverage |
|---:|---:|---:|---:|---:|---:|
| PLANET | 305 | 26 | 37 | | |
| COA | 267 | 58 | 37 | 39 | 67% |
| COA u.c | 106 | 7 | 12 | | 35% |
| JFACC | 102 | 15 | 12 | 12 | 80% |
| JFACC u.c | 86 | 9 | 6 | | 28% |
| WA | 100 | 13 | 10 | 13 | 100% |
| WA u.c | 91 | 12 | 4 | | 30% |

Table 1: Estimates of reuse of the PLANET ontology.

## Coverage and Knowledge Reuse

We wanted to measure the amount of reuse of the general PLANET ontology in each specific domain. Here we present estimates of reuse in creating new terms, since we are interested in understanding the generality and coverage of PLANET. To do this, we estimated how many axioms of PLANET were actually used in each domain, and how many new axioms were needed.

It is important to factor out domain definitions that are part of what is often described as populating the knowledge base, or knowledge base *stuffing*. For example, there may be fifty or five hundred possible tasks in a domain that share the same basic structure but this should not distort a measure of how reusable a general-purpose ontology is. For this evaluation we take these definitions out of the domain-specific ontologies and leave only those definitions that specified the basic structure of the plans. We estimated the size of each ontology by counting its axioms. We considered an axiom to be any statement about the world, including **isa** predicates, class constraints, and role constraints. We make strong use of inheritance among classes, so axioms are only stated and thus counted once.

We counted the concepts in each domain that were subconcepts of a concept in the PLANET ontology, to measure of the coverage of the domain that the ontology provided. We estimated how many axioms of the PLANET ontology were actually used in each domain by computing the "upward closure" of the definitions in the domain ontologies. The results are as shown in Table 1. Coverage is high: on average, 82% of the concepts defined in each domain are subconcepts of concepts in PLANET. However, the proportion of the ontology used by each domain is much lower, averaging 31% of the axioms. This is not surprising. First, PLANET covers a range of planning styles, including actions with preconditions and effects and decomposition patterns, but none of the domains has all of these. Second, PLANET can represent incremental decisions of planners, including commitments and untried alternatives, but the domains only represented complete plans. In general we do not expect a single domain to use a high proportion of the ontology.

There are various other ways to reuse knowledge. One can measure the reuse of ontologies by estimating how many terms are used during problem solving or reasoning. An informal analysis of the JFACC and Workarounds domains (the problem solvers for the COA domain were under development) showed that most (if not all) the new definitions would be used during problem solving, but this should be determined empirically. The ontology is also reused in modelling a new domain. Even if a term is not used in the new system, it may still have been used to understand how to model certain aspects of the domain as we discuss next.

## Related Work

In creating PLANET, we have drawn from previous work on languages to represent plans and planning knowledge (Ghallab *et al.* 1998; Wilkins and Myers 1995; Kambhampati et al 1995; Yang 1990). These languages are often constrained by the reasoning capabilities that can be provided in practice by AI planning systems. Since PLANET is an ontology, it does not make specific commitments about the language in which various items are expressed. The planning knowledge represented in these languages can be mapped into PLANET. PLANET also accomodates plans that were not created by AI planning systems, and provides a representation for the context of the planning problems that are given to these systems.

SPAR (Tate 1998) is an ongoing effort to create a standard vocabulary for plans that is compatible with other standards, such as the Process Interchange Format (PIF) and the NIST Process Specification Language (PSL). These efforts are aimed at plan representations of a more general nature, and cover aspects of plan execution. However, as a result of their generality they would require many more extensions than PLANET to represent the domains discussed in this paper.

Related work on problem-solving methods for planning (Valente et al. 1998; Nunes et al 1997)analyzes AI planning algorithms (or planning methods) and identifies the typical knowledge roles that characterize the main types of domain knowledge used by these planning methods. The main knowledge roles in this study map directly to classes in PLANET. It would be useful to add to PLANET the static vs dynamic distinctions contributed by this study.

## Conclusions

We described PLANET, an ontology for representing plans, and showed its use in three real-world domains, two where plans are created by humans and one where they are created by an AI planner. In these domains, a high proportion of the classes created were usefully covered by PLANET. PLANET is also useful in knowledge modelling by structuring important distinctions in planning domains, and can ease the task of creating new planning or plan evaluation systems. The ontology also shows promise as a tool for integrating different systems that combine to solve a planning problem.

## Acknowledgements

We gratefully acknowledge the support of DARPA with contract DABT63-95-C-0059 as part of the DARPA/Rome Laboratory Planning Initiative, with contract F30602-97-C-0118 as part of the DARPA Joint Forces Air Component Commander (JFACC) program, and with grant F30602-97-1-0195 as part of the DARPA High Performance Knowledge Bases Program.

## References

Bienkowski, M. and Hoebel, L. 1998. Integrating AI Components for a Military Planning Application. In *IAAI-98*.

Breuker, J. and Van de Velde W. (Eds.) 1994. *CommonKADS Library for Expertise Modelling*. IOS Press, Amsterdam, The Netherlands, 1994.

Cohen, P.; Schrag, R.; Jones, E.; Pease, A.; Starr, B.; and Gunning, D. 1998. The darpa high-performance knowledge bases project. *AI Magazine* 19(4).

Ghallab, M.; Howe, A.; Knoblock, C.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. Pddl — the planning domain definition language. Technical report. Available at http://www.cs.yale.edu/pub/mcdermott/software/pddl.tar.gz.

Kambhampati, S., Knoblock, C. A., and Yang, Q. 1995. Planing as Refinement Search: A Unified Framework for Evaluating Design Tradeoffs in Partial-Order Planning. *Artificial Intelligence*, 76, pp 167–238.

Knoblock, C. A. 1996. Building a Planner for Information Gathering: A Report from the Trenches. In *AIPS-96*.

Myers, K. L. 1996. Strategic Advice for Hierarchical Planners. In *KR-96*.

Nau, D. S., Gupta, S. K., and Regli, W. C. 1995. AI Planning Versus Manufacturing-Operation Planning: A Case Study. In *IJCAI-95*.

Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; and Swartout, W. 1991. Enabling technology for knowledge sharing. *AI Magazine* 36–56.

Nunes de Barros, L., Hendler, J., and Benjamins, V. R. 1995. AI Planning Versus Manufacturing-Operation Planning: A Case Study. In *IJCAI-97*.

Smith, S. F.; Lassila, O.; and Becker, M. 1996. Configurable, mixed-initiative systems for planning and scheduling. In *Advanced Planning Technology*.

Tate, A. 1998. Roots of SPAR–Shared Planning and Activity Representation. *The Knowledge Engineering Review* 13:1, pp 121–128, March 1998.

Thaler, D. E. 1993. Strategies to Tasks, A Framework for Linking Means and Ends. *RAND Technical Report*.

Todd, D. F. 1994. Strategies-to-Tasks Baseline for USAF Planning. *Internal Document, Strategic Planning Division, HQ United States Air Force*.

Valente, A., Benjamins, V. R., Nunes de Barros, L. 1998. A Library of System-Derived Problem-Solving Methods for Planning. *International Journal of Human-Computer Studies*, 48.

Valente, A.; Russ, T.; MacGregor, R.; and Swartout, B. 1999. Building and (re)using an ontology of air campaign planning. *IEEE Intelligent Systems* 14(1).

Veloso, M.; Carbonell, J.; Pérez, A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating planning and learning: The prodigy architecture. *J. Experimental and Theoretical AI* 7:81–120.

Weld, D. 1994. A gentle introduction to least-commitment planning. *AI Magazine*.

Wilkins, D. E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm*. Morgan Kaufmann.

Wilkins, D. E. and Myers, K. L. 1995. A Common Knowledge Representation for Plan Generation and Reactive Execution. *Journal of Logic and Computation*, 5(6), pp 731–761

Yang, Q. 1990. Formalizing Planning Knowledge for Hierarchical Planning. *Computational Intelligence*, 6(1), pp 12–24.