From: Proceedings of the AAAI-2000 Workshop Representational Isses for Real-World Planning Systems.

Domain Metatheories: Enabling User-Centric Planning

Karen L. Myers
Artificial Intelligence Center
SRI International
333 Ravenswood Ave.,
Menlo Park, CA 94025
myers@ai.sri.com

Abstract

Substantial improvements must be made in the usability of AI planning technologies in order for them to achieve widespread adoption. In particular, planning systems must be designed to better serve the needs of users, who generally want to play a central and ongoing role in the plan development process. In this paper, we argue that improved usability requires a new representational layer that captures metatheoretic properties of a planning domain. A *domain metatheory* would provide an abstract characterization of planning elements that highlights key semantic differences among them. This paper presents a candidate model for a domain metatheory, as well as an instantiation of that model for a travel-planning domain. The paper also describes three user-centric planning capabilities that the model enables, namely, *user directability of planning*, *generation of qualitatively different plans*, and *plan summarization*.

Introduction

Artificial intelligence (AI) planning technology provides powerful tools for solving problems that require the coordination of actions in the pursuit of specified goals. The AI community has produced several planning systems whose demonstrations on realistic problems attest to the value of automated planning techniques. Nevertheless, there has been limited success in transitioning this technology to user communities. A major reason for the lack of technology transfer lies with the difficulty of *using* planning systems. AI planners have traditionally been designed to operate as "black boxes": they take a description of a domain and a set of goals and automatically synthesize a plan for achieving the goals. This design explicitly limits the amount of influence that a user can have on the generated plans.

In many domains, users are reluctant to relinquish full control of the planning process. Several factors contribute to this reluctance. One is a belief that human creativity and experience are essential for effective planning in complex domains. Transparency is another: even in the event that a planning knowledge base could completely capture the subtleties of a particular domain, users often want to participate in plan development, in order to ensure that they understand both the structure of a plan and the process by which planning decisions were made.

For these reasons, the future of automated planning lies in the development of *user-centric* planning tools that assist human decision makers rather than replace them. These tools should both aid a user in understanding the complexity of the underlying problem, and provide guidance in determining a solution that is well suited to his or her specific needs.

The main thesis of this paper is that effective interaction between users and planning technology requires an augmentation of standard planning models to include an explicit *domain metatheory*. A standard planning domain is modeled in terms of three types of elements: *individuals* corresponding to real or abstract objects in the domain, *relations* that describe characteristics of the world and individual world states, and *operators* that describe ways to achieve objectives. The domain metatheory would capture high-level semantic attributes of planning elements, thus providing a rich vocabulary for describing characteristics of solutions and problem-solving processes. As argued in this paper, metatheories of this type would facilitate a broad range of user-planner interactions.

Different types of metatheories may suit different purposes. This paper presents a candidate model for a planning metatheory, along with an instantiation of that model for a travel-planning domain (Mayer 1997). The paper describes how this metatheory was used to support two critical user-centric planning capabilities: *user directability of the planning process* and the *generation of qualitatively different plans*. In addition, a proposal is put forth for a third capability: *plan summarization*.

A Domain Metatheory

Overview

Our candidate domain metatheory is built on three main constructs: *roles*, *features*, and *measures*.

A *feature* designates an attribute of interest for an operator that distinguishes it from other operators that could be applied to the same task. For example, among operators that can be used to refine tasks of moving from location X to location Y, there can be some that involve travel by air, land, or water; each of these media could be modeled as a feature. Because there can be multiple operators that apply to a particular task, features provide a way of abstracting from the details of an operator to distinguishing attributes that might be of interest to users. Note that features differ from operator preconditions in that they do not directly restrict the use of operators by a planner.

Related features are grouped into *feature categories*. For example, the features {Air Land Water} mentioned above define a **Transport-Media** category. Feature categories themselves can have interesting properties. Just as planning operators reflect a hierarchical structure, features and feature categories can be organized hierarchically. Certain categories may be *mutually exclusive* in that at most one feature from the category can be assigned to any given operator; that is the case for the feature category **Transit-Ownership** containing the elements {Public Private}. Other categories may support overlapping features; for example, there may be an operator that involves both Air and Land travel.

A *role* corresponds to a capacity in which a domain object is to be used within an operator. Roles map to individual variables within a planning operator. For instance, an air transportation operator could have variables `location.1` and `location.2` that correspond to the roles of Origin and Destination, respectively, as well as a variable `airline.1` that corresponds to the role of Carrier. A comparable sea transportation operator may have these same roles, although with the planning variable `cruise-ship.1` for the role Carrier.

Feature categories can have associated *measures*. A measure corresponds to an ordering (possibly partial) of features within the category with respect to some designated criteria. For example, consider the feature category **Transit-Ownership** with features {Public Private}. For the measure COMFORT, the feature Private would rank higher than Public; for the measure AFFORDABILITY, the order would be reversed.

Figure 1 presents an excerpt from the metatheory for the travel domain that shows sample feature categories and associated measures. Each block defines a feature category, with the first line listing the name of the feature category followed by its constituent elements (i.e., its *extension*). Subsequent lines declare associated measures that apply to the category, along with a ranking of the features for each. For simplicity, we show only measures that completely order the features; however, partial orders are possible.

Just as measures can be employed to rank features (and hence operators with those features), they can also be employed to rank instances. For measures on instances, an ordered set of *measure values* is defined. For each measure, a given individual (optionally) can be assigned one of these values, thus inducing a partial order over instances. In the travel domain, for example, the measure AFFORDABILITY has the values (*Extravagant Expensive Moderate Inexpensive Cheap*), in increasing order from left to right. The individual `Ritz` of class `Hotel` has the AFFORDABILITY value *Extravagant*, while the individual `Motel6` of class `Motel` has the value *Cheap*; thus, `Motel6` ranks higher than `Ritz` with respect to AFFORDABILITY.

We define the *domain* of a measure to be the set of (partially) ordered values that are ranked by the measure. For measures defined over feature categories, the domain consists of the set of features that compose the feature category. For measures defined over instances, the domain is the set of measure values that can be assigned to instances. [1]

Vacation-Scope = {Overseas National Regional}
AFFORDABILITY: (Overseas National Regional)
TIME-EFFICIENCY: (Overseas National Regional)

Accommodation = {Hotel Motel Camp}
COMFORT: (Camp Motel Hotel)
AFFORDABILITY: (Hotel Motel Camp)

Transport-Media = {Air Land Water}
AFFORDABILITY: (Water Air Land)
TIME-EFFICIENCY: (Water Land Air)

Land-Transport-Mode = {Auto Bus Shuttle Taxi Train Limo}
AFFORDABILITY: (Limo Train Auto Taxi Shuttle Bus)
TIME-EFFICIENCY: (Bus Shuttle Auto Limo Taxi Train)
COMFORT: (Bus Shuttle Taxi Train Auto Limo)

Transit-Ownership = {Public Private}
COMFORT: (Public Private)
AFFORDABILITY: (Private Public)
TIME-EFFICIENCY: (Public Private)

Transit-Capacity = {Solo Shared}
COMFORT: (Shared Solo)
AFFORDABILITY: (Solo Shared)

Figure 1: Feature Categories and Associated Measures from the Travel Domain

Discussion

There is no 'correct' formulation of a metatheory: as with the underlying planning domain, its design involves an explicit modeling process. Individual user communities may be interested in different metatheoretic properties. For example, affordability may be significant when designing a system for students, but not for high-level business executives.

The value of the domain metatheory lies with its provision of a semantically grounded abstraction of the underlying planning domain. This abstraction is built on semantic linkage among different elements within a planning domain:

- Operators may share a common feature, feature category, or role.
- Instances, operators, or operators and instances may share measures.

As shown below, this linkage enables concise high-level descriptions of plan properties. This conciseness can be exploited both by a user seeking to direct a planning system, and by a system seeking to summarize plans or planning decisions for a user.

[1] A more general model for measures on instances would support functions defined over continuous values (e.g., AFFORDABILITY as the less-than relationship over the attribute *price*).

User Directability: Planning Advice

Increased user involvement with plan generation constitutes a critical research area for user-centric planning. One natural approach would be to provide users with the ability to direct the operations of the underlying planning technology by specifying desired plan attributes. For example, a traveler could express preferences for a particular trip (e.g., modes of transportation for various legs, specific airlines to use, accommodation requirements, and restrictions on costs for various aspects of the trip), with an automated planner constructing a solution that seeks to maximize satisfaction of those preferences.

Our *advisable planning* framework embodies this model of user directability for hierarchical task network (HTN) planning (Myers 1996; 1999; 2000). It enables users to provide advice to an automated planning system in order to influence the content of the solutions that are produced. Advice consists of session-specific recommendations on how tasks are to be accomplished, in terms of specific approaches to pursue and entities to employ. Advice is specified in a high-level language designed to be natural and intuitive for users, and then operationalized into constraints that direct the underlying planning technology during plan construction. The language for expressing advice builds on the features and roles from the domain metatheory, along with the language used to represent the base-level domain theory.[2]

Advice Specification

Advice comes in two forms: *role* and *method*. Role advice constrains the use of domain entities in solving tasks, while method advice further constrains the type of approach used. Both types are formulated in terms of *role-fills* and *activities*.

Role-fills are specifications of objects to be used to fill designated metatheory roles. A role-fill may name an explicit individual, or consist of a set of constraints designating required and prohibited attributes.

Activities constitute abstract characterizations of tasks relative to the underlying planning domain, and are defined in terms of metatheory features and role-fills. Within an HTN framework, an activity maps to a plan wedge whose root has the features specified by the activity, and whose descendants satisfy all stated role-fills.

Role Advice Role advice either prescribes or restricts the use of domain entities for filling certain capacities in the plan. Role advice is characterized by the template: `<Use/Don't Use> <object> in <role> for <context-activity>`. In general, role advice consists of one or more *role-fill* specifications, a *context activity*, and a *polarity* indicating whether the advice is prescribing or prohibiting the role-fill. The following directives provide examples of role advice:

> Stay in 3-star ensuite hotels while vacationing in Scotland.
>
> Layovers longer than 90 minutes are unacceptable for domestic flights.

The first directive imposes requirements on accommodations during vacations in a given region. The second prohibits flights with long layovers. Here, we use natural language renderings of advice to aid understandability, but it is easy to map to our structured activity/role-fill model. For the first example, the context activity is defined as tasks with feature Vacation, and with role Location filled by Scotland. The advice dictates that the filler for the role Accommodation be an object that belongs to the class 3-star-hotel and have ensuite facilities listed as an attribute.

Method Advice Method advice imposes restrictions on the approaches that can be used in solving a goal or class of goals. It is characterized by the template: `<Use/Don't use> <advised-activity> for <context-activity>`. Thus, method advice consists of context and advised activities, along with a polarity expressing prescription or proscription. For example:

> Find a package bike tour starting in Athens for the vacation in Greece.
> Don't fly between cities less than 200 miles apart.

The first piece of method advice declares that the approach used for a particular portion of the trip should have certain features (*i.e.*, Bike, Package) and role constraints (*i.e.*, Start-Location is Athens). The second specifies restrictions on the approach to be taken for solving a class of transport goals.

Advice Enforcement

Models for satisfaction of advice by an HTN plan are provided in (Myers 1996), along with an algorithm for *strict enforcement* that treats advice as hard constraints. The basic approach involves adding *advice constraints* that focus the planner on choices (for operators and variable instantiations) whose metatheoretic properties are compatible with the user-specified advice.

With strict enforcement, no solution is returned in the event that the full set of specified advice cannot be satisfied. In general, users may specify advice that is not satisfiable within the limits of the problem domain. Models for advice relaxation and corresponding relaxed enforcement algorithms can be found in (Myers 2000).

Generating Qualitatively Distinct Plans

Many real-world applications have solution-rich search spaces. Air campaign planning (Thaler & Shlapak 1995; Lee & Wilkins 1996) and travel planning (Linden, Hanks, & Lesh 1997) provide two examples. For these applications, it is not difficult to find a solution; rather, the challenge for human planners is to understand the range of available options in order to ensure informed selection of a solution. One means by which to help users with this task is to provide a set of *qualitatively distinct* plans distributed throughout the overall solution space, thus providing a range of exemplars.

[2] Measures could be used as the basis for more advanced forms of advice but have not been explored to date.

Current automated planning tools can readily generate different plans, for example through repeated runs with randomized choices at decision points. The differences among such plans, however, are difficult to extract and not necessarily semantically meaningful. Furthermore, different users may have their individual notions of what constitutes 'meaningful' differences. For example, a budget traveler might like to see options with a range of costs while the business traveler might like to see options that minimize transit time. Ideally, a system for generating qualitatively different plans would allow a user to specify dimensions along which he or she would like to see variation. Recent work on mixed-initiative, interactive, and advisable planning enables users to drive the process of generating qualitatively different plans (Ferguson & Allen 1998; Tate, Dalton, & Levine 1998; Myers 1996). With these frameworks, however, the user must be involved extensively in an ongoing role to articulate desired differences and to manage the space of options.

Our work on *metatheoretic biasing* leverages the domain metatheory described above to enable automated generation of qualitatively different plans (Myers & Lee 1999). In particular, the approach capitalizes on the structure inherent to measures to create *biases* that focus the planner on solutions with certain attributes. Biases are selected in a manner designed to produce solutions from different regions of the overall solution space.

To generate *n* plans, the method partitions the domains of selected metatheory measures into *n* intervals. Each interval from a measure is assigned to one of *n bias sets*, with the different sets being used to generate different plans.[3] For example, to generate two plans using the measures AFFORDABILITY and COMFORT, the domains of these measures would be split into two subsets, with one corresponding to a high valuation and the other a low valuation. Different algorithms can be used to assign the *n* intervals from each measure to the *n* bias sets: one might establish a first bias set with high affordability and low comfort biases and the second with low affordability and high comfort biases.

Biases are enforced during planning in a heuristic manner: rather than imposing hard constraints, choices available to the planner (namely, *operator selection* and *instance selection*) are ordered to reflect their distance (according to the metatheory measures) from the stated biases. Because the enforcement of biases prioritizes choices rather than filtering them, it does not restrict the set of plans that could be produced. As such, the biases can be viewed as *relaxable* constraints on plan generation.

By capitalizing on the semantic structure of domain metatheory measures, the biasing technique provides a low-cost mechanism for the automated generation of plans with meaningful semantic differences. In particular, it enables the generation of plans that are qualitatively different *by design*, rather than relying on random search through the syntactic plan space. The experimental results in (Myers & Lee 1999) validate the effectiveness of the method for reliably generating a range of plans with meaningful semantic differences.

The biasing approach for generating qualitatively different plans can be run without user input. However, users can optionally direct the planner into desired subregions of the overall plan space by designating specific measures and subintervals of those measures that should be used for bias generation. For example, users could indicate that they want to see plans within a range of cost and time values, while insisting on traveling by airplane (rather than train, boat, or car).

Plan Summarization and Comparison

The usability of automated planning technology would be enriched greatly by an automated plan summarization capability that could convey the key aspects of generated plans. A capability of this type would enable humans to feel more comfortable with the delegation of planning tasks to an automated system, knowing that the essence of the solution will be concisely communicated to them. Similarly, the ability to perform comparisons that highlight key differences between plans would help users in navigating through large solution spaces to identify plans that satisfy their needs.

There has been limited work to date on summarization and comparison of plans, with most efforts focused on methods grounded in syntactic characteristics (Young 1999). In contrast, the domain metatheory provides the potential to abstract from the details of plan structures to concise summarizations of key decisions within plans, and to important differences among plans.

We envision an approach that employs a suite of summarization and comparison techniques to identify regularities or exceptions relative to metatheoretic concepts. Several possibilities are described here.

Role/Feature Abstraction Role abstraction involves universal quantification over values selected to fill designated roles:

> *United was chosen as Carrier for all air transportation.*

Similarly, feature abstraction involves universal quantification over approaches selected to achieve similar goals:

> *Public-transport was used throughout for transit to and from airports.*

Measure Relativization This method, a generalization of role/feature abstraction, would enable summarization relative to measures:

> *Accommodations were chosen that ranked high on affordability.*

Role/Feature Differencing This method would identify key differences between two plans at the level of filling roles and selecting among operator choices with differing features.

> *United was chosen as the Carrier for air travel in Plan-1 while Delta was chosen in Plan-2.*
>
> *Plan-1 uses Hotels for accommodation while Plan-2 involves Camping.*

[3] This overview is necessarily simplified; (Myers & Lee 1999) provides a comprehensive description of the approach.

Application of these techniques for an entire domain metatheory could be expensive and would likely produce unfocused results. Adoption of some form of *summarization profile* that specifies aspects of the metatheory that interest a user most would enable customized, more succinct summarizations.

Knowledge Costs

Fully automated planning systems are brittle in that they require complete and correct formalizations of the domain. Small omissions or errors can result in the inability of systems to yield any solutions. Providing comprehensive domain information is time consuming and expensive, and represents a significant investment for each new application. In rich application domains, those models will need to grow and evolve over time, further exacerbating the knowledge acquisition problem.

Given the inherent cost and difficulties in building and maintaining complex knowledge bases, is it practical to advocate the inclusion of metaknowledge in planning domains? Cost notwithstanding, we believe that this metatheoretic information is essential for the usability of advanced planning tools. Furthermore, while development of a domain metatheory does increase the scope of knowledge acquisition required to develop a planning application, several factors mitigate the overall cost.

Reduced Sensitivity A domain metatheory is much less sensitive to errors than a base-level planning theory because its content does not impact the set of legal solutions. For this reason, metatheory inaccuracies may lead to unexpected solutions or incompleteness when exploited by the algorithms for advisability or generation of qualitatively distinct plans, but will not impact the basic planning process. For example, within our candidate metatheory, missing features could result in an inability to find solutions that maximize satisfaction of stated user advice; however, some solution will still be returned whenever the underlying problem is solvable. When generating qualitatively different plans, the distinction between certain semantic differences may be lost, resulting only in a missed opportunity to show the user an interesting dimension of plan variability.

Ease of Formulation We believe that a good domain metatheory should be a natural by-product of a principled approach to knowledge acquisition and modeling for planning knowledge bases. For example, when defining multiple operators that overlap in their applicability, users could be required to assign features to indicate how those operators differ at a semantic level. The metatheory elicitation process would be facilitated by the use of knowledge acquisition tools such as EXPECT (Gil & Swartout 1994), which impose structure on the modeling process.

User Initiative The adoption of metatheories will enable greatly increased user involvement in the planning process. This, in turn, will lessen the requirements for correctness and comprehensiveness imposed on the underlying knowledge bases, since the user could be expected to share responsibility for both planning knowledge and plan validity. For example, in cases where a planner is unable to find a solution, interaction with the user may help to identify problems with the knowledge base that incorrectly eliminated potential branches in the search space.

Conclusions

This paper has argued that an explicit domain metatheory will play a critical part in effective user-centric planning systems. This thesis has been supported by the presentation of a specific model for a metatheory, along with a description of its role in a system that supports both user directability of planning and the generation of qualitatively distinct plans. A proposal for plan summarization methods shows promise for additional applications of domain metatheories that would improve the usability of automated planning technology.

Acknowledgments This research has been supported by DARPA Contract F30602-97-C-0067 under the supervision of Air Force Research Laboratory – Rome.

References

Ferguson, G., and Allen, J. 1998. TRIPS: Towards a mixed-initiative planning assistant. In *Proceedings of the AIPS Workshop on Interactive and Collaborative Planning*.

Gil, Y., and Swartout, B. 1994. EXPECT: A reflective architecture for knowledge acquisition. In Burstein, M. H., ed., *ARPA/Rome Laboratory Planning and Scheduling Initiative Workshop Proceedings*. San Mateo, CA: Morgan Kaufmann.

Lee, T. J., and Wilkins, D. E. 1996. Using SIPE-2 to integrate planning for military air campaigns. *IEEE Expert* 11(6):11–12.

Linden, G.; Hanks, S.; and Lesh, N. 1997. Interactive assessment of user preference models: The Automated Travel Assistant. In *Proceedings of the Sixth International Conference on User Modeling*.

Mayer, M. A. 1997. A natural language interface for the Advisable Planner. Senior Honors Thesis, Stanford University.

Myers, K. L., and Lee, T. J. 1999. Generating qualitatively different plans through metatheoretic biases. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. AAAI Press.

Myers, K. L. 1996. Strategic advice for hierarchical planners. In Aiello, L. C.; Doyle, J.; and Shapiro, S. C., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96)*, 112–123. Morgan Kaufmann Publishers.

Myers, K. L. 1999. *User Guide for the Advisable Planner*. Artificial Intelligence Center, SRI International, Menlo Park, CA. Version 2.3.

Myers, K. L. 2000. Planning with conflicting advice. In *Proceedings of the Fifth International Conference on AI Planning Systems*.

Tate, A.; Dalton, J.; and Levine, J. 1998. Generation of multiple qualitatively different plans. In *Proceedings of the Fourth International Conference on AI Planning Systems*.

Thaler, D. E., and Shlapak, D. A. 1995. Perspectives on theater air campaign planning. Technical report, Rand Corporation.

Young, R. M. 1999. Cooperative plan identification: Constructing concise and effective plan descriptions. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. AAAI Press.