

Metatheoretic Plan Summarization and Comparison

Karen L. Myers

AI Center, SRI International
333 Ravenswood Ave.
Menlo Park, California 94025
myers@ai.sri.com

Abstract

We describe a domain-independent framework for plan summarization and comparison that can help a human understand both the key elements of an individual plan and important differences among plans. Our approach is grounded in the use of a *domain metatheory*, which is an abstract characterization of a planning domain that specifies important semantic properties of templates, planning variables, and instances. The metatheory provides a semantic framework for guiding the choice and description of concepts used in summarizing and comparing plans, thus enabling results that are grounded in semantically significant concepts rather than syntactic constructs whose meaning or import is unclear. We define three specific capabilities grounded in the metatheoretic approach: (a) summarization of an individual plan, (b) comparison of pairs of plans, and (c) analysis of a collection of plans. Use of these capabilities within a rich application domain shows their value in facilitating the understandability of complex plans by a user.

Introduction

AI planning technology is being applied in increasingly more challenging application domains, resulting in the generation of plans with rich sophistication and complexity. In these complex domains, it is generally the case that a wide range of solutions is possible; part of the challenge for a human decision maker is to analyze the relative merits of various candidates before deciding on a final option. Given these advances, the development of tools that can help users understand complex plans and tradeoffs among them presents an important technological challenge.

In this paper, we describe an approach to plan summarization and comparison that is designed to help a human understand both the key elements of an individual plan and important differences among alternative plans. Our approach is grounded in the use of a *domain metatheory*. The domain metatheory is an abstract characterization of a planning domain that specifies important semantic properties of templates, planning variables, and instances. The abstraction provides the means to describe and compare plans in high-level, semantically meaningful terms.

Previous work on plan summarization and explanation has been grounded in methods that are tightly linked to either the syntactic characteristics of a plan's structure or the underlying reasoning processes used to generate it. Such approaches suffer from the problem that these structures and processes match the system's conceptualization of the domain rather than that of the user. As such, their outputs have limited explanatory value.

The concept of the domain metatheory was introduced originally to provide a language that would enable a user to *advise* a planning system, without requiring detailed knowledge of its internal workings [Myers 1996]. Advice, which describes high-level characteristics of desired solutions, is operationalized into structures and mechanisms that guide an automated planning system at runtime. Subsequently, the metatheory was also used as the basis for generating *qualitatively different* plans, by using structure within the metatheory to direct a planning system toward solutions with distinct semantic traits [Myers & Lee 1999].

A key insight underlying the work reported here is that the metatheory can be used as the basis for identifying and communicating important explanatory information about a plan. In particular, the metatheory provides a semantic framework for guiding the choice of concepts used in summarizing and comparing plans. The resultant comparisons and summaries are thus grounded in semantically significant concepts rather than syntactic constructs whose meaning or import are unclear.

Within our metatheoretic framework, we define techniques for (a) summarization of an individual plan, (b) comparison of pairs of plans, and (c) analysis of a collection of plans. These techniques look for regularities or interesting exceptions relative to key aspects of the domain metatheory. For example, a metatheory *role* corresponds to an important actor or object within a plan. In comparing two plans, one interesting dimension to consider is whether the plans fill key roles in different ways. Two plans may be similar in structure but one uses a cheap and abundant resource while the other relies on an expensive and more exotic resource.

Our approach embodies the spirit of *reconstructive explanation* [Wick and Thompson 1992], whereby an explanation is produced not by the system's own internal knowledge, but by a separate store of explanatory knowledge designed specifically with the user in mind. We believe that this style of approach is critical to ensuring that the results are of value to a user, rather than driven by the syntactic structure of the plan.

The plan summarization and comparison methods have been implemented within the PASSAT mixed-initiative

planning framework [Myers et al. 2002]. To assess their effectiveness in facilitating user understandability of complex plans, we applied the methods to a test suite from an extensive special operations domain. This usage shows that our techniques can help a user understand subtle aspects of individual plans, important differences among plans, and the structure of the overall solution space.

Domain Metatheory

Our plan summarization and comparison work assumes a *hierarchical task network* (HTN) paradigm for representing plans, similar to that described in [Erol et al. 1994]. An HTN domain theory consists of four basic types of element: *individuals* corresponding to real or abstract objects in the domain, *relations* that describe characteristics of the world, *tasks* to be achieved, and *templates* that describe available means for achieving tasks. (Templates are alternatively referred to as *methods* or *operators* in the literature.) We assume a type hierarchy for terms within the domain model. Thus, each individual has an associated type $Type(v)$, and there is a unique most-specific supertype $MinSupertype(V)$ defined for any set of individuals V .

A *domain metatheory* defines semantic properties for domain theory elements that abstract from the syntactic details of the domain knowledge. The metatheory for plan summarization and comparison is similar to that introduced for the work on advisable planning. To support summarization and comparison, however, we introduce a few extensions and refinements that provide a somewhat richer and more structured framework. The main metatheoretic concepts that we use are *template features*, *task features*, and *roles*.

Template Features

A *template feature* designates a characteristic of a template that distinguishes it from other templates that could be applied to the same task. For example, among templates that could be applied to a transportation task, there may be an air-based template that is *fast* but *expensive* with a land-based alternative that is *slow* but *cheap*. Although the two templates are functionally equivalent in that they accomplish the same task, they differ significantly in their approaches. Template features provide the means to distinguish among such functionally equivalent alternatives by capturing these characteristics explicitly.

We model template features in terms of a *feature category* (e.g., *COST*) and a *feature value* (e.g., *expensive*). Feature values are drawn from a predefined set that constitutes the *domain* of the feature category. For this work, we require that the domain for a template feature be totally ordered (that need not be true in general).

We say that a template feature f with value v *occurs in plan* P iff there is some template T applied to a task t in P such that T has the feature f with value v . In general, a plan may have multiple occurrences of a given template feature that cut across templates used to accomplish a

range of tasks. Different occurrences may have different values associated with them; duplication of values is also possible. The term $TemplFeatureInsts(f,P)$ denotes the collection of values (including duplicates) for occurrences of template feature f in plan P .

The value of template features for plan summarization and comparison is that they provide the means to identify, abstract, and contrast important evaluational properties of different strategies, such as speed or cost. In particular, template features can be used as a kind of ‘quick and dirty’ proxy for deeper, more significant evaluations of a plan.

Task Features

Task features capture important semantic attributes of a task. As with template features, task features are modeled in terms of a feature category and feature value. Here, we focus on task features that designate *types* of activities, and restrict categories to have the domain `[false true]`. For example, there may be several types of reconnaissance task: satellite reconnaissance, ground reconnaissance, and aircraft reconnaissance. Each of these tasks can be assigned the feature *RECON* with value *true*, thus providing a mechanism for abstracting over that set of tasks. (A similar sort of grouping could be achieved through the use of a class hierarchy for tasks.)

We say that a plan P *has a task feature* f iff some task t in P has the feature f with value *true*. The term $TaskFeatures(P)$ denotes the set of task features for P .

Roles

A *role* describes a capacity in which an individual is used within a template or task; it maps to a template or task variable. For instance, a template for transporting materials may contain variables `location.1` and `location.2`, with the former corresponding to the *START* role and the latter the *DESTINATION* role for the move. Roles provide a semantic basis for describing the use of individuals within templates and tasks that abstracts from the details of specific variable names. Roles also provide the means to reference a collection of semantically linked variables that span different templates and tasks (i.e., *START* roles may occur in multiple templates and tasks).

We say that a *role* r with *fill* v *occurs in plan* P iff either:

- there is some task $t(a_1, \dots, a_n)$ in P such that t has the declared role r for its i^{th} argument, and $a_i = v$, [Task Role] or
- some template T with role r declared for local variable x_i is applied to a task $t(a_1, \dots, a_n)$ in P , and x_i is bound to v [Template Role]

The term $Roles(P)$ denotes the set of roles that occur in plan P , while $RoleFills(r,P)$ denotes the collection of values (including duplicates) that occur as fills for role r in plan P .

Experimental Framework

We evaluated the effectiveness of our plan summarization and comparison techniques on a suite of nine test plans drawn from a *special operations forces* (SOF) domain. (This domain was created as part of an earlier project focused on mixed-initiative planning technology.) The SOF domain constitutes a sizable and rich test environment for evaluating our work on plan summarization and comparison: the base-level domain contains 65 predicates modeling key world properties, more than 100 tasks, and more than 50 templates spanning a hierarchy of five abstraction layers.

The original SOF domain included a limited metatheory designed to showcase advice-taking within the PASSAT system [Myers et al. 2002]. For this work, we extended the domain to include a fairly comprehensive metatheory with 13 template features, 12 task features, and more than 75 roles. The task features (see Figure 3) use the domain [false true]; the template features (see Figure 4) use the domain [low medium high].

The test plans address the high-level task of extracting a set of hostages held by a guerilla team in an urban environment. More specifically, this task requires rescuing a set of hostages being kept at Mogadishu-Town-Hall using forces based at Riyadh Airport, and then evacuating the hostages to a safe haven at Riyadh Stadium.

The SOF domain includes a number of templates that reflect different strategies for rescuing the hostages. Variations among solutions result from three sources. The first is whether the plan contains certain types of strategic and tactical activities; depending on a given situation, the commander may or may not decide to include such activities within the plan. For example, while it is not necessary to create diversions to distract the guerillas, doing so may be desirable in some circumstances. The second relates to the selection of resources to be used. In some cases, for example, it may be appropriate to use satellites to gather intelligence information while in others it may be preferable to rely on ground forces. The third relates to decisions about key parameters within a plan, such as where to establish a forward base or the drop point for inserting the assault team.

Figure 1 summarizes the nine test plans used in our evaluation. These plans were created by the developer of the SOF domain knowledge, through a combination of manual and semiautomated methods within PASSAT.¹ The plan developer was asked to create a core set of plans reflecting a representative set of strategic alternatives that a SOF commander might consider. Additionally, he was asked to create variants of the core plans by making a few key strategic changes that might correspond to handling contingencies in different ways. Given that variants of this type are commonly made in practice, we were interested in

¹ The plan developer was not involved with the research on plan summarization and comparison described in this paper. As such, the plans provide an objective test suite for evaluating the reported work.

determining how well our plan comparison techniques would be able to recognize the differences among them.

Plan Identifier	Description
<i>tiny-plan-a</i>	Very simple plan without security or support
<i>tiny-plan-b</i>	Variant on <i>tiny-plan-a</i> that uses a different type of rescue force
<i>small-plan-a</i>	Basic solution that includes reconnaissance and combat search and rescue
<i>small-plan-b</i>	Variant on <i>small-plan-a</i> that uses the same high-level strategy but differs in the lower-level realization of parts of it
<i>medium-plan-a</i>	Broadly similar to the small plans but involves refueling
<i>medium-plan-b</i>	Broadly similar to the small plans but with suppression of enemy air defenses (SEAD) activities
<i>large-plan-a</i>	Extensive plan with significant reconnaissance and support activities as well as a diversion from the main assault
<i>large-plan-b</i>	Variant on <i>large-plan-a</i> that provides increased fire support and SEAD
<i>large-plan-c</i>	Variant on <i>large-plan-a</i> with a different style of diversion

Figure 1. Summary of Test Plans

Plan Summarization

The roles and features of the metatheory provide a semantic basis for summarizing key properties of a plan. In particular, a description of how a plan fills its roles and the features that it possesses can provide valuable insight into the structure, strengths, and weaknesses of a plan.

Task Features

Task features provide a succinct summary of key activity types within a plan. In particular, such a summary can inform the user that a given plan does or does not contain critical activities such as reconnaissance or fire support.

Template Features

Template features provide a different perspective on a plan, as they designate plan characteristics that have more of an evaluational nature (e.g., cost, speed). Template features can be applied in multiple contexts within a plan, with different occurrences yielding different values. This variation reflects the fact that, for example, a given plan may use an inexpensive reconnaissance operation but an expensive rescue strategy. To enable plan-level summarization of the property represented by a template feature, we introduce the concept of *template feature value* for a feature f and plan P , denoted by $TemplFeatureValue(f,P)$. This value is defined to be the average of the values for all occurrences of f within P .

Definition 1 [Template Feature Value for a Plan] The *template feature value* for feature f and plan P is defined by $TemplFeatureValue(f,P)=Avg(TemplFeatureInsts(f,P))$.

The use of a qualitative domain for template features (as in the SOF application) introduces a complication in computing $TemplFeatureValue(f,P)$, as it is necessary to support qualitative averaging. To this end, we require for each qualitative feature f a surjective, order-preserving mapping θ_f from a designated interval of the reals $Interval(f)$ to the domain of the feature f : $\theta_f: Interval(f) \rightarrow Domain(f)$. Variation in the ‘closeness’ of values in $Domain(f)$ can be achieved by appropriate definitions of θ_f .² With this mapping, we define the average of a set V of qualitative template feature values as follows:

$$Avg(V) = \theta_f \left(\frac{\sum_{v \in V} \theta_f^{-1}(v)}{|V|} \right)$$

Roles

A description of how roles are filled within a plan can provide a concise summary of what resources are used and how, as well as key parameters to a plan (e.g., the choice of location for a forward base). Furthermore, it is possible to search for patterns in the filling of roles. So, for example, it may be useful to know that only satellites are used as reconnaissance assets, or that all transport of troops is through the use of helicopters of a particular type. We refer to such patterns as *uniformities* in the filling of roles. Here, we define two specific types of uniformity for role fills, oriented around *values* and *types*.

Definition 2 [Value Uniformity in Role Fills] A plan P *uniformly fills a role r with value c* iff $|RoleFills(r,P)| > 1$ and $v \in RoleFills(r,P)$ implies that $v = c$.

Type uniformity depends on the declaration of a type $Type(r)$ for a given role r , which indicates that all fills for role r must be of that type. Type uniformity becomes interesting when some proper subtype of $Type(r)$ generalizes all fills for a given role. For example, it can be useful to note that only satellites are used for reconnaissance within a given plan, although other types of assets (e.g., ground forces) are possible.

Definition 3 [Type Uniformity in Role Fills] A plan P *uniformly fills a role r with a type T* iff $|RoleFills(r,P)| > 1$, T is a proper subtype of $Type(r)$, and every fill value $v \in RoleFills(r,P)$ is of type T .

Value and type uniformity for roles constitute generic, domain-independent mechanisms for generalizing a collection of role fills. For a given domain, it may be

² For the SOF metatheory, every template feature has the domain [low medium high], the interval [0,1], and the mapping function $\theta: [0,1] \rightarrow [low\ medium\ high]$ where θ^l is distributed linearly across [0,1]: low maps to 0, medium to 0.5 and high to 1.

- * (Rescue-Hostage Mogadishu-Town-Hall Riyadh-Airport Riyadh-Stadium)
- * (Rescue-And-Recover Riyadh-Airport Mogadishu-Town-Hall Riyadh-Stadium)
- * (Recon Mogadishu-Town-Hall)
- * (Infiltrate Green-Oda-2 Ankara-Airport Mogadishu-Town-Hall)
- * (Produce-Landing-Plan Mh-60-G-Pave-Hawk-2)
- * (Produce-Air-Movement-Plan Mh-60-G-Pave-Hawk-2)
- * (Produce-Loading-Plan Green-Oda-2)
- * (Produce-Aircraft-Bump-Plan Green-Oda-2)
- * (Load Green-Oda-2 Mh-60-G-Pave-Hawk-2)
- * (Fly Mh-60-G-Pave-Hawk-2 Ankara-Airport Mogadishu-Stadium)
- * (Drop Green-Oda-2 Mh-60-G-Pave-Hawk-2 Mogadishu-Town-Hall)
- * (Depart Mh-60-G-Pave-Hawk-2 Mogadishu-Stadium)
- * (Establish-Observation-Post Green-Oda-2 Mogadishu-Town-Hall)
- * (Exfiltrate Green-Oda-2 Mogadishu-Town-Hall Mogadishu-Building4)
- * (Fly Uh-60a-2 Mogadishu-Town-Hall Mogadishu-Building3)
- * (Load Green-Oda-2 Uh-60a-2)
- * (Depart Uh-60a-2 Mogadishu-Building3)
- * (Provide-Fire-Support Mogadishu-Town-Hall)
- * (Take-Off Ch-53e-Super-Stallion-1 Addis-Ababa-Airport)
- * (Fly Ch-53e-Super-Stallion-1 Addis-Ababa-Airport Mogadishu-Town-Ha)
- * (Place-On-Station-Fire-Support Ch-53e-Super-Stallion-1 Mogadishu-T)
- * (Fly Ch-53e-Super-Stallion-1 Mogadishu-Town-Hall Addis-Ababa-Airp)
- * (Land-At Ch-53e-Super-Stallion-1 Addis-Ababa-Airport)
- * (Provide-Csar-Coverage Csar-Team-2 Mogadishu-Town-Hall)
- * (Prepare Csar-Cl-A)
- * (Take-Off Csar-Cl-A Baidoa-Stadium)
- * (Fly Csar-Cl-A Baidoa-Stadium Mogadishu-Town-Hall)
- * (On-Station Csar-Cl-A Mogadishu-Town-Hall)
- * (Provide-Fire-Support Mogadishu-Town-Hall)
- * (Take-Off Ah-100-1 Balikesir-Stadium)
- * (Fly Ah-100-1 Balikesir-Stadium Mogadishu-Town-Hall)
- * (Place-On-Station-Fire-Support Ah-100-1 Mogadishu-Town-Hall)
- * (Fly Ah-100-1 Mogadishu-Town-Hall Balikesir-Stadium)
- * (Land-At Ah-100-1 Balikesir-Stadium)
- * (Provide-Sead Sead-1 Ad-Dammam-Stadium Mogadishu-Town-Hall)
- * (Prepare Sead-1)
- * (Take-Off Sead-1 Ad-Dammam-Stadium)
- * (Fly-To Sead-1 Mogadishu-Town-Hall)
- * (Infiltrate Orange-Oda-1 Riyadh-Airport Mogadishu-Town-Hall)
- * (Fly-Commercial Aa7864 Orange-Oda-1 Riyadh-Airport Mogadishu-Town-Ha)
- * (Storm Orange-Oda-1 Mogadishu-Town-Hall)
- * (Exfiltrate Orange-Oda-1 Mogadishu-Town-Hall Riyadh-Stadium)
- * (Fly-Commercial Aa201 Orange-Oda-1 Mogadishu-Town-Hall Riyadh-Stadium)
- * (Provide-Fire-Support Mogadishu-Town-Hall) [Fire-Support-Naval]
- * (Station Yorktown Mogadishu-Town-Hall)
- * (Provide-Csar-Coverage Csar-Team-2 Mogadishu-Town-Hall)
- * (Prepare Uh-601-1)
- * (Take-Off Uh-601-1 Bihac-Stadium)
- * (Fly Uh-601-1 Bihac-Stadium Mogadishu-Town-Hall)
- * (On-Station Uh-601-1 Mogadishu-Town-Hall)

Figure 2. Task Decomposition View of Plan *medium-plan-b*

appropriate to introduce domain-specific generalization mechanisms. For example, in domains where locations play a significant role, it might be useful to generalize based on geographic proximity, or co-location within some designated geographic area (e.g., all air assets are pulled from bases in the same region).

Sample Plan Summary

To illustrate the value of metatheory-based plan summarization, consider the summary of the test plan *medium-plan-b* shown in Figure 2. The figure presents a task decomposition view of the plan that highlights its hierarchical structure; for simplicity, temporal sequencing information among activities has been omitted.

As can be seen, the plan is sufficiently complex that its key strategic elements are not readily apparent. Rather, some form of analysis tool is required to understand the plan. Figure 3 summarizes the task features within this plan while Figure 4 summarizes the template features and their normalized values. Figure 5 summarizes key role fills for the plan.

The summary of task features in Figure 3 makes it easy to identify the key strategic elements of the plan. The features RESCUE-AND-RECOVER and RESCUE derive from the fact that the plan describes a rescue-and-recover operation; these features are common to every plan in the test suite. At a lower level, we can see that this particular solution includes components for combat search and rescue support (CSAR-SUPPORT), fire support (FIRE-SUPPORT), reconnaissance (RECON), and suppression of

enemy air defenses (SEAD). These components are optional, as not every solution contains them.

Task Feature	Value
CSAR-SUPPORT	true
DIVERSION	false
EVACUATION	false
FIRE-SUPPORT	true
PARACHUTE	false
RECON	true
REFUELING	false
RESCUE	true
RESCUE-AND-RECOVER	true
SEAD	true
SECURITY	false
SUPPORT	true

Figure 3. Task Features for Plan *medium-plan-b*

Template Feature	Plan Value
BLUE-CASUALTY-RISK	medium
COLLATERAL-DAMAGE	low
COORDINATION-COMPLEXITY	medium
COVERTNESS	medium
DURABILITY	low
FORCE-FATIGUE	medium
FORCE-FOOTPRINT	medium
FORCE-INTEGRITY	medium
INFORMATION-QUALITY	high
LANDING-ZONE-PREP	low
ROBUSTNESS	medium
SPEED	medium
VULNERABILITY-GROUND-FIRE	high

Figure 4. Template Features for Plan *medium-plan-b*

Role	Fill Values
ASSAULT-FORCE	green-oda-2 orange-oda-1
FORCE	orange-oda-1 (2) green-oda-2 (2)
OBSERVATION-FORCE	green-oda-2

Roles related to Strategic Decisions about Locations

Figure 5. Force Usage Roles in *medium-plan-b*

Value-based Role Uniformity

Role	Value	Count
CSAR-LOCATION	mogadishu-town-hall	2
FIRE-SUPPORT-LOCATION	mogadishu-town-hall	5
FORWARD-POINT	riyadh-airport	3
INFIL-DESTINATION	mogadishu-town-hall	2
SEAD-AIRCRAFT	sead-1	2

Type-based Role Uniformity

Role	Role Type	Fill Types	Min. SuperType
EXFIL-ASSET	Asset	Commercial-flight Helicopter	Air-asset
INFIL-ASSET	Asset	Commercial-flight Helicopter	Air-asset
TRANSPORT-ASSET	Asset	Sead-aircraft Helicopter	Air-asset

Figure 6. Role Uniformities in Plan *medium-plan-b*

The template features in Figure 4 summarize key evaluational qualities of the plan. Desirable qualities include the fact that the expected quality of information underlying the plan is high, while expected collateral damage is low. On the negative side, there is high vulnerability to ground fire.

More than 30 roles occur in the plan *medium-plan-b*, some of which have multiple fills. Typically, a user would not choose to view all roles and their fills at once. Rather, at a given point in time he would be interested in knowing about a subset of these roles as he focuses on certain aspects of the plan. So, for example, a user interested in understanding the high-level strategy of a plan may concentrate on a subset of roles related to key strategic decisions, while a user interested in asset usage may concentrate on roles related to resource utilization.

Figure 5 displays the role fills related to force usage for the plan *medium-plan-b*. For fill values that occurred more than once for a given role, the number of occurrences is noted in parentheses. This summary makes it easy to see that only Green and Orange teams are used in the plan; both are used in assault roles while the Green team is also used in a reconnaissance capacity as an observation force.³

Figure 6 summarizes value-based and type-based role uniformities for the plan *medium-plan-b*. For value-based uniformity, the summary indicates the role, the fill value, and the number of occurrences. For type-based uniformity, the summary indicates the role, its type, the types of the fill values, and the most specific type that generalizes the fill values. The information on type-based uniformity is particularly useful here as it highlights the exclusive use of air assets for many key functional roles within the plan.

Plan Comparison

Our approach to comparing plans is grounded in two techniques: *feature differencing* and *role differencing*. These techniques can be useful both in terms of identifying subtle variations in similar plans, and understanding larger differences in more varied plans.

Feature Differencing

As noted above, features correspond to high-level semantic characteristics of tasks (for task features) and strategic or evaluational qualities (for template features).

Task features provide a semantic summary of key activities within a plan. Task feature differencing, which involves a comparison of task features within two plans, provides a snapshot of how the two plans differ in their key task types. This type of capability can enable a user to see easily that, for example, one plan contains reconnaissance capabilities while another does not.

³ The color in a force name is significant: colors denote units with specific skills and capabilities. For the sake of brevity, we omit detailed descriptions of the qualities associated with the various force colors.

Template feature differencing compares the normalized template feature values for two different plans in order to identify significant variations. This form of differencing makes it easy to see, for example, that one plan trades risk for increased complexity relative to another plan.

Role Differencing

Role differencing looks at variabilities in how two plans fill their roles. This type of comparison can shed insight on key differences in strategic decisions (e.g., *Where are the hostages to assemble?*) and resource usage (e.g., *What types of reconnaissance asset are used?*).

Figure 7 presents a categorization of the ways in which the fill values for a given role in two plans can differ. There, V_1 and V_2 designate sets of fill values for a role from which duplicates have been removed. It is assumed that $V_1 \neq V_2$ and that both V_1 and V_2 are nonempty. The first three entries cover situations where V_1 and V_2 are disjoint; the last two cover situations where V_1 and V_2 overlap.

The category *different single valued*, although just a special case of *disjoint types*, is useful for identifying differences in key strategic decisions for a plan. For example, for the role ASSAULT-FORCE, the plan *tiny-plan-a* uses orange-oda-2 while the plan *tiny-plan-b* uses green-oda-1. This difference is important, as noted above, because orange and green forces have significantly different core capabilities. The category *different single valued* is especially useful when the role appears exactly once within each of the two plans being compared; such a role often designates some critical parameter choice.

The category *disjoint types* requires both that the most specific supertype of the role-fill values in the two plans be different, and that neither be a subtype of the other. As an example, the plan *small-plan-a* uses only helicopters of type CSAR-HELICOPTER-CLASS-1 for combat search and rescue while the plan *large-plan-b* uses helicopters of type CSAR-HELICOPTER-CLASS-2. The category *disjoint multivalued* defines an even weaker condition, requiring only that the fill values for the two plans be different.

For overlapping values, the strongest condition is *restricted subtype*, which indicates that the most specific supertype of one collection of values is a subtype of the most specific supertype of the other collection. For example, the plan *large-plan-b* uses only assault forces of type SOF-UNIT while the plan *large-plan-c* uses a more general set of forces (of type FORCE-COMPOSITION); in contrast, the plan *large-plan-b* uses a range of watercraft to fill the role WATER-ASSET while the plan *large-plan-c* uses only values of type BOAT. *Restricted subset* weakens the *restricted subtype* condition to require only that one collection of values be a subset of the other.

Role differencing can provide insights into fundamental differences between plans, as illustrated in the next section. However, there are limitations to its usefulness.

First, the significance of role differences may be difficult to gauge in isolation. So, while the decision to use force Green-ODA-1 rather than Orange-ODA-2 to fill the ASSAULT-FORCE role is significant, as those two units

have markedly different capabilities, the difference between the forces Green-ODA-1 and Green-ODA-2 is insignificant as they have the same fundamental capabilities. This problem can be addressed by introducing a notion of ‘semantic distance’ between individuals to help identify differences that are significant.

Second, the utility of role differencing can decrease as plan size grows due to increased numbers of occurrences of a role that are not closely related. (For example, it is possible to create larger SOF plans by introducing multiple assault prongs involving forces inserted at different drop locations; doing so leads to duplication of roles used in very different contexts.) Thus, while unrestricted role differencing can be useful in small- to medium-sized plans, larger plans would benefit from some scheme to contextualize role fills to certain portions of the plan.

Disjoint: $V_1 \cap V_2 = \{\}$

Different single valued: $V_1 \cap V_2 = \{\} \wedge |V_1|=|V_2|=1$

Disjoint types: $\text{MinSupertype}(V_1) \neq \text{MinSupertype}(V_2)$
 $\wedge \text{MinSupertype}(V_1) \not\subset \text{MinSupertype}(V_2)$
 $\wedge \text{MinSupertype}(V_2) \not\subset \text{MinSupertype}(V_1)$

Disjoint multivalued: $V_1 \cap V_2 = \{\} \wedge (|V_1|>1 \vee |V_2|>1)$

Overlapping: $V_1 \cap V_2 \neq \{\}$

Restricted subtype: $\text{MinSupertype}(V_1) \subset \text{MinSupertype}(V_2)$
 $\vee \text{MinSupertype}(V_2) \subset \text{MinSupertype}(V_1)$

Restricted subset: $V_1 \subset V_2$

Figure 7. Categories of Role-fill Differences

Sample Plan Comparison

Figure 8 displays the results of applying our metatheoretic plan comparison techniques to the test plans *medium-plan-a* and *medium-plan-b*.

In looking at the results of task feature differencing, two fundamental differences emerge: *medium-plan-a* contains refueling activities and *medium-plan-b* does not, while *medium-plan-b* contains SEAD (suppression of enemy air defense) activities and *medium-plan-a* does not.

For template feature differencing, there is some variation among expected values for key evaluation criteria. Given the use of a fairly coarse-grained set of qualitative values for template feature domains in the SOF metatheory, the scope for variability is limited. A more fine-grained set of values would enable more precise comparisons.

Role differencing highlights some interesting variations in the use of resources between the two plans. Both plans include reconnaissance operations, but *medium-plan-a* relies on a satellite (*satellite-1*) while *medium-plan-b* makes use of a ground force (*green-oda-2*) as the asset used to perform the reconnaissance (see the table *Different Single Valued*). This distinction is important because the nature and quality of the intelligence that can be obtained with these two assets is markedly different. Different types of infiltration, exfiltration, fire support and transport

Task Feature Differencing:

Task Features in *medium-plan-a* but not in *medium-plan-b*:
REFUELING

Task Features in *medium-plan-b* but not in *medium-plan-a*:
SEAD

Template Feature Differencing:

Template Feature	<i>medium-plan-a</i>	<i>medium-plan-b</i>
DURABILITY	medium	low
FORCE-FATIGUE	high	medium
FORCE-INTEGRITY	high	medium
LANDING-ZONE-PREP	medium	low

Role Differencing:

Different Single Valued

Role	<i>medium-plan-a</i>	<i>medium-plan-b</i>
RECON-ASSET	satellite-1	green-oda-2

Disjoint Multivalued

Role	Values for <i>medium-plan-a</i>	Values for <i>medium-plan-b</i>
ASSET	csar-c2-b tanker-1	uh-601-1 yorktown sead-1 csar-c1-a
EXFIL-ASSET	uh-601-1	aa201 uh-60a-2
FIRE-SUPPORT-ASSET	av-8b- harrier-ii-a	yorktown ah-100-1 ch-53e-super- stallion-1
INFIL-ASSET	uh-601-2	mh-60-g-pave-hawk- 2
TRANSPORT-ASSET	tanker-1 av-8b- harrier-ii-a uh-601-1 uh-601-2	sead-1 uh-60a-2 mh-60-g-pave-hawk- 2

Restricted Subtype

Role	Type for <i>medium-plan-a</i>	Type for <i>medium-plan-b</i>
ASSAULT-FORCE	ORANGE-UNIT	SOF-UNIT
INFIL-POINT	BUILDING	POINT-LOCATION
INFIL-TEAM	ORANGE-UNIT	SOF-UNIT
LANDING-LOCATION	AIRPORT	POINT-LOCATION

Restricted Subset

Role	Values for <i>medium-plan-a</i>	Values for <i>medium-plan-b</i>
EXFIL-POINT	mogadishu-town- hall	mogadishu-town- hall mogadishu- building4
INFIL-START	riyadh-airport	riyadh-airport ankara-airport

Figure 8. Comparison of *medium-plan-a* and *medium-plan-b*

assets are used, each with their individual strengths and weaknesses (see the table *Disjoint Multivalued*).

The tables *Restricted Subtype* and *Restricted Subset* show that plan *medium-plan-a* is much less diverse than plan *medium-plan-b*, since it uses more restricted sets of entities to fill a number of key roles (i.e., ORANGE-UNIT is

a subtype of SOF-UNIT, BUILDING and AIRPORT are subtypes of POINT-LOCATION).

Overall, a user looking at the style of comparison in Figure 8 could quickly grasp the fundamental differences in strategy and resource usage between the two plans. Detailed examination of the plans themselves shows that there are additional differences in terms of unimportant low-level activities used to accomplish higher-level tasks and resource allocation. However, the metatheoretic comparison hides these nonessential differences.

Plan Space Analysis

We define two capabilities grounded in the domain metatheory for reasoning about a collection of plans: *identifying unique characteristics of a plan*, and *identifying maximally different plans*.

Identifying Unique Characteristics of a Plan

The metatheoretic differencing capabilities defined in the previous section can be used to identify three useful distinguishing characteristics of a plan P relative to a set S of candidate solutions.

1. Unique task features:

- P has a task feature not found in any other P' in S
- P lacks a task feature found in all P in S

2. Unique normalized template features: P has a normalized template feature value that differs from the value for all other solutions in S . This situation is especially interesting when all other plans share a common value for that template feature; in that case, the template feature for plan P is called *exceptional*.

3. Differing role fills: There is a role common to all plans for which some fill value in P does not occur as a fill value in other solutions in S .

Figure 9 summarizes the unique task features and normalized template features within our suite of test plans; they occurred in the plans *small-plan-b* and *medium-plan-a*. (We have not yet implemented the ability to look for differing role fills.)

The plan *small-plan-b* differs from all others in the test suite on the normalized value for the template feature BLUE-CASUALTY-RISK. In particular, its value for that feature is low while the other plans have value medium.

Plan: *small-plan-b*

Has Exceptional Template Feature Values:
BLUE-CASUALTY-RISK: low; all others medium

Plan: *medium-plan-a*

Has Unique Task Features: REFUELING
Has Exceptional Template Feature Values:
LANDING-ZONE-PREP: medium; all others low
Has Unique Template Feature Values:
DURABILITY: medium
FORCE-FATIGUE: high

Figure 9. Unique Features in the Test Suite

The plan *medium-plan-a* has several unique characteristics relative to the other plans in the test suite. First, it is the only plan with the task feature REFUELING; hence, no other plans in the test suite include refueling operations. Second, while the plan *medium-plan-a* has the normalized value `medium` for the template feature LANDING-ZONE-PREP, all other plans have the value `low`. Finally, the plan *medium-plan-a* differs from the other plans in the values for template features DURABILITY and FORCE-FATIGUE; in those cases, however, there is no common value for the remaining plans in the test suite.

Maximally Different Plans

For many applications, a human planner will want to explore a range of plans that embody qualitatively different solutions [Tate et al. 1998; Myers & Lee, 1999]. Such exploration can be useful both in terms of helping the user understand fundamental tradeoffs that are inherent to the domain, and identifying ‘out of the box’ solutions that he may not normally consider.

Our metatheoretic differencing techniques can be used to identify plans that are semantically far apart from each other, and hence are likely to have significant qualitative differences. To that end, we define a concept of *distance* between plans that builds on the concepts of *task feature*, *template feature*, and *role distance* between plans.

Task Feature Plan Distance

Task feature distance is a normalized form of Hamming distance for the task features within the plans. In particular, it is defined to be the ratio of the number of task features that appear in one but not both plans to the number of features that appear in either plan.

Definition 4 [Task Feature Plan Distance] The *task feature distance* between plans P_1 and P_2 , denoted by $TaskFeatureDist(P_1, P_2)$, is defined by

$$TaskFeatureDist(P_1, P_2) = \frac{|TaskFeatures(P_1) - TaskFeatures(P_2)| + |TaskFeatures(P_2) - TaskFeatures(P_1)|}{|TaskFeatures(P_1) \cup TaskFeatures(P_2)|}$$

Template Feature Plan Distance

Template feature distance for a pair of plans is defined to be the average distance between the values of those features that are common to both plans, normalized with respect to the range of possible values for the features. Let $TemplateFeatures(P)$ denote the set of template features that occur in plan P , and $FDist(f, P_1, P_2)$ the distance between values for template feature f in plans P_1 and P_2 .

Definition 5 [Template Feature Plan Distance] The *template feature distance* between plans P_1 and P_2 , denoted by $TemplFeatureDist(P_1, P_2)$, is defined by

$$TemplFeatureDist(P_1, P_2) = \frac{\sum_{f \in F^\cap} FDist(f, P_1, P_2)}{|F^\cap|}$$

$$F^\cap = TemplateFeatures(P_1) \cap TemplateFeatures(P_2)$$

For quantitative feature values, $FDist(f, P_1, P_2)$ is defined as

$$\frac{|TemplFeatureValue(f, P_1) - TemplFeatureValue(f, P_2)|}{|Domain(f)|}$$

For qualitative template feature values, the normalizations and differencing required to calculate $FDist(f, P_1, P_2)$ should be done within a single application of the mapping θ_f^{-1} from the qualitative values to $Interval(f)$ (i.e., rather than mapping once to compute each $TemplFeatureValue(f, P_i)$ and then again to difference them). This is necessary to minimize the discretization error from applying θ_f to map back to $Domain(f)$. Let $V_i = TemplFeatureInsts(f, P_i)$; the qualitative version of $FDist(f, P_1, P_2)$ is defined to be

$$\frac{(|V_2| \times \sum_{v \in V_1} \theta_f^{-1}(v)) - (|V_1| \times \sum_{v \in V_2} \theta_f^{-1}(v))}{|V_1| \times |V_2|}$$

Role Plan Distance

Role distance for a pair of plans is defined in terms of how distant the sets of fill values are for the roles that the two plans share. Our measure for the distance between sets of role fill values is defined to be the ratio of values that appear in one but not both sets to the total number of fill values (another normalized form of Hamming distance). We note that when possible, it may be appropriate to employ more specialized definitions that take into account the semantics of the underlying values. Such a definition could, for instance, reflect the fact that two airplanes of the same type are ‘closer’ than an airplane and a helicopter.

Definition 6 [Role Plan Distance] The *role distance* between plans P_1 and P_2 , denoted by $RoleDist(P_1, P_2)$, is defined as follows.

$$RoleDist(P_1, P_2) = \frac{\sum_{r \in R^\cap} RoleFillDist(r, RoleFills(r, P_1), RoleFills(r, P_2))}{|R^\cap|}$$

$$R^\cap = Roles(P_1) \cap Roles(P_2)$$

$$RoleFillDist(r, V_1, V_2) = \frac{|V_1 - V_2| + |V_2 - V_1|}{|V_1 \cup V_2|}$$

Metatheoretic Plan Distance

Using the above definitions, we define the *metatheoretic distance* between two plans as follows.

Definition 7 [Metatheoretic Plan Distance] The *metatheoretic distance* between plans P_1 and P_2 , denoted by $PlanDistance(P_1, P_2)$, is defined as follows, where $w_1 + w_2 + w_3 = 1$.

$$PlanDistance(P_1, P_2) = w_1 \times TaskFeatureDist(P_1, P_2) + w_2 \times TemplFeatureDist(P_1, P_2) + w_3 \times RoleDist(P_1, P_2)$$

The definition of metatheoretic plan distance assumes a set of weights, w_i , that can be used to adjust the relative importance of task features, template features, and roles in the distance specification. Because these three components

address different aspects of an overall plan, different users may be interested in biasing the plan distance calculation to stress the relative importance of these three components.

Similarly, the definitions for template feature, task feature, and role distance can be modified in a straightforward manner to support weights that enable varying degrees of emphasis for individual features and roles. Such weights could be defined either for an entire domain or customized by an individual user (on a situation-by-situation basis, if so desired).

Plan Distances for the Test Suite

The motivation for defining the concept of plan distance was to support a user in identifying semantically distinct plans. The results in Figure 10 show that for the SOF domain, our definition is effective. The figure displays the distances for plans in our test suite, using an equal distribution of weights for the task feature, template feature, and role distances (i.e., $w_1=w_2=w_3=1/3$).

The figure shows that the ‘closest’ plans correspond to core plans and their variants. In particular, the shortest plan distances found are between the two tiny plans (.08), between the various large plans (.08, .09, .15), and between the two small plans (.15). The distance between the two medium plans is appreciably higher (.31); as noted in Figure 1 and made apparent in Figure 8, these two plans are not simple variants of each other but rather contain key strategic differences. In addition, the plans that are farthest apart (the tiny vs large plans) are indeed the plans with the greatest meaningful variations among them. These results thus provide a preliminary validation of the effectiveness of the metatheoretic methods for capturing meaningful similarities and differences among plans.

Discussion

To date, research on general-purpose plan summarization and comparison methods has focused on approaches that analyze plan structures and planning processes directly. For example, [Mellish & Evans 1989] generate a textual description of a plan that references every plan element, without regard to its relative importance, thus making it difficult to understand the essence of large plans. [Young 1999] improves on that work by rating the importance of an action in a plan by counting the number of its incoming causal links; only actions with certain numbers of links are included in the plan summary.

Such syntactic approaches do not necessarily shed light on the semantic content of a plan. In particular, it is possible to have plans with significant variations in syntactic structure that are semantically similar; as well, plans with similar syntactic structure may have semantic differences that are extremely significant.

One key benefit of our metatheoretic approach to plan summarization and comparison is its emphasis on semantic rather than syntactic characteristics of plans. Thus, our comparison of metatheoretic properties grounds the results in concepts that are significant from a semantic

Plan Dist	Template Feature, Task Feature, Role Dist	Plan1	Plan2
.08	.03 .11 .08	large-plan-a	large-plan-b
.09	.00 .00 .28	large-plan-a	large-plan-c
.12	.00 .00 .35	tiny-plan-a	tiny-plan-b
.15	.03 .11 .30	large-plan-b	large-plan-c
.15	.15 .00 .31	small-plan-a	small-plan-b
.17	.12 .14 .26	small-plan-a	medium-plan-a
.21	.00 .14 .49	small-plan-a	medium-plan-b
.26	.27 .14 .37	small-plan-b	medium-plan-a
.29	.15 .14 .56	small-plan-b	medium-plan-b
.31	.14 .25 .53	medium-plan-a	medium-plan-b
.34	.12 .25 .67	small-plan-a	large-plan-a
.35	.07 .22 .75	medium-plan-b	large-plan-b
.35	.21 .67 .18	tiny-plan-b	small-plan-b
.36	.12 .25 .72	small-plan-a	large-plan-c
.37	.08 .33 .70	small-plan-a	large-plan-b
.40	.27 .25 .67	small-plan-b	large-plan-c
.40	.18 .33 .69	medium-plan-a	large-plan-a
.40	.27 .25 .68	small-plan-b	large-plan-a
.41	.18 .33 .71	medium-plan-a	large-plan-c
.41	.14 .40 .69	medium-plan-a	large-plan-b
.41	.11 .33 .79	medium-plan-b	large-plan-a
.42	.23 .33 .69	small-plan-b	large-plan-b
.42	.11 .33 .81	medium-plan-b	large-plan-c
.44	.21 .67 .45	tiny-plan-a	small-plan-b
.47	.29 .67 .46	tiny-plan-a	small-plan-a
.47	.29 .67 .46	tiny-plan-b	small-plan-a
.51	.29 .71 .51	tiny-plan-b	medium-plan-b
.51	.29 .71 .53	tiny-plan-a	medium-plan-b
.52	.42 .71 .42	tiny-plan-a	medium-plan-a
.52	.42 .71 .42	tiny-plan-b	medium-plan-a
.56	.33 .75 .58	tiny-plan-a	large-plan-c
.56	.33 .75 .60	tiny-plan-a	large-plan-a
.56	.33 .75 .60	tiny-plan-b	large-plan-a
.57	.33 .75 .62	tiny-plan-b	large-plan-c
.58	.37 .78 .60	tiny-plan-a	large-plan-b
.58	.37 .78 .60	tiny-plan-b	large-plan-b

Figure 10. Plan Distances for the SOF Test Suite

perspective, rather than concepts that are important to an automated system when generating a plan.

Our approach also supports customization to domains, individual users, or specific contexts. This can be achieved by selecting the sets of features and roles that are of interest to the user (for plan summarization and plan comparison) and by appropriate adjustment of weights (for analyzing a solution space).

Our plan summarization and comparison methods are *domain-independent*, making them applicable to a broad range of problems. In particular, we avoid domain-specific algorithms or bodies of knowledge that would limit the applicability of the method. One problem with general-purpose methods is that their generality often comes at the cost of depth. This tradeoff applies to our approach, in that more precise quantitative analysis tools could be developed for an individual domain that provide deeper summarization and comparison capabilities.

Our methods for plan comparison and summarization are not intended to eliminate the need for more discriminating tools. Rather, we envision the metatheoretic approach being valuable in the early stages

of planning, both in terms of enabling a user to quickly understand the main features of a plan, and to perform an inexpensive analysis of what differentiates alternative candidate plans. After developing some preliminary understanding of the plan space, a user may then wish to perform more expensive and time-consuming quantitative analyses to assess plans in detail.

The existence of a well-designed domain metatheory is critical for the successful application of our plan summarization and comparison methods. As noted elsewhere [Myers, 2000], the design of the metatheory should be a by-product of a principled approach to modeling a planning domain. Still, it remains a bit of an art to design a metatheory appropriately.

The explanatory capability of our methods when applied to larger plans could be improved by introducing a capability for contextualization that could localize application of the summarization and comparison techniques to meaningful subportions of a plan. This localization could enable more interesting regularities or trends within plans to be identified. The hierarchical structure of HTN plans provides an obvious way to generate candidate contexts, namely, subplans appearing below a given task node. Within that framework, however, identifying the most appropriate contexts for a given situation remains an interesting challenge.

Conclusions

AI planning tools must provide effective explanation capabilities in order for them to gain acceptance for real applications. To date, there has been relatively little effort devoted to developing such capabilities. Furthermore, the work that has been done has focused on syntactic elements of plans and planning processes, despite the fact that such syntactic characteristics may not correspond to important semantic features.

This paper defines an approach to plan summarization and comparison that builds on the notion of a domain metatheory. The approach has the benefit of framing summaries and comparisons in terms of high-level semantic concepts, rather than low-level syntactic details of plan structures and derivation processes. We defined a set of techniques that instantiate this approach and evaluated them within the context of a rich special operations planning domain. The evaluation showed that the techniques are effective in helping a user understand subtle aspects of individual plans, importance differences among plans, and the structure of the overall solution space.

Acknowledgments This work was supported by NASA grant NCC 2-1267. The author thanks the following people for their contributions to this work: Mabry Tyson for his assistance in making modifications to PASSAT to support the metatheoretic summarization and comparison capabilities, Peter Jarvis for developing the suite of SOF test plans used to evaluate the plan summarization and

comparison techniques, Michael Wolverton for discussions on explanation technology and feedback on the ideas.

References

- Erol, K., Hendler, J., and Nau, D. 1994. Semantics for Hierarchical Task-Network Planning. Technical Report CS-TR-3239, Computer Science Department, University of Maryland.
- Mellish, C. and Evans, R. 1989. Natural Language Generation from Plans. *Computational Linguistics* 15(4).
- Myers, K. 1996. Strategic Advice for Hierarchical Planners. In Proc. of the 5th Intl. Conf. on Principles of Knowledge Representation and Reasoning, Cambridge, Massachusetts.
- Myers, K. L. and Lee, T. J. 1999. Generating Qualitatively Different Plans through Metatheoretic Biases. In Proc. of AAAI-99, AAAI Press, Menlo Park, CA.
- Myers, K. L., Tyson, W. M., Wolverton, M. J., Jarvis, P. A., Lee, T. J., and desJardins, M. 2002. PASSAT: A User-centric Planning Framework. In Proc. of the 3rd Intl. NASA Workshop on Planning and Scheduling for Space, Houston, TX.
- Tate, A., Dalton, J., and Levine, J. 1998. Generation of Multiple Qualitatively Different Plan Options. In Proc. of Artificial Intelligence Planning Systems.
- Wick, M. R. and Thompson, W. B. Expert System Explanation, *Artificial Intelligence*, 54(1-2), 1992.
- Young, R. M. 1999. Using Grice's Maxim of Quantity to Select the Content of Plan Descriptions. *Artificial Intelligence*, 115(2).