Automating Planning using Natural Language

Kalyan Moy Gupta[1,2] and David W. Aha[2]

[1]ITT Industries; AES Division; Alexandria, VA 22303
[2]Navy Center for Applied Research in Artificial Intelligence;
Naval Research Laboratory (Code 5515); Washington, DC 20375
firstname.lastname@nrl.navy.mil

Abstract

We propose **PLANL,** an approach for **PLA**nning with **N**atural **L**anguage, to accelerate the development of automated planning systems, enable plan sharing across multiple planners, and facilitate natural language interaction. PLANL uses generative sublanguage ontologies (GSOs) to robustly and accurately translate planning knowledge descriptions into representations such as STRIPS or hierarchical task networks. GSO's accomplish this through a novel ability for efficiently representing and resolving polysemy. Unlike alternative approaches, PLANL does not have a proprietary plan representation. Instead, it exploits existing plan representations and selects a linguistically motivated conceptual vocabulary for them.

1. Introduction

Developing automated planners requires intensive and lengthy effort, the resulting systems cannot share planning knowledge, and they do not support natural language (NL) interaction. The first two limitations have been partially addressed by sharable ontologies (Brenner & desJardins, 2002; Gil & Blythe, 2000). However, these ontologies are not linguistically motivated. Thus, translation from NL plan descriptions to a computable plan representation is problematic. Others have used ontologies capable of linguistic representation, but they cannot represent polysemous words efficiently and require a proprietary plan representation (e.g., SNePS [Shapiro, 2000]).

Significant gains can be realized by using linguistically motivated expressive ontologies. We propose a novel approach for automating **PLA**nning with **N**atural **L**anguage, called **PLANL**, which:

- Represents planning knowledge by reference to a generative sublanguage ontology (GSO). GSOs are linguistic ontologies inspired by generative lexicon theory (GLT) (Pustejovsky, 1995), which support robust NL processing (Gupta & Aha, 2003; 2005).
- Simplifies system-human interaction and accelerates knowledge engineering by encoding NL descriptions of plan knowledge, plans, and world states into their formal representations.

The next section introduces PLANL, and describes GSO and its use in interpreting planning knowledge descriptions. We then conclude with directions for future research.

2. Planning with Natural Language

Automated planners represent planning knowledge, goals, and world states in a formal knowledge representation language that are subsets of propositional and first-order predicate calculus (Weld, 1994). For a planner to work, the world and goal state descriptions must share their vocabulary and expressions with the domain theory.

PLANL has two main components. First, the *interpreter* converts NL descriptions of domain knowledge, world states, and planning goals into formal computational descriptions (e.g., a logic program) by using a GSO. Second, the *planner* generates a plan to accomplish a specified goal using the domain knowledge.

2.1 Generative Sublanguage Ontologies

While linguistic ontologies are intrinsically suited for NL interpretation, they may lack the expressiveness needed to represent planning knowledge (e.g., WordNet [Fellbaum, 1998]) and/or lack a theory for efficiently representing polysemous terms (e.g., SNePs). Consequently, these approaches can fail during interpretation.

GSOs efficiently represent polysemous terms and support sense ambiguity resolution. GSO concepts have a predicate argument representation within an object oriented framework and can express plans in established plan representation formalisms (e.g., hierarchical task networks (HTNs) [Nau *et al*., 2003]).

GLT supports *systematic polysemy*, in which potentially unanticipated but related meanings of a term can be systematically *generated* from a *well-defined conceptual structure*, and its applicable meaning can be selected based on a term's context. It provides a set of formally defined *generative operators* to select senses. GSO extends and implements GLT for constrained domains (i.e., sublanguages) (Gupta & Aha, 2003; 2005).

A GSO's conceptual structure includes four elements: (1) *Arguments*, a set of typed variables that a concept accepts as parameters when it is instantiated. These are used to represent events and relations; (2) *Qualia*, a set of relationships that a concept has with other concepts, including relations necessary for resolving polysemous expressions and compound noun interpretations; (3) *Event structure*, a set of temporal and causal relationships among processes and states used for interpreting polysemous and light verbs; and (4) *Inheritance*, a specification of type-subtype relationships among concepts.

Three generative operators enable sense generation and selection during interpretation: (1) *Type coercion*, a principled operation for argument type shifting that is used to recover from type failures in novel circumstances; (2) *Co-composition,* an operation involving two concepts where the slot of one co-specifies the other to generate a sense that is not overtly expressed; and (3) *Selective binding*, which resolves polysemous underspecified adjectives by selecting the behavior slot of the object they qualify.

Besides the conceptual structure, GSOs include a library of terms with their syntactic and morphological features and pointers to the conceptual structure(s) that represent their meanings.

2.2 Interpreting Domain Descriptions with GSOs

PLANL's interpreter encodes domain descriptions into a logic representation by *generating a syntactic parse(s), retrieving and instantiating GSO concepts* corresponding to terms used in the descriptions, and *resolving references by* mapping the arguments with GSO instances.

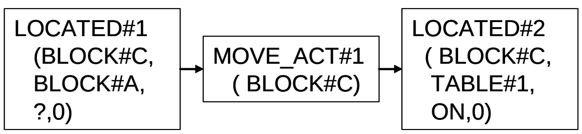

Figure 1. A GSO representation of a blocks world move task

Figure 1 shows the GSO representation of the NL blocks world domain knowledge description "Move C from A onto Table". It shows that BLOCK#C, which was LOCATED#1 on BLOCK#A, was moved by MOVE_ACT#1 and is now LOCATED#2 on TABLE#1. The GSO event structures capture the post condition description of an action, which is a standard way of describing actions in most plan representations. Also, the choice of predicates and their language was determined by the GSO rather than a non-linguistic concept vocabulary (e.g., see [Weld, 1994]).

2.3 Applying PLANL

PLANL will be used in two phases, namely *knowledge acquisition* and *plan generation.* Knowledge acquisition first updates the GSO to include domain specific terms and concepts (e.g., "block" and "table" in a blocks world). It then updates planning domain knowledge using NL descriptions. For example, "Move C from A onto Table if there is nothing on C" is an NL domain description in a blocks world. Likewise, PLANL can also resolve unknown named entities (e.g., "block A") via user descriptions of such entities (e.g., "A is a block"). PLANL will allow domain experts to use NL descriptions rather than a formal plan representation language.

To generate plans, users will specify initial and goal states with NL descriptions. For example, "A is on Table, C is on A, and B is on Table" is a state description in a blocks world. PLANL then generates the plan for these states using one of the existing planners such as STRIPS, HTN (Nau *et al*., 2003), or Task-Method Knowledge (TMK) (Murdock *et al*., 2003) engine. Clearly, describing the initial and goal states in NL is comparatively easy, flexible, and convenient for anyone lacking familiarity with plan representation languages and provides a user friendly alternative.

3. Conclusion and Future Work

We introduced PLANL, a novel approach for planning with natural language enabled by GSOs. PLANL makes knowledge sharing and reuse feasible, and may accelerate knowledge engineering and simplify user-system interaction at no extra cost. We are currently implementing PLANL, and will evaluate it on planning tasks that use representations such as HTNs and task-method-knowledge models.

Acknowledgement

We thank NRL for funding this research.

References

Brenner, M., & desJardins, M. (Eds.) (2002). *Planning with and for Multiagent Systems: Papers from the AAAI Workshop* (Technical Report WS-02-12). Menlo Park, CA: AAAI Press.

Fellbaum, C. (1998). WordNet, An electronic lexical database. Cambridge, MA: MIT Press.

Gil, Y. & Blythe, J. (2000). PLANET: A sharable and reusable ontology for representing plans. In Y. Gil & K. Myers (Eds.) *Representational Issues for Real-World Planning Systems: Proceedings of the AAAI Workshop* (Technical Report WS-00-07). Menlo Park, CA: AAAI Press.

Gupta, K.M., & Aha, D.W. (2003). Nominal concept representation in sublanguage ontologies. *Proceedings of the Second International Workshop on Generative Approaches to the Lexicon* (pp. 53-62). Geneva, Switzerland: University of Geneva.

Gupta, K.M., & Aha, D.W. (2005). Interpreting events using generative sublanguage ontologies. Unpublished manuscript.

Murdock, J.W., Aha, D.W., & Breslow, L.A. (2003). Case-based argumentation via process models. *Proceedings of the Fifteenth International Conference of the Florida Artificial Intelligence Research Society* (pp. 134-138). St. Augustine, FL: AAAI Press.

Nau, D., Au, T.-C., Ilghami, O, Kuter, U, Murdock, J.W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, **20**, 379-404.

Pustejovsky, J. (1995). *The generative lexicon*. Cambridge, MA: MIT Press.

Shapiro, S.C. (2000). An introduction to SNePS 3. In B. Ganter & G.W. Mineau (Eds.), *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Berlin: Springer.

Weld, D.S. (1994). An introduction to least commitment planning. *AI Magazine*, **15**(4), 27-61.