

B-SURE: A BELIEVED SITUATION AND UNCERTAIN-ACTION REPRESENTATION ENVIRONMENT

JOHN K. MYERS

ATR Interpreting Telephony Research Laboratories
Sanpeidani, Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
myers@atr-la.atr.co.jp

Abstract

This paper presents a system that is capable of representing situations, states, and nondeterministic nonmonotonic-outcome actions occurring in multiple possible worlds. The system supports explicit representations of actions and situations used in intentional action theory and situation theory. Both types and instances are supported. Situations and states before and after nonmonotonic actions can be represented simultaneously. Agents have free will as to whether to choose to perform an action or not. Situations and actions can have expected values, allowing the system to support decision-making and decision-based plan inferencing. The system can perform global reasoning simultaneously across multiple possible worlds, without being forced to extend each world explicitly. The resulting system is useful for such natural language tasks as plan recognition, intentions modeling, and parallel task scheduling.

1. Introduction

The key to good reasoning is a powerful representation system that is able to accurately model details of a problem. Once a good representation has been established, problem computations often become straightforward.

Recent advances in situation theory [BP83, Bar89] and the theory of intentions [Bra87] have offered many new insights on significant problems found in natural-language understanding. However, these theories offer philosophical approaches only, and do not give instructions for building concrete representation and reasoning engines.

At the same time, the software systems that have been built for reasoning and representation fall short in any number of areas. Production systems and semantic networks can follow chains of inferences, but can only represent one possible world at a time—they cannot reason with states that are both possibly true and possibly not true, while keeping the chains of resulting inferences separate. Most planners work with limited possible worlds, but cannot reason and perform inferences across multiple worlds at the same time. The classical ATMS¹ can represent and reason with multiple timeless possible worlds, but cannot represent actions [dK86]—in particular, nonmonotonic actions where a retracted state is both believed to be true in the world before the action takes place, and believed to be *not* true in the world representing the situation after the retracting action has taken place, cannot be represented. In addition, the ATMS only represents propositions that are instantiated constants or Skolem constants; it does not represent uninstantiated variables. A modified ATMS that can represent nonmonotonic transitions between worlds has been developed [MN86], but this system does not explicitly represent situation types and instances, action events, nor nondeterminism. Most plan inference systems have ignored free will and the

explicit representation of the right to choose actions, e.g. to choose to be uncooperative. Almost all previous systems have ignored the nondeterministic quality of real-world actions that necessitates commitment in intentions. Real actions can result in one of several possible outcome situations, whereas almost all previous systems are completely unable to model nondeterministic outcomes. Only decision-analysis systems have modeled expected values of actions, and they do not support inferencing. See [BL85] for an excellent summary of issues.

The B-SURE (Believed Situation and Uncertain-action Representation Environment) package is an implemented system that supports representation, planning, decision-making, and plan recognition using probabilistic and uncertain actions with nondeterministic outcomes in multiple possible action worlds. Situations, states, and action events are all represented explicitly, using types (variables) and instances. The B-SURE system is implemented as a series of extensions to a classical ATMS. The resulting system is very useful, and is being used in plan recognition, intentional agent, and scheduling research.

2. Situation Theory

In [BP83], situations are divided into the categories *abstract* and *real*, and also into the categories “*states of affairs*” and “*courses of events*”. Abstract situations denote situations that are mental representations. All the situations discussed in this paper are “abstract situations”. Real situations denote situations as they actually are in the real world. Since it basically never makes sense to talk about real situations in the computer, there is no need to supply these in a representation environment. “States of affairs” correspond to situations that are static, called simply *situations* in this paper. “Courses of events” correspond to situations that describe actions that are being executed, called *action events* or *actions* in this paper. Barwise and Perry also make use of “relations” defined over “individuals” and “space-time locations”. This paper takes as primitive the expression of a relation, which will be termed a *state*. The user is free to mention individuals or space-time locations in state descriptions as desired. State descriptions may be represented using *logical forms*, *feature structures*, or *other methods*—since the contents of states are not used by B-SURE except for output, it does not matter. States, situations, and actions are assigned one of the belief values {*definitely believed true*, *possibly believed true*, *not believed true*, *believed not true*, *not believed*}, otherwise known as {*actual*, *possible*, *hypothetical*, *inconsistent*, *null*}, corresponding to the amount of support offered by the system’s underlying ATMS

¹Assumption-Based Truth Maintenance System [dK86]

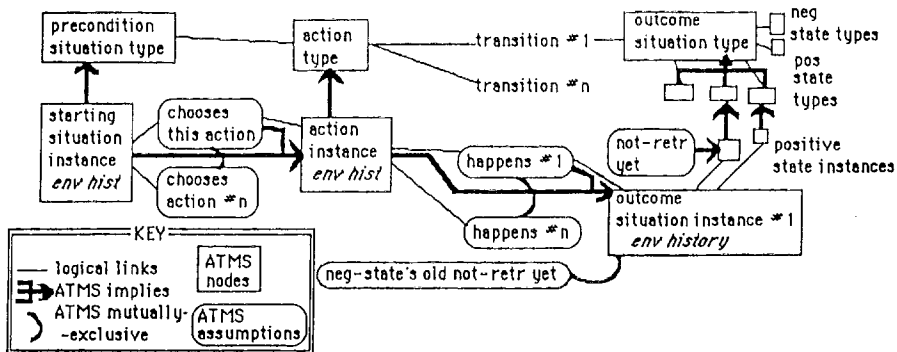


Figure 1: Structure for Representing Nondeterministic Actions

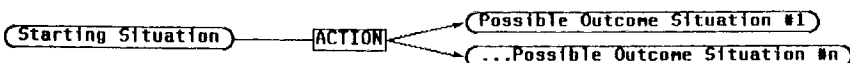


Figure 2: Compact Graphical Representation which Omits States and Types

representation (see [Mye89] for more information).

3. Intentional Action Theory

One model of intentions states that an intention is a choice to perform an action, plus a commitment to obtaining its desired outcome[CL87]. With deterministic action outcomes, there is no real need for *endeavoring* [Bra87], since once the action has been started, it is guaranteed to finish properly. Many planners in fact operate in this "fire and forget" mode. However, once it is acknowledged that action execution is in fact nondeterministic and can have undesirable outcomes, the need for endeavoring becomes clear. The planner must predict the likelihood of possible outcomes happening, and judge which action sequence offers the best chances. It must interactively maintain a history of past endeavors and results, and modify its future behavior based on current outcomes. Acting intentionally becomes significantly more interesting and realistic with the explicit representation of possible chains of nondeterministic actions.

4. Previous Efforts

DeKleer [dK86] presents the first ATMS. Morris and Nado [MN86] present an ATMS that can represent nonmonotonic transitions, but do not handle probabilities, uncertainties, explicit situation types, state types, nor action events. The research of Allen (e.g. [AK83, All87]), who uses a predicate-calculus representation, offers some of the best multiple-worlds (deterministic) action representation in this field. Charniak and Goldman [CG89] use probabilities and Bayesian nets to represent the truth value of probabilistic statements and attack story understanding. Although nondeterministic-outcome actions are not represented, and Bayesian nets cannot support global inferencing with nonmonotonic actions, their work is important. Norvig and Wilensky [NW90] comment on problems of probabilistic statements. The most similar work is recent research by Rao and Georgeff (e.g. [RG91]), who use a modal logic instead of an ATMS to represent nondeterministic actions.

5. B-SURE Entities & Implementation

The underlying ATMS works with *nodes*, *assumptions*, and *implications* (justifications). See [dK86].

A state consists of a proposition about the world. States are primitives. A *situation* is a set of positive and negative (withdrawn) states. An *action event* represents the state that "execution of the action has started". States, situations, and actions have types and instances. See figure 1. (The abridged representation of figure 1 is shown in figure 2.) Existence of an instance in a world always implies existence of its type. A *chooses* node is an assumption associated with an action instance that represents whether an agent chooses to execute that action or not. The chooses assumption together with the starting situation instance imply the action instance. Since an agent typically can only execute one action in a given situation, the situation's ensuing chooses assumptions are rendered mutually exclusive (pairwise "nogood"). Action types have precondition situation types. Action instances are instantiated from types by first verifying that the precondition situation type is believed true in that world. Action instances transition from a starting situation instance to one of a number of known nondeterministic outcome situation instances. Actions have *transitions*. A transition has an outcome situation and a probability or an uncertainty. An *uncertainty* is defined as a probability random variable of range [0,1] together with an associated second-order probability distribution. Uncertainties are initialized using maximum-entropy theory, and get updated as outcome observations are taken, to enable the system to learn and estimate possible probabilities. See Section 6. Uncertainties are used to represent *confidence* in *likelihood* values and to make decisions regarding information-gathering activity. The calculus of uncertainties is too complex to explore further here, and is not required for understanding the main capabilities of the representation; probabilities are sufficient. Transitions can be types or instances.

A transition instance is defined as a *happens* assumption. An action instance, together with a happens assumption, imply the corresponding outcome situation instance. Typically only one outcome situation can occur from a given action instance, so the action's happens assumptions are made mutually exclusive. A situation type is implied by its state types. When an outcome situation instance is instantiated, all of its new positive states are instantiated and all of its old negative states are retracted. A positive nonpermanent state instance is implied by a *not-retracted-yet* assumption. The outcome situation instance remembers these. Situation and action instances store an explicit *environment history* of all added state, chooses, and happens assumptions that are currently believed true in that possible world's timeline. A negative state is retracted by making the situation instance and the state's "not-retracted-yet" assumption mutually inconsistent, and deleting the state's assumption from the outcome situation's environment history. A state type or instance or situation type's belief value in a particular world is found by testing that node against a situation instance's environment history. Situation types and instances can have values. Actions can have costs. The expected value of an action is determined by summing the transition probabilities times the expected values of the outcome situations, when known, and subtracting its cost. The expected value of a nonvalued situation instance is determined by maximizing the expected values of the possible subsequent actions, when known. In this manner, decision theory determines the course of action with the maximum expected value at any one situation, for a planning agent. This can be used to predict the probable next course of action of a planning agent by an observing agent performing plan recognition (actually, "decision recognition") [Myc91].

6. Probability Estimation

The probability of an outcome situation i occurring following performance of an uncertain action is estimated using the new estimator $\frac{k_i+1}{m+n}$ instead of $\frac{k_i}{m}$, where m is the total number of previously observed trials of that action type, k_i is the previously observed number of i th situation-type outcomes, and n is the number of known possible outcome situations from that action. The new estimator is optimal. It represents the center of mass of all possible probabilities, instead of the maximum-likelihood mode; it converges faster and on average is more accurate than the old estimator; and, it can be used accurately with small sample numbers and small success counts [Myc92].

7. Maintaining an Interactive History

One important advantage of the B-SURE system is that not only can it be used for hypothetical reasoning about future events, but the same structures can then be used as a history mechanism for interactively monitoring and representing the history of the actual events as they occur. A user system should start out in a known situation, which is presumed actual. Typically, the user system will use B-SURE to explore many different nondeterministic-action sequences and

make decisions as to which actions are the best ones to perform. The system will then start executing the first action in the chosen sequence. At this point, the user system should instruct the B-SURE system to **presume** the chooses assumption associated with the chosen action being executed, which will change its truth value from "possibly believed true" to "definitely believed true". If the chooses node has already been made inconsistent with other chooses nodes (because the user-system or agent could only perform one action at a time), those other nodes are automatically rendered "believed not-true" at this point. The presumption of the chooses node renders the associated implied Action Event instantiation "definitely believed true" at this point, also. This represents the fact that the action has started and is currently being executed.

When the action finishes, it is necessary for the B-SURE system to realize which outcome occurred. This is typically performed by the system setting up a *recognition demon* that is attached to a separate state or situation type that, when true, reliably indicates that a given outcome has occurred. When the demon fires, it presumes the outcome's happens assumption. It is important to ensure that one and only one recognition demon fires. Alternatively, the user can control presuming the happens nodes directly. When a single happens assumption is presumed, it automatically renders its sibling happens assumptions "definitely believed not-true".

The combination of the happens node being presumed and the action event node already being believed true renders the appropriate resulting situation instance believed true. Note that if any instance becomes true, so does its associated type node as well.

At any one point in time, the states, situations, and action event instances that have happened in the world already are believed true; and the situations and events that have not happened yet but could happen are believed possible. In this way, the system maintains a timeline history of the situations and action events that have in fact occurred, while allowing hypothetical planning and exploration of possible future events in the same data structure.

It is not necessary for the system to maintain only a single timeline history. It is possible to maintain disjoint histories, to represent e.g. progress made by different processing agents, progress made in different domains, or progress made at different hierarchical levels of abstraction. It is possible to maintain forking (nondisjoint) histories if this makes sense, and the mutual exclusion options have been turned off (see Section 10).

Counterfactuals The system maintains the structures of past possibilities that did not happen. Although these are not believed true, it is possible for the user to explore these structures and perform reasoning on what *could* have occurred had certain actions been chosen or certain nondeterministic outcomes happened, by supplying an extra *counterfactual* assumption to justify the desired action or sit-

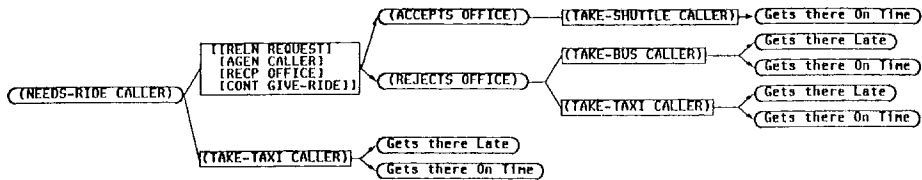


Figure 3: Modeling a Plan/Decision Inference Problem in Getting to a Conference On Time

uation instance. It is even possible to add to these structures, if necessary. This can be used to explicitly represent newly-received past counterfactual information (e.g., "If you had applied for the conference last June, the cost would have been 35,000 yen") and the associated reasoning derived from such assertions. Such reasoning has traditionally been very difficult to represent, because of the negative truth values.

8. Decision Inference Example

A researcher is calling a conference office from the train station and wants to get to the conference on time. He has a choice between asking for taxi directions, or requesting the office to send a shuttle-bus out directly to give him a ride. The shuttle will take him directly to the conference on time. If he requests and the office turns him down, he has a choice between taking a taxi, and taking the regular bus. These cost different amounts of money and have different chances of getting to the conference on time. See figure 3. The plan inference system must predict which paths of information he will explore, i.e. what he will say next; and then which decisions he will make for his actions. This is done using "decision inference", by understanding which action trees offer the best expected value based on the value and chances of outcomes. Note that the shuttle-bus, the taxi, and the regular bus will all three allow the researcher to possibly obtain his desired goal, but there are definite preferences. The system should not remain uncommitted. See [Mye91] for more details.

9. Intentional Communication Example

A recent analysis of 12 actual interpreted telephone conversations revealed that 31% of the utterances were spent in requests for confirmation and repetitions of information such as telephone numbers, name spellings, and addresses, that were not completely understood the first time [OCP90]. This means that the traditional plan-recognition model of assuming that the hearer automatically understands the semantic content of the speaker's utterance is fallacious. The speaker, and the system too, must consider the case in which the hearer does not understand an utterance. Since the speaker wants and intends to communicate specific information², the speaker will endeavor to ensure that the information is communicated, by repeating an utterance when it is not understood. Thus, speaking an utterance is a nondeterministic ac-

tion; it is unclear whether the hearer will understand or not. Intentional utterance acts are therefore modeled as nondeterministic-outcome actions by B-SURE. Different courses of the conversation can be represented depending upon the outcomes of the utterance acts. See Figure 4.

10. Process Scheduling Example

The application of the BEHOLDER³ limited-resource parallel scheduling system to translation systems is being researched. A hypothetical model system is used for testing. The system will accept an input candidate from a speech recognition module, and attempt to quickly transfer the result directly to output. If required, a morphological analyzer will derive multiple possible analyses candidates for each input candidate. A pattern matcher will then recursively apply a body of patterns to each analysis candidate. Each pattern has a series of transfer-driven translation templates; each template has a series of prototypical example bindings. The highest-ranking structure of matching nested patterns and their bindings are sent to a template matcher. The distances between the pattern bindings and the template examples for each pattern in the structure are compared using a thesaurus. The template with the closest match for each pattern will be used to assemble a translation.

It is the responsibility of the BEHOLDER system to schedule this activity in an opportunistic fashion on multiple processors. There is no need to continue to explore a branch if a good translation has been found. The BEHOLDER system must use value-of-information theory and decision theory to determine which process branches to explore next and when to stop.

The BEHOLDER scheduler uses the B-SURE system to keep track of which processes are running and which have been executed. Using this representation, it can plan ahead and decide how useful it is to expand a particular path of execution. As processes are started, the chooses nodes are presumed. Figure 5 shows a simulated run where the direct transfer, the morphological analysis, one pattern match, and one template match have been run. The template match has examined two examples so far.

Since in this case more than one action can be executed at a time, and one action can legally have more than one possible outcome, it was necessary to modify the B-SURE system to allow local disabling of the

²Note that people do not always decide to intend to endeavor to do everything that they want. Intending is quite different from wanting.

³Beneficial Entity for Heuristically Ordering processes under Limited resources and Decision-making for Execution in Real-time.

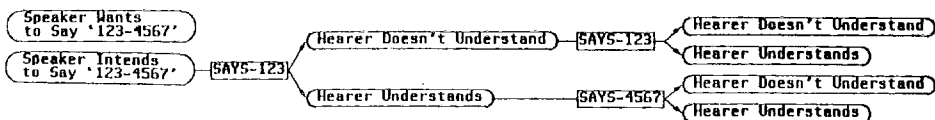


Figure 4: Modeling an Intention to Communicate a Telephone Number Correctly

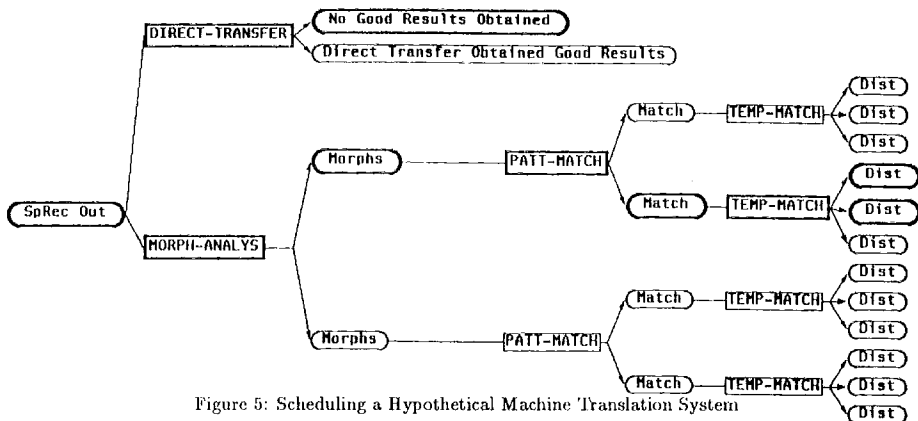


Figure 5: Scheduling a Hypothetical Machine Translation System

mutually-exclusive actions and outputs features.

11. Conclusion A powerful situation representation tool is required for representing past, present, and future nonmonotonic actions, when the actions can have nondeterministic outcomes. The B-SURE environment offers such a tool. Being able to model realistic actions allows exploration of significant problems in situation modeling, plan inference, intentional actions research, and value-of-information theory as applied to parallel process scheduling.

References

- [AK83] James F. Allen and Johannes A. Koonen. Planning using a temporal world model. In *IJCAI'83*, pages 741-747, Karlsruhe, 1983.
- [All87] James Allen. *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, CA, 1987.
- [Bar89] Jon Barwise. *The Situation in Logic*. Center for the Study of Language and Information (CSLI), Stanford, CA., 1989.
- [BL85] Ronald J. Brachman and Hector J. Levesque. *Readings in Knowledge Representation*. Morgan Kaufmann, Los Altos, CA, 1985.
- [BP83] Jon Barwise and John Perry. *Situations and Attitudes*. The MIT Press, Cambridge, 1983.
- [Bra87] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard Univ. Press, Cambridge, MA, 1987.
- [CG89] Eugene Charniak and Robert Goldman. A semantics for probabilistic quantifier-free first-order languages, with particular application to story understanding. In *IJCAI'89*, pages 1074-1079, Detroit, MI, 1989.
- [CL87] Philip R. Cohen and Hector J. Levesque. Intention = choice + commitment. In *AAAI'87*, pages 410-415, Seattle, WA, 1987.
- [dK86] Johan de Kleer. An Assumption-based TMS. *Artificial Intelligence*, 28(2):127-162, March 1986.
- [MN86] Paul H. Morris and Robert A. Nado. Representing actions with an Assumption-based TMS. In *AAAI'86*, Philadelphia, PA, 1986.
- [Mye89] John K. Myers. An assumption-based plan inference system for conversation understanding. In *WGNL Meeting of the IPSJ*, pages 73-80, Okinawa, Japan, June 1989.
- [Mye91] John K. Myers. Plan inference with probabilistic-outcome actions. In *Conf. Proc. Information Processing Society of Japan*, volume 3, pages 168-169, Tokyo, March 1991.
- [Mye92] John K. Myers. An introduction to planning and meta-decision-making with uncertain non-deterministic actions using 2nd-order probabilities. In *First Inter. Conference on AI Planning Systems*, College Park, Maryland, June 1992.
- [NW90] Peter Norvig and Robert Wilensky. Abduction models for semantic interpretation. In *COLING-90*, volume 3, pages 225-230, Helsinki, Finland, August 1990.
- [OCP90] Sharon L. Oviatt, Philip R. Cohen, and Ann Podlozuy. Spoken language in interpreted telephone dialogues. Technical Report AIC-496, SRI International, Menlo Park, CA, 1990.
- [RG91] Anad S. Rao and Michael P. Georgeff. Asymmetry thesis and side-effect problems in linear-time and branching-time intention logics. In *Proceedings of IJCAI-91*, Sydney, Australia, 1991.