

Our current system is being advanced in several directions. But first, we present some motivations.

3 Motivations

Research in planning and acting has progressed independently of the research in knowledge representation and reasoning. Traditional planners end up using three different levels of representations (each using a different representation language)—a representation for world model (typically a FOPL); a representation for operators/actions (like the operator schema of STRIPS [Fikes and Nilsson 1971], or the operator description language—ODL of SIPE [Wilkins 1988]); and a representation for plans (like NOAH's [Sacerdoti 1977] and SIPE's procedural networks). As a consequence, the system has to perform reasoning at three different levels—reasoning within the world model; reasoning about actions (used in the planning component); and reasoning about plans (as done by *procedural critics* of NOAH and SIPE). Facts stored in the world model correspond to the agent's beliefs. The kind of reasoning done on these beliefs is limited to basic retrieval, and sometimes, using simple inference rules (which may or may not be expressed in the same language! See [Wilkins 1988]'s deductive operators.) simple consequences of current beliefs can be derived. The state of the art in knowledge representation and reasoning is much more advanced than that. Current knowledge representation and reasoning systems are capable of dealing with issues in natural language understanding, representing beliefs of the agent as well as others, belief revision using truth maintenance procedures, and other subtle issues. Some of these representations also deal with beliefs about agents performing actions and events taking place.

In the past, most efforts of researchers of the SNePS research group have centered around representation and reasoning about beliefs derived from natural language interaction with the user. Our work extends the SNePS approach to modeling cognitive agents by integrating the notions of acting and planning. The basic motivations underlying our approach can be summed by the following quote from [Georgeff 1987]:

Another promising approach to providing the kind of high-level goal-directed reasoning capabilities, together with the reactivity required for survival in the real world, is to consider planning systems as rational agents that are endowed with the psychological attitudes of belief, desire, and intention. The problem that then arises is specifying the properties we expect of these attitudes, the way they interrelate, and the ways they determine rational behavior in a situated agent.

4 Towards Integrated Acting and Inference

In most current architectures reasoning is performed by some inference engine and acting is done under the control of an acting executive. In order to achieve our goals, we have come to the conclusion that inference and acting need to be more tightly coupled. A survey of most systems will reveal that it is somewhat awkward to do acting in reasoning (or logic-based) systems (but it is convenient to talk about representational and reasoning issues using them), and it is awkward to study reasoning and representational issues in systems designed for acting/planning. Our approach is based on the viewpoint that logical reasoning rules implicitly specify the act of believing. Thus the inference engine can be viewed

as a *mental actor*. This enables us to establish a closer relationship between rules of inference and rules of acting (or planning). Believing is a state of knowledge; acting is the process of changing one state into another. Reasoning rules pass a *truth* or a *belief* status from antecedent to consequent, whereas acting rules pass an *intention* status from earlier acts to later acts. A reasoning rule can be viewed as a rule specifying an act—that of believing some previously non-believed proposition—but the believe action is already included in the semantics of the propositional connective. McCarthy (1986) also suggested that inference can be treated as a mental action. As mentioned in [Shapiro 1989], when a rule fires, the agent forms the intention of believing its consequences. This suggests that we can integrate our models of inference and acting by eliminating the acting executive. While this clarifies the notion of an inference rule as specifying an act, we need to reexamine representations for plans and acts and the role they play under the influence of forward/backward chaining procedures. For that purpose, we have introduced a more general notion of a transformer (see [Kumar 1989] for details).

A *transformer* is a representation that specifies a belief or act transformation under the influence of a transformation procedure. It has two parts—($\langle a \rangle$, $\langle b \rangle$), where both $\langle a \rangle$ and $\langle b \rangle$ can specify either a set of beliefs or some act. Basically a transformer is a more general representation that captures the notions of reasoning as well as acting. The transformation procedure can use the transformer in forward as well as backward chaining fashion. Using a transformer in forward chaining is equivalent to the interpretation “after the agent believes or intends to perform $\langle a \rangle$, it believes or intends to perform $\langle b \rangle$.” A transformation procedure using backward chaining on a transformer yields the interpretation “if the agent wants to believe or perform $\langle b \rangle$, it must first believe or perform $\langle a \rangle$.”

Since both $\langle a \rangle$ and $\langle b \rangle$ can be sets of beliefs or an act, we have four types of transformers—*belief-belief*, *belief-action*, *action-belief*, and *action-action*. The idea behind defining transformers is to have a unified notion of reasoning and acting. Next, we show how our existing representations for reasoning rules, acts, and plans, can be classified as transformers. We will also identify how forward/backward chaining through them can facilitate reasoning, acting, planning, and reacting.

Belief-belief transformers are reasoning rules of SNePS. A reasoning rule can be used in forward, backward, or bidirectional inference. The meaning and use of the representation remains the same. SNIP, the SNePS inference procedure already knows how to interpret these.

Belief-act Transformers under forward chaining provide the interpretation— “after the agent believes $\langle a \rangle$, it forms the intention of performing $\langle b \rangle$ ”. Such a transformer models reactivity. E.g., a reactivity specification like “In case of fire, leave building.” When a sensory input yields a belief that there is a fire, a forward chaining procedure will use the transformer to schedule the act of leaving the building on the queue of intentions.

Used during backward chaining, i.e., “if the agent wants to perform $\langle b \rangle$, it must believe $\langle a \rangle$,” the transformer represents a specification of a precondition of an action. The transformation procedures should ensure that backward chaining is blocked in the case of reactive transformers and forward chaining is blocked in the case of precondition transformers.

Action-belief Transformers are used to represent effects of actions, as well as plan decompositions for complex goals.

Forward chaining through effect transformers specifies the updating of beliefs after performing an action, and backward chaining through them can be used in planning (i.e., "if the agent wants to believe $\langle b \rangle$, it must intend to perform $\langle a \rangle$."

The goal decomposition transformer used by the backward chaining procedure specifies plan decompositions for achieving some goals. Forward chaining through it is blocked.

Action-action Transformers are modeled using a control action that sequences two acts. Such a transformer is used only in the forward chaining direction and represents the agent's intentions to first perform $\langle a \rangle$ and then perform $\langle b \rangle$. It wouldn't make sense to backward chain through such a transformer. However, there is a complex act decomposition transformer, which when used in backward chaining, will result in specification of decomposition of complex acts.

5 The SNePS Acting And Inference Package

We have informally described the notion of a transformer as a general representation that can provide reasoning, planning, and acting capabilities under the interpretation of forward and backward chaining procedures. The resulting architecture of such a model is depicted in Figure 2. The natural

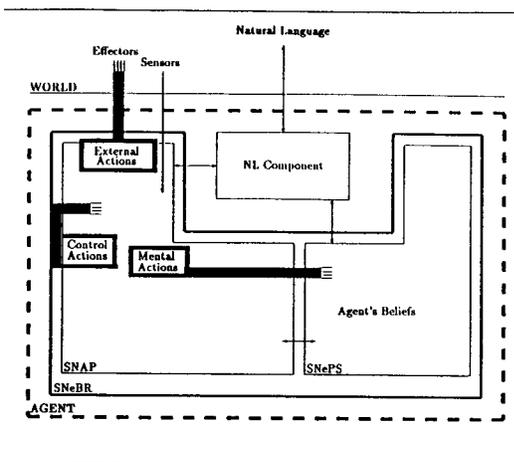


Figure 2: Architecture of the integrated acting and reasoning system (being implemented).

language component and representation of agent's beliefs remains the same as that of the SNePS actor. The syntax of rules and other planning and acting-related propositions remains the same. However, they now belong to the more general class of transformers. Transformers are interpreted and used by the SNePS acting and inference package (SNAP). SNAP maintains the agent's intentions which can be inferences (mental actions) and/or external acts. Implementation of the integrated architecture is nearly complete.

6 Analysis of the Architecture

We now characterize our architecture along the dimensions listed in the symposium guidelines.

Generality: The architecture is a generic tool for modeling rational cognitive agents. Agents modeled using this architecture are capable of interacting in natural language. They can represent and reason about beliefs, actions, and plans,

and can be made to react to sensory input. Other than tasks of natural language understanding and deductive reasoning, the architecture is not specialized for any specific domain or tasks.

Versatility: The natural language component is written using an ATN. To facilitate natural language interaction in a specific domain one has to supply an appropriate lexicon and a grammar for sentences in that domain. The effector behavior interface is programmed in CommonLisp. Currently we have demonstrated its capabilities in a simulated blockworld, and as a prototype user interface development tool to a commercially available geographic information system.

Rationality: The ATMS is an integral part of the system. All plans and acts are deductively derived from the agent's beliefs and goals. Since the deductive system is sound, rationality is ensured.

Adding new knowledge can be done via the natural language component as well as the sensory interface (when it exists).

Ability to learn: As mentioned earlier, the agent initially starts with an empty set of beliefs and can be instructed about any domain. Thus it possesses the capability of learning by instruction. At the moment this is the only machine learning technique used. [Choi 1990] has implemented a scheme where the agent can learn and subsequently use new rules from existing ones.

Taskability: As mentioned above, all tasks are described to the agent in natural language. A user can query the agent on how it would perform tasks (in which case plans may be derived) and/or request the agent to perform them (in which case plans will be derived and the agent will form the intention to carry them out).

Scalability: The agent is capable of receiving more instruction and subsequently using that knowledge to influence its future behavior. We have successfully demonstrated this in microworld domains. There is nothing inherent in the architecture as such that would limit scalability.

Reactivity is inherently defined and built-in in the architecture of the system at a primitive level. Forward chaining through beliefs representing reactivity transformers is responsible for reactive behavior during plan execution. Thus individual plans do not have to account for unpredictable situations. The agent's activity is always triggered in reaction to some input.

Efficiency: Our primary concerns have been cognitive modeling rather than real-time performance. The system performs in a reasonably efficient fashion. All the demo tasks modeled run in real-time. However, there is no guarantee as far as constrained real-time performance is concerned. Much work needs to be done towards this end.

Psychological validity: Ours is a cognitive modeling-based approach as opposed to solving specific problems related to specific AI tasks (e.g. plan critiquing etc).

7 On Sharing Knowledge and Control

All knowledge (world model, actions, goals, plans) is represented in the form of agent's beliefs in the SNePS semantic network formalism. This enables a single component to perform reasoning about different kinds of knowledge. Having an ATMS as an integral part of our system has facilitated several benefits. We have been able to exploit the idea of

maintaining assumptions underlying derived beliefs to design cleaner (and in our view correct) representations for situational effects of actions as well as conditional plans.

The deductive engine is designed for efficient retrieval of derived beliefs, i.e. beliefs once derived are stored as long as their support is still present, thus subsequent retrieval needn't repeat the deductive process. Of course this works correctly only in conjunction with the ATMS.

Another major advance of our approach is the representation of actions and plans. All actions, preconditions, effects, and plans are represented in the same formalism. We do not require a different language and a representation to perform different tasks relating to these.

Integrating inference and acting using the transformer formalism facilitates traditional reasoning as well as planning representations. In addition, low level (stimulus-response) type of reactivity forms a natural part of the formalism. This in our opinion is another important criteria that contributes to the appeal of the idea of integrated acting and inference.

8 Other Approaches

HOMER [Vere and Bickmore 1990] integrates limited natural language understanding and generation, temporal planning and reasoning, plan execution, simulated symbolic perception, episodic memory, and some general world knowledge. The actions of HOMER are represented using state transition semantics which is a frame-like description of verbs augmented by a description of the effect of the action on the world state. Thus, the agent maintains a dual representation of executable actions—a linguistic model; and a planner's model. Their approach is to amalgamate various task modules in a single system. Our goals are similar. However, as discussed above, we are taking a unified approach to integrating various tasks.

SOAR [Laird *et al.* 1987] is a general architecture for problem-solving and learning. Our approach is based on the premise of modeling a rational cognitive agent. Problem-solving and learning are two among other tasks such agents should be capable of performing. Our agent's problem-solving and learning capabilities are not as sophisticated as SOAR's. However, our approaches are quite different in that ours is based on a deductive system that can reason about beliefs (which represent facts, rules, acts, and plans).

References

- [Choi 1990] Joongmin Choi. Fast implementation of non-standard quantifiers in sneps. In Deepak Kumar, Hans Chalupsky, and Syed S. Ali, editors, *Current Trends in SNePS—Proceedings of the 1990 Workshop*, Forthcoming 1990.
- [Fikes and Nilsson 1971] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5:189–208, 1971.
- [Georgeff 1987] Michael P. Georgeff. Planning. In *Annual Reviews of Computer Science Volume 2*, pages 359–400. Annual Reviews Inc., Palo Alto, CA, 1987.
- [Kumar *et al.* 1988] D. Kumar, S. Ali, and S. C. Shapiro. Discussing, using and recognizing plans in SNePS preliminary report - SNACTOR: An acting system. In *Proceedings of the Seventh Biennial Convention of South East Asia Regional Confederation*, pages 177–182, New Delhi, India, 1988. Tata McGraw-Hill.
- [Kumar 1989] D. Kumar. An integrated model of acting and inference. In D. Kumar, editor, *Current Trends in SNePS—Semantic Network Processing System: Proceedings of the First Annual SNePS Workshop*, pages 55–65, Buffalo, NY, 1989. Springer-Verlag.
- [Laird *et al.* 1987] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987.
- [Martins and Shapiro 1988] J. P. Martins and S. C. Shapiro. A model for belief revision. *Artificial Intelligence*, 35(1):25–79, 1988.
- [McCarthy 1986] John McCarthy. Mental situation calculus. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge—Proceedings of the 1986 Conference*, page 307, 1986.
- [Sacerdoti 1977] Earl D. Sacerdoti. *A Structure for Plans and Behavior*. Elsevier North Holland, New York, NY, 1977.
- [Shapiro *et al.* 1989] S. C. Shapiro, D. Kumar, and S. Ali. A propositional network approach to plans and plan recognition. In *Proceedings of the 1988 Workshop on Plan Recognition*, page 21, Los Altos, CA, 1989. Morgan Kaufmann.
- [Shapiro 1989] S. C. Shapiro. Formal foundations of an intensional propositional semantic network. Presented at the Workshop on Formal Aspects of Semantic Networks, Santa Catalina Island, CA, 1989.
- [Vere and Bickmore 1990] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.
- [Wilkins 1988] David E. Wilkins. *Practical Planning—Extending the Classical AI Planning Paradigm*. Morgan Kaufmann, Palo Alto, CA, 1988.