Toward an Amalgamated Model of Inference and Action

Stuart C. Shapiro
Department of Computer Science
and
Center for Cognitive Science
State University of New York at Buffalo
Buffalo, New York 14260
shapiro@cs.buffalo.edu

November 12, 1991

Chapter 1

Project Description

1.1 Introduction and Objectives

We propose to investigate the relationships among beliefs, plans, effective acts, and sensory acts, and between reasoning and acting, in the context of a computational rational agent.[1] We expect this investigation to result in an integrated intelligent agent architecture with a representational formalism, semantics, and inference/acting mechanism that unifies these subjects while clearly maintaining the proper distinctions among them.

The behavior of modeled rational agents should be driven by their beliefs, goals, and intentions. They should be capable of reasoning as well as acting in the real world. This research addresses the problem of designing and implementing representations and reasoning mechanisms required to model rational cognitive agents. We propose to design and explore representations of an agent's beliefs, actions, plans, and intentions. Using these representations, we propose to develop an intelligent agent architecture with appropriate reasoning and acting techniques that would allow the agent to reason about, act, and react based on its representations. This work will proceed by investigating the relationship between beliefs, inference, intentions, and acting. The underlying thesis of this research is that an approach based on the relationship between inference and acting will provide a better framework for modeling rational agents because of the following:

- The modeled agent will be able to represent and reason about its beliefs and those of others.
- The agent will be able to represent and reason about actions and plans the same way as it does about its beliefs.
- Using these representations, the agent will be able to plan its future actions.
- The agent will be able to perform the intended actions in a real world.
- It will use its representations and beliefs to react to events happening around it.
- The agent will be able to discuss its actions, plans, and beliefs with other agents in the world.

Past research in knowledge representation and reasoning has largely ignored issues relating to planning and acting and similarly research in planning/ acting has proceeded without concerns for the issues addressed by the knowledge representation and reasoning researchers. As a result, existing good knowledge representation and reasoning systems are bad planners/actors and vice versa. This research will attempt to synthesize an integrated representation, reasoning, and acting system that will provide appropriate mechanisms to deal with sensory acts, external events, and how the behavior of a rational agent can be affected by them. It will build upon and, where necessary, modify the existing representational framework of SNePS[39, 45] which is considered a highly acclaimed knowledge representation and reasoning system.

---

[1] A computational rational agent is a computer model of an agent that can reason about its beliefs, goals, and actions, and is capable of affecting (and reacting to) the external world.

1.2 Desiderata For Rational Agents

Any model of a rational cognitive agent should be able to do the following:

- Reason about its beliefs and those of others. For example, it should be able to understand sentences like "If a block is on a support then the block is not on another support." " All men are mortal." "John believes that Socrates is a man." " Block A is on top of block B.", "Landuse is a polygon coverage", etc., and use its understanding of these to answer queries like "Who is mortal?" " Is block A on top of block C?" "Is Socrates mortal?", etc.
- Plan future actions and events. For example, it should be able to use the information contained in these sentences: "A plan to achieve that a block is held is to pick up the block." "A plan to pile a block on another block on a third block is to put the third block on the table and then put the second block on the third block and then put the first block on the second block." "If a polygon coverage is displayed then a plan to identify a region is to say *identify* and then say the coverage and then tell *Move the mouse over the coverage until cross hairs appear and then choose an area by clicking on it* and then send *polys* * and then interpret the region on the coverage."
- Actually perform actions in the external world upon request. For example, "Pile A on B on C." " Pile A on B on C and then pick up the red block." "Identify this region."
- Perceive and react to events happening around it. For example, "If you drop something, pick it up." "In case of a fire, sound the alarm and leave the building."
- Perceive and react to actions performed by other agents. For example, when "John picked up block A" is reported it should at least update its model of the world.
- Understand its own acts and those of others, as in "Before picking up a block, make sure that it is clear." "Before crossing the street, make sure the walk signal is lit." "After plotting a coverage the coverage is displayed." "John put block A on block B."
- Reason about its actions and plans.
- Discuss its actions and plans with other agents around it. For example, "How would you plot a polygon coverage?" "How would you identify a region?" "What happens when someone picks up a block?"

It is clear that for an agent to be able to plan, act, react, and reason about its actions, it needs to be endowed with a set of beliefs, goals, and intentions. Beliefs can change based on sensory input from the dynamically changing environment and lead to new intentions. Reports of events happening in the world can lead to new (or revised) beliefs, which in turn can lead to the formation of new intentions. This suggests that a closer relationship needs to be established between reasoning about beliefs and actions, and the process of planning and acting. We propose to evaluate existing models of inference (i.e., the representations of rules of inference, and the mechanisms of forward and backward inference) as well as current models of planning and acting (i.e., the representations of operators and plans, and plan execution and monitoring). Such a study will yield a better model of knowledge representation, reasoning, and acting and solve most of the problems mentioned above.

1.3 Domains and testbeds

The examples cited above are taken primarily from three domains— a blocksworld, where blocks and supports are the objects manipulated by the agent's arm; a multi-modal user interface to a state-of-the-art geographical information system (GIS) called ARC/INFO, where the modeled agent acts as a liason between a human user and the GIS; and the general domain of knowledge representation for natural language. Since we are proposing to develop representational and reasoning formalisms for a generic architecture of a rational cognitive agent we plan to use the modeled agent in several diverse domains. There has been recent interest in developing simulated domains for experimenting agent architectures[53, 32]. Such testbeds provide dynamic,

System	World Model	Operators/Actions	Plans
STRIPS	FOPL wffs	Operator schema	Linear operator lists
NOAH	QLISP expressions	SOUP functions	Procedural Network
NONLIN	???	Task Formalism	Procedural Network
SIPE	FOPL wffs	Operator Definition Language	Procedural Network
OPLAN	???	Task Formalism	Procedural Network

Table 1.1: Representational formalisms employed by classical Planners

yet controllable environments in which agents can be placed so as to facilitate empirical studies of their modeled behavior. We propose to use at least one such existing doman, the Tileworld[32], in addition to the ones mentioned above. Our modeled agents will be capable of performing all of the tasks mentioned in the previous section in all of the testbed domains.

1.4 Related Work

Early research on planning [10, 38, 51, 56, 52] was centered around the idea of representing actions as operators. Initially, a plan was represented simply as a list of instantiated operators[10]. This evolved into hierarchical representations of actions and non-linear and conditional plan representations[38, 51, 56]. Some mechanisms for reasoning about plans were also developed[38, 51, 52]. This research on planning has come to be called *classical planning*. Some of the more recent approaches to planning and acting have attempted to solve the problems concerned with reactivity and sensory inputs using alternative control paradigms. They have tried to use more advanced notions of belief, knowledge, and reason maintenance in their modeling strategies. However, the scope of most of these models is restricted to planning and acting. Research in belief representation and reasoning for natural language as well as in general still proceeded in isolation from most of the planning work. First, we will survey some classical approaches, which will then be followed by an overview of some current techniques.

1.4.1 Classical Planning

We will begin by examining the so called classical AI planning paradigm[57]. Under this paradigm, a planning system is modeled using a world model, a set of acts, and a truth criterion procedure. Classical planning systems traditionally use three different levels of representations:

- representations of world models
- representations of actions
- representations of plans

Three different representational formalisms (schema) are usually employed. A consequence of such an approach is that the modeled agent has to use three different "reasoners" to perform reasoning:

- reasoning about the world model
- reasoning about actions
- reasoning about plans

Reasoning about a world model is assumed to correspond to reasoning about the agent's beliefs. The planner (or the planning module) reasons about actions when formulating plans. Reasoning about plans is normally done by *critics*. This involves reasoning about the structure of plans (or about an agent's intentions). What follows is a survey of some planning systems to illustrate this point.

Table 1.1 summarizes the representational formalisms used by some of the representative classical planning systems. STRIPS [10] uses a resolution theorem prover for answering queries about the world model and a means-ends-planner for reasoning about operators while constructing plans. Plans generated by STRIPS were linear lists of instantiated operators and there was no plan reasoning component. Reasoning about plans in NOAH[38] is done using various *procedural critics* using a mechanism called TOME (Table Of Multiple Effects) for identifying the conflicts. The TOME is also used for replanning purposes. The declarative *task formalism* of NONLIN[51] is based upon the operator schemas of STRIPS. The structure of plans is similar to that of NOAH. Reasoning about plans is done using a TOME and a GOST (Goal Structure) to detect and resolve conflicts. As an improvement over NOAH, it has a declarative representation of operators; it allows typed preconditions; it allows alternatives to be explored; and it introduces a better question-answering procedure that deals with queries in a partially ordered network of nodes. O-PLAN[52] uses similar but more generalized representations than the ones used by NOAH and NONLIN. The control structure of O-PLAN uses a blackboard scheme. SIPE's [55, 56, 57] representations are richer and more flexible than NOAH's. Actions are described in an *Operator Description Language (ODL).* Procedural networks similar to NOAH's are used for representing plans. SIPE can represent some world knowledge using operators in the form of *domain rules.* However, these rules only aid in making the operators more general and are used (by the question-answering procedure called *the truth critereon procedure*) only in deducing context-dependent effects of actions. Reasoning about plans is done using an improved version of TOME-based procedural critics.

In all these systems, actions are represented by *operators.* The only purpose that an operator serves is that of specifying changes from one state to another when the corresponding action is performed. Such a representation is useful only to a planner. It is totally divorced from the set of beliefs of the modeled agent. Since the formalism used to represent operators is different from that used to represent beliefs the agent cannot have any explicit beliefs about operators. Consequently it does not *understand* actions as it does its "beliefs." It can use and perform actions using the planning and execution modules. If an agent is to be able to reason about other agent's actions, it should be able to represent them and even discuss them. Having different representations for beliefs and actions leads to different reasoning modules. The same argument goes against having a different representation for plans.

In contrast to these systems, we intend to have one reasoning module that is able to reason about beliefs, actions, and plans. In order to do this, we intend to represent all of them in a uniform manner while maintaining appropriate semantic distinctions. We intend to design and implement a system capable of having beliefs about actions and at the same time able to use them to plan and act. However, this is only the first step towards an integrated model of acting and inference.

1.4.2 Current Approaches

An attempt to represent world states, plans, and reasoning rules in a uniform way was first made by McDermott [25]. He developed a theory of planning and acting, which is implemented in the NASL problem solver. The state of the world, tasks and plans, the state of the problem solver, and the rules governing them are represented in a database of predicate-calculus statements. Plans are represented in a declarative task network schema. The interpreter works on assigned tasks and uses a theorem prover for deductive retrieval of facts as well as instantiated plan schemas (decompositions). The NASL interpreter is capable of reasoning about these relationships so as to execute subtasks in the specified order.

The interaction between knowledge and action was first explored by Moore [27, 28], who used modal logic with a possible-worlds semantics to develop a formal integrated theory of knowledge and action. He argues that it is necessary to explicitly reason about what knowledge is needed to carry out a plan and how that knowledge can be obtained. He explained that one needs to consider knowledge prerequisites in addition to physical prerequisites for reasoning about actions. Using his logic, one is able to reason about the knowledge prerequisites (for example, "knowing the combination for a safe in order to open it") by knowing what action to take (e.g., "dial the combination of the safe") and deciding whether one knows how to perform an action ("open a safe at any given time"). The theory is also able to handle how agents infer new knowledge as a result of an action (like being able to infer the acidity of a solution by observing the results of a litmus paper test).

Allen (see [3, 1, 2]) uses a temporal reasoning framework to reason about actions and plans. In this framework, a world model consists of all of the agent's knowledge of the past and present, and predictions about the future expressed in an interval-based temporal logic. Actions are represented in a STRIPS-like fashion along with temporal augmentations. A plan is a collection of assertions viewed as an abstract simulation of some future world. In a complete plan ("useful solution"), the initial and final states have causal connections. The task of planning constitutes decomposing actions and doing constraint propagation over temporal relationships to determine causal connections between initial and the final states. Thus, reasoning about actions is done in the classical sense (à la STRIPS) and reasoning about plans is accomplished via constraint-based temporal reasoning. Reasoning about temporal constraints allows the problem-solver to determine constraints on action ordering (as opposed to using procedural critics à la NOAH). Domain rules for the world model are also specified using temporal domain constraints. Allen's formalism allows for representation of overlapping and simultaneously occurring actions, and actions involving non-activity.

Drummond[7, 8] has used concepts from net theory to develop the notion of *plan nets.* The plan net formalism was designed to allow for descriptions of an action's sensory results (e.g., a weight sensor in the robot arm would record the weight of an object when the robot performs a *pickup* action on that object). In addition, the representation of action and belief is able to describe iterative and conditional behaviors. Operators and plan nets are defined using a formal language. The representation of operators facilitates the specification of physical and mental results of the denoted actions. Drummond also describes a mechanism for analyzing plan behaviors. He does not address the issue of plan generation. And, the discussions apply only to single-agent plans.

Nilsson[31]is developing an architecture called *action networks* for modeling agents whose behavior is guided by their goals and changing environmental conditions. He calls such agents *teleo-reactive.* The modeled agent is supposed to have real-time control of some environment with which it is connected via sensors and effectors. The control program is an action network modeled using combinational logic circuits. The agent's beliefs are represented by a *belief structure*, which is a collection of binary-valued beliefs. Similarly, there is a set of propositions that forms the *goal structure* of the agent. Actions are represented by a special kind of logical gate called an *action unit.* An action network is created for each task. Nilsson is developing a programming language called *ASTRAL* that can be used to describe actions. Action networks can then be created from ASTRAL programs just as procedural nets are created from SOUP code in NOAH. This architecture is being developed to solve the problems of "real-time AI". The work is in preliminary stages and it is too early to draw any kind of conclusive results from it. One of the major concerns that needs to be solved, Nilsson mentions, is how the connection between a general purpose reasoning system and such an acting system is to be achieved (see [31] p. 49).

Vere [53] has constructed a "basic agent"(HOMER) that integrates limited natural language understanding and generation, temporal planning and reasoning, plan execution, simulated symbolic perception, episodic memory, and some general world knowledge. HOMER is a simulated autonomous underwater cognitive agent that behaves as a submarine in a simplified simulated Seaworld. The temporal task planner and reasoner are based on DEVISOR V[54]. The actions of HOMER are represented using *state transition semantics* which is basically 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 for executable actions— a linguistic model; and a planner's model.

Another approach discussed in [19] suggests that we should view STRIPS as a form of logic, and STRIPS operators as rules of inference in this logic. Here, the notion of planning is governed by the operators representing the rules of inference and that of acting is defined by associating action-functions with each operator. These functions, when interpreted, will model the appropriate state transitions for each action. While it is good to take this approach to prove formal properties about the model, it is not clear as to how these ideas relate to the physical realization of such logics, in particular, how the rules of inference are used by the inference mechanisms to exhibit planning (and decision making). Moreover, there is no mention of how they relate to the inference rules of the world model.

1.4.3 BDI-proposals

Recently there have been attempts at developing models of rational agents that are attributed with the psychological attitudes of belief, desire, and intention (BDI).

Review Cohen and Levesque........[6]..

In the PRS system[12, 11, 14, 13], Georgeff and Lansky use the notion of a *process* (also called *Knowledge Areas*) to represent plans (or plan schemas). A process is a declarative procedure specification that can be used to generate a sequence of world states (called *behaviors*). A process is modeled by a transition network whose nodes represent control points and whose arcs are labeled by *action descriptions.* A *process assertion* expresses a declarative fact about the effects of performing a certain sequence of actions under certain conditions. Some of the preconditions of process assertions can be used to specify *reactive invoking conditions*; i.e., the associated process will be invoked as soon as the system believes the invoking condition. This is responsible for PRS's reactive behavior. An *interpreter* manipulates these representations by maintaining a stack of invoked processes that represents the system's intentions.

Add Bratman stuff here as a lead....[4]....

Pollack *et al* [5, 32] have proposed a high-level specification of the practical reasoning component of an architecture for a resource bounded rational agent. Their work concentrates mainly on the control paradigm of an architecture of a practical reasoning agent that has to perform means-ends-analysis, weigh competing alternatives, and address the problem of resource boundedness during deliberation.

Review Rao and Georgeff.....[34]...

Review John Pollock....[33]...

1.4.4 Observations

In the research surveyed above, there are clear distinctions in the approaches used and the problems attacked. Moore and Lifschitz are concerned about formal aspects of reasoning about knowledge and action. Their models stress the semantics of the representations used. Allen, Drummond, Nilsson, Georgeff, and Pollack are exploring different architectures or models to approach the problem of reasoning about actions and beliefs. The choice of models used is motivated by different specific requirements demanded of the modeled agents. Allen developed a temporal logic to be able to reason about actions and events in a temporal framework; Drummond uses net theory to account for acquisition of sensory information and also to model conditional and iterative behaviors; Nilsson is using action nets to model teleo-reactive agents; and Georgeff uses procedural networks for modeling real-time process control agents. McDermott developed an informal theory of planning and acting and described it using the NASL interpreter. Vere is trying to build autonomous modeled agents that integrate various state-of-the-art AI faculties. Pollack's architecture is designed to use the agent's plans to control the amount of further deliberation in resource bounded situations.

While all the approaches represent major advances in planning, acting, and reasoning about actions and beliefs as described, the need for an integrated reasoning and acting system as pointed out in Section 3.2 still remains unfulfilled. Moore's theory provides a model for reasoning about knowledge and action, but that is only part of the picture. Such a reasoning agent should ultimately perform and discuss the actions it reasons about. There is no attempt to formalize reactivity in the model, which is a major concern for "effective agents." We propose to provide a model of acting where actions of the agent can lead to other actions, and where acquisition of new beliefs may also lead to actions.

A few things about BDI architectures...

1.4.5 Our Previous Work

SNePS, the Semantic Network Processing System [39, 45, 43, 42, 46] has so far been designed and developed to be a system for representing the beliefs of a natural-language-using cognitive agent. It has always been the intention of its developers that a SNePS-based "knowledge base" would ultimately be built, not by a programmer or knowledge engineer entering representations of knowledge in some formal language or data entry system (although this can also be done), but by a human informing it using a natural language (NL) (generally supposed to be English), or by the system reading books or articles that had been prepared for human readers. Because of this motivation, the criteria for the development of SNePS have included: it should be able to represent anything and everything expressible in NL; it should be able to represent generic, as well as specific information; it should be able to use the generic and the specific information to reason and infer information implied by what it has been told; it cannot count on any particular ordering among pieces of information it is given; it must continue to act reasonably even if the information it is given includes circular definitions, recursive rules, and inconsistent information. Our achievements on these accounts, too numerous to mention here (some of them are summarized in [46]), have led us to claim that SNePS is a fully intensional, propositional semantic network processing system and stands as the only implemented system that has been in existence for more that two decades [37].

Recently we have been involved in extending the SNePS paradigm to model rational cognitive agents who can also plan and act. Our representations of plans and acts described in [40, 16, 49, 50] are such that an agent can be made to reason about beliefs as well as actions and plans in a uniform manner. The acting subsystem, called SNACTor (for the SNePS actor) is described in [16, 49, 50, 17, 18]. The architecture of the SNePS actor is as shown in Figure 1.1. The SNePS actor operates in a world inhabited by itself (i.e., a single-agent world). The agent has beliefs that are stored as SNePS propositions in the agent's belief space (called a *context*, see [23]). For example, the proposition "A is on top of B" is represented by a node (say M23) whose structure is $M23 \equiv \{\langle \mathtt{rel}, \{ON\}\rangle, \langle \mathtt{arg1}, \{A\}\rangle, \langle \mathtt{arg2}, \{B\}\rangle\}$. A pictorial representation of M23 is shown in Figure 1.2.[2] The node is stored in the agent's belief space as a *supported wff* containing an *origin tag*, an *origin set*, and a *restriction set* (see [20, 35, 45, 44]). Thus, $\langle M23, HYP, \{M23\}, \{\}\rangle$ represents the fact that M23 is a supported wff in the agent's belief space. It is a hypothesis whose origin set contains the

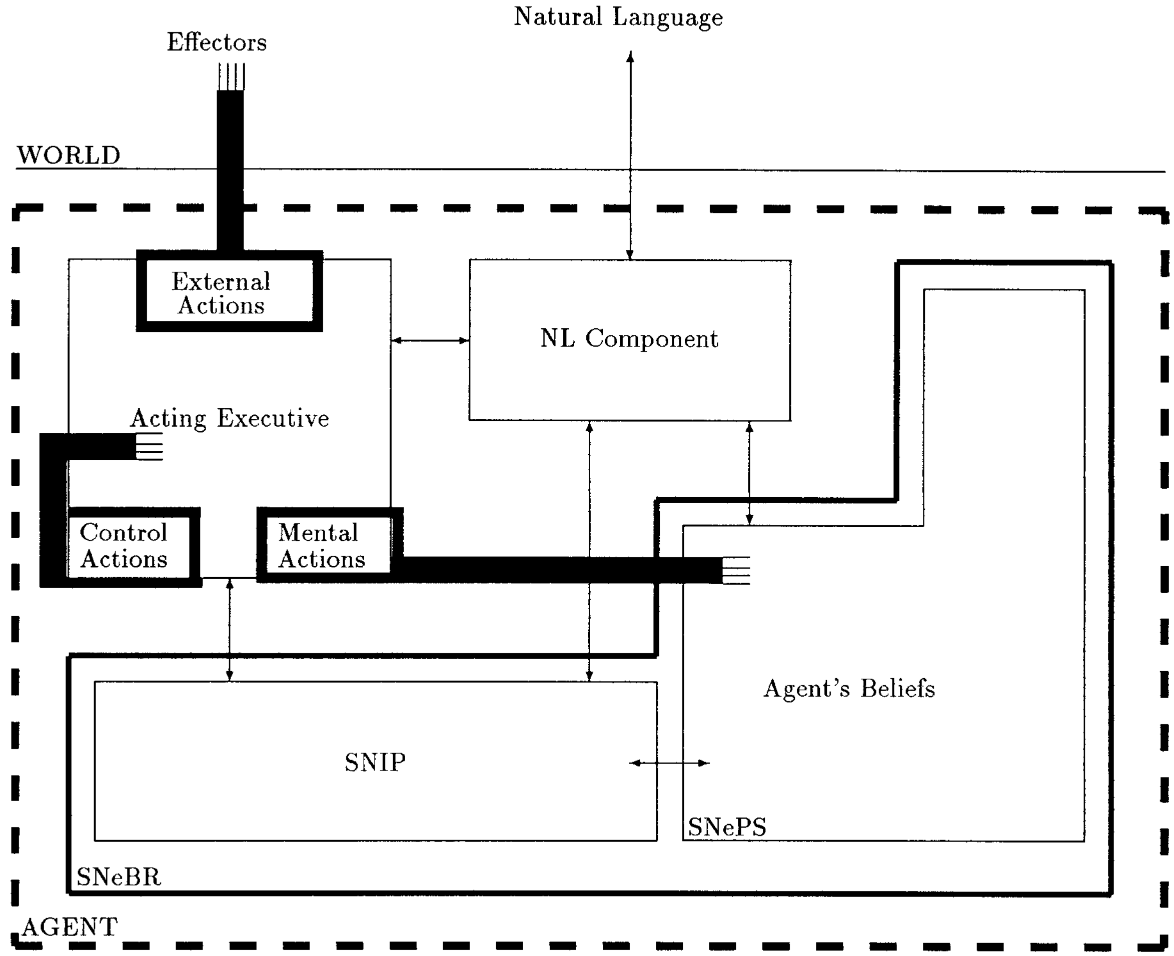

Figure 1.1: Architecture of The SNePS Actor

---

[2] We will refrain from pictorial representations of subsequent representations to save space.

node itself, and its restriction set is empty. This enables SNeBR—the SNePS system for Belief Revision, an assumption-based truth maintenance system [22, 21, 23], to ensure that the agent's belief space is always consistent.[3] An important feature of this representation is that M23 is interpreted as a "mental entity" (or an "intensional object") (see [41]). The agent can have beliefs about all intensional objects. All interaction with the agent is done using the natural language component. Sentences are parsed by a grammar (written as an ATN), translated into SNePSUL(the SNePS User Language) commands, and form beliefs in the agent's belief space. World-model rules for reasoning in the agent's belief space are also translated and represented as agent's beliefs. An inference rule in SNePS is a structured proposition node of the form[4]
$\{\langle$**&ant**$, \{$<antecedent-beliefs>$\}\rangle, \langle$**cq**$, \{$<consequent-beliefs>$\}\rangle\}$
Roughly, the above rule is a specification of antecedent and consequent beliefs using appropriate quantifiers and connectives. SNIP, the SNePS Inference Package, can do forward, backward, or bidirectional inference using the same set of rules.

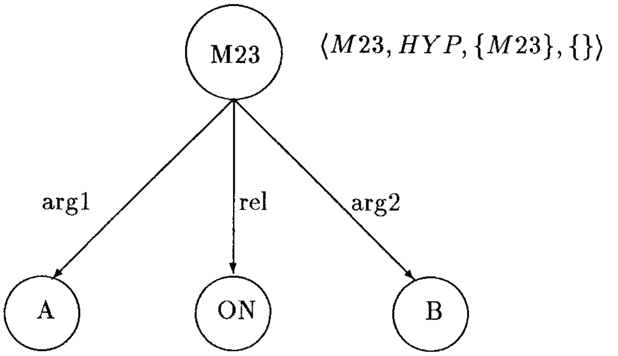

Figure 1.2: A Pictorial representation of the supported wff M23

We treat acts and plans as mental objects. This enables the agent to discuss, formulate, use, recognize, and reason about acts and plans (see [50]). This is a significant advance over operator-based descriptions of plans. Our representations for acts, goals, and plans build upon and add to the intensional propositional representations of SNePS. This framework enables us to tackle various tasks in a uniform and coherent fashion.

We classify actions as being *external*—those that affect the outside world; *control*—those that affect the acting executive; and *mental*—those that affect the set of beliefs. Plans (or complex acts) comprising a set of external and control actions are represented as structured nodes. Decompositions of plans/goals are specified using the following propositions[5]
$\{\langle$**plan**$, \{$<some-plan-i>$\}\rangle, \langle$**goal**$, \{$<some-goal>$\}\rangle\}$
$\{\langle$**plan**$, \{$<some-plan-j>$\}\rangle, \langle$**act**$, \{$<complex-act>$\}\rangle\}$
These propositions represent that <some-plan-i> is a plan for achieving <some-goal>, and that <some-

---

[3] During the course of acting, beliefs are removed and added. This is done using SNeBR operations. For example, one of the things SNeBR takes care of is that when a belief is removed as a consequence of performing an action, all propositions derived using that belief are also removed.

[4] This linear representation of SNePS rules is designed to facilitate our current discussion. In SNePS rules one can have universal, existential, and numerical quantifiers over variables. The connectives available are and-entailment, or-entailment, numerical entailment, and-or, thresh, and non-derivable. The predicate used here represents only the typical antecedent-consequent type of rules. However, the discussion applies to all SNePS rules in general. See [39, 45, 43] for details on the SNePS representation of rules.

[5] The exact syntax and semantics of these representations can be found in [16, 50]

plan-j> is a decomposed plan for doing <complex-act> respectively. Effects of acts are represented as
{⟨**act**, {<some-act-i>}⟩, ⟨**effect**, {<effects-of-act-i>}⟩}
specifying that <effects-of-act-i> are the effects of <some-act-i>. This, when used by the acting executive, specifies mental actions of believing to be performed so as to update the set of beliefs after performing an act. Acts can also have preconditions that are specified as
{⟨**precondition**, {<preconditions-of-act-i>}⟩, ⟨**act**, {<some-act-i>}⟩}
Figure 1.3 shows the network representation of a *Puton* action. Requests to perform an action are serviced by the acting executive (see Figure 1.1). The request (which is represented as an act node) gets scheduled on an acting queue maintained by the executive. This represents the agent's intentions. Plans are structured using *control actions* that, when interpreted, affect the queue of intentions. Our repertoire of control actions includes sequencing (**snsequence**), conditional (**snif**), iteration (**sniterate**), and a few others (see [16]). *External actions* affect the external world via their respective associated procedures. The acting executive uses SNIP to derive plans, plan decompositions, and the effects and preconditions of actions. It schedules *mental actions* to believe effects of actions. It also schedules acts to achieve preconditions of actions in case they are not satisfied.

Figure 1.3: The SNePS Representation of *Puton*

SNIP, the SNePS Inference Package[15, 9], is implemented on top of MULTI (a LISP based multiprocessing system) [26]. MULTI is a simulated multiprocessing system that maintains a queue of parallel processes, executing processes from the process queue until the queue is empty. An inference is carried out by SNIP by activating nodes[6]. The activated nodes pass messages to send/request/recieve information to each other in order to accomplish an inference. The underlying system of logic is the SWM system [23] which, as a natural deduction system, is based on a set of introduction and elimination inference rules in addition to *Modus Ponens* and *Generalization*. As mentioned earlier, SNIP can perform forward, backward, and bidirectional inference.

1.4.6 Observations On The SNePS Actor

The SNePS actor is a model of an agent that is a "pure effector" living in a world populated only by itself. However, a rational agent has to be able to act in a real world inhabited by other agents (effector systems).

---

[6] In the case of forward inference, the first node activated is the belief to be added to the belief space of the agent. In the case of backward inference, it is the node representing some query.

Also, the world is being affected by the constant occurrence of natural phenomena. To account for these, rational agents should be endowed with sensory capabilities so they can update their beliefs about the world. Regardless of the sensory interface between the agent and the world, sensory information results in the formation of new beliefs. These beliefs are acquired on a continuous basis. Of course there can also be sensory activity intended by the agent in order to gain some information. A model of a rational agent should be able to account for both these types of sensory activity.

A model of attention is required to filter beliefs acquired by sensory activity. As a consequence, some beliefs will simply be added to the agent's belief space. However, adding of some beliefs may require the agent to perform some reactive actions. An agent's reactions are governed by its "desires". Thus, there is a need for a model that can use the beliefs and desires of an agent to create new intentions to perform some actions.

1.5 Proposed Approach

By exploring how beliefs and goals can lead to intentions and actions, we propose to extend current models of knowledge representation and reasoning systems. This will involve a clarification of relationships between representations for beliefs, acts, and plans, and the traditional models of inference and acting. Though the need for integrating inference and acting has been recognized, and some attempts have been made to formalize various aspects of knowledge and action, integrated systems built around these theories are yet to appear.

In the SNePS actor (and the framework of classical planning and acting), the modeled agent is always under the control of an acting executive. Once the agent is asked to do something, it would do it; only after it is done and has nothing left to do in its intention queue would it wait for another command. Also, if in the middle of carrying out its intentions, the world changes, it would have no way of detecting that change, however crucial it may be to the completion of its current plans. (The STRIPS triangle tables can be used to successfully complete a plan, but the agent will never know that some of its actions are correcting earlier failures.) The agent may intend to do something and try to do it, but a failure of that act would result in undefined consequences. To deal with such issues, a mechanism is needed that would enable an agent to reason about or respond to a new belief in the middle of doing whatever its current intentions are. Reacting to a new belief may result in a new immediate intention for performing some action (which may or may not be related to its current plan). This requires a mechanism that would allow us to model the forming of an intention as a result of some new belief entering the agent's current set of beliefs. Currently in SNePS, adding a new belief can lead to forward inference by chaining through the antecedents of the rules that get triggered (or are already active because of some earlier bidirectional inference). However, this forward inference will only lead to the formation of new beliefs, since reasoning rules in SNePS only pass a belief status from the antecedents to the consequents (or the other way round in the case of backward inference).

There are some similarities between the architecture of the SNePS actor and SNIP, the inference package. As first suggested by [29], inference can be looked at as the sequence of actions performed in applying rules to derive beliefs from other beliefs. In the SNePS actor model, this can be represented as a mental action. Thus SNIP can be viewed as a *mental actor*.

Ensuing from the earlier observations and the above discussion that inference can be viewed as a specialized form of acting is the suggestion that acting and inference are closely related. This is main motivation underlying our proposed research.

1.5.1 Integrating Acting And Inference

Earlier, we introduced the notion of mental actions, which are used to update or retrieve an agent's beliefs. This may enable 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 has also suggested that inference can be treated as a mental action. As mentioned in [41], when a rule fires, the agent forms the intention of believing its consequences. This suggests that we may be able to integrate our models of inference and acting by eliminating the acting executive. The inference queue can serve as the queue of intentions. While this may clarify the notion of an inference rule as specifying an act, we will still need to reexamine representations for plans and acts and define the role they might play under the influence of inference procedures (forward/backward chaining). For that purpose, we propose to investigate the more general notion of a *transformer.*

1.5.2 Transformers

A *transformer* is a representation that specifies a belief or act transformation under the influence of a transformation procedure. It has two parts—($< a >, < b >$), where both $< a >$ and $< b >$ can specify either a set of beliefs or some act. Basically a transformer is a more general representation that may aid to capture the notions of reasoning as well as acting. We will need to redefine inference in terms of a transformation procedure that will be able to use transformers in forward as well as backward chaining fashion. Using a transformer in forward chaining is equivalent to the proposed interpretation "after the agent believes or intends to perform $< a >$, it believes or intends to perform $< b >$." A transformation procedure using backward chaining on a transformer yields the interpretation "if the agent wants to believe or perform $< b >$, it must first believe or perform $< a >$."

Since both $< a >$ and $< b >$ can be sets of beliefs or acts, we will 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. In this research we will explore the possibility of reclassifying our representations as transformers. Then we will be able to determine the role they will play in planning, acting, and reasoning.

1.5.3 The SNePS Acting And Inference Package

Using the idea of a transformer to define the notions of planning, acting and inference in a unified framework is expected to yield a simpler integrated intelligent agent architecture. An architecture of such a model is depicted in Figure 1.4. On comparison with Figure 1.1 we find that the natural language component and the SNePS modules for agent's beliefs will still be present. The acting executive will be integrated into the SNePS acting and inference package (SNAP). We will preserve the current natural language component and representation of agent's beliefs. We will also preserve the syntax of rules and other planning and acting-related propositions. However, they will belong to the more general class of transformers. We expect no need to have a separate acting executive and inference engine. Transformers will be interpreted and used by the SNePS acting and inference package. We propose to modify SNIP to use the inference queue as the integrated acting and inference queue. In addition to existing SNIP processes (that facilitate inference), we will also have act processes that, when scheduled on the queue, represent the agent's intentions to perform those actions.

1.5.4 Examples Revisited

As outlined in Section 2 above, let us review the tasks we would like a modeled rational agent to perform and how they may be represented and accomplished using the proposed model.

Sentences like "If a block is on a support then the block is not on another support" and "All men are mortal" may be represented using the *belief-belief* transformers. If the belief "Socrates is a man" is added to the agent's set of beliefs, forward chaining will enable the belief "Socrates is mortal" to be added. Questions like "Who is mortal?" and "Is Socrates mortal?" can be answered using backward chaining. Similarly, given the belief that "Block A is on top of block B" questions like "Is block A on block C?" can be answered by backward chaining.

The sentences "If you drop something, pick it up" and "In case of a fire, sound the alarm and leave the building" will be represented using *belief-act* transformers. As discussed earlier, forward chaining through such transformers upon acquiring the required beliefs ("dropped block A", "there is a fire") will lead to the formation of an appropriate intention to do the specified acts ("pick up block A", "sound the alarm and leave building"). Note that backward chaining will have to be blocked through these!

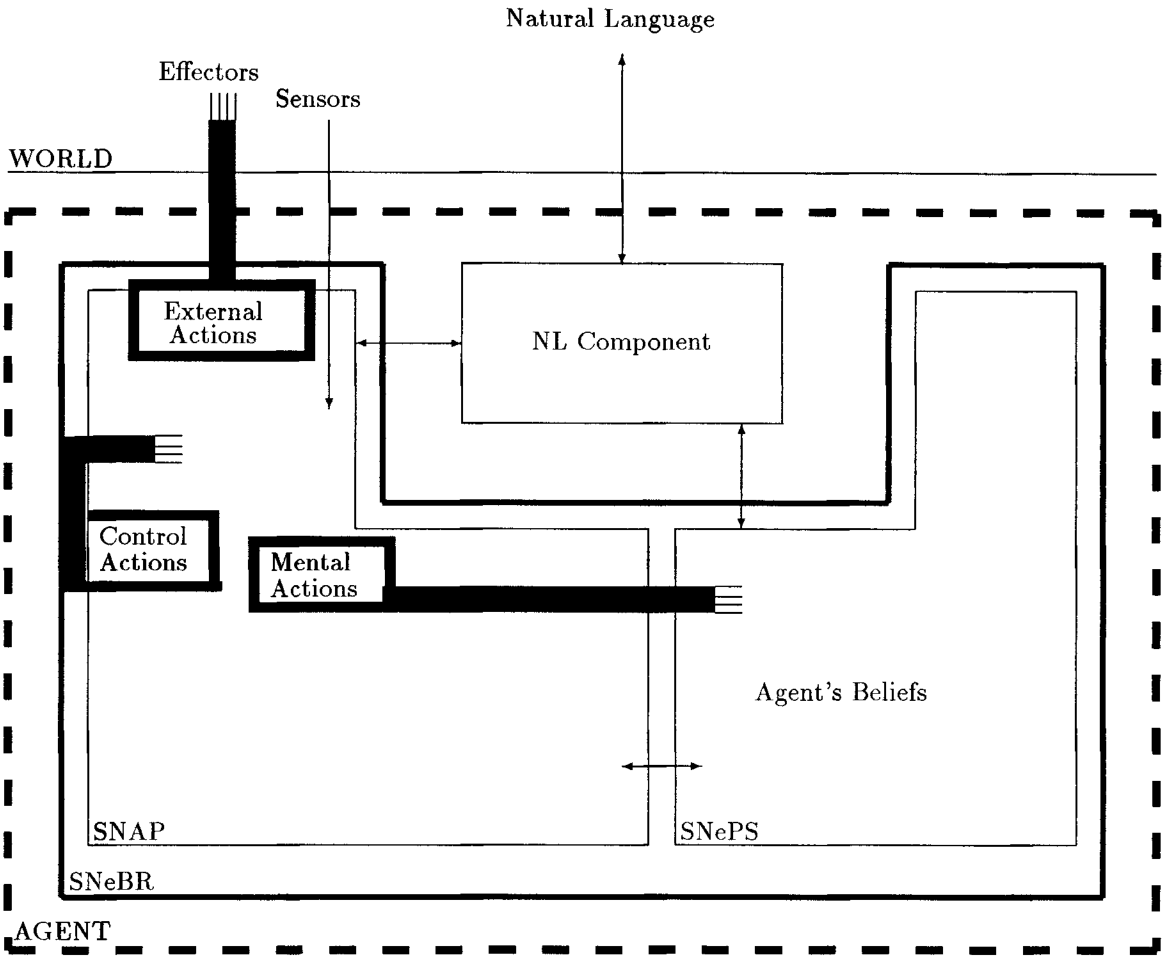

Figure 1.4: Architecture of the integrated acting and reasoning system.

The precondition proposition will be used to represent the sentences "Before picking up a block, make sure that it is clear" and "Before crossing the street, make sure the walk signal is lit" respectively. Similarly, the sentence "After putting a block on another block the latter block is not clear" will be represented using the effect proposition. Used in forward chaining, these representations will specify the effects of performing actions. Plans to achieve goals as specified in sentences "A plan to achieve that a block is held is to pick up the block" or decompose complex acts into simpler ones as in "A plan to pile a block on another block on a third block is to put the third block on the table and then put the second block on the third block and then put the first block on the second block" will be expressed as before.

1.6 Proposed Research

We have proposed to investigate the relationships among beliefs, plans, effective acts, and sensory acts, and between reasoning and acting, in the context of a computational rational agent. As mentioned above, this research will lead to an integrated acting and inference system. The relationship between inference and acting as outlined above will be the chief motivation underlying our research.

We propose to design and explore appropriate mechanisms that would facilitate representations of an agent's beliefs, actions, plans, and intentions. We propose to develop appropriate reasoning and acting techniques that would allow the agent to reason about and act based on its representations. These representations and reasoning and acting mechanisms would provide reactive capabilities for the agent to respond to changes in its environment while it is in the process of carrying out its tasks.

We propose to use transformers as unified representations for capturing the notions of acting and inference. We propose to give a semantics for transformers such that the logic underlying inference as well as acting is formally specified. This implicitly involves laying the formal foundations of node-based inference in SNePS, something that has not yet been done. Since we are proposing to define inference as mental activity, we will have to use the notions of intention and acting to specify the semantics of our representations. Thus, we expect, that the ideas of rational agency will be included in the formal foundations of the resulting model. We will investigate the possibility of modeling Moore's logic of knowledge and actions in our framework. This will inherently extend as well as provide a working implementation of his theory. We will also compare our work with the various theories of imperative logics[36, 30].

Based on our formal model, we propose to modify the current implementation of SNePS along the lines of Figure 1.4. This will require that we get rid of the SNePS actor, and modify SNIP to conform to the definitions of the transformers. As mentioned earlier the design of SNIP is based on a multi-processing regime. We will attempt to preserve this feature while developing the proposed architecture.

We will design and implement a facility for concurrent input of external sensory activity. Once this is implemented, We will be able to demonstrate how beliefs and goals of an agent can be transformed into intentions and actions using the integrated model. We will demonstrate all the capabilities outlined above in a modeled agent.

1.7 Plan Of Work

If this proposal is funded, we expect the research personnel to be Stuart C. Shapiro (P.I.), Deepak Kumar (graduate research assistant the first year, post-doctoral associate the second year), and Henry Hexmoor (graduate research assistant both years). Shapiro and Kumar have previously cooperated on research on representing and reasoning about acts, which has been the work that has lead to the current proposal. Hexmoor has recently joined Shapiro's department as a graduate student, after having worked on planning research at another laboratory. (See the biographical sketches.) Kumar will have principal responsibility for the development of the proposed theory. Hexmoor will be principally responsible for implementation and testing. His experience with planning under a different paradigm will valuable for rigorous testing of our ideas. Shapiro will provide overall supervision and direction, and will help with both theory development and implementation, both areas in which he has extensive experience.

In Artificial Intelligence research, theory does not always precede implementation, rather experimenting with an implementation often gives direction to theory development. We intend to follow this methodology, and begin both by implementing and experimenting with transformers, as well as by developing their theory.

For our work, we will be using, and, where necessary, modifying SNePS 2.1. SNePS 2.1 is implemented in Common Lisp, and runs on any platform that supports Common Lisp. In particular, it is running on all the AI research computers in our department. Most of these computers are SUN workstations. We also have a few ageing Lisp Machines. After some investigation, we identified SUN Microsystems' Symbolic Programming Environment (SPE) as the most reasonable development environment on a general-purpose platform, and expect to use this for our proposed work. Our department already has already obtained a site license for SPE for use in our AI courses.

Chapter 2

Bibliography

Bibliography

[1] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–54, 1984.

[2] James F. Allen and Henry A. Kautz. A model of naive temporal reasoning. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*, pages 251–269. Ablex Publishing Corp., Norwood, NJ, 1985.

[3] James F. Allen and J. A. Koomen. Planning using a temporal world model. In *Proceedings of 8th IJCAI 1983*. IJCAI, 1983.

[4] M. Bratman. *Intentions, Plans and Practical Reason.* Harvard University Press, 1987.

[5] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence*, 4(4), 1988.

[6] P. R. Cohen and H. J. Levesque. Persistence, Intention, and Commitment. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans - Proceedings of the 1986 Workshop*, pages 297–340, Los Altos, CA, 1987. AAAI and CSLI, Morgan Kauffmann.

[7] M. Drummond. Refining and extending the procedural net. In *Proceedings of the 9th IJCAI 1985*, pages 528–531. IJCAI, August 1985.

[8] Mark E. Drummond. A representation of action and belief for automatic planning systems. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans - Proceedings of the 1986 Workshop*, pages 189–212, Los Altos, CA, 1987. AAAI and CSLI, Morgan Kauffmann.

[9] Carlos P. Ferreira, Nuno J. Mamede, and Joao P. Martins. SNIP 2.1 — The SNePS Inference Package. Technical report, Instituto Superior Technico, 1989. In preparation.

[10] 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.

[11] M. P. Georgeff, A. Lansky, and P. Bessiere. A procedural logic. In *Proceedings of the 9th IJCAI*, 1985.

[12] M. P. Georgeff, A. Lansky, and M. Schoppers. Reasoning and planning in dynamic domains: An experiment with a mobile robot. Technical Note 380, AI Center, SRI International, 1987.

[13] Michael P. Georgeff. An embedded reasoning and planning system. In Jay Webber, Josh Tenenberg, and James Allen, editors, *Advance Proceedings of The Rochester Planning Workshop-From Formal Systems to Practical Systems*, pages 79–101, October 1988.

[14] Michael. P. Georgeff and Amy. Lansky. Procedural knowledge. Technical Note 411, AI Center, SRI International, 1987.

[15] R. G. Hull. A new design for SNIP the SNePS inference package. SNeRG Technical Note 14, Department of Computer Science, SUNY at Buffalo, 1986.

[16] 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.

[17] Deepak Kumar and Stuart C. Shapiro. Architecture of an intelligent agent in SNePS. *SIGART Bulletin*, 2(4):89–92, August 1991.

[18] Deepak Kumar and Stuart C. Shapiro. Modeling a rational cognitive agent in SNePS. In B. Barahona, L. Moniz Pereira, and A. Porto, editors, *EPIA 91: 5th Portugese Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence 541*, pages 120–134. Springer-Verlag, Heidelberg, 1991.

[19] Vladimir Lifschitz. On the semantics of STRIPS. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans - Proceedings of the 1986 Workshop*, pages 1–10, Los Altos, CA, 1987. AAAI and CSLI, Morgan Kauffmann.

[20] A. S. Maida and S. C. Shapiro. Intensional concepts in propositional semantic networks. *Cognitive Science*, 6(4):291–330, 1982. Reprinted in R. J. Brachman and H. J. Levesque, eds. *Readings in Knowledge Representation*, Morgan Kaufmann, Los Altos, CA, 1985, 170–189.

[21] J. P. Martins and S. C. Shapiro. Belief revision in SNePS. In *Proceedings of the Sixth Canadian Conference on Artificial Intelligence*, pages 230–234. Presses de l'Université du Québec, 1986.

[22] J. P. Martins and S. C. Shapiro. Theoretical foundations for belief revision. In J. Y. Halpern, editor, *Theoretical Aspects of Reasoning About Knowledge*, pages 383–398. Morgan Kaufmann Publishers, Los Altos, CA, 1986.

[23] J. P. Martins and S. C. Shapiro. A model for belief revision. *Artificial Intelligence*, 35(1):25–79, 1988.

[24] John McCarthy. Mental situation calculus. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge—Proceedings of the 1986 Conference*, page 307, 1986.

[25] D. McDermott. Planning and acting. *Cognitive Science*, 2:78ff, 1978.

[26] * D. P. McKay and S. C. Shapiro. MULTI: A LISP based multiprocessing system. Technical Report 164, Department of Computer Science, SUNY at Buffalo, 1980. (Contains appendices not in item 15).

[27] R. Moore. Reasoning about knowledge and action. Technical Report 191, SRI AI Center, 1980.

[28] Robert C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex Publishing Corp., Norwood, NJ, 1985.

[29] E. J. Morgado and S. C. Shapiro. Believing and acting: An approach to meta-knowledge and meta-reasoning. SNeRG Technical Note 11, Department of Computer Science, SUNY at Buffalo, 1983. Superseded by 59.

[30] Hector-Neri Casta neda. *Thinking and Doing.* D. Reidel, Dordrecht, Holland, 1975.

[31] Nils J. Nilsson. Action networks. In Jay Webber, Josh Tenenberg, and James Allen, editors, *Advance Proceedings of The Rochester Planning Workshop-From Formal Systems to Practical Systems*, pages 21–52, October 1988.

[32] Martha E. Pollack and Marc Ringuette. Introducing the Tileworld: Exprimentally Evaluating Agent Architectures. In *Proceedings of the Eighth National Conference in AI 1990*, pages 183–189. AAAI, July 1990.

[33] John L. Pollock. *How to Buid a Person: A Prolegomenon.* MIT Press, Cambridge, MA, 1989.

[34] Anand S. Rao and Michael P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In *Principles of Knowledge Representation and Reasoning— Proceedings of the Second International Conference(KR91*, pages 473–485. AAAI, IJCAI, CSCSI, April 1991.

[35] * W. J. Rapaport. Syntactic semantics: Foundations of computational natural-language understanding. Technical Report 86–24, Department of Computer Science, SUNY at Buffalo, 1986. Superseded by 102.

[36] Nicholas Rescher. *The Logic Of Commands.* Monographs in MOdern Logic. Routledge and Kegan Paul Ltd., London, 1966.

[37] Charles Rich. Special Issue on Implemented Knowledge Representation and Reasoning Systems—Letter from the Guest Editor. *SIGART Bulletin*, 2(3), June 1991.

[38] Earl D. Sacerdoti. *A Structure for Plans and Behavior.* Elsevier North Holland, New York, NY, 1977.

[39] S. C. Shapiro. The SNePS semantic network processing system. In N. V. Findler, editor, *Associative Networks: The Representation and Use of Knowledge by Computers*, pages 179–203. Academic Press, New York, 1979.

[40] S. C. Shapiro. Representing plans and acts. In *Proceedings of the Third Annual Workshop on Conceptual Graphs*, pages 3.2.7–1 – 3.2.7–6. The American Association for Artificial Intelligence, Menlo Park, CA, 1988.

[41] 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. Superseded by [47]., 1989.

[42] S. C. Shapiro. Case studies of SNePS. Technical Report 90–29, Department of Computer Science, SUNY at Buffalo, 1990. Superseded by [48].

[43] S. C. Shapiro and The SNePS Implementation Group. *SNePS-2 User's Manual.* Department of Computer Science, SUNY at Buffalo, 1989.

[44] S. C. Shapiro and J. P. Martins. Recent advances and developments – the SNePS 2.1 report. In D. Kumar, editor, *Current Trends in SNePS–Semantic Network Processing System: Proceedings of the First Annual SNePS Workshop*, pages 1–13, Buffalo, NY, 1989. Springer-Verlag.

[45] S. C. Shapiro and W. J. Rapaport. SNePS considered as a fully intensional propositional semantic network. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier*, pages 263–315. Springer–Verlag, New York, 1987.

[46] S. C. Shapiro and W. J. Rapaport. The SNePS family. Technical Report 90–21, Department of Computer Science, SUNY at Buffalo, 1990.

[47] Stuart C. Shapiro. Cables, paths and "subconsious" reasoning in propositional semantic networks. In John F. Sowa, editor, *Principles of Semantic Networks*, chapter 4, pages 137–156. Morgan Kaufmann, San Mateo, CA, 1991.

[48] Stuart C. Shapiro. Case studies of SNePS. *SIGART Bulletin*, 2(3):128–134, June 1991.

[49] Stuart C. Shapiro, Deepak Kumar, and Syed S. Ali. A propositional network approach to plans and plan recognition. In *Proceedings of the AAAI-88 Workshop on Plan Recognition.* Morgan Kaufmann, 1989. To appear.

[50] Stuart C. Shapiro, Beverly Woolf, Deepak Kumar, Syed S. Ali, Penelope Sibun, David Forster, and Scott Anderson. Discussing, using, and recognizing plans–Annual Report for 1988. Technical report, North-East Artificial Intelligence Consortium, 1989.

[51] A. Tate. Generating project networks. In *Proceedings 5th IJCAI*, pages 888–93, 1977.

[52] Austin Tate. Goal structure, holding periods and "clouds". In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans - Proceedings of the 1986 Workshop*, pages 267–278, Los Altos, CA, 1987. AAAI and CSLI, Morgan Kauffmann.

[53] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.

[54] Steven A. Vere. Temporal scope of assertions and window cutoff. In *Proceedings of 9th IJCAI*, pages 1055–1059. IJCAI, 1985.

[55] D. E. Wilkins. Representation in a domain-independent planner. In *Proceedings of the 8th IJCAI*, 1983.

[56] D. E. Wilkins. Domain-independent planning: Representation and plan generation. *Artificial Intelligence*, 22:269ff, 1984.

[57] David E. Wilkins. *Practical Planning–Extending the Classical AI Planning Paradigm.* Morgan Kaufmann, Palo Alto, CA, 1988.