SRI International

Technical Note No. 481 • May 1990

A SURVEY OF AI APPROACHES TO THE INTEGRATION OF INFORMATION

Thomas D. Garvey, Associate Director
Artificial Intelligence Center

Originally prepared as Report 78211 for the Conference on Infrared Sensors and Sensor Fusion (SPIE-87)

SRI International

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Abstract

The integration of information is a central issue for Artificial Intelligence research and development. The *inference process* in AI is the fundamental mechanism for combining information, and a significant aspect of most AI systems is the means by which they manage their overall workload by focusing processing attention and controlling which inferences are drawn and when it is appropriate to draw them. Several perspectives on the control of inferential processes and their access to information have evolved. One view of the problem treats the task as a goal-driven perceptual process, where specific information is explicitly sought from the world through selected sensor modalities, translated into a common "vocabulary," fused with other relevant information, and finally translated back into an understanding of critical aspects of the environment. Another view, centers on a flexible structure known as the blackboard architecture for enforcing control and communication activities. In this paper, we first review briefly a variety of AI inference techniques, focusing primarily on logical inference and uncertain reasoning methods. We conclude with a survey of approaches used to control inference processes, to mediate their access to real world information, and to schedule their activities.

1 Introduction and Overview

A motorist stops a pedestrian in a suburban area and asks for directions to a particular address. The pedestrian tells the motorist to continue down the street for several blocks, until he passes a shopping center on the left, take the first right, then a left, and his destination will be across from the train station. After repeating the directions to ensure he has them, the motorist continues on his way.

A cat hears a faint rustling in the tall grass a few feet away. Crouching down, she continues to scan the area. When she sees the flash of motion out of the corner of her eye, she springs.

The radar seeker head in a missile scans a wedge-shaped area in front of it. When a significant return from the radar's signal is detected, the radar processor analyzes the signal and, based on its amplitude and spatial extent, determines that it is a target. It narrows its field of view to include just the area containing the target, and begins a turn toward the target area.

Each of these activities requires that information from one or more sources be integrated in order to provide a picture of the environment that can be effectively used to accomplish the goals and objectives of the "system" acquiring the information. The motorist must integrate the sounds of the speaker into a model in order to interpret the utterances; this may be aided by pointing gestures on the part of the pedestrian. An interpretation of the utterances yields an abstract, specialized map of the area which must, in turn, be integrated into the spatial model of the environment already possessed by the motorist. The cat integrates sounds of movement with a visual indication of motion. The sound provides *cueing* information which allows a more precise system, with a narrower field of view, to be used effectively. The radar seeker must integrate electromagnetic energy until it determines that a possible target is present. This information is then integrated with a file of target descriptions to determine whether it is of interest.

An effective understanding of the environment based on the integration of information is central to any sort of purposive behavior. Humans excel in their ability to integrate information from a wide variety of sources, simple and complex, and draw accurate and far-ranging conclusions from them. This is primarily due to their ability to effectively integrate data from distinct sources into an all-encompassing framework. Artificial intelligence (AI) researchers use human activities as models of numerous types of intelligent behavior, including sensing and perceiving, interpretation and reasoning, planning and problem-solving, and language understanding and speech. Each of these general areas involves the integration of information as a central theme. AI approaches to problems in these areas typically use one or more *inference* methods as the basic mechanism for integrating information.

Inference, a fundamental AI operation, is the process of drawing a new conclusion from two or more pieces of information. The most fundamental logical inference technique, *modus ponens*, uses a statement of fact and a related implication statement, to generate a new statement of fact. For example, the integration of the statement, SUNNY-DAY, and the rule (SUNNY-DAY *implies* GO-TO-THE-BEACH), asserts GO-TO-THE-BEACH.

Complications arise when it is important to control which conclusions get drawn, or more importantly, to ensure that certain conclusions get drawn when necessary. One of the most important tasks in any AI system is to control the exploration of a potentially huge space of possible inferences, by focusing towards "useful" ones.

In this paper we shall discuss some of the methods that AI practitioners use for drawing inferences, both boolean, where all conclusions will be either true or false, or weighted by their likelihood of being true. We will then discuss general methods for controlling an inference process. We will conclude with a brief discussion of system issues.

1.1 The Role of Artificial Intelligence

An AI approach[1] offers a number of benefits for problems involving the integration of various types of information, both in the development of a system and as a component in the subsequent operation of the system. These benefits include:

- **Flexibility** – By making inferences and decisions explicit, an AI system is (typically) easier to change and update than a conventional system accomplishing an equivalent task. Furthermore, if the target environment is highly dynamic or complex (or poorly understood), an AI system can offer the flexibility necessary to adapt to the particular conditions holding when it is employed.

- **Ability to handle complex problems** – Recognition and assessment of real-world activities can result in highly complex interactions among pieces of data, with multiple interpretations possible in general. AI offers effective representations and computational methods for managing this complexity.

- **Understandability** – By making decisions and chains of inference explicit, an AI system is often able to provide explanations for its findings and actions that are not feasible with algorithmic approaches.

The utility of AI techniques for use in a fielded system depends roughly on the complexity of the sources of information that must be accessed and the environmental states that may need to be recognized. These may be characterized as follows:

[1]In this paper, we shall use the term *AI* approach to indicate a solution method based primarily on the use of explicit AI formalisms, which typically include inference and search. The terms *algorithmic* approach and *conventional* approach will be used interchangeably to mean non-AI methods.

- The sources of information and their organization are well understood with respect to the information provided and the types of errors expected. An engine control system that monitored temperature and pressure in order to regulate fuel flow would fall into this category.
- The number and types of sources are bounded, but the specific data provided and the possible types of errors in the data are not predictable in detail. A military battlefield information system would be an example of this.
- The sources of information and their errors are essentially unbounded. A system for interpreting natural language utterances about widely varying topics would fit into this partition.

In the first case, techniques for integrating information may generally be prespecified, along with methods necessary for eliminating noise and refining estimates. Activities that are based upon this types of information may be thought of as *instinctive*. The role for AI is relatively small in these applications. In the third case, the amount of background knowledge, and the uses to which it might be put are essentially unpredictable. These problems, while certainly requiring intelligent behavior, are significantly beyond the current operational state-of-the-art in AI. In the second case, however, there is enough prior knowledge about the environment and available information sources to constrain the activities of the system, while requiring a flexible, intelligent capability for combining, manipulating, and interpreting information. Problems in this class seem most appropriate for the application of AI techniques, and will form the focus of this paper.

A number of other factors will influence the detailed choice of methods used to acquire and integrate information. These include, for example, the use to which the information will be put, the number, types and quality of information sources, the degree to which the sources are understood in advance, the complexity of the environment being monitored, the dynamics of the environment, and the degree and quality of environmental knowledge available to the integration system.

1.2 Knowledge, Information and Representation

The types of knowledge and information that may be available or useful to an information-integration process are likely to be quite varied. Raw data in an image might represent intensity, color, range, texture, surface shape, surface type, elevation, optical density, x-ray absorbance, and geometric relationships. Different types of data will suggest different representations. For example: intensities are simply represented as numeric values of continuous quantities; texture might be best represented as a certain statistical distribution of intensity; color might be represented either as a vector of intensity values for the primary colors, red, green, and blue, or in terms of hue, saturation, and brightness; simple surface shapes might be represented as superquadrics[ref pentland]; complex surfaces may be represented by collections of small patches or by articulated combinations of simple surfaces; and surface type might be represented by a set of symbolic names. The knowledge used for understanding the information in the image might be derived from a variety of experience and prior knowledge and include declarative statements such as, "The color RED signifies danger," imperative knowledge such as "To find intensity edges, convolve the image with the Laplacian operator," procedural knowledge such as that embodied (at a low level) in an image-operator description, or (at a high level) in an analysis *script*, and causal knowledge such as "The intensity at a point is a combination of the incident intensity, the local surface orientation, and the reflectance function of the surface."

In addition, the information available may have a variety of other qualities. In particular, information is often uncertain, incomplete, inaccurate, and ephemeral (i.e., it ages). It may be acquired out of sequence with other information. It will frequently vary in "granularity" or level of abstraction. The sources of information may be either dependent or independent, inconsistent with other sources, of varying reliability, and few or numerous. The amount of information that must be handled may be appropriate for the available processing resources and problem requirements, potentially overwhelming, necessitating some means for throttling the data to a manageable level, or insufficient for the task, requiring some means for either inferentially filling information gaps or an effective method for acquisition of additional information.

A wide variety of applications have information integration components where the information sources and the necessary interpretations are bounded and amenable to AI approaches. Examples include autonomous vehicle navigation, personnel identification systems, access-control and monitoring systems, business management systems; battle management systems, intelligence analysis; photinterpretation, cartography and map-making, medical diagnosis, and military threat detection and warning systems.

2 Generic System Architecture and Definitions

In general, we shall treat the architecture shown in Figure 1 as an informal, generic perception/action model, that links sensing and effecting through an inferential information integration module (IM). This model joins a collection of information *producers* to a set of information *consumers* (users). Each producer will be associated with a single source of environmental information, but will have one or more distinct output channels (a multi-channel producer will be said to be *multi-modal*). These channels will be treated as distinct (although often highly correlated) information inputs to the IM. Inputs to the producers from the IM will include control information, commands, and parameters. In a *multi-level* or *hierarchical* system, each producer may itself be composed of a complete perception/action process as described here.

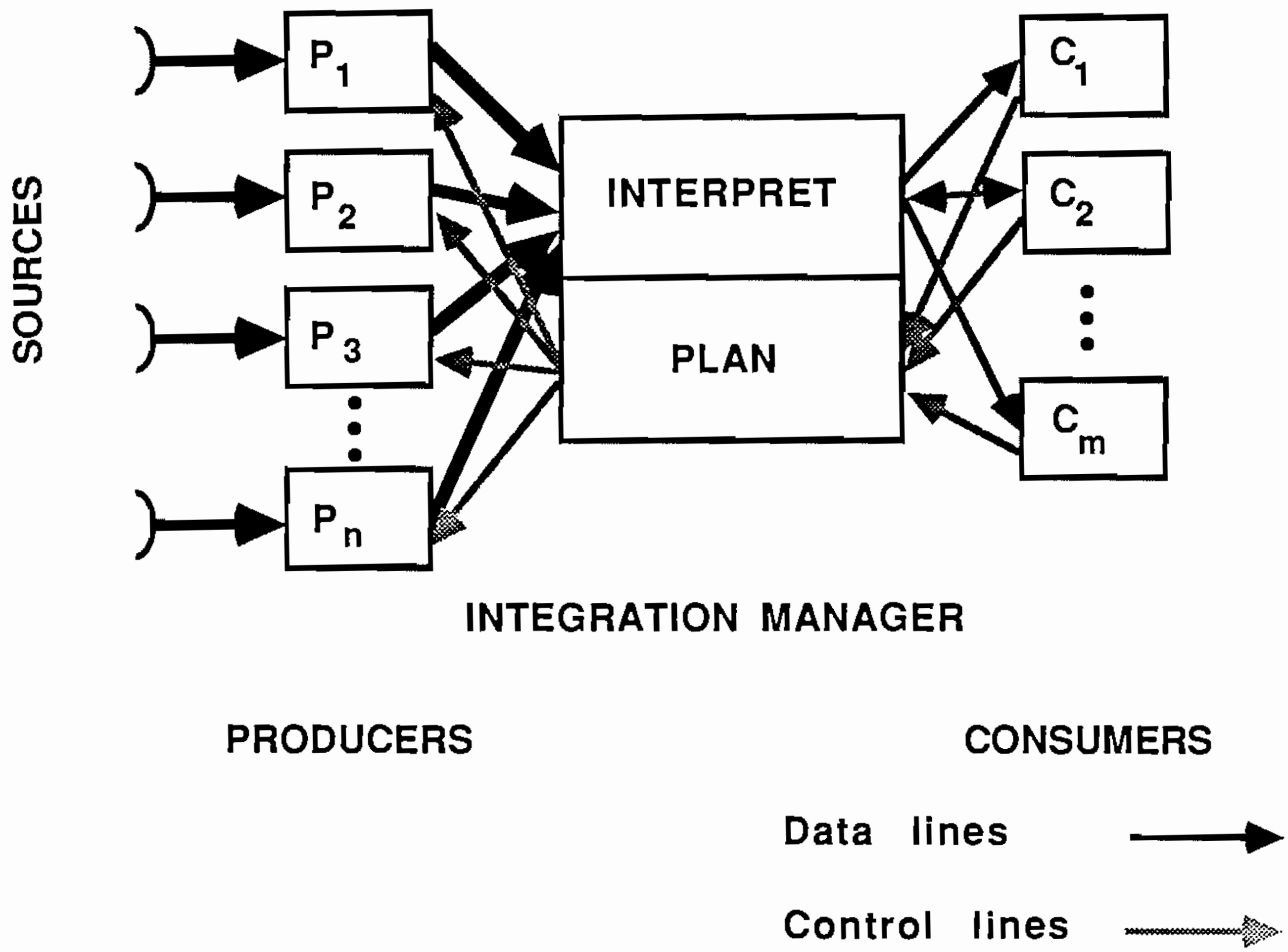

Figure 1: The Prototypical Perception/Action Process

Integrated outputs from the IM are provided to the consumers, who will determine their behavior based on the information they receive. Consumers may make information requests of the IM. These requests are further processed by the IM, and appropriate instructions will then be issued to the producers. A consumer may also be a producer. Feedback occurs through adaptive information requests from the consumers.

The IM may be functionally partitioned into two components, an integration and interpretation part and a planning and control part. The integration and interpretation component processes and combines information received from the producers before updating appropriate databases and making the processed information available to the consumers. The planning and control unit interprets consumer information requests in light of overall system goals and requirements, plans and schedules producers' activities, and closes the loop by feeding necessary control and processing information back to the producers.

This simple architecture captures the essence of a wide variety of systems. Three principle sub-architectures of the generic architecture cover most current and contemplated systems: the single-source, single-mode (channel) system, the single-source, multimode system, and the multisource system.

A radar tracker for a missile is an example of the single-source, single-mode system. It integrates reflections from a target into a model of the target's motion dynamics. It continuously maintains its model by predicting a new position, comparing acquired information with its predictions, and updating its model. Even this simple system must control its access to information,[2] integrate the information into a model, and adjust its model to accept new information.

A radar system that computes both range and velocity information is an example of a single-source, multimode system. In this case, signal processed information is integrated into two separate dynamic models, which in turn yield two distinct (correlated) channels of information. These channels may then be used by completely separate consumer processes. The human optic system is another single-source, multimode system which produces several distinct channels of information providing spatial, topological, and temporal information.

The multisource, multimode system is exemplified by many human organizations.[3] Intelligence gathering and interpretation, both by Government and commercial organizations, is a key instance of this type of system. A simpler example would be a combined threat warning system consisting of (say) a radar intercept receiver, a laser warning receiver, and an IR detector. The integrated results of processing could include information for warning a crew member of a possible threat, as well as detailed information for use by an automatic countermeasures system.

[2] A common radar technique is to use a *range gate* to limit when it looks for returned pulses.

[3] As well as by cats!

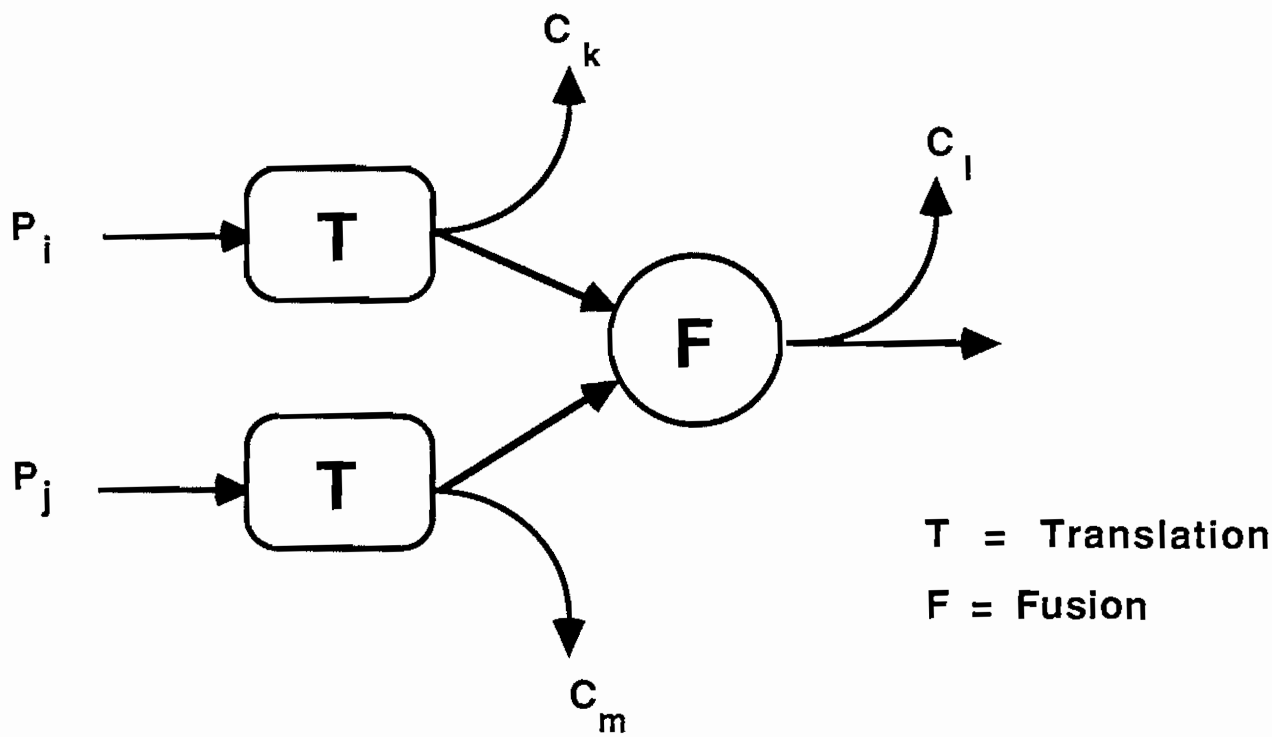

Figure 2: The Primitive Information Integration Process

In Figure 2, we illustrate a prototypical view of an information integration process. Two distinct information manipulation processes are shown: *fusion* (*integration* or *combination*) processes and *translation* (*interpretion*)[4] processes. This simple network can be thought of as an *inference module.* Inputs come from producers; outputs got to consumers. These modules may be combined by identifying the consumers for one module as the producers for the next. A common view of the information integration process as moving from "signal to symbol" emphasizes the concept of the data moving through ever more complex (and abstract) interpretations.

2.1 The Information-Integration Process

Information acquired from external sources must normally be combined with a model and *interpreted* in order to draw conclusions about the world. For example, a radar system makes a direct measurement of the intensity of the electromagnetic field at its antenna; an interpretation of the electromagnetic radiation may yield the conclusion that a real radar pulse was received. A further interpretation of the pulse information

[4]Translation can also be thought of as a process of *synchronizing* information.

could lead to the determination that an aircraft was the source of the energy. This interpreted information is the output typically provided to a user. It is rare that raw information (particularly that which is received directly from sensors) describes the world in ways that are of direct interest to a user. In one sense, the information is almost always in the wrong "language," and a *translation* operation is required to convert the information to a statements that are meaningful; that is, to a form that can be integrated with the current model of the situation. This leads to an important observation: information is not combined directly with other information, but is integrated into an evolving *model* of some aspect of the situation. The current characteristics of the model are a function of information received earlier.

The process of integrating information, then, can be envisioned as one of first translating the information to an appropriate framework, and then combining it with other information in the same framework. This process is appropriate regardless of the number and types of the sources. A key ingredient in the process of integrating information is the inferential framework which defines the interactions between different types and sources of data.

The process of acquisition and assimilation of information may require significant effort, and therefore, must be effectively managed to ensure adequate information for achieving goals. Acquisition and integration of new information into a model is a goal-driven, purposeful process with attendant requirements for planning and reasoning capabilities, in order to be carried out effectively.

The information-integration process described above raises issues that must be addressed in the design and implementation of such a system. First, based upon the nature of the information available to the process and the uses to which the derived information is to be put, the designer must select an appropriate representation of that information (for example, information expressed as a series of precise statements will be represented differently from information expressing weighted judgements). Second, based on the information sources, the process goals, and the characteristics of the environment, the appropriate means for controlling the process must be determined. Third, the necessary means for manipulating the available information and drawing inferences from it for interpretation and translation, as well as the means necessary for converting from one representation to another as appropriate must be determined. The remainder of this paper discusses these issues.

3 Inference

A fundamental AI computation is that of combining information in order to *infer* a conclusion. Inference procedures are designed to work on a particular representation of available information. A variety of representations and methods of inference have proven useful for AI problems. The selection of a particular method is ordinarily based on the type of information available to the reasoning process, the types of outputs required, the computational complexity of the available mechanisms, and the degree to which assumptions are necessary and possible in order to carry out the computation.

The basic inferential modes addressed by AI researchers are logical inference and uncertain inference. Formal logical inference methods center on propositional and the predicate logics. They are particularly useful for reasoning about situations where available information is *boolean*, that is statements are either true or false. Logical methods are typically based on formal properties which make it possible to characterize the results of their operations. For example, in a predicate calculus system, it is guaranteed that any statement which can be deduced from axioms (statements that are true by assumption or definition) using valid rules of inference is a theorem, and therefore true. Formal logical systems are *consistent*; if a statement is a theorem, its negation cannot be. They are also *complete*; any statement that can be proved from a consistent set of axioms is a theorem in the system. At the same time, predicate calculus is not *decideable*; there are statements which can be neither proven or disproven. Logical systems traditionally have traded ease of expressiveness and computational simplicity for formality.

Uncertain inference methods are based on various models of probability theory. Such method typically have an underlying propositional structure which allows statements to be made. In addition, they provide a means for representing uncertainty about whether the statement is strictly true and sometimes provide ways of representing imprecision or ignorance.

3.1 Logical Inference

The style of logical reasoning most commonly employed is *deductive*, and can be best exemplified by *theorem proving*. A proof of a theorem is derived by creating a chain of inferences from axioms to the theorem of interest. In this regard, theorem proving involves a search process, where candidate statements are combined in an attempt to get "closer" to the target theorem. The key characteristic of deductive reasoning is that the results, by the nature of the computations, are known to be valid and consistent.

3.1.1 Propositional logic

The propositional calculus[27] deals with constant statements (or *propositions*), which are known to be either true or false. Propositions may be either atomic propositions consisting of a single statement, or more complicated statements composed of other propositions joined by valid connectives. The legal connectives in the propositional calculus are conjunction (represented as AND or "$\vee$"), disjunction (OR or "$\wedge$"), negation (NOT or "$\neg$"), and implication (IF or "$\Rightarrow$").

The fundamental rule of inference in propositional logic is *modus ponens*, which states that $\mathcal{B}$ is a direct consequence of $\mathcal{A}$ and $\mathcal{A} \Rightarrow \mathcal{B}$. Thus, an indication from a visual sensor, LIGHT-IS-RED, may be combined with an implication (LIGHT-IS-RED $\Rightarrow$ DANGER) to infer DANGER.

3.1.2 Predicate calculus

By virtue of its restriction to constant statements, propositional logic is limited in its ability to express complex concepts[10, 3]. While retaining the connectives from propositional calculus, the first-order predicate calculus provides additional expressive power, by permitting the use of terms, variables, functions, predicates, and quantifiers.

Predicates denote properties and relations among objects, and can take on a value of true or false when their arguments are specified. An *atomic formula* is a predicate with the terms that constitute its arguments. A *literal* is a single (negated or unnegated) predicate and its terms. A *term* is either a constant, a variable, or a function application. A *variable* is a symbol that stands for an unspecified object. *Functions* represent arbitrary, fixed expressions whose arguments are terms.

For example, the expression $BLUE(x)$ is a literal that states that x, an unspecified variable, is blue. The expression, $BLUE(SKY)$ is true (in general) for California skies, while $BLUE(TREE)$ is false (in general). Quantifiers come in two forms, universal and existential. Universal quantifiers (written "$\forall$" and pronounced "for all") permit the statement that something is true for all possible values of a variable. Existential quantifiers (written "$\exists$" and pronounced "there exists") allow the assertion that a statement is true for at least one possible value of a variable. Existential and universal quantifiers may be combined in the same expression.

Thus, we may represent the statement, "Where there's smoke, there's fire," by the predicate calculus statement,

$$\forall x(SMOKE(x) \Rightarrow \exists y(FIRE(y) \wedge SAME\text{-}LOC(x,y)))$$

Predicate calculus uses the *Rule of Universal Instantiation* in addition to *modus ponens* as its fundamental rules of inference. *Modus ponens* is exactly the same for propositional and predicate logic. Universal instantiation merely states that universally quantified variables may be replaced by constants in a theorem, and a new theorem will result. Universal instantiation of x with MARY in the formula $\forall x(HUMAN(x) \Rightarrow \exists y(MOTHER\text{-}OF(x,y)))$ yields $(HUMAN(MARY) \Rightarrow \exists y(MOTHER\text{-}OF(MARY,y)))$.

Systems for automated theorem proving typically use a rule of inference called *resolution*. The resolution principle allows the deduction of the statement $\mathcal{U} \vee \mathcal{V}$ (the *resolvent*) from the two statements $\neg\mathcal{A} \vee \mathcal{U}$ and $\mathcal{A} \vee \mathcal{V}$, where $\mathcal{A}, \mathcal{U}$, and $\mathcal{V}$ are arbitrary, valid formulas. $\mathcal{A}$ and $\neg\mathcal{A}$ are the literals resolved upon.

In order to apply resolution to logical statements, one statement must contain the literal to be resolved upon and the other the negation of the literal. The two literals must contain the same predicate (otherwise, there is no possibility of resolving them), however there is no guarantee that the terms of the predicates will be identical. In order to determine whether there is a substitution which, when applied to each of the literals, will make them identical (except for the negation), a process called *unification* is performed. Unification attempts to match the two literals by finding the most general substitution that will render them identical. Once this substitution is applied (using the rule of Universal Instantiation) to each of the statements involved (that is to all literals in the staements, not just the ones resolved upon), resolution may proceed, and the new resolvent computed.

Unification and resolution are another example of the basic information integration principle mentioned in the beginning of this paper. That is, the process consists first of a translation step, where the two "sources" are transformed so that they are both concerned with the same thing. Second, an inference step is performed that completes the integration. Planning[39] and robot control systems[9] have been developed using formal theorem-proving techniques as the information integration mechanism.

The concept of programming a computer by specifying goals as theorems to be proven, situational facts as axioms, and rules as implications has been attractive to AI researchers ever since effective theorem proving procedures were developed. Prolog[4] one of the most popular *logic-programming* languages currently available makes this possible. Prolog uses a *clause* form of logical representation, containing only constant terms and universally quantified variables. While Prolog uses a simple syntactic, top-down (discussed below) scheme to determine the order of processing facts and rules as a default, the system provides built in procedures for altering the flow of processing under programmer control.

3.1.3 Other logical representations

The basic logical representations as discussed above consist of formulas and statements in propositional or predicate logic. Other representations of great value to AI developers include production rules, frames, semantic nets, and state transition networks.

A *production rule* is the most common representation of information used in expert systems development[2]. It is roughly equivalent to an implication in the propositional or predicate logic. It consists of *evidence* (also called the left-hand-side or LHS) and a *hypothesis* (also called the right-hand-side or RHS). Establishing the existence (i.e., the truth) of evidence enables a production rule system to infer the hypothesis. Furthermore, a system that needs to establish the truth of the hypothesis can seek to establish the truth of the evidence – this is termed backward-chaining, and will be discussed below. Production rules with appropriate updating formulas can be used for uncertain reasoning; this will also be discussed below.

Frames group knowledge about particular objects and situations. Properties of an object and interrelationships among objects are represented as *slots* in the frame; values for the properties are stored in the slots. Frames provide useful mechanisms for focusing attention, as they collect relevant information into a single element. Default values are often provided for slots.

Semantic nets are quite similar to frames in concept. A semantic net uses arcs to represent properties and relations (which are slots in frames) and nodes to represent entities and values (the contents of slots).

State transition networks are graphical representations of sequential machines. States are represented by nodes; conditions for state change are represented as predicates (expressions that evaluate to true or false) on the arcs connecting the nodes. State transition networks provide useful computational mechanisms for handling temporal changes. The evolution of a system may be expressed through changes in a state transition network.

3.1.4 Other logics

The first-order predicate calculus does not allow relationships among predicates, beliefs, temporal relations, or statements of possibilities. In addition, there is no means for deleting assertions from the database. For this reason, a variety of alternative logics and inference methods have been explored. AI workers are actively researching problem-solving and inference methods based on these logics, which we will briefly describe here (for more details see [35]).

Modal logics are concerned with necessity and possibility. They are primarly used to represent statements of belief. A modal logic of particular interest is *epistemic logic*, a formalism suitable for representing states of knowledge[29].

Temporal logics deal with the representation of time and statements whose truth-value is tied to the temporal interval over which it is evaluated. Temporal logic is important for planning and interpreting situations where time is a critical factor.

Higher-order predicate logics can express properties of predicates. Certain problems, particularly those involving concepts of equality, can be worked more easily in a higher-order predicate logic.

Non-monotonic logics address the problem of non-monotonic changes in the database. In a standard logical system, it is assumed that the axiom base is consistent (that is, it contains no contradictions), and therefore all theorems derived from the axiom set are true. In practical situations, it is common to treat the axiom base as the current database of facts about the world. All consistent new facts that are added, then, cause *monotonic* changes to the knowledge base (i.e., the "amount" of knowledge is always increasing). When a fact is inserted that contradicts an earlier axiom, a *non-monotonic* change has occurred, and all theorems derived from the invalidated axiom must be updated. Non-monotonic logics assist in this updating process.

3.2 Uncertain Reasoning

Most real-world activities involve varying degrees of uncertainty about the true situation. This type of information is not well- or easily described in formal logical terms. As a result, several schemes for representing and reasoning about uncertain information have evolved. Most of these approaches begin with classical, Bayesian probability theory as a base, and then extend it either formally or heuristically to handle situations that are difficult or impractical to address using the pure theory.

Most systems that deal with uncertainty use a propositional framework to represent interrelated statements about situations of interest. The chosen uncertainty representation is then overlaid on the propositions in this framework. Most developers of uncertain reasoning system will follow some or all of the following steps: framing the problem, creating a background knowledge structure representing key problem elements and their interrelations, creating a structure for analysis of situational data and, finally, using the system to interpret and analyze data acquired about the situation. Differences in approach arise primarily from particular assumptions about the nature of the underlying information being interpreted. These assumptions motivate the choice of an *updating mechanism* for uncertain information which, in turn, dictates the requisite infrastructure to support the choice.

In this section, we shall focus on several types of uncertain reasoning formalisms. We shall highlight *evidential reasoning* as a particular form of uncertain reasoning developed specifically to address problems in interpreting real-world information.

3.2.1 Statistical reasoning

The simplest form of uncertain reasoning is statistical inference. A primary motivation for the use of statistics is to summarize and describe populations of events and situations, based on relatively small subsets of those events. An important statistical method is the process for estimating population parameters with prespecified confidence levels and intervals[28] . Such estimators are useful for determining whether acquired data indicate a possibly significant event. For example, a radar detector must accumulate and average several samples of possible radar pulses before determining (with acceptable confidence) that there is a target in its field of view. The integration process carried out by the radar (and by mosts statistical operations) combines information into the current *model* of the target. This model will, for example, include information about the current position and vector velocity of the target. The model is *predictive* in nature, as it enables a new estimate of position to be derived. This information can be used to control acqnisition of information (by using a "range gate," for example), as well as to verify the current model (see [34] for a detailed discussion). Kalman filters integrate position data into a model of an objects's state parameters in order to estimate future positions. This information is (typically) used to track the object.

A familiar form of statistical inference is pattern recognition or pattern classification[6]. A typical approach to classifying input features as belonging to particular objects begins by creating a partitioned feature space. An ideal feature space would consist of well-spaced, compact clusters; interpreted features would map to single clusters and the recognition problem would be trivial. In real life, feature measurements associated with one object typically overlap the feature measurements for other objects. To add to the difficulty, the measurements themselves may be corrupted, leading to further imprecision. This situation requires the developer to select a partition of the feature space that will minimize the likelihood and the costs of wrong decisions. This transforms the problem into a statistical decision theory task.

3.2.2 Probabalistic inference

The primary means for estimating the probability of a hypothesis based upon the measured probability of supporting evidence is the familiar *Bayes' Rule of Conditioning*. While formally defined using strictly *a priori* probabilities and conditionals, Bayes' rule is often used to *update* belief in a hypothesis based on new evidence. Bayes' rule is expressed mathematically as

$$p(A \mid B) = \frac{p(B \mid A)\, p(A)}{p(B)}.$$

This shows how observing the value of $p(A)$ changes the prior value $p(B)$ to the *a posteriori* likelihood, $p(A \mid B)$.

An interesting situation occurs when more than one piece of evidence has been acquired, and a new *a posteriori* likelihood based on the combination is desired. In this case, input descriptions typically consist of the individual conditional probabilities and the individual priors. For example, starting information might consist of the following *a priori* specifications:

$$p(B), p(C), p(B \mid A), \text{and } p(C \mid A).$$

New probabilities are measured for events B and C, and $p(A \mid B \wedge C)$, the *a posteriori* likelihood is desired. Bayes' rule provides

$$p(A \mid B \wedge C) = \frac{p(B \wedge C \mid A)\,p(A)}{p(B \wedge C)}.$$

Note that in order to compute this expression, values for $p(B \wedge C \mid A)\,p(A)$ and $p(B \wedge C)$ must be available or computable. Since determining all of the required conditional and prior probabilities needed to solve such updating problems exactly is onerous if not impossible in complex situations, various assumptions are typically made to facilitate the computation. Frequent assumptions are *conditional independence* and the *principle of insufficient reason*. Conditional independence states that $p(B \wedge C \mid A) = p(B \mid A)\,p(C \mid A)$. That is, given conditioning statement A, B and C are independent and their probabilities may be multiplied to compute the joint conditional probability. If B and C are strictly independent, then by definition, $p(B \wedge C) = p(B)\,p(C)$.[5] These independence assumptions allow us to rewrite the above expression

$$p(A \mid B \wedge C) = \frac{p(B \mid A)\,p(C \mid A)\,p(A)}{p(B)\,p(C)},$$

which is now composed of known quantities.

Often, necessary probability values are unavailable when computations are to be performed. In such cases, it is common to employ the principle of insufficient reason to assign missing probability values. Simply put, if the probability of a disjunction of events is known, but the probabilities of the individual components is not, and there is no particular reason to expect that one event is more likely than any other, then the principle of insufficient reason dictates that equal probabilities, totalling to the original probability, be assigned to the individual components. A more sophisticated version of this approach is the *maximum entropy principle*[20]. In this approach, probability values are selected that maximize the entropy (or the "disorderedness") of the assignment. The use of a maximum entropy computation corresponds to making a *minimal committment* in estimating unknown probabilities.

It is important to note that for many real-world problems, most uncertainty methods based on classical Bayesian probability will require information that is not available, and must be estimated. Often these estimates will turn out to be close to the "correct" probabilities, particularly when the space of possibilities is well understood, and the estimation procedure is matched to the situation. However, one must not lose sight of the fact that they are only estimates, may well be incorrect, and must be accounted for in the final results.

[5] In most practical systems, *ratios* of quantities such as *odds* are used for the probability computations. This scheme avoids the need to handle *joint*, prior probabilities explicitly.

3.2.3 Evidential reasoning

Information acquired about real-world situations provides *evidence* about the possible states of the world that might have given rise to it[15, 16]. This evidential information is typically uncertain, usually incomplete (that is, it contains residual *ignorance*), and it may contain errors. Based on a formal theory, the *Dempster-Shafer Theory of Evidence*[32], evidential reasoning[6] makes a formal departure from classical probability to address these issues. In particular, evidential reasoning allows belief[7] to be associated directly with disjunctions of events. That is, rather than forcing probabilities to be distributed across the set of possibilities, evidential reasoning maintains the association between the measure of belief and the disjunction. This approach avoids the need for assumptions for values of missing data. When beliefs of components are later needed, they are underconstrained as a result of the disjunction and an interval representation is needed to capture the true constraints. This interval enables the explicit modeling of both what is known (although with uncertainty) and what is unknown.

The fundamental entity in evidential reasoning is the *frame of discernment* (commonly called the frame and indicated by "Θ"). Θ consists of a set of mutually exclusive, exhaustive statements which represent concepts of interest to the developer. Propositions are made up of elements of the power set (i.e., the set of all subsets) of Θ, indicated by 2^{Θ}. Belief can be assigned to any proposition, including to Θ itself; any belief assigned to Θ expresses total ignorance to that extent. An *evidential mass function* (or just mass function) represents the distribution of a unit of belief across selected (focal) elements of 2^{Θ}. A *body of evidence* is the frame of discernment and a particular mass function.

For manipulating bodies of evidence, Evidential reasoning provides explicit formalisms for both combination and translation, the two aspects of the information integration problem discussed in the beginning of this paper. *Dempster's rule of combination* is used to combine two distinct bodies of evidence over a common frame of discernment to yield a new body of evidence. *Compatibility relations* are used to translate statements from one frame to another. Because Dempster's rule is both commutative and associative, multiple (independent) bodies of evidence can be combined in any order without affecting the result. If the initial bodies of evidence are independent, then the derivative bodies of evidence are independent as long as they share no common ancestors.

Evidential reasoning supports a number of primitive operations for reasoning from evidence. All of these operations have a formal basis in the Dempster-Shafer math-

---

[6] *Evidential reasoning* is a term coined by SRI International [24] to denote the body of techniques specifically designed for manipulating and reasoning from evidential information.

[7] Although *beliefs* are not strictly probabilities, we will use the terms interchangeably in this informal paper.

ematical theory of evidence and have intuitive appeal as well. Thus, both flexibility and understandability are retained without sacrificing validity.

- **Fusion**—This operation pools multiple bodies of evidence into a single body of evidence that emphasizes points of agreement and deemphasizes points of disagreement.
- **Discounting**—This operation adjusts a body of evidence to reflect its source's credibility. If a source is completely reliable, discounting has no effect; if it is completely unreliable, discounting strips away all apparent information content; otherwise, discounting reduces the apparent information content in proportion to the source's unreliability.
- **Translation**—This operation moves a body of evidence away from its original context to a related one, to assess its impact on dependent hypotheses. For example, a body of evidence pertaining to the activities of an object can be translated to estimate observeables that ought to be associated with it.
- **Projection**—This operation moves a body of evidence away from its original temporal context, to a related one. For example, evidence about an object's state parameters can be projected to estimate future locations.
- **Summarization**—This operation eliminates extraneous details from a body of information. The resulting body of evidence is slightly less informative, but remains consistent with the original.
- **Interpretation**—This operation calculates the "truthfulness" of a given statement based upon a given body of evidence. It produces an estimate of both the positive and negative effects of the evidence on the truthfulness of the statement.
- **Gisting**—This operation produces a single statement that captures the general sense of a body of evidence, without reporting degrees of uncertainty.

Evidential reasoning techniques have been automated in a system called **Gister**[8][26]. Gister provides graphical facilities for constructing a background knowledge base, creating frames of discernment, defining compatibility relations among them, and interactively creating and evaluating analyses of situational information. The steps in using Gister to develop an evidential reasoning solution to a problem are exactly those listed in the beginning of this section. First, background knowledge is structured. The various "vocabularies" are selected and represented as frames of discernment. Compatibility relations linking these frames are specified next. In order to provide a structure for analyzing situation data, an *analysis graph* is developed. The analysis

[8]Gister is implemented in Lisp on the Symbolics 3600-series Lisp Machines.

graph describes exactly the evidential operations that are to be performed on the input bodies of evidence. Information from a body of evidence flows through the network of operations specified in the analysis graph in a *dataflow* fashion, until all operations are complete. Information computed for selected output frames may then be examined in order to determine the results of the analysis.

Gister provides a graphical, interactive aid for *argument construction*[23], the creation of explanations for evidence received about the environment. An *argument* can be thought of as a way of explaining evidence. It is often the case, particularly in highly complex domains such as intelligence analysis, that it is not obvious how to interpret information until that information is in hand. Essentially, there is no single style of explanation that can account for all possible inputs. Traditional expert systems, on the other hand, typically use models which can be thought of as the generic argument for explaining all inputs to the system. Argument construction may be viewed as the analogue to a logical proof in the field of uncertain reasoning. Gister gives the developer flexibility in operating on evidence, in creating arguments, and in evaluating alternative explanations until a suitable one is found.

Evidential reasoning relaxes some of the extensive information requirements of classical Bayesian probability theory, while maintaining its formal appeal, and is therefore more natural for a wide range of problems. In addition, a wide variety of primitive evidential operations with a rigorous theoretical basis have been defined to facilitate manipulation of evidence. Evidential reasoning has been implemented as the basic computational mechanism in Gister, an interactive, graphical shell.

3.2.4 Heuristic methods

Early work in expert systems addressed the problem of reasoning from uncertain data using the production rule formalism. The very nature of expert systems, which attempt to suppress insignificant information and focus on the data deemed useful by an expert, makes it effectively impossible to use formal probabilistic methods to update hypotheses based on new evidence. The work on Mycin[33], an expert system to diagnose and recommend treatment for certain blood diseases resulted in an informal approach which used *certainty factors* between -1 and 1 to represent degrees of belief. A goal of the early research that led to the Prospector mineral exploration consultant[7] was to formalize the updating process, using classical probability.

Prospector was forced to assume conditional independence in order to update hypotheses. Although it led to an inconsistent (that is, non-invertible) procedure, this was an acceptable compromise because of the nature of the probability values which were typically associated with the presence of ore bodies. These probabilities tend to be very small, and errors due to the updating rule were minimal[5].

4 Control of the Integration Process

Inference procedures are typically quite simple and straightforward to perform. In general, each inference generates a new piece of information to add back into the data base. Since it is simple to fill a database with irrelevant data from uncontrolled inference, a key problem in AI is to control these processes effectively in order to reach correct, useful, and desireable conclusions. This means that the processing framework that controls the acquisition of information and the selection and application of inference procedures is of critical importance. In this section, we discuss certain basic AI control paradigms along with architectures which have evolved specifically to address problems of real-world, real-time information system control.

In order for a system to behave *purposively*, it must know what its (or its developer's) goals are. These goals which may be well understood by the developer must be made explicit and meaningful to the system itself. In addition, the system must have some means for recognizing when it is making progress towards its goals. Typical informational goals include detecting the presence of certain activities, identifying the activity and the actors, measuring interesting features of an activity (such as its location, status, and state variables), discriminating among possible identifications, and predicting future activities. In addition, however, *meta-goals* may be operative. These might include requirements for real-time operation, a need for explanation of results, a requirement for effective man-machine interaction, a need for "quiet" (i.e., passive) operation, and a desire for the system to learn and adapt to new situations. These meta-goals will likely have as great an impact on the selection of an approach as do the nominal informational goals.

4.1 Top-down/Bottom-up Methods

Two basic control paradigms used in AI are *top-down* and *bottom-up* methods. Top-down methods are also termed *goal-driven*, *model-driven*, or *backward-chaining* techniques. Bottom-up methods are also called *data-driven* or *forward-chaining* techniques.

Top-down methods begin with a statement of the problem or goal and attempt to solve it by finding subproblems that can be solved. For example, an expert system may contain a rule that states, "If a fruit is red and round then it is an apple." Inverting the rule provides a means for finding apples – try to find fruits that are both red and round. This is the source of the word "back-chaining;" the system chains backward through its rules, until it finds a subgoal it can solve. *Hierarchical* methods are top-down (typically) methods that break a problem into ever finer parts, solving problems at one level before filling in details at the next.

Since certain subgoals may be impossible to achieve, failure of the solution procedures must be expected from time to time. In this case, the system (usually) has no recourse except to *backtrack* to an earlier choice point (normally the most recent) and try another alternative. Should it exhaust all of its options, then the procedure fails and the top-level goal also fails. Each time the system chooses a new alternative to consider, it typically must switch its evaluation context to that of the new alternative. Context switching, a direct result of backtracking, can be an extremely costly operation and a typical design goal is to minimize the number of times it occurs. Effective procedures try to use knowledge to make good choices in the first place and to learn enough from those failures that do occur to eliminate alternatives that might have otherwise been considered.

Clearly, top-down processing can involve *searching* a potentially huge database of possibilities. Much of early AI research was concerned specifically with controlling search. A number of formal and heuristic search techniques such as *alpha-beta search* were developed and are still part of the AI developer's tool kit.

Bottom-up processing draws conclusions by reasoning forward from data to conclusions. Using the same rule about apples mentioned above, a system could infer that it had an apple once it had determined that it had a fruit that was also red and round. Data-driven methods are useful for determining the implications of new information, however, by their nature they are difficult to focus.

A typical approach to interpreting real-world information is to use a combination of top-down and bottom-up techniques. This, for example, was the procedure used in Prospector. The user would enter data that had been collected in the field. Implications derived from this data in a forward-chaining manner provided an initial context of hypotheses for evaluating the likelihood of ore bodies of interest. This initial context helped Prospector, in a top-down fashion, to frame queries to the user in order to refine its initial hypotheses. At any time, the user could enter new data to be interpreted in a data-driven fashion. This *hybrid* method facilitated an interactive paradigm known as *mixed initiative*, where the system has the initiative part of the time and can ask questions; other times the user would take the initiative and volunteer information.

Generally speaking, top-down, model-driven approaches are useful when the models are restrictive and the data is noisy. Noisy data causes problems to a data-driven system. If there are no safeguards for data quality (and in a data-driven mode there is rarely enough information to ensure quality in a noisy environment) it is quite likely that corrupted data will generate errors which may be costly to rectify. Noisy data requires effective data acquisition and data management techniques based on an understanding of the system's goals and its environment.

4.2 Perceptual Reasoning

As mentioned previously, the acquisition and interpretation of information may be considered a *perceptual* task and, therefore, is a purposeful activity, undertaken to support definite system goals. In that regard, a perceptual system must be capable of interpreting system objectives and using them to focus the system's resources in order to optimize the collection and interpretation of information. By updating the underlying model of the situation from new information, and altering its detailed information goals, a system can adapt to new situations.

A straightforward architecture designed to control a perceptual process in a top-down mode is the *perceptual reasoning loop*[25, 13] shown in Figure 3. This architecture consists of three functional modules, ANTICIPATE, PLAN, and INTERPRET. This architecture was first elaborated for use in a simulation of an electronic warfare (EW), multisensor situation assessment system[14, 12], but it serves as a general, functional architecture for perception.

The ANTICIPATE module attempts to predict aspects of the situation that might be expected, but for which there is no direct evidence. These include events that will take place due to the passage of time or due to interactions among actors. Other anticipated events include activities associated with previously identified events. The goals of the EW system were to detect, identify, and locate possible threats to the aircraft. Examples of the types of events anticipated included undetected components of identified systems and system mode changes due to actions of the platform. For instance, the fact that certain types of surface-to-air missile systems were defended by anti-aircraft guns enabled the system to consider looking for the guns before they became a threat. Similarly, knowledge of the standard operating procedures of the missile systems allowed the system to predict likely mode changes (say from an acquisition mode to a target-tracking mode – definitely an interesting development) based on the computed range to the missile. A state-transition network was used to represent a process model; anticipation of mode changes was made on the basis of state changes in the network, which were in turn predicated on the satisfaction of conditions placed on the arcs. Based on the predicted events likely to be of interest, the ANTICIPATE module would make up a set of information requests for the PLAN function.

The PLAN module was given the task of determining an optimal plan for satisfying as many information requests as possible. There was always competition for sensor resources, so it was, in general, impossible to satisfy all the requests made. The first step in the PLAN function was to order the information requests by their priority. This priority was computed based on two key parameters, a determination of the overall effect that acquiring the information would have on the system's goals, and the degree of inportance of the information (which was typically based on the lethality of the threat being considered). For each available sensor a probability model of its performance (modified for the current environment) was used to determine the utility of using that particular resource to acquire the desired information. The utility values were used by a heuristic, dynamic-programming allocation routine to make the ultimate assignment of sensor resources to information requests. In the course of

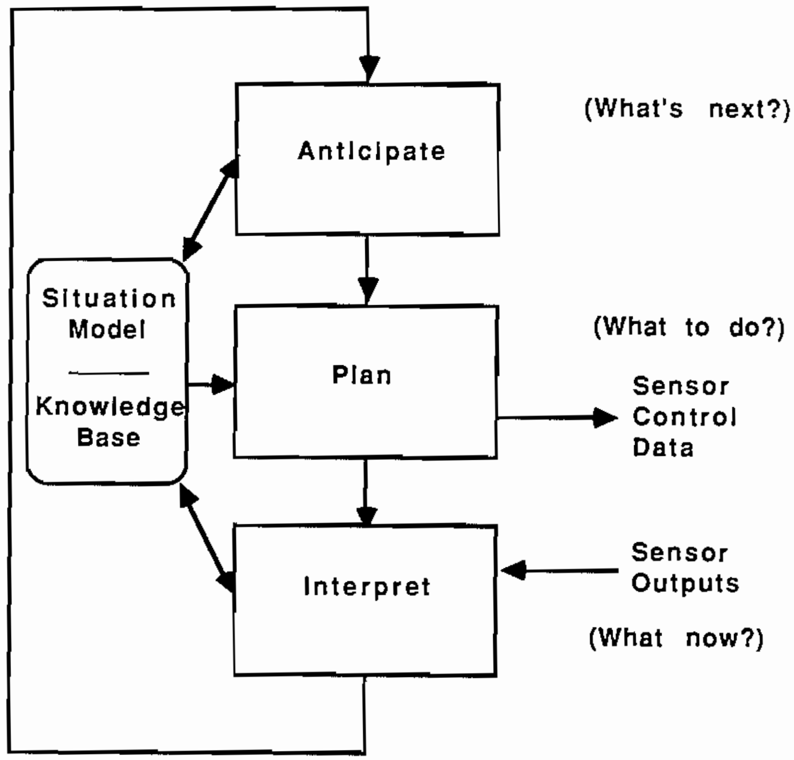

Figure 3: The Perceptual Reasoning Loop

computing utility values, a determination was made of the optimal control parameters for the sensors. These control data were communicated to the (simulated) sensors, and they were operated in the simulated environment to acquire their data.

The INTERPRET module analyzed the acquired sensor data in terms of the expected information. Information that matched expectations typically resulted in increased likelihoods, while information that was expect, but not acquired typically resulted in decreased likelihood. Evidential reasoning was used to integrate new information into the situation model. In addition, data which was received that was unanticipated was interpreted in a data-directed mode to determine possible unexpected threats. The updated situation model was the primary, *short-term* database, used in turn by the ANTICIPATE module.

This system was highly adaptive and could adjust to varying environmental conditions by altering its selection of sensors. It exhibited various interesting behaviors, but one of the most useful was a *cueing* capability that use information received from one sensor to point another sensor.

4.3 The Blackboard Architecture

The *blackboard architecture* provides a very flexible control structure for interpretation and problem solving. The name "blackboard" attempts to capture the notion of a collection of asynchronous processes writing messages on a blackboard which are in turn read by anyone looking at the blackboard, and acted upon by those able to do so. A less anthropomorphic view would characterize the architecture as a mediated, limited-broadcast communication structure.

As shown in Figure 4, the three basic components of a blackboard system (as implemented in AGE[30, 18]) consist of the blackboard itself, a controller, and the knowledge base. The blackboard is a hierarchical data structure organized to represent the problem domain as a hierarchy of analysis levels. Facts about objects are stored in the knowledge base; production rules represent the knowledge for using these facts and information on the blackboard. Related rules are grouped together into *knowledge sources*.

The controller selects knowledge sources for activation, based on the contents of the blackboard. By specifying different control structures, the developer can explore different problem-solving strategies. The default control mechanism has two distinct functions, *inference generation* and *focus of attention*.

The primary contents of the blackboard are organized into one or more *hierarchical hypothesis structures*. These structures are oriented toward particular parts of the problem. Each level in a hypothesis structure is integrated with levels above and below by links joining hypothesis elements in the various levels. Links that represent support from above are called *expectation links*; those representing support from below are called *reduction links*. Hypothesis elements can be thought of as *abstractions* or *summarizations* of lower level elements, and *components* of higher level ones.

A solution is built incrementally by rules that add or modify the hypothesis elements or relationships among them. Hypothesis formation is a process where the rules do one of the following: interpret data at a lower level, instantiate a more general model at a higher level, or generate expectations that must be verified by data.

Blackboard systems have been developed for a variety of applications including speech and natural-language processing[22, 8], multi-sensor signal analysis[31], and image understanding[17].

4.4 Planning Methods

Early AI work addressing the integration of information from multiple sources focused on the task of locating objects in unregistered range and color images[11]. This research resulted in a system that developed plans for locating specific objects based on their expected appearance in each sensor modality (the information used was hue,

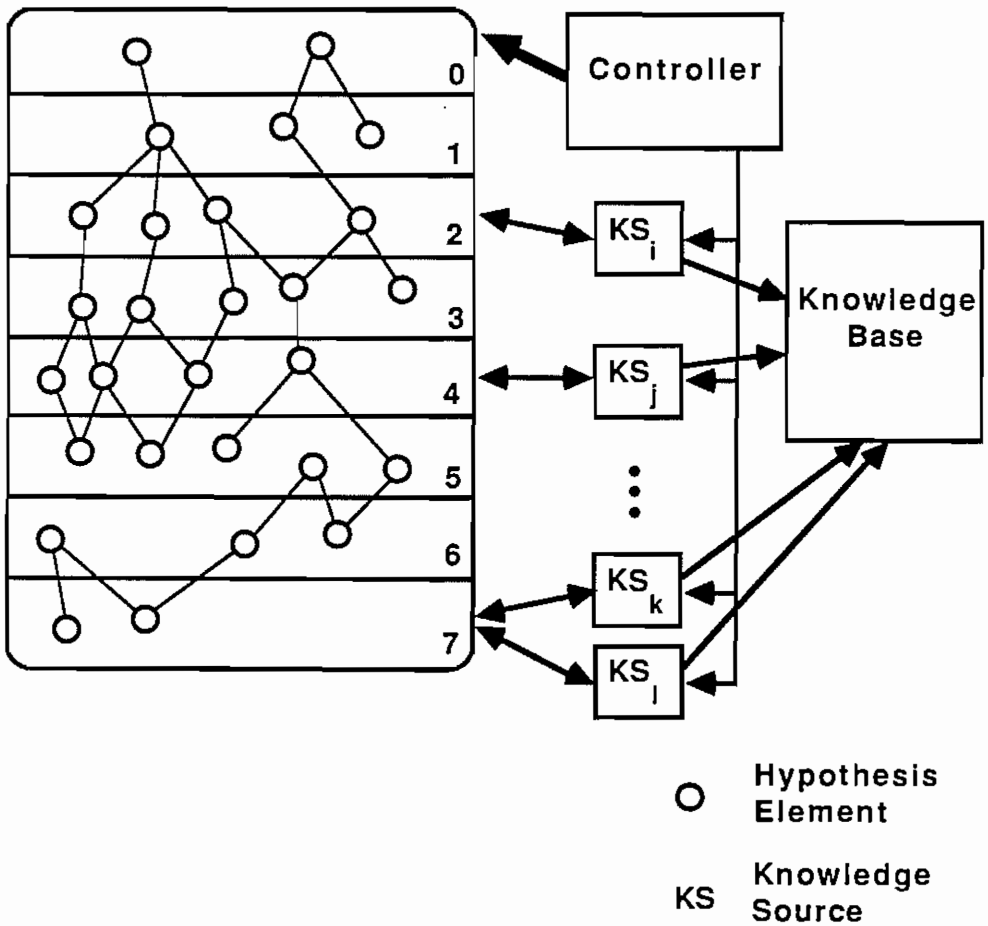

Figure 4: The Blackboard Architecture

saturation, intensity, surface orientation and extent, and reflectance at the wavelength of the laser used in the range sensor) and its expected relation to other objects.

More recent work[36, 38, 37] has taken the view that planning itself is an evidential process. That is, a system does not know for sure what the outcome of using a particular procedure in a plan will be, and can only estimate the likely outcome based on evidence about the operation. This is particularly true of processes that acquire information from the environment. The work uses evidential reasoning to draw conclusions about which process to invoke next in a scene interpretation task.

4.5 Local/Global Methods

Information in an image is both *local* and *global*. Local information includes point properties such as intensity and range and properties of small patches such as texture and surface orientation. Global information may include facts such as several pixels appear to be in a single line or that surface slope is constant over a wide area. It is clear to vision researchers that being able to recognize such global concepts and bring them to bear on the local computations would make the overall interpretation more robust. Unfortunately, detecting global similarities is quite expensive and, furthermore, requires certain local determinations to be made first.

Relaxation[19] methods provide a control architecture that introduces global information into a local interpretation in an iterative manner. They do this by computing local properties from a combination of current interpretations for a pixel and its immediate neighbors. These local computations may be boolean or probabilistic. A single iteration involves performing local computations and updating at each pixel in the image. A number of iterations are performed until the system "relaxes," and further changes will be minimal. Relaxation has been shown to be equivalent to constrained optimization, and offers particular utility for *enhancing* certain image features.

Another method that attempts to bring global information to bear on local interpretations is *simulated annealing*[21]. In this approach, an *energy function* is defined that is (typically) based on local interpretations and an overall, global interpretation. In effect, the initial system of data is brought to a very high "temperature," generating disorder in the possible local interpretations. Very slowly, the system is "cooled" and the energy of the system is reduced. This occurs by randomly selecting pixels and making local changes in interpretation that reduce the overall energy. Changes that will result in a slight increase in overall energy are made on a probabilistic basis, where the probability is a function of the system temperature – when the system is at a high temperature, there is a greater likelihood that energy-increasing changes will be permitted. What this means is that an interpretation of a pixel that leads to a local energy minimum may yet be changed, allowing it to "pop" out ofthe local minimum and continue seeking the true minimum. Over time, the system converges toward interpretations that result in a minimal, ground energy state. Simulated annealing is being explored as a technique for computing stereo disparity in two images[1], and for many other applications.

5 Summary and Conclusions

In common to all purposive activities is a continuing need for up-to-date environmental information. Directly perceived information is used to detect new events, to monitor dynamic events, and to measure parameters of the environment. By exploiting redundancy, a situation "mosaic" may be created from a diverse collection of information sources each with a partial view of the situation. In addition, the combination of information from multiple sources will provide a means for overcoming the imprecision, inaccuracy, and occasional errors inherent in most sensing processes. As situations in which such activities occur become more complex, greater quantities and greater diversity of information sources are typically needed. The integration of information by inference, is a fundamental process in intelligent systems – both natural and artificial. In this brief survey, we have discussed a number of AI representations, inference methods, and control strategies for inference procedures.

References

[1] Stephen T. Barnard. Stereo matching by hierarchical, microcanonical annealing. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1987. In press.

[2] Bruce G. Buchanan and Edward A. Feigenbaum. *Rule-Based Expert Systems.* Addison-Wesley Publishing Company, 1984.

[3] Eugene Charniak and Drew McDermott. *Introduction to Artificial Intelligence.* Addison-Wesley Publishing Company, 1985.

[4] W. F. Clocksin and C. S. Mellish. *Programming in Prolog.* Springer-Verlag, 1981.

[5] Richard O. Duda. Private communication.

[6] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[7] Richard O. Duda, Peter E. Hart, and Rene Reboh. A rule-based consultation program for mineral exploration. In *Proceedings of the Lawrence Symposium on Systems and Decision Sciences*, pages 306–309, Berkeley, California, October 1977.

[8] Lee D. Erman, P. E. London, and S. F. Fickas. The design and an example use of hearsay-iii. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 409 – 415, 1981.

[9] R. E. Fikes, P. E. Hart, and N. J. Nilsson. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251 – 288, 1972.

[10] Martin A. Fischler and Oscar Firschein. *Intelligence - The Eye, the Brain, and the Computer.* Addison-Wesley Publishing Company, 1987.

[11] Thomas D. Garvey. *Perceptual Strategies for Purposive Vision.* PhD thesis, Stanford University, 1976.

[12] Thomas D. Garvey and Martin A. Fischler. *Machine Intelligence Based Multi-Sensor ESM System.* Final Report SRI Contract 6804, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California, October 1979. (Distribution limited to US Government Agencies).

[13] Thomas D. Garvey and Martin A. Fischler. Perceptual reasoning in a hostile environment. In *Proceedings of the First Annual National Conference on Artificial Intelligence*, pages 253–255, American Association for Artificial Intelligence, 445 Burgess Drive, Menlo Park, California 94025, August 1980.

[14] Thomas D. Garvey and John D. Lowrance. *Machine Intelligence for Electronic Warfare Applications.* Final Report SRI Contract 1655, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California. November 1983. (Distribution limited to US Government Agencies).

[15] Thomas D. Garvey, John D. Lowrance, and Martin A. Fischler. An inference technique for integrating knowledge from disparate sources. In *Proceeding of the Seventh Joint Conference on Artificial Intelligence*, pages 319–325, American Association for Artificial Intelligence, 445 Burgess Drive, Menlo Park, California, August 1981.

[16] Thomas D. Garvey, John D. Lowrance, and Martin A. Fischler. An inference technique for integrating knowledge from disparate sources. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 457–464, Morgan Kaufman Publishers, Inc., 95 First Street, Los Altos, California, 1985.

[17] A. R. Hansen and E. M. Riseman. *Computer Vision Systems*, pages 129 – 163. Academic Press, 1978.

[18] Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat, editors. *Building Expert Systems*, pages 326 – 334. Addison-Wesley Publishing Company, 1983.

[19] Robert A. Hummel and S. W. Zucker. On the foundation of relaxation labeling processes. *IEEE Pattern Analysis and Machine Intelligence*, 5(3):267 – 287, May 1983.

[20] E. T. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106:620–630, 1957.

[21] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, (220):671 – 680, 1983.

[22] B. T. Lowerre. *The HARPY Speech Recognition System.* PhD thesis, Carnegie-Mellon University, Pittsburgh, Pa., 1976.

[23] John D. Lowrance. Automating argument construction. In *Proceeding of the Workshop on Assessing Uncertainity (November 13-14, 1986)*, Stanford University and the Navy Center for International Science and Technology, Department of Statistics, Stanford University, Stanford, California, March 1987.

[24] John D. Lowrance and Thomas D. Garvey. Evidential reasoning: a developing concept. In *Proceeding of the Internation Conference on Cybernetics and Society*, pages 6–9, Institute of Electrical and Electronical Engineers, October 1982.

[25] John D. Lowrance and Thomas D. Garvey. *Evidential Reasoning: An Implementation for Multisensor Integration.* Technical Report 307, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California, December 1983.

[26] John D. Lowrance, Thomas D. Garvey, and Thomas M. Strat. A framework for evidential-reasoning systems. In *Proceeding of the National Conference on Artificial Intelligence*, pages 896–903, American Association for Artificial Intelligence, 445 Burgess Drive, Menlo Park, California, August 1986.

[27] Elliott Mendelson. *Introduction to Mathematical Logic.* D. Van Nostrand Company, 1964.

[28] David S. Moore. *Statistics – Concepts and Controversies.* W. H. Freeman and Company, 1979.

[29] Robert C. Moore. *A Formal Theory of Knowledge and Action*, chapter 4. Ablex Publishing, 1985.

[30] H. P. Nii and N. Aiello. Age (attempt to generalize): a knowledge-based program for building knowledge-based programs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 645 – 655, 1979.

[31] H. P. Nii and E. A. Feigenbaum. *Rule-Based Understanding of Signals*, pages 483–501. Academic Press, 1978.

[32] Glenn Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, New Jersey, 1976.

[33] Edward E. Shortliffe. *Computer-Based Medical Consultations: MYCIN.* Elsevier, New York, New York, 1976.

[34] Merrill I. Skolnick, editor. *Radar Handbook*, chapter 4. McGraw-Hill Book Company, 1970.

[35] R. Turner. *Logics for Artificial Intelligence.* Halsted/Wiley, 1985.

[36] Leonard P. Wesley. Evidential knowledge-based vision. *Optical Engineering*, 25(3):363–379, March 1986.

[37] Leonard P. Wesley. *Evidential-Based Control in Knowledge-Based Systems.* PhD thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, 1987. In preparation.

[38] Leonard P. Wesley, John D. Lowrance, and Thomas D. Garvey. *Reasoning about Control: An Evidential Approach.* Technical Report 324, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California, July 1984.

[39] David Wilkins. Domain-independent planning: representation and plan generation. *Artificial Intelligence*, 22(3):269–301, 1984.