Two expert systems apply the Procedural Semantic Networks formalism to assess left-ventricular behavior and diagnose cardiac rhythm disorders.

Building Knowledge-Based Systems: The PSN Experience

John Mylopoulos, Tetsutaro Shibahara, and John K. Tsotsos, University of Toronto

Knowledge representation is recognized today as a central problem in artificial-intelligence research because the current design paradigm for expert systems stresses the need for a knowledge base that stores expert knowledge and provides associated knowledge-handling facilities. This paradigm sharply contrasts with earlier ones that emphasized general-purpose heuristic search techniques.

Research in knowledge representation may involve a study of how we can represent particular semantic notions, such as time, causality, beliefs, and intentions. On the other hand, it may take the form of a language design project, where the language is intended for knowledge representation and the "programs" written in the language are knowledge bases storing knowledge about some domain of expertise. A third alternative is that research in knowledge representation may involve developing a programming environment, in the sense of Unix or Interlisp, for building and using knowledge bases. The Procedural Semantic Networks, or PSN, project began at the University of Toronto in 1976 as a language design and implementation project. Since last year, however, we have been paying increasing attention to programming environmental issues.

Knowledge-representation languages have been classified traditionally as *declarative* or *procedural,* depending on whether their basic features come from mathematical logic or data structures, on one hand, or from programming languages.[1] Declarative representation languages, such as Prolog and KL-One, encourage perspicuity and support the maintainability of a knowledge base. Procedural representation languages are particularly well suited for heuristic knowledge, and their use can lead to efficient searching on the part of an expert system. The general framework for procedural representation languages offered by production systems has been used successfully in designing several expert systems.

Since Minsky's influential "frames paper,"[2] many attempts have been made to integrate features of declarative and procedural representation languages, a topic that persists in ongoing research. PSN is one such attempt that focuses on the integration of semantic network and procedural notions.

An overview of PSN

The following description of the main features of PSN emphasizes the background and philosophy of its design. An initial design for PSN was proposed by Levesque[3] and outlined by Levesque and Mylopoulos.[4] Further work on the language and its implementation has been summarized by Patel-Schneider and others.[5]

Semantic networks. Semantic networks were first proposed in the mid-1960's as a (rather crude) model of human memory. Each semantic network can be viewed as a directed, labeled graph whose nodes represent entities or concepts—such as, the person John Smith, the word "apple," or the concept of student—and their links represent binary relationships—for example, John Smith's relationship to his wife, the semantic relationship between the words "apple" and "fruit," or between the concept of student and those of person and student-number. The main feature of semantic networks is their

0018-9162/83/1000-0083$01.00

associative viewpoint, which admits an obvious graphical representation and can be used to define conceptual or implementational access paths. Some of these paths can be, and indeed have been, used to organize semantic networks along different dimensions. Traditionally, their major drawback has been their designers' overdependence on intuitions derived from node and link labels rather than on a formal semantics.[6] PSN deals with this issue by associating programs with every generic object in a PSN knowledge base, programs which specify how to operate on instances of that generic object. This approach is very much in the spirit of abstract data types and some of the ideas of the frame proposal.

Basic features. A PSN knowledge base consists of *objects,* which can be tokens, classes, links, relations, and programs. *Tokens* represent particular entities—such as the person John Smith, the number "7," and the string "story"—and are interrelated through *links* representing binary relationships between entities. *Classes* represent generic concepts—like person, number, and string—while *relations* represent such generic relationships as the brother-of relation between a male and another person. Each token is an instance of at least one and usually several classes, while each link is an instance of at least one relation.

Figure 1 shows a simple configuration involving two tokens, John and 23; two classes, Person and Number; a relation, Age; and a link relating John and 23. The figure also indicates the INSTANCE-OF relationships between John, 23, and the John-23 link as well as those between Person, Number, and Age. Four programs provide the "meaning" of each class by specifying respectively how to *insert, remove,* and *fetch* instances of the class, and how to *test* whether an object is an instance of the class. Similarly, each relation has four associated programs that specify how to insert or remove instance links, how to fetch all the objects in the range of the relation associated with a particular instance of the domain, and how to test if two objects are interrelated through an instance of the relation.

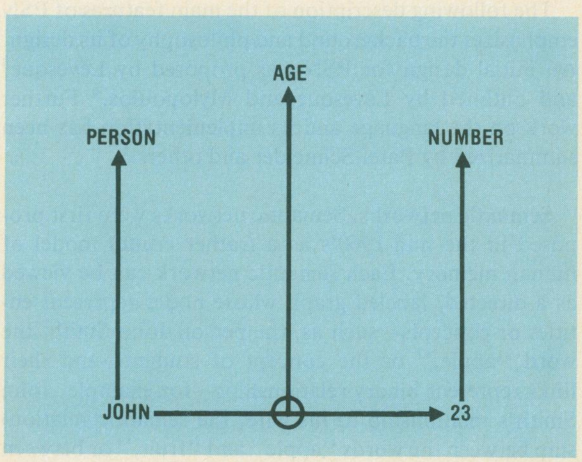


Figure 1. Two tokens, two classes, a relation, and a link in a PSN configuration.

A large knowledge base, like a large program, needs to be structured by intuitive organizational principles to be understood by its designers and users. Moreover, it is the responsibility of the knowledge-representation language to provide these principles. PSN offers three such principles in the primitive relations INSTANCE-OF, PART-OF, and IS-A.

We have already seen INSTANCE-OF links between tokens and classes that constitute their *types.* A token can be an instance of several classes (for example, John Smith may be an instance of Person and Student). In PSN, the INSTANCE-OF relation is used to relate *all objects*—tokens, classes links, relations, or programs—to their types. Thus we can have metaclasses that have other classes as instances, as indicated in Figure 2. Indeed, every object in a knowledge base must be an instance of at least one class or relation. Several built-in metaclasses make this possible:

- *Object* has all objects, including itself, as instances.
- *Class* has all classes, including itself and *object*, as instances.
- *Relation* has all relations as instances.
- *Program* has all programs as instances.

These metaclasses can be viewed as constituting the top of the INSTANCE-OF dimension. The depth of the dimension depends on the application and may range from two tokens, classes, or built-in metaclasses to any depth.

Aggregation is a second mechanism useful in structuring a knowledge base. Each concept can be considered an aggregate of simpler concepts, composed together in a particular way. For example, a person can be conceptualized as an aggregate of his physical characteristics or, alternatively, as an aggregate of his social characteristics, such as name, address, and insurance number. This organizational principle is realized in PSN through *slots,* which are associated with a class and specify the parts of the concept it represents. Each slot has a type and other constraints delimiting the objects that can be bound to the slot. By viewing a class one level higher than the classes defining the types of its slots, we can define a second organizational dimension, PART-OF, which places composite classes nearer the top and atomic classes at the bottom.

A third organizational dimension relates a concept to its specializations or generalizations. Thus, Person is related to Student, Employee, Male, etc., while Student is related to Graduate, Undergraduate, Full-time, etc. Respecting a long-established tradition, we call the relation between a class and its generalizations IS-A (so STUDENT IS-A PERSON). A partial order relation between classes, IS-A defines a generalization hierarchy, or *IS-A hierarchy.* A key issue for IS-A hierarchies is the kind of *slot inheritance* they support. Thus, if Person has slots Name and Age, these slots are inherited by its specializations, such as Student. Of course, Student can have additional slots such as "student-number" and "supervisor."

We mentioned earlier that, like everything else, programs are objects. In fact, each program is a class whose slots determine its parameters, prerequisites (conditions that must be true before each execution, and actions (that must be carried out during each execution). Since a program is a class, it has its own generalizations, or programs, from which it can inherit slots specifying parameters, prerequisites, and actions. Patel-Schneider and others[5] discuss this issue.

Another structuring mechanism is introduced to the PSN framework by Tsotsos[7] because of the interest in using PSN knowledge bases for recognition purposes. This mechanism, also proposed by Minsky,[2] involves *similarity links* for classes. These links can suggest other classes to be tried when the match fails between a given class and input data. After a failure, an *exception* is raised; the identity of the exception depends on the nature of the failure. This exception determines which similarity link should be used to suggest other classes to be tried for matching.

Implementation. The current implementation of the language is layered in the sense that each layer offers a set of representational features that includes features of the inner layers. Thus, PSN/0 offers the ability to construct knowledge bases structured only in terms of the INSTANCE-OF relation. PSN/1 adds IS-A and a simple form of PART-OF to further facilitate the organization of knowledge bases. PSN/2 introduces a more sophisticated and expensive version of PART-OF, along with similarity links and exceptions. The two systems described below use PSN/2. Further layers are being contemplated, for example, to treat programs as classes (at this point, they are treated as atomic objects) and to represent temporal and causal notions (currently, they are treated as constraints on slots values). An important property of layered implementations – and one that we are striving toward—is that a PSN/i knowledge base should also be a PSN/j knowledge base for $j<i$.

Realizing that the construction of a large knowledge base requires a friendly user interface, a number of programs have been designed and implemented by students in the University of Toronto's Department of Computer Science that are intended to help a PSN user visualize the knowledge base he is constructing and to facilitate its construction. One program by Hugh Meyers implements a number of display and editing functions, along with run-time support and version-control facilities. A second set of programs—the combined efforts of Ron Gershon, Yawar Ali, and Michael Jenkin—offers a restricted natural-language front-end facility for queries on the contents of the knowledge base. The replies take the form of small amounts of PSN code, simple natural-language statements, or raster graphics color displays for answers involving knowledge-base structure, such as the IS-A or PART-OF hierarchies, the similarity net, or temporal interrelations among classes.

Applications: Alven and CAA

Two large knowledge-based systems built at the University of Toronto use PSN as a knowledge-representation language. The Alven project,[7,8] begun in 1976, is currently being implemented a second time with a much expanded knowledge base and facilities that would make it suitable for clinical testing. Research on extending Alven's representation and control structure for application to electrocardiography began in 1979, resulting in a prototype design and implementation of the CAA system.[9]

The application environment. Because the performance of the human heart is complex, analysis of the heart's behavior requires several knowledge sources. The domain of cardiology, rich in temporal and causal relationships, is ideal for experimenting with representation and control strategies that manipulate such concepts.

Of the many medical tests that can gather information on cardiac performance, two involve cineradiography and electrocardiography. In cineradiography, X-ray image sequences are obtained of the left ventricle as it pumps in the patient's chest. Analysis of these images is performed mostly by human observation with little or no

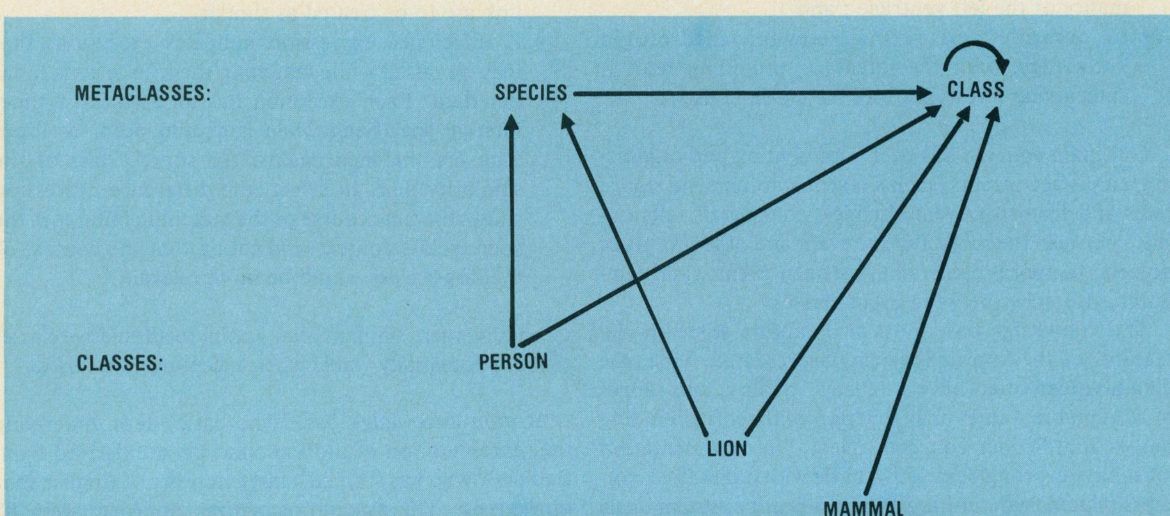


Figure 2. In PSN, metaclasses can have other classes as instances.

computer assistance, even though the motions of the left ventricle are more complex than the human visual system can objectively and consistently analyze.

We are studying left ventricles that have had corrective surgery—either coronary bypass or valve prosthesis. The goal of the study is to appraise their postoperative function and, therefore, the effectiveness of surgery. During corrective surgery, nine tiny tantalum helixes (markers) are implanted into each left ventricular wall, all roughly on the same plane. In addition, two clips are attached to the aorta as points of reference. After surgery, follow-up examination involves relatively simple cineradiography. Our first application attempt was in this domain, resulting in the design and construction of the Alven system, a system that analyzes the motion of the left ventricular wall.

The electrocardiogram, or ECG, can be viewed as the resulting signal of an "antenna" receiving the aggregate of the electrical activity of the heart, so there is no simple correspondence between signal features and individual electrical discharges in the heart. An important category of cardiac abnormalities that can, in principle, be detected from ECGs involves arrhythmias—that is, cardiac rhythm disorders. Detection of arrhythmias typically requires ECGs gathered over 24 hours. The Causal Arrhythmia Analysis, or CAA, system attempts to apply physiological knowledge of the heart conduction system together with signal knowledge to the task of recognizing and describing arrhythmias.

The Alven system. Alven is an expert computer-vision system for assessing the performance of the human heart's left ventricle. Input to the system consists of an image sequence obtained by cineradiography of the human left ventricle; however, the methodology is applicable to other forms of dynamic cardiac imagery. The goal of the Alven system is three-fold:

- to experiment with and study methodologies for understanding visual motion in the context of a rich temporal domain;
- to consistently and objectively evaluate the performance of the left ventricle ; and
- to compile and refine relevant cardiologic knowledge, with the intent of providing both a clinical and a research tool for cardiologists.

Our main concern has been representing and organizing temporal concepts, such as event start and end states, rates and forms of event changes, changes of relationships between the subject of the event and other events or objects; temporal constraints between events; and temporal differences between event classes.

The knowledge base must also capture some special features of the knowledge being represented. First, the data involved often have a "fuzzy" quality, and ranges of acceptable values may overlap with corresponding ranges for distinct concepts. Next, the representation must have a strong descriptive basis with qualitative concepts related to quantitative ones, since the system deals with real signal data. Finally, the knowledge base must include concept descriptions at different levels of abstraction, depending on the level of expertise of the person entering the knowledge.

Alven events are represented as PSN classes, while the semantic components of events—such as motion subject, volume changes, length changes, velocities, and trajectories—are represented as slots, their expected characteristics constrained. For each constraint, exceptions included are raised whenever a matching failure occurs, i.e., when the given constraint is dissatisfied. Moreover, the IS-A and PART-OF relations are used to represent different levels of abstraction.

Time is handled using an interval-based scheme: each event has a time interval slot as one of its components. In turn, the class Time-Interval has three slots, each representing an event's start time, end time, or duration. Duration is required because, in many cases, the start and end times for an event are unknown, and only the constraints on the duration are present. Such a scheme permits flexible reference to any point on a time-line of events, the representation of temporal constraints on any of these three time-interval features for a particular event, and the ability to form relations between such events as During, Overlapping, Before, and Simultaneous.

Perhaps the most useful representational feature for recognition is the similarity link, which serves as an exception-handling mechanism.

Perhaps the most useful representational feature for recognition is the similarity link. These links connect event classes by their patterns of temporal differences and similarities, serving as an exception-handling mechanism. Besides its source and target classes, each similarity link carries two types of information:

- A similarity expression specifies conditions under which the source and target classes of a similarity link are to be treated as similar.
- A difference expression indicates exceptions that may be raised while matching the source class to input data. Each exception has an associated time-interval specifying when it should occur as input data are matched against the source class of the similarity link. In effect, the difference expression shows the time course of the matching failures in the source-class context, indicating that the context of the target class would be more relevant.

Generally, then, similarity links point to alternative classes that are potentially viable when exceptions are raised.

Motion knowledge base. The basic ideas underlying the representation of motion concepts are derived from Badler,[10] who has studied a large number of English motion verbs. The system recognizes approximately 80 general-motion concepts, including a variety of two-dimensional translations and rotations; changes of area, length, and shape; and compound motions defined as sets of simultaneous, overlapping, or sequential motions.

Using this general-motion knowledge base, a domain-specific set of classes has been constructed, with the general motions as components. The domain of left ventricular performance has a broad spectrum of temporal concepts, including

- specific temporal patterns of volume change, long axis dimension change, or perimeter change,
- motions relative to the center of the left ventricle, such as inward or outward, and the extent of such motions,
- motion onset asynchronies, i.e., specific patterns of start-time delays, both normal and abnormal, between different objects, and
- motions of one part of the left ventricular wall in relation to another.

These concepts, plus all the standard terms used within cardiology—such as *hypokinesis* (abnormally diminished motion), *dyskinesis* (motion in the wrong direction), or *akinesis* (no motion, best indicated by no circumferential shortening)—are defined in the knowledge base. Normal left ventricles as well as 10 classes of common abnormalities are defined, forming a large knowledge base of approximately 450 classes.

To properly compare the sizes of the knowledge base in expert systems, we must keep in mind the "knowledge granularity" of each system. We have found that Alven classes generally contain more information than representational units of expert systems using production systems as their knowledge-representation language.

Control strategy. Our recognition control structure is based on the paradigms of (1) competition and cooperation among hypotheses and (2) hypothesize and test. The key feature is that the control structure is driven by the structure of the knowledge base. Recognition proceeds from the general to the specific along IS-A, from the primitive to the aggregate along PART-OF, across sets of mutually exclusive events along the similarity relation, and, finally, forward through time.

Hypotheses are ranked according to certainty factors. A modified relaxation labeling process[11] is used to update the certainty factors representing the notion of "good temporal continuity" of observed events. This process is based on conceptual adjacency, which specifies which hypotheses are competitors or complementary and in what respect. The compatibilities between hypotheses necessary for the relaxation labeling process are derived dynamically, depending on which conceptual adjacencies are present between hypotheses active during the input image sequence. The best, or highest ranked, hypotheses are used to derive the expectations for the next image. The generation of expectations incorporates constraints from each level of the motion's description. If image characteristics are not found as predicted, the IS-A relation is used to generalize the motion hypothesis. Such generalization relaxes the constraints while widening the image search space in a predictable manner.

The CAA system. Because ECG analysis has received considerable attention over the years, current systems can properly interpret up to 80 percent of all ECGs. However, the techniques used to build such systems, including signal transforms and simple contour analysis, do not seem to apply in the analysis of the remaining 20 percent—particularly when rhythm disorders are present in the signal.

The goal of our research in this domain is not only to provide a system capable of analyzing some of those remaining cases, especially those involving arrhythmias, but also to make three specific contributions to expert system technology, namely

- to represent and use causal knowledge in predicting events in the signal domain, which are observable, as well as events in the physiological domain, which are not directly observable but can be inferred by using causal knowledge;
- to create two knowledge bases, one each for the signal and physiological domains, and to explore the use of "projection" as a transduction mechanism between the two knowledge bases; and
- to represent statistical information, existing in abundance for many medical domains, in terms of PSN metaclasses, thus permitting default reasoning for those classes for which more specific descriptions are not available.

CAA has adapted much of the Alven control structure, including its use of IS-A, PART-OF, and similarity relations in activating hypotheses. Perhaps the most interesting CAA feature is the representation of causal information—that is, the description of a flow of influence from one distinguishable event to another. CAA causal links interrelate events in two important ways: first, they specify the existential dependency of an effected event on its causative event(s); second, they impose temporal constraints between causative and effected events. Thus, the effected events cannot occur without the occurrence of the corresponding causative events, with effects temporally following their causes. Several types of causal links are useful for our ECG domain:

- Transfer: the subject of the event normally completes the current event or state and proceeds to the following event.
- Transition: the subject is forced to terminate its current event and proceed to a new event.
- Initiation: the causative event, due to a given subject, triggers a new event of another subject.
- Interrupt: the causative event, due to a given subject, interrupts and forces the termination of an event by another subject.

These "one-shot" causal links are based on work described by Rieger and Grinsberg.[12]

Event prediction and statistical estimation. The role of causal links is to provide local integrity conditions for the events they relate. Thus, once an event has been identified, the system uses causal links to look ahead or back in time for other causally related events and predict their probable temporal locations. These events are then confirmed if their observable counterparts can be found in the predicted temporal locations of the signal.

If a predicted event is not directly observable, the system provides the event with default slot values. For example, when the start-time of a nonobservable event is determined using other observed events, the duration (and thus the end-time) is estimated by retrieving statistical information about the event and then fitting it to the current context.

Projection between knowledge bases. The CAA stratified knowledge base includes a morphological knowledge base, which describes such aspects of ECG waveforms as their shape and duration, and a physiological knowledge base, which maintains information about the cardiac conduction system. Conduction events are related to their morphological counterparts through a projection mechanism we have developed.

Waveform recognition starts with the detection of prominent waveforms in ECG signals, resulting in a starting set of hypotheses. Alven's recognition strategy is then applied to this set to generate and evaluate additional morphological hypotheses. As these hypotheses are generated, the system seeks to establish corresponding physiological hypotheses through *projection links* relating a morphological event class to a collection of possible physiological event classes with binding conditions for each. Since the overall arrhythmia recognition process must start with a rhythm hypothesis, which includes a sequence of beats, the system uses this process to examine each projected class and decide whether the class must be included in the current global hypothesis as a component hypothesis. Using this process, the system recursively hypothesizes consecutive beat events and rates the degree of consistency by testing them against the corresponding wave sequence. Shibahara[13] provides a detailed account of CAA.

We have presented a macroscopic view of the PSN project whose goal is to develop and test linguistic and other tools that facilitate the construction of large knowledge bases. Obviously, much remains to be done before this goal is achieved.

The fundamental idea that underlies this knowledge-representation project is that organizational issues are critically important in constructing and using large knowledge bases. We believe that our careful treatment of organizational principles and their application in organizing different kinds of knowledge and in developing more effective and natural recognition algorithms contributes to knowledge-representation and expert-systems research. ■

Acknowledgments

A number of colleagues have made important contributions to this project. Hector Levesque, now with the Fairchild AI laboratory, has been a driving force for the Toronto knowledge-representation group as well as a principal designer of PSN. Dominic Covvey, with his medical expertise and contacts, contributed greatly in the development of Alven and CAA. Gordon McCalla—now at the University of Saskatchewan, Saskatoon—and Lou Velli were part of the early phases of the project. Peter Patel-Schneider, Bryan Kramer, and Yves Lesperance helped in the design of PSN, while Andrew Gullen is responsible for the current implementation.

This research has been supported by the Canadian and Ontario Heart Foundations, by the Natural Sciences and Engineering Research Council of Canada, and by a contract sponsored by the Department of Supply and Services of Canada and administered by the Defense Research Establishment Atlantic.

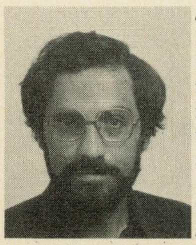
References

1. J. Mylopoulos and H. Levesque, "An Overview of Knowledge Representation," *On Conceptual Modelling*, M. Brodie, J. Mylopoulos, and J. Schmidt, eds., Springer Verlag, New York, 1983.
2. M. Minsky, "A Framework for Representing Knowledge," *The Psychology of Computer Vision*, P. Winston, ed., McGraw-Hill, New York, 1975.
3. H. Levesque, "A Procedural Approach To Semantic Networks," masters thesis, Dept. of Computer Science, University of Toronto, 1977.
4. H. Levesque and J. Mylopoulos, "A Procedural Semantics for Semantic Networks," *Associative Networks*, N. Findler, ed., Academic Press, New York, 1979.
5. P. Patel-Schneider et al., "PSN: An Extensible Knowledge Representation Scheme," *Proc. Canadian Soc. Computational Studies Intelligence Conf.*, 1982.
6. W. A. Woods, "What's in a Link: Foundations for Semantic Networks," *Representation and Understanding: Studies in Cognitive Science*, D. Bobrow and A. Collins, eds., Academic Press, New York, 1975.
7. J. Tsotsos, "A Framework for Visual Motion Understanding," PhD thesis, Dept. of Computer Science, University of Toronto, 1980.
8. J. K. Tsotsos, "Temporal Event Recognition: An Application to Left Ventricular Performance Evaluation," *Proc. Int'l Joint Conf. Artificial Intelligence*, 1981.
9. T. Shibahara et al., "CAA: A Knowledge-Based System with Causal Knowledge to Diagnose Rhythm Disorders in the Heart," *Proc. Canadian Soc. Computational Studies Intelligence Conf.*, 1982.
10. N. Badler, "Temporal Scene Analysis: Conceptual Descriptions of Object Movements," PhD thesis, Dept. of Computer Science, University of Toronto, 1975.

11. S. Zucker, "Production Systems with Feedback," *Pattern-Directed Inference Systems*, D. Waterman and S. Hayes-Roth, eds., Academic Press, New York, 1978.

12. C. Rieger and M. Grinsberg, "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanics," *Proc. Int'l Joint Conf. Artificial Intelligence,* 1977.

13. T. Shibahara, "CAA: Computer Diagnosis of Cardiac Rhythm Disorders from ECGs," PhD thesis, Dept. of Computer Science, University of Toronto (forthcoming).

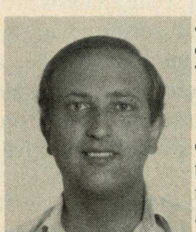


John Mylopoulos has been with the Department of Computer Science at the University of Toronto in Toronto, Ontario, Canada, since 1970. His research interests include knowledge representation and the design of knowledge-based systems.

Mylopoulos completed his undergraduate studies at Brown University in 1966 and his graduate studies at Princeton University in 1970, both in electrical engineering.

Tetsutaro Shibahara is working toward a PhD in computer science at the University of Toronto. He was previously an assistant professor in the Information Science Laboratory for Biomedicine at Kyushu University Hospital, Fukuoka, Japan, where he contributed to the installation of one of the largest medical data processing systems in Japan. His special interest is in building intelligent knowledge-based systems that emulate highly technical human activities, such as those in medical diagnoses, signal/pattern recognition and analysis, and computer-aided design and control.

Shibahara received an undergraduate degree in electrical engineering from Yokohama National University in 1967, and an MS in electrical engineering from the University of Illinois in 1971. He is a member of the Information Processing Society of Japan and a student member of the ACM and the IEEE Computer Society.



John K. Tsotsos is an assistant professor of computer science at the University of Toronto and Canadian Heart Foundation Research Fellow at Toronto General Hospital. His research interests include computer vision, knowledge-based systems, and application of artificial intelligence to biomedical image analysis and diagnosis.

From the University of Toronto, Tsotsos received a BASc in engineering science in 1974, and an MSc and PhD in computer science in 1976 and 1980. He is a member of the Canadian Society for Computational Studies of Intelligence, the ACM, and the IEEE-CS.

The authors' address is Dept. of Computer Science, University of Toronto, Ontario, Canada.