



Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

P. Bernus, Z. Létray

Intelligent systems interconnection:
what should come after open systems interconnection?

Computer Science/Department of Interactive Systems

Report CS-R8742

September

Bibliotheek
Centrum voor Wiskunde en Informatica
Amsterdam

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

69K 14, 69K 16, 69C 13, 69C 24, 69L 60

Copyright © Stichting Mathematisch Centrum, Amsterdam

Intelligent Systems Interconnection: What Should Come After Open Systems Interconnection?

Péter Bernus[†]

*Centrum voor Wiskunde en Informatica
P.O.Box 4079, 1009 AB Amsterdam, The Netherlands*

Zoltán Létray

*Computer and Automation Institute, Hungarian Academy of Sciences
P.O.Box.63, H-1502 Budapest, Hungary*

Abstract: The battle for the best knowledge representation language seems to be terminated. Not because anyone had found one, but because fairly obviously there is none. We propose to turn our attention to general architectures of knowledge representation. If theories to be represented become specified, we can use such general constructs in the development of individual architectures. Theories are generally used by some agent for specific purposes. Agents need to communicate with each other, in order to make use of their theories. We claim that a theory of computer-agenthood should describe communication and conversation among agents and the way they interact with the environment. A functional architecture for intelligent systems interconnection (ISI) is proposed. Subsuming functional layers of representations and theories are identified, without making commitments for particular choices in any one layer. We attempt to show the road how how to bring together represented conscious and unconscious, as well as not represented inherent knowledge. We do it in order to combine intelligence with effectiveness.

1980 Mathematics Subject Classification (1985 Revision): 68T30 [see also 68T05], {for entailment structures of conversation theory see 03B45}

1987 CR Categories: I.2.4 (knowledge representation formalisms and methods); related fields: I.2.6,C.1.3 and C.2.4; application: J.6 (CAD)

Keywords and phrases: representation languages, agents, CAD, conversation theory, knowledge representation architectures

Note: Paper presented at the First Eurographics Workshop on Intelligent CAD Systems, Noordwijkerhout April 1987. This paper also appeared in[1] by Springer-Verlag.

1. Introduction

Many attempts and proposals aimed at finding a universal, "best" knowledge representation language. The scope is very broad and encompasses several dimensions, depending on the choices between procedural or declarative, intensional or extensional representations [21] or the choice of blending them into so called multiparadigm systems [3], or choosing the right expressive power. There have been tasks around for which a given representation language was poor and others for which it was an overkill. We need

[†] The author is member of the Bart Veth IICAD group, which consists of (in alphabetical order) V. Akman, P. Bernus, W. Eshuis, P.J.W. ten Hagen, J. Rogier, T. Tomiyama, and P. Veerkamp.

to recognise that intelligent systems need more than one representation scheme. Many examples suggest [7, 13] that intelligent behaviour may not be achieved by single physical entities acting in one scene and having one single goal only. Intelligent behaviour supposes, instead, *controlled, self-conscious actions* with appropriately concentrated or diversified motivations. The actions are carried out in environments possibly populated with other similar agents and are caused by internal acts on varying higher levels. Represented (conceptualised) conscious and unconscious as well as not represented but implemented knowledge must be brought together in order to give an account on *effective intelligent behaviour*.

We claim that the integrated effect of mental and physical actions is decomposable into a cooperative function of communicating agents. Computational limitations and efficiency requirements do not allow one single theory to deal with real world problems in intelligent manner. Intelligence can only be achieved if different theories about the same subject matter compete. Agents, on one hand have to implement a theory of communication and conversation in order to confront the theories they represent. On the other hand an agent not only is a part of an environment containing other such agents, but itself must be treated as a possible set of cooperating subagents.

The interconnection of agents together with the representational task is functionally abstracted into a multilayered architecture. We show how representational layers are formed in computer based implementations and how is it possible to build succeeding strata of theories on their top. In Section 3.1. we identify four such representational layers:

1. Senses and actions layer.
2. Computer representation layer.
3. Logic representation layer.
4. Abstractions layer.

With the help of this the theory of the agent can be built up. The agent's theory can possibly be clustered, one such sub-theory being the *theory of communicating agents*. Section 3.2 presents the stratification of such a theory, giving an account on:

5. Model management layer.
6. Situation and discourse modelling layer.
7. Motivations and agenthood layer.

These seven layers form the *Intelligent Systems Interconnection (ISI) proposal*. Brachman similarly stratified the possible meanings of constituents in a knowledge representation tool [4] while he investigating semantic nets. This technique was successfully used for representing a theory of activity and time in [18].

As it has already been mentioned there are no single prescribed or specific choices made in the general architecture. This is necessary because practically the functionality of every layer can be made specific depending on the actual set of agents to communicate through these layers. In the discourse modelling layer, for instance, different languages can be used (special purpose interface languages, graphic languages and last but not least only natural language).

In the layer of logic representation(s) the chosen kind of logic cannot be frozen once for all. Agents should not carry the overhead of the implementations of obscure logics if they do not need them, but the trickiest modal logic should also be introducible into this

layer in case a more demanding application (agent) needs it.

The architecture presented in this article serves two purposes. One is analysis of the requirements which an intelligent CAD system's central modelling language should meet. Here the essential intellectual activity underlying the motivations of the involved agents is design in a broader sense. Therefore one of the agents or a group of agents represent a design theory.

The second aim is to investigate what are the *useful abstractions* to help us describe theories of intelligent agents. In this sense the present general model is an attempt. We hope to stimulate other people to make their criticisms and try to contribute to the formulation of a more widely acceptable, well specified architecture and conceptual framework — essential for commensurable research results in this area.

2. Functional architecture for knowledge representation

2.1 Implementations, representations and theories

A CAD system must operate on the principles of a theory which gives sufficient account on (interactive) design. An implemented CAD system is at the ultimate a model of that theory. A theory needs a language to be represented in. In a sense, we could say that there is no theory without a representation language. The two, however, are different things which is apparent from the *same* theory being representable in several *different* languages. (Two theories are considered the same, if they have the same set of possible models in sense of model theoretic semantics [11].)

The concepts of a possible design theory (such as models, theory of mechanical machines, strategies, paradigms, goals, criteria etc.) should be representable in such a language. The distance between the available computer languages (for possible implementation of a computer model of the design theory) and the above concepts is fairly big. Therefore the expressive power of the available representation languages must be extended with great care. There might be several reasons for us to do so, be it for extending the possible set of representable (and in their representation distinguishable) theories, or for reducing the complexity of the representation. A good overview of possible meanings of expressive power can be found in [12]. We shall collectively call them useful enrichments. The result of enrichment is a language, of which the concepts subsume the concepts of the simpler language.

In practice the search for an adequate representation language can proceed in the opposite direction also. Given concepts of a theory in a language (e.g. natural) we may wonder about what kind of general concepts would suffice instead. Assume that we are able to describe the way the subsumed concepts can be used to create the subsuming ones. There are several ways — essentially all different kinds of *abstractions*. One is specialisation of subsumed concepts to form subsuming ones; another is aggregation of subsumed concepts into a single subsuming one; etc. (see [8] for more).

Suppose, that we use a logic language for representation (i.e. there are axioms which assign truth values to the well formed formulae of the language). In this case the theory in question (here theory of design) should be represented in forms of axioms as well. For a language with first-order predicate calculus see for instance [6]. In case we already have a logic language, but its expressive power is not high enough, we can do the following:

- a.) Extend the syntax of the language (that will need part of b. also)
- b.) Extend the logic by
 - inventing new truth values
 - modifying/extending the set of independent postulates which pin down those well formed formulae of the language which invariably have the same truth value
 - modifying/extending the rules of inference.
- c.) Add defined symbols to enrich the language, so that one can express the same thing in several different ways.

Given the so enriched language it will be possible to express our theory in it. It is, however, natural to seek for some structures in the concepts of our theory and (as mentioned above) for layers of subsuming concepts. The language is used for creating the representation of a "basic theory". The concepts of such a basic theory can be utilised to *define* concepts of a theory based (layered) on it. A theory of mechanisms for example can be represented in terms of concepts and their epistemological relations, while a theory of concepts can be represented in terms of a language with an appropriate logic (see [4]). Of course the actual forms of definition (the different kinds of abstraction) are to be operations expressible in the basic representation language (thus determining the largest set of real world models our theory can possibly have, no matter what layering strategy was used to describe the theory). It must be noted, that concept definition is sometimes very demanding concerning the expressive power of the representational language.

The layering of theories is useful, and in our view unavoidable, because of the complexity and unstructuredness of the representation resulting otherwise. From the viewpoint of higher level (layered) theories the more basic theory's concepts can be considered as enrichments of the representation language, or alternatively, as just another representation language (provided that the underlying theory carries along all the representational features of the "real" representation language). The layering upwards into strata of theories has a counterpart downwards, where several switches of representation language take place before the representation is actually tied to a physical implementation. This especially holds for computer models.

By extending the expressive power of the representation language, the completeness status of the language changes also. The gain of being able to say something that was impossible to state before should be weighted against the added complexity of theorem proving necessary to carry out when the represented theory is set to work. One attempt to trim the trade-off is discussed in [5].

We think that the decision on what order of logic and what kind of modalities to include depends on the complexity of the conceptual relations within a problem area. The abstraction techniques for defining new concepts based on known ones have for instance an important effect.

Care must be taken also, because the decision has consequences to the applicable inference control strategies as well. In one case one might wish to get guaranteed response within predictable time for specific problems. In other situations, like searching for an adequate assembly which satisfies given functional constraints, we need to contend with a relaxation strategy or appropriate heuristics. The complexity of this second task is a result of functional abstraction in the reasoning process.

Consequently representation languages of different agents can and in fact must be different. Applicable inference control strategies are also problem- or situation-dependent and thus not common to all agents.

The functional architecture of agents in general must be layered for reasons of

- gaining the appropriate expressive power and system qualities,
- mastering the complexity of the agent's theory.

From the above discussion it seems natural to admit the multiple agent assumption (i.e. agents should consist of multiple subagents): only competing theories of the real world can resolve the antagonism of the need to deal with complicated problems and the need to act in limited time. Let us call the users of the theories in question agents. The agents, having to compete with each other, need to implement a theory about themselves (i.e. a psychology). This is what enables them to reason about activities and knowledge of their own. (Accepting agents as implemented psychological theories follows [7].) Competition involves at least two parties, so the reasoning has to cover activities and knowledge of the opposite parties as well. Therefore the multiple agent assumption must be accompanied by a theory of the communication of such agents. Information exchange with the rest of the environment through perception and manipulation should also be accounted for. For purposes of design theory (see [20]) only a limited ability of communication is needed, if compared with such "societal" communication behaviour as learning, teaching, or negotiating and other fancier conversation situations.

It is instructive to look at the representational problem from the point of view of conversation theory [9,15-17] where the representation of an agent's theory is not only related to the communication channel through which conversation with other agents can take place, but also to the agent's perception and motoric system. We think that the much argued question of whether intelligent beings "really" use representations or representations are only external observers' fabrications, may become a non-issue in this light. The world "really" becomes meaningless in this question, since the use of representations is optional for a large part of the theories. If an interrelated set of attributes of biological or artificial structures *can* be accepted as a model for a theory, then tokens of that theory do not *need* intermediate representations. Electric impulses in a computer just happen to persist without being represented in the computer itself, but that does not exclude the possibility of representing impulses as concepts in a computer simulation language. On the other hand the representation of the impulses in a computer becomes a self-knowledge only if the mapping from the theory of electric impulses to the real impulses can directly or indirectly be established. The presented arguments can be applied to the mapping function as well: it should be an attribute of the agent — not necessarily represented but by all means exhibited. The mapping function can be implemented directly through identity relations or indirectly through sensory means. In the light of this discussion we see that the need for a particular agent to have representations of unconscious concepts depends on the physical quality of the agent (flesh and blood, wires and n-p transitions etc.). An agent can act without representing the act itself, but hardly can reason about the act without having represented it first (be the "act" physical or mental).

Since percepts and manipulative capabilities of present day computers are desperately limited (see efforts for getting off the ground in [14]), the reinforcement between representations and real world models in computers lies quite ahead in time.

It would be interesting to find real world models where the representation of percepts and of motoric controls were identical to the percepts and controls. Example: if an established connection causing a motoric action is at the same time the implementation of the action's conceptual representation.

The lowest level of an agent's functional architecture is thus the sensimotoric layer (interpreted in CAD as the implementing computer together with its I/O system). Through this layer can denotations finally get connected to the real world and not to artifact models.

The whole theory of design (just as our human knowledge about the surrounding world) will not be able to rely entirely on such first hand denotations, but on knowledge engineered, human made, built-in representations or to some extent on "second-hand" denotations acquired through communication with other agents (including humans). The semantic integrity of the CAD system will, on the ground of the previous reasons, rely on the users and maintainers of the system.

2.2 Choices in the general architecture

The presented views allow for a multitude of different types of knowledge representation systems to be built (and thus theories to be represented). One aspect of the permissiveness lies in the allowable physical architectures. If we consider agents as autonomous systems, this autonomy is always relative to the environment, i.e. the only thing common to two agents is their environment. But exactly in computer based implementations the definition of what should be considered as environment is left to us (the external observers).

The trivial implementation is to use separate computers for separate agents and connect those machines. It is also possible to implement agents in a multitask environment and provide intertask communication. Even one language with corouting possibility can be considered environment. The need for parallel processing for agents depends on the discourse situations they are expected to enter. Many discourse situations can essentially be described without using parallelism. We think, that future CAD systems will need a little more than that: a sequentially described discourse between agents may do, but in-between agents should be capable of processing in parallel (e.g. a human user plus the interface to the human user as one agent and other agents of the CAD system).

The choice of representation language remains with the agent itself, and is basically dependent on the agent's theory. Even on the level of equivalent expressive power there are basic choices, which influence the effectivity and the maintainability of the agent's representation of its theory. In general we expect an agent not only to represent its theory in a static way, but also its functions: e.g. a simulator program should represent both the target of simulation (say a conveyor) and also the functions of the simulator (say being able to gather statistics). If so, the control of the theorem proving task (or equivalent procedure) must also be representable in general. Different kinds of conceptual dependencies, like abstractions, can have an important role in reasoning. Inheritance is one special kind of inference that may need control.

For the design of a representational language, the language concepts should form an orthogonal system. The same holds for independent theory-layers. Orthogonality assures locality of effects caused by modification and causes longer life expectations and less vulnerability for a system in general. Orthogonality is defined here as both of the following

criteria being met:

- meaningful transformations on the data in that layer should be expressible using combinations of the functions in that layer (completeness criterion of the functions), whereas
- combinations of functions within one layer must not be restricted but by the meaning of the composite transformation that the combination is intended to perform, e.g. no restriction from lower level layers (of implementation or theory) is permitted to penetrate the layer in question — unless this restriction is functional (independency criterion).

3. A layered knowledge representation architecture

3.1 Intra agent representation of knowledge

We identified layers 1-4 in the functional architecture of an agent.

The *Senses and Actions layer* is responsible in the end for the computer interpretation of the agent's theory. Any theory in the layer's language will be related to the computer's pieces, states and relations over pieces and states, including the tokens of information passing through the I/O system of the computer (procedural primitives for locomotion and other motoric, as well as manipulative functions over manipulables, and means for sensing and creating implementations of messages, implementations of memory and primitives for learning and unlearning — if appropriate). There is no unique set of canonical primitives; it all depends on the "creature".

In computer systems this heterogeneous representational vehicle is a sort of assembly language which directly relates to the physical parts of the machine: I/O registers, memory, interrupts, graphic I/O, etc. In order to hide all this from the higher level representational tasks, these things are abstracted into a general purpose representation language (high level programming language).

The *Layer of Computer Representations* is built on top of the former one and features everything that is necessary from the point of view of good software engineering. It supports good system design and maintenance practices and makes it possible for computer based systems to survive their original implementation environment. The change to this higher level of representation layer improves systems qualities and at the same time expressive power can also be added (e.g. emulation of parallel computation on sequential machines).

While the concern for introducing a new layer on top of layer 1 was "good engineering", tuning the knowledge representation language to the needs of an agent may require the introduction of a third layer with adequate expressive power and computational efficiency.

We call this the *Layer of Logic Representations* because it is straightforward to compare knowledge representation languages by formulating their logic. This is not to say that the representation language of every agent must by all means be a logic language.

Especially in this layer it is necessary to maintain extensibility of the language: the completeness criterion of orthogonality will take care of not building in theories into the representation language, while the independency criterion separates representational issues

from inference control. The representation should not know about the inference control and the inference control mechanism(s) must not know anything specific from the representations (except what the possible inference operations are). Heuristics should be separated from general inference control functions.

In addition, in multiparadigm systems the automatic propagation of propositions among representational parts goes along with limitations in expressive power, but the behaviour of the system is predictable. If we tolerate the inconsistency between parts, we can circumvent these limitations but only at the cost of unpredictability in general (task-specific debugging of representations).

Augmenting the richness of the representation language by abstractions is a common way and has much been discussed by different authors [4, 8]. The *Layer of Abstractions* enriches the language by defining conceptual dependencies (such as aggregation, generalisation, etc.) needed to economically express theories. The expressive power of the language is not effected, for the definition of these conceptual dependencies would otherwise have not been possible.

3.2 Intelligent Systems Interconnection (Inter-Agent Communication)

Layering can be continued in many different directions, depending on the actual theory to be built up. Additional system layers are called layers of theories. After realising that we need multiple agents in order to implement intelligent systems which cope with real world problems (e.g. Intelligent CAD systems), we must consider how they are interconnected. Any agent taking part in the communication should have a sub-theory of communication which is valid for the given "society" of agents. This sub-theory is a possibly multilayered architecture of theories, generally accepted by all the participants (general architecture). The value of general architectures (such as the famous ISO-OSI model [22]) convinces us of the utility of putting considerable effort into the elaboration of such framework for the interconnection of intelligent agents.

Stated in a simplified way, the intelligent systems interconnection (ISI) architecture should be a framework for the description of how intelligent agents talk to each other. The approach of the architecture is functional, allowing space for different specific cases as well as for variations in the actual implementations. For practical purposes we need to limit the generality of the architecture to the needs of interconnecting agents within a CAD system (including users as agents). It should be noted, however, that not every aspect of information exchange can be accounted for by communicating via language discourses. Sometimes a common referential model is also needed to which both agents can point to.

If an agent's theory is complete, it has not too much to communicate with others. This is (especially in CAD) not the case: incomplete theories ("models of the solution") must be dealt with. The *Layer of Model Management* could be the basis for dealing with incomplete information, building alternative approaches and representing assumptions. In a given moment several (maybe even contradictory) states of affairs can be accepted as attainable by augmenting the agent's knowledge. Information exchange among agents needs to be accompanied by the agent's possibility to reason about the effects of receiving new information and reason about what acts can be taken to get some new information.

Nonmonotonic assumptions can help the agent in that reasoning. In the end, these assumptions should be checked by directly consulting the real world (through

manipulations and percepts — e.g. a CAD system for programs could run the program and measure its performance), or asking other agents about the status of the given assumption. It is also possible that the underlying theory is not complete. Exchanging theories among agents is a most interesting phenomenon (learning, unlearning, reinterpretation, etc.) a part of which can be described by conversation theory [15-17].

Using the theory of multiple models, a theory compartmented in the *Situation and Discourse Modelling Layer* could be worked out. We think that conversation theory provides substantial basic material for working out this layer's exact specification.

The present layer introduces the notion of situation. It should provide concepts which describe, how situations are effected by actions and communication. Conversation, data capturing and interpretation, message generation, and other actions all influence

- the models of the situations
- the models underlying the situations.

These concepts can be used to talk about the roles, goals and motivations of an agent. So again we can work out a layered theory which sits on top of the former. We call this new layer *Layer of Motivations and Agenthood*. Primitives of self-knowledge [7] could be defined, for self-reference and reference to other agents in their quality of being implemented psychological theories. The agent can take acts rooted, via conceptual or direct links, in the senses and actions layer. Whenever the theory of agenthood proves an act, this act (by the agent) takes place.

The multiple models of the model management layer make it possible to divide the agent into sub-agents, and let them have their own theories of the subject matter. The theory of agenthood can use the situation modelling layer's theory to model the fact, that not only one agent is acting on his behalf.

Language understanding and generation is one of the very complex acts that an agent can indulge in. Potentially every layer of theories as well as representations can take part in it. Understanding and generation is not a function defined in one of the layers. It is rather a part of the total architecture. The roots are in the senses and actions layer for perception and articulation and the results are used by higher level actions. Such actions can be for example morphological or syntactic transformations. The agent is allowed to use representations, as in analysis using symbolic computations. In other cases he may not make use of them, as in construction based on memory. (Note, that memory is not necessarily representational.)

Higher level semantic functions carried out on concepts have a place in the basic theory-layer (model management). Some semantic functions require the use of situations for understanding and selecting adequate response (thus should be defined in that layer). Interactions of discourses and motivations, on the other hand, go into the layer above (motivations and agenthood).

In the same way that different theories call for different representations, there should not be a single language for communication, either. Humans also, beside using natural language, make use of several different sorts of languages, be them graphical, written or spoken, general purpose, or tightly coupled to a discipline.

Some interaction between agents does not need the highest level functions of these, and even mere context-free question-answering can take us far enough for exchanging information between compatible but incomplete representations of the agent's theories. In

the other extreme we take into account all the aspects of the communication. These are the syntactic construct, the functional structure (how it conveys the semantic information) and the interstructural aspect (how it functions in the given communication situation) [10].

In practical CAD projects (until appropriate theories are worked out) substantial simplification is necessary. The communication between agents (especially those which are built out of extant tools) can be managed by a "design manager", which possesses a "general model" as in the General Modeling System in [2] or a metamodel" as in the Integrated Data Description Schema of the IIIICAD system [20]. The design manager, implemented as an agent, represents all the relevant information about the object of design, and works basically in query answering mode with other agents (or slightly generalised, he acts as a moderator-mediator between agents).

To simplify matters it is possible to accept the representation language of the design manager as a language for communication also. In that case the specific ISI implementation will have much in common with the representational architecture of the "design manager". While agents in general represent their theories about the target of design, the design manager will represent a theory of design (for details see [19]).

The layers to build on top of the representational ones (1-4) must therefore contain a theory of design evolution [20]. A sub-theory of it should describe the communication between the agents, or at least between any agent and the design manager.

Conclusion

- a.) We argued that intelligent agents should be decomposable into interconnected agents. Each of them should have it's own theory, representation language and specific implementational details.
- b.) We proposed a functional layering approach for knowledge representation with four layers of representation and three for one particular theory, that of communication. Layering the knowledge representation architecture is not a goal *per se*. Switching representation languages once or more times could be replaced if appropriate technology was available for different implementations. Layering of theories is much more inherent in the knowledge to be captured, because of the basic complexity of conceptual structures.
- c.) We also proposed that conversation theory could be incorporated into this multilayered architecture and provide artificial agents with the possibility of directly relating their theories to the real world (not only to theories of other agents).
- d.) We conclude, that intelligent CAD systems should be integrated in the following way.
 - d.1.) Create computer based agents which contain different extant CAD tools.
 - d.2.) Represent the theory of the tool incorporated by the agent.
 - d.3.) In every agent implement the same *theory of communicating agents*.
 - d.4.) Implement a design manager agent.

Acknowledgements

The idea of ISI has first been laid down in February 1986 at a preliminary oral presentation at the Winter School on Conceptual Modeling in Visegrád, Hungary under the title: "The Idea of a FRPRSM System for Advanced System Analysis and CAD" — by

the same authors. At that time FR stood for Frames, PR for production rules and SM for Smalltalk (referring to object modelling). The present article is a refinement of these ideas reflecting work done at CWI in the framework of the IIICAD project at the Interactive Systems Department. We would like to thank the editors for their useful comments on the version of this paper which appeared in the preprints of the Noordwijkerhout workshop.

References

1. *Intelligent CAD Systems 1 — Theoretical and Methodological Aspects*, eds. P.J.W ten Hagen and T. Tomiyama, Springer-Verlag, Berlin (1987).
2. BERNUS, P. AND HATVANY, J., "Computer Aids to the Design of Integrated Manufacturing Systems," *Computers in Industry* 1(1), pp. 11-19 (1979).
3. BOBROW, D.G., "If Prolog is the Answer, What is the Question? or What it Takes to Support AI Programming Paradigms," *IEEE Trans. on Software Engineering SE-11*(11), pp. 1401-1408. (November 1985).
4. BRACHMAN, R.J., "On the Epistemological Status of Semantic Networks," pp. 3-50. in *Associative Networks: Representation and Use of Knowledge by Computers*, ed. N.V. Findler, Academic Press, New York (1979).
5. BRACHMAN, R.J. AND LEVESQUE, H.J., "Krypton: A Functional Approach to Knowledge Representation," *Computer*, pp. 67-73. (October 1983).
6. COPI, I.M., *Symbolic Logic*, Macmillan Publ. Co., New York (1979).
7. DOYLE, J., "A Society of Mind," CMU-CS-83-127, Carnegie-Mellon Univ., Dept. Comp. Sci. (1983).
8. FOX, M.S., "On Inheritance in Knowledge Representation," pp. 282-284. in *Proc. IJCAI*, Tokyo (1979).
9. GERGELY, T. AND SZOTS, M., "About Representation of Semantics," pp. 227-234. in *Progress in Cybernetics and Systems Research, Proc. of the Symposium of Austrian Soc. of Cybernetic Studies (Vol.11)*, eds. R. Trappl, N.V. Findler and W. Horn, Hemisphere Publ. Co. (1982).
10. HANKISS, E., *The Literary Work as a Complex Model*, Magvetö, Budapest (1985). (in Hungarian)
11. HAYES, P.J., "The Second Naive Physics Manifesto," pp. 1-36. in *Formal Theories of the Commonsense World*, eds. J.R. Hobbs and R.C. Moore, Ablex Publ. Co., Norwood, NJ (1985).
12. ISRAEL, D.J. AND BRACHMAN, R.J., "Some Remarks on the Semantics of Representation Languages," pp. 119-146. in *On Conceptual Modeling: Perspectives from Artificial Intelligence, Databases and Programming Languages*, eds. M.L. Brodie, J. Mylopoulos and J.W. Schmidt, Springer Verlag, New York (1984).
13. MINSKY, M., "K-lines: a Theory of Memory," *Cognitive Science* 4, pp. 117-133. (1980).
14. NEGROPONTE, N., *The Architecture Machine*, MIT Press, Cambridge, MA (1970).
15. PASK, G., *Conversation, Cognition and Learning*, Elsevier, Amsterdam (1975).
16. PASK, G., "Developments in Conversation Theory," *International Journal of Man*

- Machine Studies* 13(4), pp. 357-411. (November 1980).
17. PASK, G., "Concepts, Coherence and Language," pp. 421-427. in *Progress in Cybernetics and Systems Research, Proc. of the Symposium of Austrian Soc. of Cybernetic Studies (Vol.11)*, eds. R. Trappl, N.V. Findler and W. Horn, Hemisphere Publ. Co. (1982).
 18. SATHI, A., FOX, M.S., AND GREENBERG, M., "Representation of Activity Knowledge for Project Management," *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-7(5)*, pp. 531-552. (September 1985).
 19. TOMIYAMA, T. AND YOSHIKAWA, H., "Extended General Design Theory," pp. 95-130 in *Design Theory for CAD*, eds. H. Yoshikawa and E.A. Warman, North Holland, Amsterdam (1986).
 20. TOMIYAMA, T. AND TEN HAGEN, P.J.W., "Organisation of Design Knowledge in an Intelligent CAD Environment," in *Expert Systems in Computer Aided Design*, ed. J. Gero, to be published by North Holland, Amsterdam (1987).
 21. TOMIYAMA, T. AND TEN HAGEN, P.J.W., "Representing Knowledge in Two Distinct Descriptions: Extensional vs. Intensional," CWI Report, Centre for Mathematics and Computer Science, Amsterdam (1987).
 22. ZIMMERMANN, H., "OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection," *IEEE Trans. Commun. COM-28*, pp. 425-432 (April 1980).