

An Architecture for Combining Ways to Think

Push Singh and Marvin Minsky, MIT Media Lab, Cambridge, MA, {push, minsky}@media.mit.edu

Abstract—Why have AI researchers not been able to give computers human-like ‘common sense’, the ability to think about ordinary things the way people can? In our view, the source of the difficulty is that they too often seek after types of cognitive architectures, kinds of representations, and methods of inference that are based on some single, simple process, theory, or principle. Despite their elegance, no single one of such techniques can capture the diversity of mechanisms needed to reason about the broad range of commonsense domains—for example, those that require reasoning about temporal, spatial, physical, psychological, social, and self-reflective matters. In this paper we describe aspects of an architecture that we are developing to support the construction of AI systems resourceful enough to combine the advantages of many different ways to think about things, by making use of many types of mechanisms for reasoning, representation, and reflection.

1. INTRODUCTION

Why have AI researchers not been able to give computers human-like ‘common sense’, the ability to think about ordinary things the way people can? In our view, the source of the difficulty is that they too often seek after types of cognitive architectures, kinds of representations, and methods of inference that are based on some single simple process, theory, or principle. Despite their elegance, no single one of such techniques can capture the diversity of mechanisms needed to reason about the broad range of commonsense domains—for example, those that require reasoning about temporal, spatial, physical, psychological, social, and self-reflective matters. Ordinary commonsense thinking spans so many different types of problems and depends on so many forms of knowledge that more unified frameworks, ones that primarily make use of a single type of representation and mode of reasoning, are stretched beyond their capacity. Just as biological systems have no single, simple principle for their operation, we expect that cognitive systems will contain just as numerous and heterogeneous a variety of components.

So rather than seeking a ‘unified theory’, we seek instead to develop an architecture that can support a great diversity of cognitive processes. In this paper we briefly describe aspects of an architecture that we are developing to support the construction of AI systems resourceful enough to combine the advantages of many different ways to think about things, by making use of many types of mechanisms for reasoning, representation, and reflection.

2. THE OVERALL ARCHITECTURE

The central idea behind our architecture is that the source of human resourcefulness and robustness is the diversity of our cognitive processes: we have many ways to solve every kind of problem—both in the world and in the mind—so that when we get stuck using one method of solution, we can rapidly switch to another. Thus, at the top level, our architecture is organized as follows. When the architecture encounters a problem, it first uses some knowledge about ‘problem-types’ to select some ‘way-to-think’ that might work. However, its first approach is likely to fail in various ways. Then if certain agents that we call ‘critics’ notice particular ways in which that approach has failed, they either suggest ways to adapt that method, or suggest alternative ways-to-think, as shown in Figure 1 below.

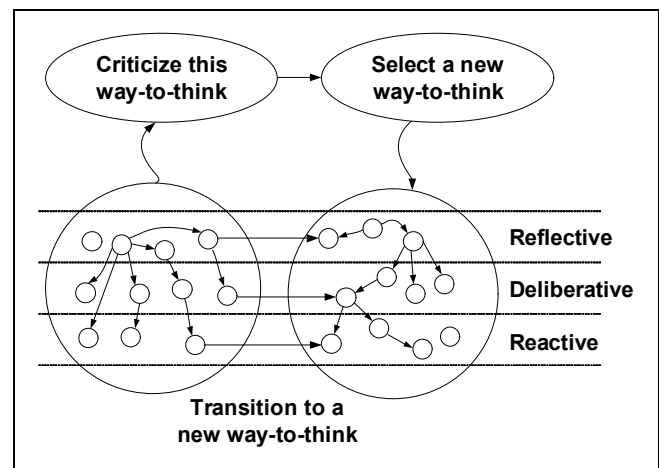


Figure 1 - Transitioning to a new way-to-think. Small circles represent agents and other resources specific to that way-to-think, spanning the many levels of the architecture.

Each such way-of-think disposes the system towards using certain types of knowledge, methods of reasoning, and other kinds of resources to solve the problem at hand.

3. MAJOR FEATURES OF THE ARCHITECTURE

3.1 Many Ways-To-Think

In our view, ordinary commonsense thinking involves not a single mechanism for inferencing, but rather a tremendous array of more specialized ways-to-think. Each way-to-think is suited for certain problem-types, such as finding the best path to walk between two locations, guessing whether someone likes you or dislikes you, or determining after solving a problem what exactly you did that helped to solve it, and what was just a waste of time. In addition to specialized ways-to-think, there are many more general ways-to-think that correspond to well-known heuristic problem solving strategies, such as:

Knowing How—We don't see most problems as problems at all, because we already know how to solve them.

Analogy—Try to adapt a method you've used before. Few problems ever seem utterly novel because they remind us of similar ones.

Dividing and Planning—When we can't solve a problem all at once, break it down into smaller parts, and regard those parts as separate goals or 'stepping-stones'.

Simulation—Mental experiments in virtual worlds can help when actions are dangerous.

Proof by Contradiction—Try to show that your problem cannot be solved, and then look for a flaw in that argument.

Reformulation—When other methods fail, try to find a new way to describe the problem.

Simplifying—First ignore the parts of the problem that seem difficult. Then restore them, one at a time. This may not solve the problem itself, but may help us to understand it.

Elevation—If you are bogged down in too many details, describe things in some more abstract way. If your description seems too vague, then switch to a more concrete representation.

Meta-Reflection—Ask, "Why does this problem seem so hard?" or "What could I be doing wrong?"

Impersonation—If your own ideas don't seem adequate, think of someone who is better at this, and imagine what that person might do.

Cry for Help!—If none of your usual methods work, try using someone else's resources!

3.2 Multiple Reflective Layers

Existing architectures have largely focused on levels in

which they only react or deliberate, but to make machines more versatile, they will need more knowledge about themselves, that is, on higher, more reflective levels. They need processes for noticing problems in their own activities, knowledge about how to debug those problems and—when that fails—knowledge about how to transition to different, alternative ways of thinking. Our architectural design presently includes four reflective levels beyond the deliberative level, as shown in Figure 2.

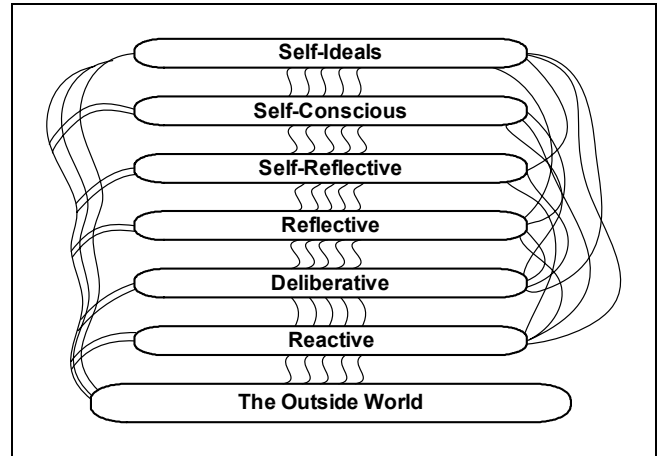


Figure 2 - Architectural Tower of Reflection

Each reflective level is concerned with coordinating, managing, and responding to problems in the levels beneath. Our present reflective levels include:

Reflective—Thinks about the recent deliberations of deliberative level, such as whether a subgoal has gotten the system closer to a supergoal or further away.

Self-Reflective—Thinks about its activities with respect to large-scale models of its abilities and limits, for example, what kinds of things the system knows, how it typically behaves in similar situations, and so forth.

Self-Conscious—Thinks about itself in relation to others entities, for example, to compare its own skills and experiences with those of others.

Self-Ideals—Thinks about itself with respect to its highest level and longest term goals, perhaps by imagining what one of its 'imprimers' (role-models) would think of its activities.

3.3 Varieties of Mental Critics

We have been formulating an ontology of the types of reflections that a mind might make of itself. We have focused in particular on the *negative*—what are the ways that a mind might criticize its own activities? We have produced a catalog of 'mental critics' that we envision as populating the upper reflective levels of the system. Some

examples of these include the following:

Credit to Wrong Action—Credit was given to a particular action for producing an outcome, when it was really produced by another action.

Assumed False Preconditions—An action has failed, and the critic realizes that a precondition for that action did not hold.

Unable to Decide—Several methods seem to apply to the current problem, but the system has not decided on one.

Wasted Reasoning—While formulating a plan of action, the situation has changed and the problem no longer needs to be solved.

Lack of Experience—The system has had only a few experiences dealing with this type of problem.

Ignored Relevant Object—The system had expected a particular outcome from a given action, but this did not happen because of some object or condition that had not been noticed.

Transient Conditions—The system had been depending on certain conditions to hold for a period of time, but in fact those conditions only held more briefly.

Misremembering—The system’s remembered description of an event has turned out to not have been accurate.

See Singh [1] for a longer list of such mental critics.

3.4 Parallel Representations

Our architecture is designed to keep updating multiple representations of knowledge in parallel. This enables it to quickly switch between different ways-to-*think* because, instead of starting all over, each newly activated way-to-*think* will find an already-prepared representation. This means that the system will rarely get stuck because those alternate ways-to-*think* will be ready to take over when the present one runs into trouble, as shown in Figure 3.

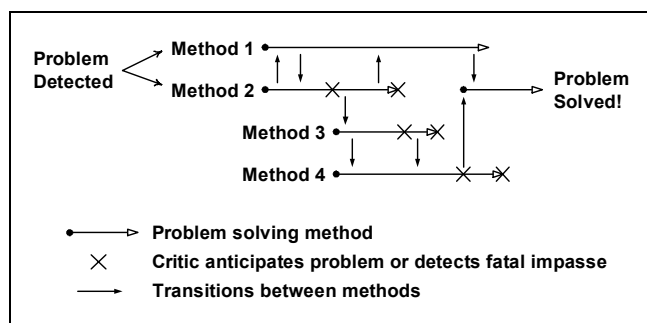


Figure 3 - Switching between parallel methods or ways-to-*think*.

We plan to do this not by employing any single principle for reformulation, but instead by using a family of processes that we together refer to as ‘panalogy’ (a term that derives from ‘parallel arrays of analogous representations’.) These techniques support the synchronizing and sharing of information between different ways-to-*think* concerned with the same or similar problems. We can also make more partial changes like the representation language they are using, the types of assumptions they are making, the methods that are available to them for solution, and so forth. By actively maintaining correspondences between representations across ways-to-*think*, we can rapidly switch from one way-to-*think* to another. Here are some of the methods of ‘parallel representation’ that we are considering:

Context panalogy—Maintain multiple contexts for running processes. Method and slot names dynamically bind to different values in the different contexts. Each slot can refer to multiple representations of its value, and each method can map to different ways of solving the given problem. Each slot lookup or method call is performed in several different ways simultaneously.

Event panalogy—Maintain the correspondences between the slots of frames representing the before and after state of events and actions, as in Minsky’s ‘paranome’ idea applied to transframes [2].

Model panalogy—Maintain descriptions of different models or interpretations of a situation, like seeing a cardboard box as simultaneously a folded up sheet of cardboard and as a rigid cube [3]. Each of these interpretations may suggest different inferences or courses of actions.

Theory panalogy—Maintain mappings between different logical theories of the same domain. This may require translation tables that allow descriptions in one representation to be translated into the other. This is similar the notion of contexts as described by McCarthy [4], which uses ‘lifting rules’ that make explicit the assumptions to add and remove from assertions when transferred it from one context to another.

Realm panalogy—Maintain correspondences between different ‘mental realms’. For example, Lakoff and Johnson [5] have argued that the knowledge and skills we have for reasoning about space and time are also used to help reason about social realms, and there are pervasive analogies between these seemingly very different domains.

Structure panalogy—Maintain connections between fragments of compositional descriptions, so as to build a larger model from multiple, incomplete partial models. For example, one might approximate a human skeleton with just a dozen bones rather the actual 206 bones of a normal adult, or as a set of sub-skeletal structures consisting of the

bones of the head, neck, chest, etc. This is related to Minsky's frame array idea [2] where many views of an object are linked by their common parts to together form a more realistic or complete model than any individual view could form.

Ambiguity panalogy—Maintain connections between ambiguous senses of predicates. For example, the preposition 'in' can refer to a wide range of different relations far more specific than any division provided by ordinary dictionary senses. Rather than selecting a particular such relation when formulating a problem, we can instead maintain the ambiguity between those predicates, so that we can draw on our understanding of all those related senses to answer questions about how one thing could be 'in' another.

3.5 Heterogeneous Reasoning Methods

Making effective use of multiple reasoning mechanisms in the same system requires a deep understanding of how different reasoning methods compare, yet today we still have little idea of what the strengths and weaknesses of the various AI methods really are. Thus an important thrust of our work is to develop a theory of 'problem-types' that lets us think more effectively about when and when not to apply different AI methods.

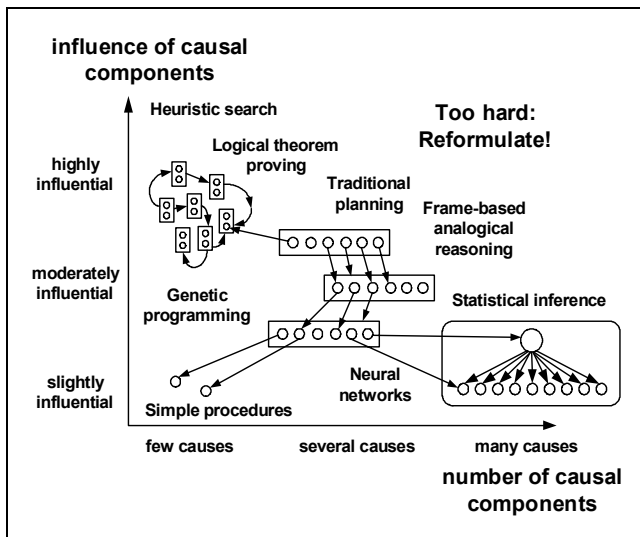


Figure 4 - The causal diversity matrix. Each common AI technique such as logical or statistical inference is matched to problem-types with particular causal structures.

As an initial step in this direction, Minsky offers as an example the 'causal diversity matrix' [6], the theory matrix presented in Figure 4. Here each distinct method (such as logic, case-based reasoning and statistical inference) is assessed in terms of its competence at dealing with systems with particular kinds of distributions of causal components. For example, recognizing the textures in a visual scene

requires 'summing' many small pieces of evidence, usually distributions of nearby pixels where no one pixel is terribly important, and thus statistically based techniques are appropriate. On the other hand, logical techniques are sensitive to the slightest inconsistency, hence may be more suitable for reasoning about computer programs, where changing a single bit could drastically change the performance of the whole system. Minsky discusses such issues at length in [7].

Our goal is for the reflective levels of the architecture to be able to make reasonable selections about which problem solving techniques to use in the current situation. Such a theory of problem-types would let the architecture itself classify new problems according to their general features, so that it could then select types of methods, forms of representations, or entire architectural reconfigurations suitable for solving those types of problems.

Of course, this is just a coarse first step toward developing a heuristic theory of when and how to apply different AI techniques in different contexts. Modern methods of planning, theorem proving, constraint satisfaction, case-based reasoning, analogy making, statistical inference, neural network propagation, and so on, are all techniques suitable for solving different classes of problems, and we need to find ways to exploit their advantages and avoid their disadvantages.

How could such a theory be further elaborated? In addition to the numbers and strengths of causal conditions, here are some other dimensions we are exploring along which to organize problem-types:

- *Observability*—Is this a situation for which we can directly observe the important variables and processes?
- *Causal structure*—What are the degrees of independence between, and the relations among the causal components?
- *Uncertainty*—How uncertain are the results of our actions and observed events?

Similarly, we need to classify representations and ways-to-think along multiple dimensions:

- *Tractability of inference*—How quickly can we solve the desired set of problems using this representation?
- *Ease of reflection*—How easy is it for other processes to say useful things about each representation or way-to-think? For example, neural networks are very opaque, while semantic networks are fairly transparent.
- *Expressivity*—How expressive is this representation? Does it support objects, useful quantifiers, and so forth?

3.6 Multiple Reasoning Domains

A primary design goal for our architecture is to support reasoning across multiple domains within a single problem solving situation, as in the “blocks world” scenario depicted in Figure 5.

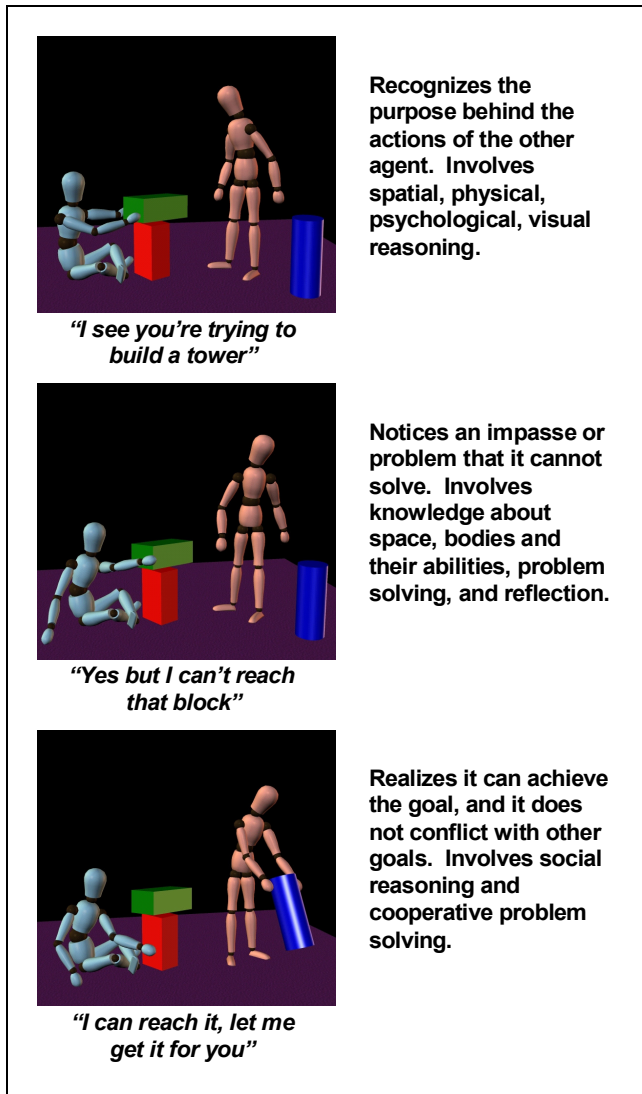


Figure 5 - Problem solving across mental realms.

Even the limited world of Figure 5 presents many kinds of problem-types that span a wide range of mental realms. But no present-day AI system can demonstrate even the following range of commonsense skills:

Spatial—Reasoning about the ways in which objects and the parts of objects are oriented and situated in relation to one another. (Which of those blocks is closest to me?)

Physical—Reasoning about the dynamic behavior of real objects with masses and interacting surfaces. (What would happen if I removed that middle block from the tower?)

Bodily—Reasoning about the capabilities of one’s physical body. (Can I reach that block without having to get up?)

Visual—Reasoning about the world that underlies what can be seen. (Is that a cylinder-shaped block or part of a person’s leg?)

Psychological—Reasoning about one’s goals and beliefs and those of others. (What is the other person trying to do?)

Social—Reasoning about the relationships, shared goals and histories that exist between people. (How can I accomplish my goal without the other person interfering?)

Reflective—Reasoning about one’s own recent deliberations. (What was I trying to do a moment ago?)

Conversational—Reasoning about how to express one’s ideas to others. (How can I explain my problem to the other person?)

Educational—Reasoning about how to best learn about or teach someone else about some subject. (How can I generalize useful rules about the world from experiences?)

Such realms are not cleanly separated but are highly cross-connected; the physical realm makes intimate use of the spatial realm, and the social realm makes intimate use of the psychological realm. Furthermore, each such realm is further composed of sub-realms concerned with narrower, more specialized issues. It’s not clear whether there is a ‘best’ such decomposition of commonsense reasoning abilities into realms, but it has been a useful heuristic for us to break apart knowledge and reasoning processes in this manner.

Each such realm may require its own special representations. These include representations such as occupancy arrays that describe the spatial relationships between objects and the paths between them, social networks that describe who knows who in a social group, plot units that describe how the goals of multiple agents interfere and support each other, and so forth. In addition to such well-understood representations, we are also formulating new types of representations to describe the processes within minds themselves, such as the mental critics we described earlier.

3.7 Learning and Endowment of Knowledge

Most modern work in machine learning consists of techniques that rely on some uniform heuristic for credit assignment and adaptation to new examples. However, in our view human-level learning requires an architectural approach that is intimately connected to the rest of thinking. Our efforts have been focused on the ‘credit assignment problem’—how can a system decide how to

improve itself, especially as it gets more complicated? As the system grows, solving this problem requires a more and more elaborate self-model, so that the reflective layers of the system can try to predict the effects of changes it might make to itself. Powerful learning is enabled by a powerful ability to reflect.

We all agree that learning is of value, but we don't all agree on where to start. Some researchers would like to start with nothing; however an architecture that comes with no knowledge is like a programming language that comes with no programs or libraries. With the exception of Cyc, most groups have ignored the problem of initially populating their architectures with useful knowledge. Thus, in addition to more autonomous forms of learning, we have been exploring new ways to build commonsense databases, so that we can distribute our architecture with useful knowledge resources. We have been successful in gathering a great deal of knowledge from the general public via our 'Open Mind' web sites, which to date have accumulated nearly 750,000 pieces of knowledge (about a half of that in Cyc, at a much lower cost, but with much less pre-specified structure) [8, 9, 10]. We are also exploring other methods for building commonsense databases by using learning techniques in physical and social simulations. We are putting much effort into this because we believe that approaches that start out with too little knowledge will not achieve enough versatility in any practical length of time.

4. CONCLUSIONS

One might question the need for an architecture with so many components. Our view is that present-day theories still explain too few aspects of the many things that human minds do; therefore, we should elaborate rather than simplify, by building theories with more parts, not fewer. This general philosophy pervades the architecture, with its many layers, representations, ways-to-think, critics, analogies, and other ingredients. Later, once we have built an adequate architecture, it may make sense to try to simplify things.

FURTHER INFORMATION

Aspects of this architecture are presented in the meeting reports for the IBM Research Symposium on Commonsense Computing and St. Thomas Commonsense Symposium [11, 12]. This architecture will be described more fully in Minsky's forthcoming book *The Emotion Machine* [13].

ACKNOWLEDGEMENTS

We would like to thank Aaron Sloman and Erik Mueller for many valuable discussions about these ideas. This work is supported by the sponsors of the MIT Media Lab.

REFERENCES

- [1] Push Singh (2003). A preliminary collection of reflective critics for layered agent architectures. To appear in *Proceedings of the Safe Agents Workshop (AAMAS 2003)*. Melbourne, Australia.
- [2] Marvin Minsky (1986). *The society of mind*. New York: Simon and Schuster.
- [3] Push Singh (1998). *Failure-directed reformulation* (M.Eng. thesis). Department of Electrical Engineering and Computer Science, MIT.
- [4] John McCarthy (1993). Notes on formalizing context. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*.
- [5] George Lakoff and Mark Johnson (1990). *Metaphors we live by*. Chicago, IL: University of Chicago Press.
- [6] Marvin Minsky (1992). Future of AI technology. *Toshiba Review*, 47(7).
- [7] Marvin Minsky (1991). Logical vs. analogical or symbolic vs. connectionist or neat vs. scruffy. *AI Magazine*, Summer 1991.
- [8] Push Singh (2002). The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. Palo Alto, CA.
- [9] Push Singh, Thomas Lin, Erik Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu (2002). Open Mind Common Sense: Knowledge acquisition from the general public. *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. Lecture Notes in Computer Science*. Heidelberg: Springer-Verlag.
- [10] Push Singh and Barbara Barry (2003). Collecting commonsense experiences. In *Proceedings of the Second International Conference on Knowledge Capture (K-CAP 2003)*. Florida, USA.
- [11] John McCarthy, Marvin Minsky, Aaron Sloman, Leiguang Gong, Tessa Lau, Leora Morgenstern, Erik Mueller, Doug Riecken, Moninder Singh, and Push Singh (2002). An architecture of diversity for commonsense reasoning. *IBM Systems Journal*, 41(3), 530-539.
- [12] Marvin Minsky, Push Singh, and Aaron Sloman (forthcoming). The St. Thomas commonsense symposium. To appear in *AI Magazine*.
- [13] Marvin Minsky (forthcoming). *The emotion machine*.