Computing commonsense

P Singh, M Minsky and I Eslick

How can we build systems with 'commonsense', the thinking skills that every ordinary person takes for granted? In this paper, we describe a multi-agent architecture for enabling commonsense reasoning which is in development at the Media Lab. The system reasons about the kinds of fundamental entities that show up in nearly all situations — such as people, objects, events, goals, plans and mistakes. The architecture supports multiple layers of reflective reasoning, mechanisms for coherent reasoning across multiple representations, and large-scale control structures called 'ways to think'. We first describe the main features of our architecture and then discuss its application and evaluation to an artificial life scenario.

1. Introduction

Computing devices have become indispensable to modern life, but they remain largely ignorant about the people they serve and the world that they so deeply permeate. When machines fail — by crashing, allowing spam into our mailboxes, or failing to retrieve documents we seek — they demonstrate a profound lack of basic intelligence. Modern computers lack the ability to innovate when presented with a new situation; more, they lack even the knowledge that we, or they, exist at all. We believe the next epoch in computing systems will arise when we can give machines the capacity for more self-awareness and 'commonsense' — the ability to think, learn, and act in the world with the resourcefulness and flexibility exhibited by people.

Over the years, many complex problems have been largely solved, from chess-playing programs to logistics and planning, but invariably these solutions employ domain-specific heuristics and representations developed by the programmer. When conditions fall outside the predefined parameters of their representations, the programs are unable to generate new heuristics or modify existing logic to achieve their goals. The failure of the field of artificial intelligence to make significant progress towards machines with human-level intelligence has resulted from this emphasis on domain-specific problems and specific mathematical techniques. We need to set our sights on problems where the solution cannot be accomplished with fixed heuristics or particular mathematical models.

Commonsense reasoning is one of the oldest and most difficult problems in artificial intelligence. McCarthy [1] proposes that a program has commonsense 'if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything that it is told and what it already knows'. Commonsense inferencing involves many types of reasoning, including analogical, statistical, logical, and heuristic methods. Solving even a small set of commonsense scenarios provides an ideal anvil for the development of robust reasoning systems, because the scenarios require broad-based knowledge, many different reasoning procedures, multiple representations and the ability to generalise to new situations via analogy, rule induction and other kinds of model-learning processes.

At the Media Lab we are designing a system for 'computing commonsense', an AI operating system for the next generation of computing devices. This effort encompasses both large-scale knowledge acquisition, such as Singh's Open Mind Common Sense project [2], and Lenat's Cyc project [3], and work on the architecture for reasoning described here. This system will be applicable to a broad spectrum of applications, from high-end servers to consumer devices such as cell phones, cars and houses. We envision applications such as:

- telephones that understand the social relationships among participants,
- computers that see bugs and failures in their own processing and repair and update themselves,

- houses that optimise energy usage while adjusting to the daily habits of their occupants,
- augmented interfaces based on rich modes of communication such as natural language,
- virtual actors that engage more naturally with a target audience.

And there are surely many important applications we cannot yet see. Widely deployed commonsense reasoning will alter how we think about what computing devices can be useful for.

Related papers describe the methods we are developing to engineer large commonsense knowledge bases [4], build toolkits for reasoning with this knowledge [5], and put these reasoning systems to use in practical applications [6]. This paper will focus on architectural aspects of large-scale commonsense systems — given large quantities of multiple-represented commonsense knowledge and multiple methods for reasoning with this knowledge, how can we fuse these heterogeneous resources and techniques into an integrated system that is more flexible and resourceful than any individual technique?

We are developing an architecture for commonsense reasoning that supports this kind of cognitive diversity. Our architecture is a multi-agent system able to reason about the most basic and pervasive kinds of concerns, such as people, objects, events, goals, plans and mistakes. It can also engage in the kind of self-reflection needed to think about and improve its own abilities over time. In this paper, we emphasise the higher order and novel features of this architecture. These organisational techniques will imbue the architecture with robustness in the face of new and varying problems.

commonsense reasoning will alter how we think about what computing devices can be useful for

While we are inspired by human capabilities and experiences, we do not see this architecture as a sufficient performance model of human reasoning or human cognitive development. We also do not require that the model reproduce the entire range of human mental and linguistic feats. Rather, it will address the commonsense reasoning problem by demonstrating the utility of, and the strategy for, employing multiple representations in pursuit of automated reasoning.

In the subsequent sections, we will elaborate on the following features of the architecture.

- Agents

  The architecture consists of a large collection of agents. Each is a specialised type of knowledge or a fragment of a cognitive process. At any time, only a subset of the agents is active, which produces a particular way of thinking.

- Thinking

  Ways of thinking are selected by special, self-reflective agents called critics and selectors.

- Panalogies

  When the current way of thinking fails to make progress, the architecture can switch to another, more effective way of thinking by making use of panalogies, which are connections between the representations used by different agents.

- Agent organisation

  Agents can be organised along two main axes:

  — vertically stacked layers provide for self-models and reflective reasoning, enabling debugging and repair of lower-level behaviours by upper-level processes,

  — realms can cluster domain-related agents together such as physical reasoning, emotional knowledge, social scenarios, etc.

We will further discuss applications of the architecture to an artificial life scenario where simulated people face realistic physical and social problems, enabling quantitative evaluation and experimental reproduction of the architecture's performance.

2. Ways of thinking

Our architecture consists of a vast diversity of agents, numbering perhaps in the millions. Our use of the word agent implies a fine degree of granularity; each agent is roughly on the scale of a small unit of knowledge such as a rule, a small semantic network, or a subroutine in a computer program. Agents can be regarded as types of programs that perform the kinds of functions specifically involved in mental activities, such as expecting, predicting, repairing, remembering, revising, debugging, acting, comparing, generalising, exemplifying, analogising and simplifying. Nothing about these agents needs to be common or uniform. We can base each agent on a different type of process with its own distinct kinds of purposes, languages for describing things, ways of representing knowledge, and methods for producing inferences.

How can we organise this seemingly unmanageable diversity? The basic principle that we use is that at any time only a subset of these agents are active — and each of these states produces a specific 'way to think'. This is illustrated in Fig 1.

In other words, the architecture is not a single kind of 'machine', based on a single type of algorithm or method of reasoning. Instead, in different contexts it becomes a different machine by turning on different subsets of agents. Each such subset results in a fundamentally different way of thinking about things. Some examples of these ways to think include:

- solving problems by making analogies to past experiences (e.g. Carbonell [7]),

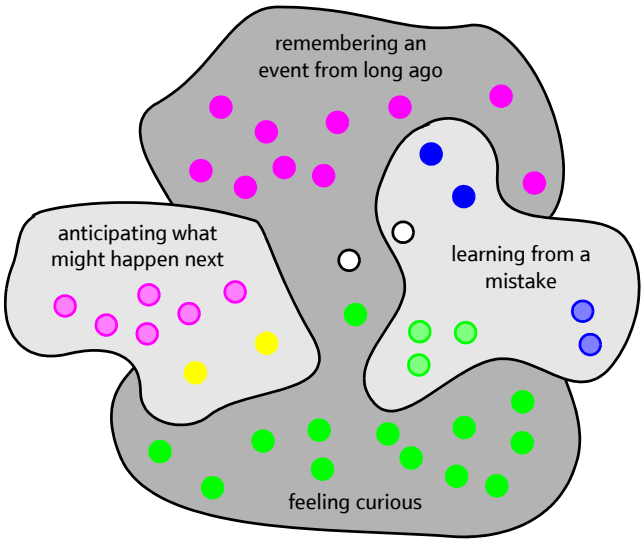


Fig 1 Each way to think results from the activity of a particular subset of mental agents.

- predicting what will happen next by rule-based mental simulations (e.g. Kuipers, [8]),
- constructing new 'ways to think' by building new collections of agents (e.g. Minsky, [9]),
- explaining unexpected events by diagnosing causal structures (e.g. Davis, [10]),
- learning from problem-solving episodes by debugging semantic networks (e.g. Winston [11], Sussman [12]),
- classifying types of situations using statistical inference (e.g. Pearl, [13]),
- getting unstuck by reformulating the problem situation (e.g. Amarel, [14]),
- inferring the state of other minds by reusing self-models (e.g. Minsky, [15]).

These ways to think are not fixed. Because each of these ways to think is the result of the activity of a set of agents, new ways to think can be assembled by forming new collections of agents. This idea originally evolved from the k-lines of Minsky's Society of Mind theory [16] — k-lines are special agents that turn on other sets of agents, which is a simple but powerful mechanism for disposing a mind towards engaging relevant kinds of problem-solving strategies, retrieving particular fragments of knowledge, and selecting or prioritising sets of goals, memories of particular experiences, and the other mental resources that might help the architecture in coping with a situation.

Each such way to think is more or less self-contained, and the mind can be seen as a distributed collection of such ways of thinking with no 'central control' — the original vision of the mind presented in the Society of Mind.

3. Trains of thought

What controls which ways to think are active at any moment? When does the system switch to new ways to think? In our architecture, there are special agents concerned primarily with selecting ways to think. We call these agents critics and selectors. It is easiest to conceptualise them as chronic or persistent questions and concerns that direct cognitive activity. These critics and selectors act together to produce trains of thought, which are sequences of ways to think. The selection takes place against a background that incorporates the current scenario or event, the active goals, and any recent contextual state. Example traces include the following.

- What is going to happen next following this event? (Will my goals be influenced by this event?) (Have I ever experienced anything like this event before?)
- What would explain this event? (Is it surprising that this event was initiated by some particular person?) (Are there other potential explanations besides the most obvious one?)
- What is the best thing for me to do now? (What would my parents/wife/friends think of me if I did that?) (Would doing that thing benefit or hurt me in the long run?)
- What can I learn from this event? (Would that learning result in my becoming better at some goal G?) (Am I likely to remember this event in the future?)
- What can I learn from this failure? (Should I fail more in this manner to learn more things of this type?) (What could I have done to avoid this failure?)
- How long will it take to perform this action? (Can I complete it prior to some deadline?) (Have I spent too much time on this problem already?)
- What sorts of things might go wrong while performing this action? (Would any of those things damage me?) (Can I prepare for any of those failures?)

Each of these mental questions invokes a way of thinking that attempts to answer it. However, when a way of thinking begins to fail, the architecture can switch to another more appropriate way to think. This happens through the operation of critic agents that recognise general classes of failures and impasses during thinking, and as a result invoke selectors that choose alternative ways to think, as shown in Fig 2.

Critics are invoked collectively in the architecture while selectors are invoked individually. All critics triggered by the current state are turned on, attempting to pick a selector. The current way to think dictates which critic and associated selector is chosen. This may be done by a simple heuristic, such as invoking the highest valued selector indicated by the sum of all active critics that try to select it. A more sophisticated method involves training a learning device within each way to think that captures the aggregate historical summary of which critics or selectors were most useful in specific contexts. The system context and/or goal state will dictate how decisions are made — a learning goal may lead to trying new things, perhaps at random, while a problem-solving high-level goal will prefer the event most likely to succeed.

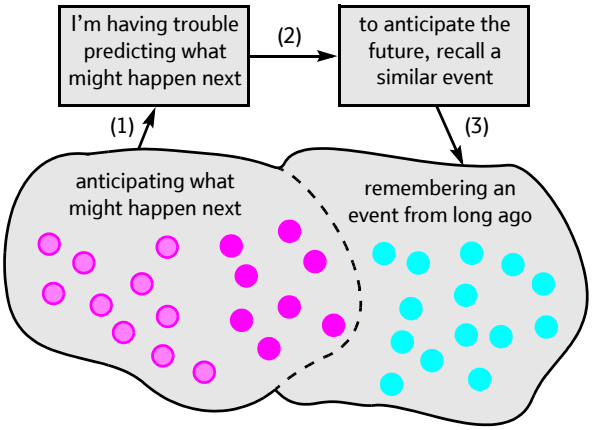


Fig 2 When one way of thinking is beginning to fail, mental critics recognise the failure, and mental selectors invoke alternative ways to think.

This model is central to our methodology for encapsulating high-level problem-solving heuristics like those in Polya's [17] classic work on mathematical methods. At the moment many of our critics and ways to think are described in pre-formal English statements rather than in terms of more well-defined representations, although we are working on the latter problem (see Singh [18] for examples of how to represent reflective critics). Further, these catalogues of critics intuitively invoke expectations in the programmer, helping to guide implementation and debugging of the supporting agents. This is one powerful way to help human designers conceptualise abstractions about the massive meta-management issues inherent in the architecture's behaviour.

Reflecting on one's own subjective experiences helps to form expectations about the behaviour of the architecture. As such, trains of thought are analogous to psychological notions of mental attention. Yet, many people will experience a subjective impression of parallel activities taking place across different models of a situation or different relevant memories. The new ideas, memories or models seem to 'pop out' of the background and into our sphere of attention. The analogous phenomenon in our architecture is the unified action of multiple representations operating under a set of panalogies.

critics and selectors determine 'ways to think'

4. Multiple representations

As before, when a way to think becomes ineffective the architecture tries to switch to another. Normally, this would require a certain 'start-up time' during which the agents of the new way to think gather the information they need to do their jobs. However, our architecture performs transitions more efficiently by having special support for multiple representations, allowing agents that represent similar information to synchronise automatically what they know. When an agent writes to a representation, it also updates the representations of related agents in parallel, including the ones used by agents that are at the moment quiescent. Thus, when the architecture selects a new way to think, instead of having to start from scratch, it will find many of its agents already prepared for the situation.

We do not use any single technique for synchronising representations across multiple agents, but instead make use of a family of processes for synchronising and sharing information. We refer to these together as a panalogy (a term that derives from 'parallel analogy'). Here are some of the methods of panalogy we use.

- Event panalogy

  Maintains the correspondences between the elements of event descriptions across multiple representations. For example, when we imagine the consequences of buying a fancy new car, we can rapidly switch between considering the effects of that purchase on our social status (which it may improve) and on our financial situation (which it may hurt).

- Model panalogy

  Maintains descriptions of different models or interpretations of a situation, like seeing a door simultaneously as both a portal and as an obstacle. Each of these interpretations may suggest different inferences or courses of actions, and if we discover that in fact we have a key to the door, inferences based on that interpretation are already available for use.

- Theory panalogy

  Maintains mappings between different theories of the same domain. For example, we may choose to describe a theory of time where events are treated as atomic points on a timeline, and another theory of time where events are treated as occurring over intervals on a timeline. When one theory is unable to answer a given question about, for example, the order in which some set of events occurred, we can try to switch to the other.

- Realm panalogy

  Maintains analogies between different 'mental realms'. Lakoff and Johnson [19] have argued that the knowledge and skills we use 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.

- Abstraction panalogy

  Maintains connections between different abstract descriptions. 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. We can link each of these different abstractions to form a more realistic or complete model than any individual abstraction could form.

- Ambiguity panalogy

  Maintain links between ambiguous senses of predicates. For example, the preposition 'in' can refer to a wide range of relations far more specific than any division provided by ordinary dictionary senses. Rather than

selecting any particular such relation when describing a situation, we can instead maintain the ambiguity between those relations. We can then draw on our understanding of all those related senses to answer questions about how one thing could be 'in' another.

There are two implementation choices implicit in the panalogy model. Firstly, we could use a late-binding model in which the final state of the failing agency updates the selected agencies of the next way to think. Secondly, we could update related representations as the source representation is modified, thus providing opportunities for critics and selectors to be activated as a consequence of local computations in related, but inactive, agents.

As an example, imagine a scenario where the architecture is embodied in a mobile robot. The current way to think is computing 'get to physical location X'. The selected agency performs path planning through a spatial map of the current physical environment. One state in the path planning may take the robot near another robot, Robbie. An event panalogy would automatically propagate the adjacency information to other agencies. One such agency could be a logical rule-based system that computes social preferences based on logical rules such as 'near(self, talkative world-agent) → potential(<current state>, delay, high)'. The panalogy would write an assertion to the agent's current state such as 'near(self, Robbie)' and a local update of the agency's state would result in the conclusion that a delay would be experienced under that state.

In the lazy evaluation scenario, the critic would not be activated until the path was chosen, causing invalid paths to be explored. If the panalogy update were aggressive, then the critic would be activated earlier, causing a new way to think to be selected. In both cases the most likely next way to think would backtrack and retry the search ('try to find a better path') to avoid the unfavourable adjacency condition. One of the flexibilities of our architecture is that this implementation choice can be a parameter of the current way to think merely by inclusion or exclusion of the 'lazy panalogy evaluation' agent in the current way to think.

5. Multiple layers of reflection

An important feature of our architecture is that we designed it to be a highly self-aware system. In contrast, most architectural designs in recent years have focused mainly on ways to react or deliberate, giving agents no special ability to reflect upon their own behaviour or to improve the way they think. However, to achieve increased versatility, machines will need better ways to recognise and repair the obstacles, bugs and deficiencies that result from their activities. In our architecture, the agents are organised into six layers, as shown in Fig 3.

Reflection is important because in a system that regularly faces new situations, existing communities of agents will sooner or later find themselves to be deficient when faced with a new context or problem. Thus we need agents whose role is to debug and improve the behaviour of other agents in the mind.

Fig 3 The mind can be divided into layers, each managing and reflecting upon the layers beneath.

We have been formulating [18] a catalogue of the kinds of agents involved in reflecting upon the behaviour of other agents — in other words, agents that bridge these layers. Mental critics are an important type of agent whose job is to notice problems in the agents in the layers beneath, and the architecture will be teeming with them. We have a catalogue of such mental critics.

- **Reactive critics** detect problems in the outside world and turn on ways to react to those problems. Much of the behaviour of animals can be described by networks of such critics and many modern robots have hand-coded heuristics that incorporate this knowledge. For example:

  I hear a loud noise → move to a quieter place
  I feel hungry → follow the smell of food
  I am far from something I want → walk towards it
  I feel scared → run quickly to a safe place

- **Deliberative critics** search for solutions. When faced with a problem without a directly applicable stimulus-response rule, we build a model of the situation, for example, as a network of goals, actions, and their effects, in which we can search for a solution. But in the case of a machine, even the simplest problems may result in large search spaces, and the architecture needs to learn techniques to simplify the model or to search more effectively:

  Action A did not quite achieve my goal → try harder, or try to find out why
  Action A worked but had bad side effects → try some variant of that action
  Achieving goal X made goal Y harder → try them in the

opposite order
These events do not chain → change one of their end-points to match

- **Reflective critics** — when faced with a very hard problem that we are not making much progress on, we may need to reflect on the techniques that we are using to solve that problem:

  The search has become too extensive → find methods that yield fewer alternatives
  You have tried the same thing several times → some manager agent is incompetent
  You overlooked some critical feature → revise the way you described the problem
  You cannot decide which strategy to use → formulate this as a new problem

- **Self-reflective critics** — when reflecting on why the methods we use fail to help, then we may criticise ourselves:

  I missed an opportunity by not acting quickly enough → invoke a critic that warns me whenever I am about to do that
  I can never get this exactly right → spend more time practicing that skill
  I let my other interests take control → tell one of my friends to scold me when I get distracted
  I do not seem to have the knowledge I need → quit this and go to graduate school.

With mental critics, we need not build architectures under the impossible constraint that all agents always produce the perfect inference or suggestion for course of action, because it is impossible to anticipate all of the conditions under which the architecture may be used. Instead, when there are failures in reasoning, the architecture examines recent activity and attempts to diagnose and resolve the problem.

6. Multiple realms of thinking

The range of problem types that people cope with is enormous, and at first glance seems to consist of a haphazard collection of commonsense knowledge and inference skills. As architects and designers of such a system, how does one organise and reason about all these agents?

While the range of things that an adult person knows about is vast, there is a more limited class of things that we expect all people to be able to think about. Based on this assumption, we have been enumerating a list of mental realms, those general commonsense domains that all people have at least some expertise in, even as children. In fact, a person who lacks the ability to reason about one of these mental realms would be considered cognitively deficient. Some of the important fundamental mental realms include:

- spatial — reasoning about the ways in which objects and the parts of objects are oriented and situated in relation to one another (are those objects close enough to reach?),
- physical — reasoning about the dynamic behaviour of real objects with masses and interacting surfaces (would this object break if it hit the ground?),
- bodily — reasoning about the capabilities of one's physical body (can I throw this object to them?),
- psychological — reasoning about one's goals and beliefs and those of others (do they know that I am trying to give them this object?),
- social — reasoning about the relationships, shared goals and histories that exist between people (do they want this thing I am handing them?),
- reflective — reasoning about one's own recent deliberations (what was I trying to do a moment ago?),
- dominion — reasoning about the objects and entities under one's control (do I really want to transfer ownership of this object to them?).

This is of course only a subset of the major commonsense realms, and each of these realms contains many more specialised sub-realms for dealing with more specific types of problems. For example, in the 'bodily' realm, the knowledge we may have about manipulating objects with our hands may be separate from the knowledge we may have for using our legs to walk over complex terrains.

These realms are not entirely separate, and in fact are highly connected through the use of panalogies. One example of how multiple realms can be used when reasoning about a typical problem is shown in Fig 4.

Realms are distinguished not only by the knowledge they include, but also by the methods of reasoning they support. Problems within certain realms may use specialised representations and reasoning methods to improve efficiency. For example, in the spatial realm, a specialised planner designed for a three dimensional Euclidean space may be more suitable to solving spatial path-planning problems than a more general technique that can search an *N*-dimensional space. This specialised planner can also make assumptions; for example, it can partially discover the shape of the search space through the vision system.

7. A periodic table of commonsense agent types

How can we best organise these ideas of realms and layers? We find it useful to merge the previous two dimensions of reflective layers and realms into a kind of 'periodic table', as in Fig 5.

Here we have organised our architecture into a matrix in which each cell consists of populations of agents that think about certain mental realms at different levels of reflection.

Beginning with the social realm, a single vertical slice through all the agents (from the robot point of view) yields insight into the organisational utility of this scheme. At the reactive level, there are processes for recognising that someone is smiling at you, for smiling back at them, and so on. At the deliberative level, there may be models of how people react to different

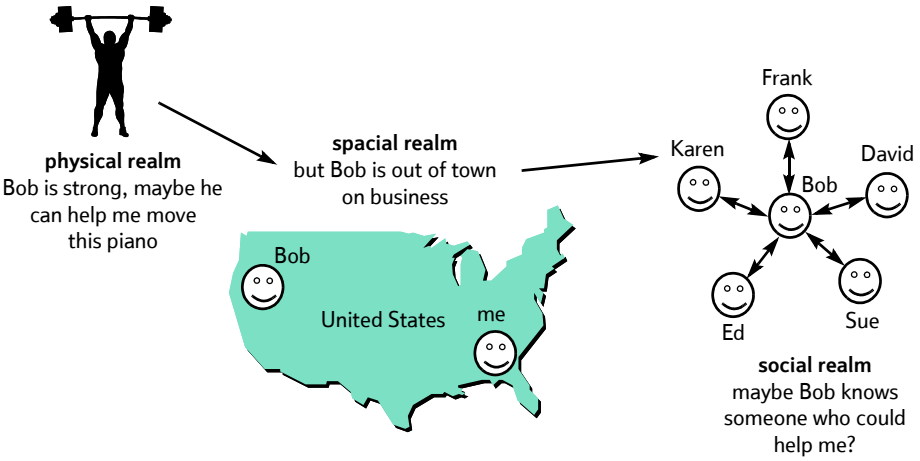


Fig 4 When one way of looking at a problem does not work, we can quickly switch to another.

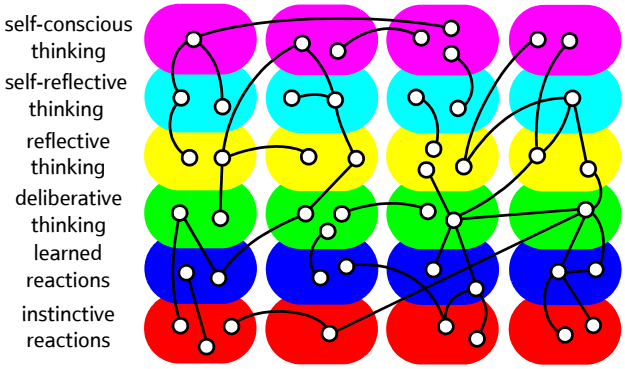


Fig 5 We can divide the architecture into a matrix of cells. Examples include the physical-deliberative cell and the social-reflective cell.

sorts of social actions; for example someone who smiles at you has no malicious intent, or perhaps they recognise you. At the reflective level there may be processes for understanding why we made a mistake about classifying our relationship with someone else. For example, it is not always the case that someone who smiles at you knows you, but you may have jumped to that conclusion by mistake. At the self-reflective level, you may decide that you are no good at remembering people's faces and need to do something about that problem. At the self-conscious level you may think that others think less of you for making that mistake. As a result, you experience mild embarrassment, which is a way of thinking that helps to form a new critic within the social realm.

This raises the general question of how learning occurs within the architecture. We do not provide a treatment here, but observe that the architecture is always undergoing change based on experience. The primary mechanisms of learning in the architecture are the formation of new critics and selectors and the evolution of k-lines into new ways to think. A given agent may learn local rules, such as a probabilistic critic agent that chooses a selector based on the current way-to-think or other source of context. The lower layers have the most plasticity; the learning rate decreases as you go up the layers. This has the important consequence of reducing the consequences of an incremental change on the rest of the system. A bad reflex might be fatal, but is rarely so. Faulty logic in the upper layers may result in a robot driving off a cliff because it is the fastest way to a destination. At the top levels, knowledge and operations must remain very stable to avoid rapid or damaging changes in the robots' goals and behaviours.

8. An evaluation scenario

We are developing a concrete implementation of the architecture at the MIT Media Lab. To enable the exercise of the features described above, we are constructing an 'artificial life' scenario in which two robots, simulating the behaviour of people in a virtual world, work together to build structures using simple objects like sticks and blocks, as shown in Fig 6.

While this domain may seem simple, its sparseness obscures a wide range of issues. In particular, most of the mental realms we have discussed so far show up in some form in this domain. Because the world is physically realistic, the robots must reason about the effects of gravity on objects and the forces that must be applied to move them. Because the robots have synthetic vision systems, they must reason about whether

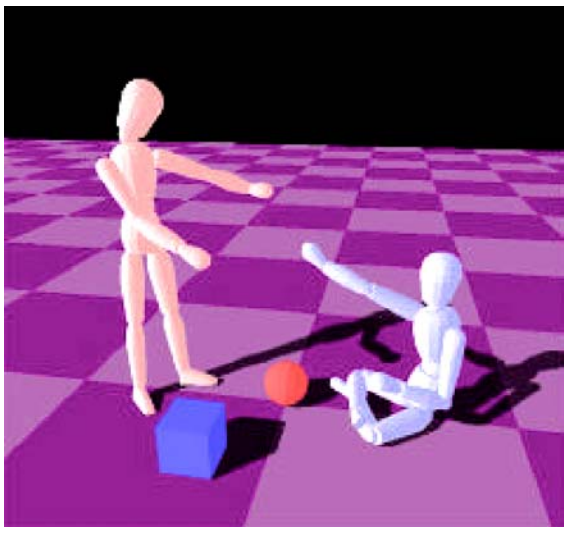


Fig 6 A simulated world.

objects that seem to have disappeared behind bigger ones are in fact really still there. Because there are two robots, they must reason about the social challenges that arise, such as various kinds of conflict, competition, and opportunities for co-operation. To solve problems in this world requires reasoning simultaneously in the physical, social, psychological, and other mental realms.

the architecture is designed to be highly self-aware

For our architectural work to provide a scientifically valuable contribution it must support a methodology to characterise its performance on a set of problems; specifically it must be subject to quantitative analysis, objective comparisons and be reproducible by other researchers. Our simulated world will run on all three major operating systems, have provisions for metering and statistics gathering, and support multiple language bindings. Our aim is to enable a larger body of researchers to engage problems in an experimental domain that is easily reproduced in different researchers' laboratories. In this way, the community will accumulate a collection of scenarios wherein agents can be enabled and disabled to evaluate their impact on a wide range of commonsense tasks. Only when we can catalogue and characterise the questions a representation can answer, along with the performance and cost parameters required to use it, can we begin to have a science of commonsense computing.

Let us examine a simple scenario that hints at the range of commonsense reasoning that happens in even the simplest social exchange. Consider the storyboard in Fig 7, in which two robots named Blue and Pink work together to build a tower.

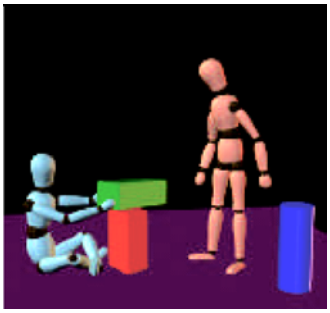
Consider the first two frames of this storyboard. Let's examine Blue's thoughts as it reaches for a block, fails, then realises that Pink may be able to help. Even this seemingly simple problem requires reasoning across multiple realms and multiple levels: reactive, deliberative, and reflective processes over the physical, bodily, spatial, perceptual, and social realms.

Blue wants to build a tower three blocks high. It tries and fails to reach the blue block:

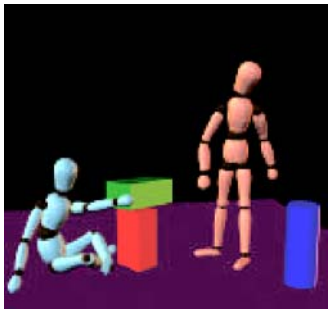
- deliberative physical (goals) — I want to build a tower of blocks three high,
- deliberative physical (situation) — I have built a tower two blocks high,
- self-reflective physical (known methods) — what do I know about assembling structures?
- deliberative ownership (goals) — I need to obtain a new block to build a tower three high,
- reflective priorities (expectation) — if I am building a tower, then I will not be able to achieve my other goals,
- deliberative physical (situation) — there are three blocks in this room,
- deliberative resources (situation) — those three blocks are ingredients for the desired tower,
- reflective visual and spatial (question) — perhaps there is another hidden block,
- reflective procedural (memory) — I have never built a tower exactly three blocks high,
- reflective debugger (method) — abstracting my goal may result in finding a suitable method,
- reflective debugger (method) — replace 'three' by 'several',
- deliberative spatial (expectation) — placing a block on top of a tower will make the tower higher,
- deliberative bodily (situation) — there is a blue block nearby that I can possibly reach,
- deliberative spatial (expectation) — I might not be able to reach that blue block,
- reactive bodily (action) — produce appropriate muscle actuations to produce desired hand trajectory,
- deliberative bodily (situation) — my arm is at full length and I do not have the blue block in hand,
- reflective bodily (critic) — the current method has completely failed,
- self-reflective bodily (method) — find another method.

Blue sees Pink and asks for help:

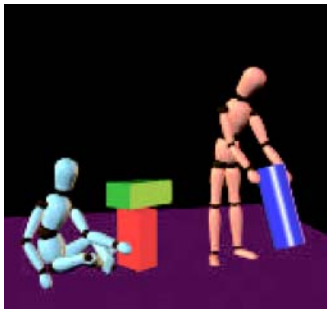
- self-reflective bodily (critic) — no method is available for easily reaching that block from here,



Pink: "I see you are building a tower."



Blue: "Yes, but I cannot reach that blue block."



Pink: "I can reach it. Let me get it for you."

Fig 7 Building a tower together.

- self-reflective social (selector) — switch to social way of thinking for obtaining help,
- deliberative spatial (situation) — Pink is nearby,
- deliberative social (situation) — Pink may be able to help me,
- deliberative spatial (inference) — Pink seems to be near enough to the block to reach it,
- deliberative communication (method) — ask Pink for help,
- deliberative psychological (goal) — Pink may have other things to do,
- deliberative social (situation) — Pink may want that block for itself,
- deliberative linguistic (situation) — Pink says it notices I am building a tower,
- deliberative social (inference) — Pink understands my larger goal,
- deliberative social (inference) — it would cost Pink very little to help me,
- reactive linguistic (action) — say 'I cannot reach that blue block',
- reactive gestural (action) — point at blue block.

So we see that even this simple seemingly trivial exchange exercises many of the cells in our periodic table of commonsense agents. The details here are greatly simplified; presumably every step involves many more agents than are listed, and many iterations of thought must be involved in producing and refining the solutions to the sub-problems encountered by those agents.

9. Conclusions

This paper elaborates on similar discussions in Singh and Minsky [20, 21]. There are many facets of the architecture not covered here that revolve around grounding the implementation — specific mechanisms for learning, the form and role of goals, managing a history of critics and selected ways to think, management of time and memory, etc. More details about our architectural design are available in Minsky's forthcoming book *The Emotion Machine* [15], and in McCarthy et al [22] and Minsky et al [23].

One might question the need for this form of architectural description with its many lists, catalogues, and other accumulations of components. There may even be those who find such approaches inelegant, and would prefer something simpler, perhaps based on some new mathematical principle or universal method of learning or reasoning. We believe any approach seeking to build something with as many features as the human commonsense reasoning system will need to consist of a great accumulation of specific skills, for many of the same reasons that a typical modern computer requires many thousands of small programs and files to operate. Can we realistically expect something comparable to the human mind to be reduced to some simple algorithm or principle given the range of things it must be able do? Even man-made systems, such as a modern workstation with its hardware and operating system, cannot succumb to any kind of universal principle or mathematical analysis.

We hope that our architectural design and virtual world experiments will help alter the way AI researchers picture what an AI system should look like, provide an environment for sharing and characterising techniques and convince researchers to value systems less according to some ethereal notion of elegance and more based on their speed, flexibility, and all-round resourcefulness.

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 McCarthy J: 'Programs with commonsense', in Mechanisation of Thought Processes', Proceedings of the Symposium of National Physics Laboratory, pp 77—84, London, HMSO (1959).

2 Singh P: 'The public acquisition of commonsense knowledge', in Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access, Palo Alto, CA, AAAI (2002).

3 Lenat D: 'CYC: A large-scale investment in knowledge infrastructure', Communications of the ACM, 38, No 11, pp 33—38 (1995).

4 Singh P, Barry B and Liu H: 'Teaching machines about everyday life', BT Technol J, 22, No 4, pp …… (October 2004).

5 Liu H and Singh P: 'ConceptNet — a practical commonsense reasoning tool-kit', BT Technol J, 22, No 4, pp …… (October 2004).

6 Lieberman H, Faaborg A, Espinosa J and Stocky T: 'Commonsense on the go — giving mobile applications an understanding of everyday life', BT Technol J, 22, No 4, pp …… (October 2004).

7 Carbonell J: 'Derivational analogy: A theory of reconstructive problem solving and expertise acquisition', in Michalski R, Carbonell J and Mitchell T (Eds): 'Machine learning: an artificial intelligence approach', Morgan Kaufman Publishers, San Mateo, CA (1986).

8 Kuipers B: 'Qualitative simulation', Artificial Intelligence, 29, pp 289—338 (1986).

9 Minsky M: 'K-lines, a theory of memory', Cognitive Science, 4, pp 117—133 (1980).

10 Davis R: 'Diagnostic reasoning based on structure and behaviour', Artificial Intelligence, 24, pp 347—410 (1984).

11 Winston P H: 'Learning structural descriptions from examples', PhD thesis, Department of Electrical Engineering, MIT (1970).

12 Sussman G J: 'A computational model of skill acquisition', PhD thesis, Department of Mathematics, MIT (1973).

13 Pearl J: 'Probabilistic reasoning in intelligent systems', San Mateo, CA, Morgan Kaufmann (1988).

14 Amarel A: 'On representations of problems of reasoning about actions', in Michie D (Ed): 'Machine intelligence', 3, No 3, pp 131—171, Elsevier (1968).

15 Minsky M: 'The Emotion Machine', (forthcoming).

16 Minsky M: 'The society of mind', New York, Simon and Schuster (1986).

17 Polya G: 'How to solve it: a new aspect of mathematical method', Princeton, NJ, Princeton University Press (1957).

18 Singh P: 'A preliminary collection of reflective critics for layered agent architectures', Proceedings of the Safe Agents Workshop (AAMAS 2003), Melbourne, Australia (2003).

19 Lakoff G and Johnson M: 'Metaphors We Live By', University of Chicago Press (1980).

20 Singh P and Minsky M: 'An architecture for combining ways to think', Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems, Cambridge, MA (2003).

21 Singh P and Minsky M: 'An architecture for cognitive diversity', in Davis D N (Ed): 'Visions of Mind', IDEA Group Inc (2004).

22 McCarthy J, Minsky M, Sloman A, Gong L, Lau T, Morgenstern L, Mueller E, Riecken D, Singh M and Singh P: 'An architecture of diversity for commonsense reasoning', IBM Systems Journal, 41, No 3, pp 530—539 (2002).

23 Minsky M, Singh P and Sloman A: 'The St Thomas commonsense symposium: designing architectures for human-level intelligence', AI Magazine, 25, No 2, pp 113—124 (2004).

Push Singh is a doctoral candidate in MIT's Department of Electrical Engineering and Computer Science, and conducts his doctoral research as a member of Marvin Minsky's Society of Mind group at the Media Lab, where his work is focused on finding ways to give computers human-like commonsense. He created the Open Mind Common Sense Web site to acquire a corpus of commonsense statements from the general public; a semantic network called OMCSNet was mined from this corpus, and has been used for a number of research projects at MIT. He received his BS and MEng in electrical engineering and computer science from MIT.

A philosopher and scientist, Marvin Minsky is universally regarded as one of the world's leading authorities in the field of artificial intelligence, having made fundamental contributions in the sectors of robotics and computer-aided learning technologies. In recent years he has worked chiefly on imparting to machines the human capacity for commonsense reasoning. His book Society of Mind is considered a basic text for exploring intellectual structure and function, and for understanding the diversity of the mechanisms interacting in intelligence and thought. He received his BA and PhD in mathematics from Harvard and Princeton, respectively. In 1951 he built the SNARC, the first neural network simulator. His other inventions include mechanical hands and other robotic devices, the confocal scanning microscope, the 'Muse' synthesiser for musical variations (with E Fredkin), and the first LOGO 'turtle' (with S Papert).

Ian Eslick is a master's student at the MIT Media Lab developing knowledge representations and algorithms for learning in the context of large-scale architectures for commonsense reasoning.

Before joining the Media Lab, he was a director of engineering for Broadcom Corporation, which he joined through the acquisition of Silicon Spice, an embedded communications and semiconductor company he co-founded in 1996.

He earned his BS and MEng in electrical engineering and computer science from MIT.