Design and Implementation of an Agent Architecture combining Emotions and Reasoning

Janos Tapolczai

December 5, 2014

Modern AI is generally divided into two schools of thought, separated by the used level of abstraction. We have, on one hand, the biologically inspired, low-level modelling which uses neural networks to imitate the workings of brains. On the other, we have a logic-oriented, high-level approach that tries design and implement ideal ways of thinking. In recent decades, the latter approach has triumphed and has displaced computational model based on neural networks. In this practically oriented thesis, I shall make the case that this was perhaps too hasty, and that, despite their many applications, purely logic- and planning-based algorithms are insufficient to model authentic intelligence. This work is composed of two parts; the first is an empirical argument for placing models of AI on evolutionary foundations, under the assumption that we can best understand the workings of a complex artefact like the animal brain by tracing its history; the second is a description of an implementation of a kind of affective agent based on these principles, serving as a proof-of-concept that one can create simple, animal-like intelligence with them.

Contents

1 Introduction

The history of AI is marked by vacillations between two paradigms: the biological and the ideal one. These sometimes move closer to each other, and then apart again, due to the fact that they are naturally in tension, yet also inextricably bound together. The biological model, which subsumes the connectionist and the cybernetic ones, seeks to imitate biological organisms on the lowest level; cybernetics, now unpopular, tried to emulate intelligence via feedback loops; connectionism wants to build complex cognition out of interconnected, simple parts. Its most famous instance — neural networks — and the approach in general, have, time and time again, failed to produce even the rudimentary intelligence we seen in animals. The high hopes of the pioneers of the field that the mathematical modelling of some primitive, biologically inspired computing machinery, cleverly pieced together, could mimic the cognition of humans, were sorely disappointed. Then, after years of lacklustre results, Marvin Minsky published his devastating proof of the theoretical weakness of perceptrons — a then popular type of neural network — in his book *Perceptrons: An Introduction to Computational Geometry* [58]. This led to the abandonment of the approach among AI researchers and, arguably, to the AI winter of the 1970s. In hindsight, we can say that, despite their unimpeachable brilliance, their position in history imposed upon them a fatal naivete and an optimism born from the ignorance of the harrowing complexity of an organ that had been, in one form or another, a good few hundred million years in the making. The idea that neural networks mimicking the smallest-scale structures in the brain and consisting of few thousand nodes, trained over the course of minutes or days, could hope to emulate human intelligence, turned out to be false.

Though it had been developed coterminously with the connectionist approach, the ideal approach came to prominence after the former's disappointments. We can identify it largely with *symbolic* computation, whose proponents falls into the two camps (allegedly first identified by Roger Schank): the *neats*, who favour provably correct methods, and the *scruffies*, who are willing to use whatever works [54, pp. 421–424]. Both, to some degree, kept a penchant for mathematical formalisms, but they abandoned the goal of re-tracing the workings of biological brains. The neats especially, in an Aristotelian manner, tried to automate how one *ought to think*. How humans accomplished their tasks was no longer of importance; only what they accomplished was — and preferably, that *what* was abstract reasoning. Out of this school thought came many applications that have proved very useful: automated planning, in the forward and backward variety; default and common-sense reasoning; answer-set and logic programming; Prolog; knowledge-based systems that assist experts in their decision-making, in a way, informed an uninformed search algorithms. Many of these can now outperform the best humans in specialized tasks. Knowledge-based in medicine now rival doctors in the quality of their diagnoses [13, p. 592, Table 31-1]. Automated driving system can navigate through traffic accident-free [28]. In chess, the greatest human players have trouble keeping up with computers, as Gary Kasparov famous loss to Deep Blue showed in 1997 [63]. Yet, impressive as they are — and they are impressive —, these *weak AIs* have not allowed us to piece together the big picture.[1] As good as they are, one would be hard-pressed to call them intelligent in the colloquial sense. As good as any chess program or automated car is, they are still just machines (again in the colloquial sense). It would indeed be absurd to propose that one could exchange one for the

[1] As a side note, it should be said that that the term "weak AI" may be unfairly denigrating. Even if we came into possession of human-level strong AI, so-called weak AIs would still easily outperform it in special areas, just as they outperform humans today.

other. While Gary Kasparov was able to drive home after his match against IBM's Deep Blue, the notion of "car" was not even part of his opponent's conceptual universe. Equally, the control system in Google's self-driving car would not be able to play chess, or assemble a shopping list, read a book, smell, interpret emotions, or do any number of things of which humans and animals are capable.

In recent years, the schism between the biological/connectionist and the ideal/symbolic schools of thought has grown quite pronounced, they are, on a deeper level, joined at the hip. The simple reason for this is that the brain is both a "messy" biological organ, and a high-level computation device. It is neither an amorphous mass of neurons, as connectionism models it, nor is it merely a neutral hardware for idealized reasoning, to which the symbolic approach is wont to reduce it. It is both, and in this thesis, I submit that any strong AI must find find a bridge between the two views and integrate them into one whole.

This thought is not new; such a bridge already exists in the form of the *integrated approaches*. This vague category comprises things like James Albus's Hierarchical Control Systems [2], Rodney Brook's Subsumption Architectures [11], Carnegie-Mellon's 4CAPS [81], and, in theory, the architecture laid out in Minsky's The Emotion Machine [57]. The chief commonality of these systems is that they proceed from an engineering perspective, not from the mathematical one of symbolic computation, or the biological one of connectionism. Yet, they attempt to combine the high-quality results of the mathematical methods with the dynamism of the biologically inspired systems. It is to this category to which I hope to make a small contribution by proposing that one must not only consider the mind not just as messy, but something worse: as *evolved*. I submit that we can best make sense of the seemingly random jumble of features, defects, idiosyncrasies, and quirks of brains by tracing their history. By going through the developments step-by-step, by proposing simple components that came about one by one and grew over time, what is seemingly random and senseless can begin to make sense. By re-creating them, we can hope to create *authentic* intelligences — ones that match biological ones not just in raw processing power, but also in kind.

Structure of thesis. After Section 2 **Related work**, the thesis consists of two large segments. The first one is the theoretical argument and empirical data supporting it. It deals with the origin and purpose of neural systems, their evolution (insofar as is known), the components of which they are likely comprised, and how these could have come about. The constituent sections are

- Section 3 **Preliminary considerations**, containing the general evolutionary story,
- and Section 4 **Schema of cognition**, which hypothesizes about the cognitive structure of humans.
- After that, Section 5 **Mathematical notation**, in which we introduce a bit of notation to ease talking about the systems, and
- Section 6 **Selected subsystems**, in which a number of proposed subsystems are sketched.

The second segment puts the material of the first into practice and consists of Section 7 **Proposed architecture** and Section 8 **Implementation**. Therein, I describe a kind of affective agent architecture that utilizes both emotions and reasoning to navigate a toy world populated by others like it. The last section contains the result of that experiment.

I would like to remark that everything in this work is, at best, a rough outline;: though (conjectured to be) basically correct, and, I hope, useful, we do simply not have the huge amount of data about real brains to construct truly faithful models of them, or to verify even hypothesis in detail.

2 Related work

Cognitive architectures. This thesis falls into the category of cognitive architectures and the integrated approach to AI, pioneered by people like Rodney Brooks and his subsumption architecture, and [11], Douglas Hofstaedter, who famously wrote about many aspects of AI in Gödel, Escher, Bach [38], and who created the Copycat analogy-making program [39]. Another important work is the Hierarchical Control System of James Albus [2], in which cognitive tasks are organized hierarchically and delegated by nodes on higher levels to those on lower ones (this is similar to the mesh-like organisation of components described in Section 5, and to the layered structure of Minsky's *The Emotion Machine* [57]). The organisation described by him in [1], is, moreover, very similar to the one in Section 7, with world simulator, belief generator, sensory perception, and knowledge base (herein called "memory") modules being mostly analogous. Another large and notionally similar system is Carnegie-Mellon's 4CAPS [81], which posits small, relatively simple components, individually doing simple tasks, and having only limited computational resources. Most of 4CAPS's stated principles can be recognized in the coming sections [81, Operating Principles of 4CAPS]:

> 0. Thinking is the product of the concurrent activity of multiple brain areas that collaborate in a large-scale cortical network.
>
> (...)
>
> 1. Each cortical area can perform multiple cognitive functions, and conversely, many cognitive functions can be performed by more than one area.
> 2. Each cortical area has a limited capacity of computational resources, constraining its activity.
> 3. The topology of a large-scale cortical network changes dynamically during cognition, adapting itself to the resource limitations of different cortical areas and to the functional demands of the task at hand.
> 4. The communications infrastructure that supports collaborative processing is also subject to resource constraints, construed here as bandwidth limitations.
>
> (...)

The probably earliest example of a cognitive architecture was Allen Newell's and Herbert A. Simon's *Logic Theorist*, created in 1955 [19, p. 44]. Simon's theory of bounded rationality [32] — the idea of finding a merely satisfactory solution instead of a (provably) optimal one — is very similar to the loop between belief generation and evaluation described in Section 8. In both cases, agents with limited information search heuristically for the first solution that they find acceptable. Unlike exhaustive search methods (e.g. A*), this does not guarantee the best possible results, but it is much more cost-effective and closer to the way real humans solve problems. In spirit, this is also similar to the *Procedural Reasoning System* of Michael Georgeff

et al. [41], which is based on the belief-desire-intention model [67, 9]. Much theoretical work has been done on BDI, but it is only tangentially related to this thesis.

Sloman. Many of the fundamental ideas in this thesis can be found in Alan Sloman's works [74, 75, 76, 77, 78], especially in *Beyond shallow models of emotion*. Therein he formulated the crtierion of evolvability in the context of cognitive architectures and postulated the possibility that nervous systems may be chaotic (but not unorganized). The agent architecture in Section 8 substantially resembles his, though it was not taken from there. The similarity is, however, indicative of a great deal of shared thought.

Implementation. In terms of software engineering, our model has similarities, both to the Actor model, and to publish/subscribe architectures [8] — although more as a concession to practicality and less because of a similarity to their theories. The theoretical basis of our implementation is the postulate that the components of the brain function as white boxes and that other components may listen in on their activity, so to speak. Since this is diametrically opposed to the traditional idea of the procedure/function as a black box, which nigh every programming language follows, we compromise and model the cognitive structure as a mesh of loosely coupled components communicating via passing. This description is reminiscent to the Actor model developed by Carl Hewitt et al. [37], although there are differences[2]: in the Actor model, the topology of the network may change through the creation of new actors, and messages are always passed from one source to known targets (via addresses). In our model, on the other hand, there is no topology in a strict sense; messages are put into a global message storage and every component is free to consume any message it deems relevant. Senders do not know who will read their output, and consumers do not know the sources. This arrangement can be seen as a particularly loose variant of a publish/subscribe architecture, in which the source and the target of a message are completely unaware of each other, and in which there are no specific channels to which one may subscribe. The only criterion by which messages may be accepted or rejected is their content.

We also make use of already existing solutions — specifically answer-set programming and the ACTHEX solver DLVhex. The internal world simulation of our agents makes use of the non-monotonic reasoning provided by ASP and ACTHEX. Answer-set programming was created by Gelfond and Lifschitz [49]. Soon after them, Subrahmanian made the connection between ASP and planning [79]. Together with Eiter and others, he later developed the ACTHEX language which allowed provided a framework for decision making in logic programming via external input and output atoms [24, 23, 25].

Nouvelle AI. Lastly, the overall goal, if not the method, of this thesis echoes that of the *nouvelle AI* of, again, Brooks, who claims that

> the Von Neumann model of computation has lead Artificial Intelligence in particular directions. Intelligence in biological systems is completely different. [12]

The nouvelle AI approach stands in contrast to traditional AI in that it does not aim for human-level performance at specific tasks, but rather for the faithful reproduction of the behaviour of

[2] Although I do not describe the implementation in the language of the Actor model, a translation into it would be quite easy. Such a translation would require using only very rudimentary features of the model, however, and as that is not the focus, I forego the task.

lower animals like dogs [17]. Brooks is very likely morally correct in his desire to abandon the von Neumann model in favour of biologically modelled computation, although I am more willing to compromise for expediency's sake. A proof-of-concept implementation being my goal, and programming languages treating everything but their preferred philosophy as second-class, I deemed it easier to imitate biological computation in spirit rather than in every detail.

3 Preliminary considerations

In this section, we will go over the foundational ideas that, while serviceable on their own, will underlie the work in the second part of this work. The information will primarily concern biology, computational models, and the brain as a product of evolution. Biologists will find all of it terribly basic, but this document is not intended for them; it is intended for computer scientists — who, I feel, have not truly taken to heart the consequences of the routes our nervous systems have taken through history for their present state. Sure enough, we have things *called* "evolutionary algorithms" and "machine learning", but names such as these invite us to a perilous confusion of labels with the real things. Inspired though such mathematical abstractions may be by biological processes, they are not the equivalents of these processes. Re-creating the end-products of biology demands an understanding of biology on its own terms, not through the lens of misguidingly named mathematical abstractions. Providing the basis of such an understanding will be our aim for the next couple of pages.

Historical and designed artefacts. In order to understand how our brain works or could work, we must possess conceptual clarity — we must conceive of it, not as a product of one-time engineering, but as a historical artefact. Unlike "perfect" systems, like Peano arithmetic and the λ calculus, those which grew historically does not make sense if one only looks at their current snapshot. One will find nonsensical solutions, and kludge piled on top of kludge in a futile attempt to correct some early erroneous design decision. The system as a whole will operate according to incomprehensible schemata and be ensouled with a mad logic of which no sane engineer would ever conceive. Of all such systems, the human brain might well be the most complex one; the task of understanding it correspondingly harrowing. Sloman asked whether the the brain might have no architecture at all [75, p. 5]:

> Another question on which there is disagreement is whether the provision of a large set of capabilities, such as those listed above, necessarily involves the creation of an *intelligible* design, with identifiable components performing separate tasks, or whether the functionality could sometimes (or always?) emerge only in a very complex and incomprehensible fashion from myriad interacting components.
>
> For example, experimenters using genetic algorithms to evolve neural sets to control a robot sometimes create networks that work, but which seem to be impossible to understand (not unlike some legacy software which has grown over many years of undisciplined developments).

It the classical sense, it probably does not, but we ought to be cognizant that "the classical sense" was induced by tradition and the limits of human cognitive ability. We might dismissively describe the brain as a jumbled, tangled chaos of neurons, but the fact that we do not recognize a structure by no means implies that one does not exist. Things, contrary to what is oft espoused,

do not "just work"; if they reliably produce complex results, they must have an architecture inside them, independent of our ability to recognize or understand it as such. We merely need to relax the notion of "architecture" to include structures that result from incremental change and the creative combination of pre-existing parts. While the results of such processes are often extremely unintuitive and often even incomprehensible to us, we at least have a way of understanding them by re-tracing their evolution. Doing so is laborious and requires a huge amount of data (which we currently do not have), but this approach of regarding brain function as through-and-through Darwinian (as opposed to having been pieced together) might bear results that have, so far, eluded the other schools of thought in the field.

What, one might now ask, is the consequence of such a view? The first is that each new feature in the developmental history had to have been useful on its own. The second is that it allows the distinction between what I will herein call EFFICIENT systems and CLEAN systems. Since, at each stage of its evolution, the organism that carried the brain had to be viable, the end product is by definition guaranteed to be "efficient". Because of that same fact, however, it is all but guaranteed not to be "clean": for one, it was not possible to snap whole new components into the system; it would have also been impossible to combine old components in the elaborate and precise ways in which a human engineer might use parts. Worse, old components were almost certainly not discarded when new and better ones came into being. A good exposition of this process in humans can be found in Paul MacLean's seminal work *The Triune Brain in Evolution* [52].

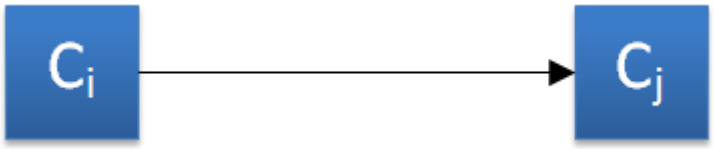


Figure 1: Relationship between the components of an organism without a nervous system.

3.1 Origin of nervous systems

The evolution of nervous systems in organisms dates back to the development of primitive electrical signalling in eukaryotes, using calcium action potentials[3] and sodium channels [48]:

> Voltage-dependent sodium channels are believed to have evolved from calcium channels at the origin of the nervous system.

These sodium channels predated modern-day neurons, but served the same fundamental purpose of acting as control systems. We can readily conceive the benefits of imparting a control system onto an organism with the following thought experiment: let us imagine a microscopic organism without any sort of nervous system — all of its behaviour is hard-coded and mechanical. It can take in nutrients through its cell walls or through an opening; parts of it can contract or expand in response to stimuli like light or pressure; homeostatic conditions can influence its chemistry. Figure 1 shows this schema: if we enumerate the constituent parts or *components* of an organism as $\{C_1, \ldots, C_n\}$, the organism's behavior is caused by signals

[3] See any textbook on evolutionary biology.

Figure 2: Relationship between the components of an organism possessing a nervous system. F can be understood as a simple signal transformer or a central coordinating mechanism.

being sent between C_i and C_j (the case $i = j$ is possible). Such an organism suffers from three disadvantages: (a) reactions are localized, as two of its components might be too far apart to communicate in a timely manner or at all; (b) its repertoire of behaviours is necessarily simple and (c) it is not very adaptable.

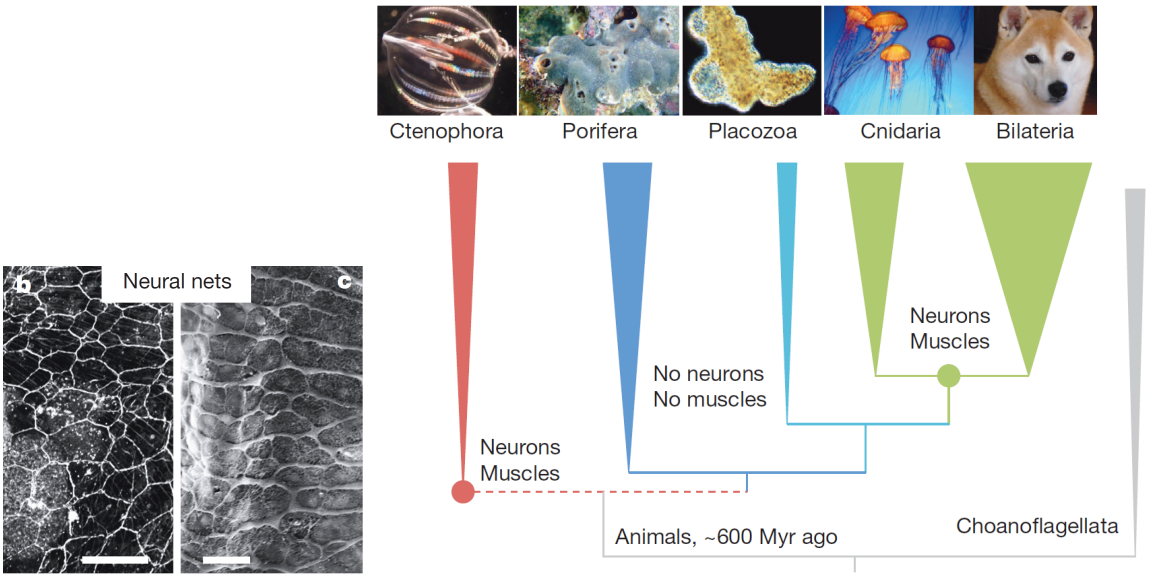
Precursors to nervous systems ameliorated (a) first via action potentials, which were intracellular electrical signals [48] (emphasis mine):

> Another key animal innovation was the nervous system, which is present in all but a few animals (i.e., sponges and placozoans). *Rapid, specific, long-distance communication among excitable cells* is achieved in bilaterian animals and a few jellyfish (cnidarians) through the use of action potentials (APs) in neurons generated by voltage-dependent sodium (Na_v) channels. Voltage dependent calcium (Ca_v) channels evolved in single-celled eukaryotes and were used for intracellular signaling. *It has been hypothesized that Na_v channels were derived from Ca_v channels at the origin of the nervous system* [the results in the paper support the hypothesis] (3), thereby conferring the ability to conduct action potentials without interfering with intracellular calcium. This view was reinforced by the apparent lack of sodium currents in sponges (4).

The introduction of dedicated, long-distance[4] signalling cells between parts of an organism created the possibility of not only transmitting, but also modifying information. The moment an organism's parts do not communicate directly biochemically/mechanically, but over transmissions lines, evolutionary processes acting upon these lines are able to mutate them so that they change the signals. The first changes might consist of amplifying, diminishing, or distributing signals. Over time, the nerves may come to act as transducers on the stream of signals; in some rudimentary sense, they may begin to compute functions. Schematically, we see this in Figure 2, where a function F is interposed between two components. Not all components of an organism are created equal, of course. The first and most important use of nerve cells was the communication between sensory organs and the movement apparatus of the organism, and the bulk of nerve cells were located close to the sensory organs, where they processed information. A mere handful of neurons are not able to compute much, but they must have conferred considerable advantage to their owners.

The history of these developments is not entirely clear, but action potentials are present in all animals (with the exception of sponges) and in plants [47, 30]. A step up from mere stream transducers are the nerve nets that permeate the entire bodies of cnedaria (jellyfish) and the

[4]The term "long-distance" may very well mean "long-distance within a single cell". František and Mancuso argue in [4] that neural analogues already existed in prokaryotes (bacteria and archaea; organisms without cell walls and nuclei) and unicellular eukaryotes.



(a) Nerve nets in ctenaphora. From [59, p. 110].

(b) Phyolgeny of the animalia. Even Ctenophores, macroscopic marine invertebrates which predate both jellyfish and bilateria, have nervous systems in the form of distributed nerve nets. From [59, p. 110].

nerve cords that run along the bodies of bilateria (animals with left and right sides). In Figures 4 and 3b we see them in the phylogeny of the kingdom animalia. Both can process signals in a sophisticated way, and enable the performing of varieties of complex tasks, although the sets vary widely from species to species.

Central nerve cord and cephalisation Nerve nets, while interesting, are not our aim. Unlike jellyfish, bilateria have a central nerve cord which runs from their front to their back. At various points alongside the cord, we find ganglions — thickenings containing larger amounts of nerve bundles. In all animals but worms, the frontal ganglion further thickened until it came to contain the overwhelming majority of the organism's neurons — forming the head. While substantial neural activity was occurring before this time, it is only here that it becomes to proper to speak of brains, and where we can begin to analyse macroscopic structures like lobes.

Very briefly: vertebrate brains are subdivided into hinbrain, midbrain, and forebrain, having evolved in this order. The forebrain (cerebrum) is responsible for all higher functons and is again divided into six lobes, which we see in Figure 6. The functions performed by these lobes are not precisely understood, but a number can be clearly associated to one lobe. The frontal lobe, for instance, is responsible of conscious thought; the temporal lobe processes auditory and olfactory signals; the occipital lobe deals with sights.[5] While some functions, like memory, are not neatly localisable, we can nonetheless see in the anatomy of vertebrate and mammalian brains the accruing of large groups of functions: motor control, smell, hearing, sight, reason, emotions. The question of organisation remains, however: it is one thing to say that we have hearing and smell, but what, if anything, ties these experiences together? We, after all, perceive the data from all our senses as one integrated experience. Here, views diverge. The common-

[5] Interestingly, the occipital lobe is at the *back* of the head.

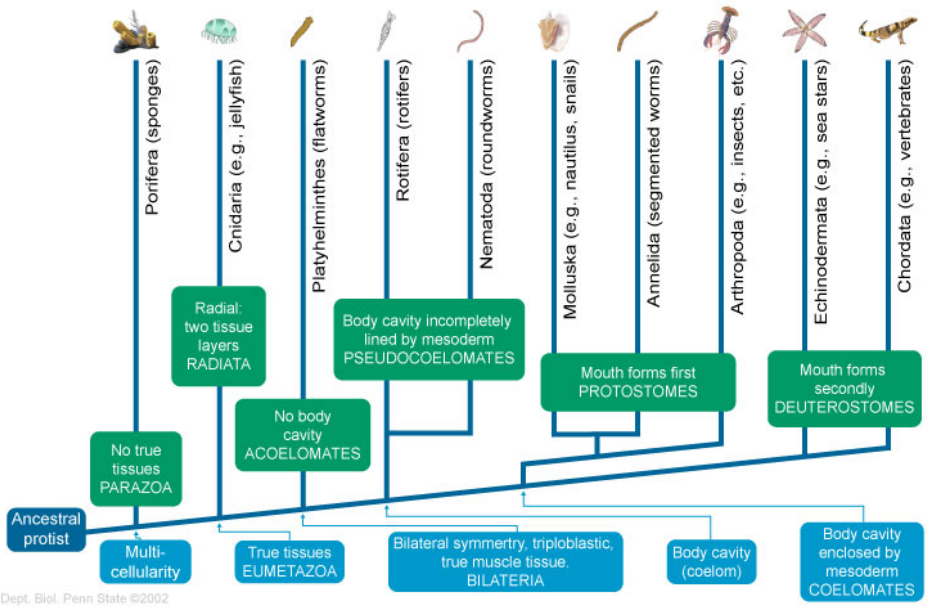


Figure 4: Phyolgeny of the animalia. Note the cnidaria and bilateria; both of these have types of nervous systems. From [82].

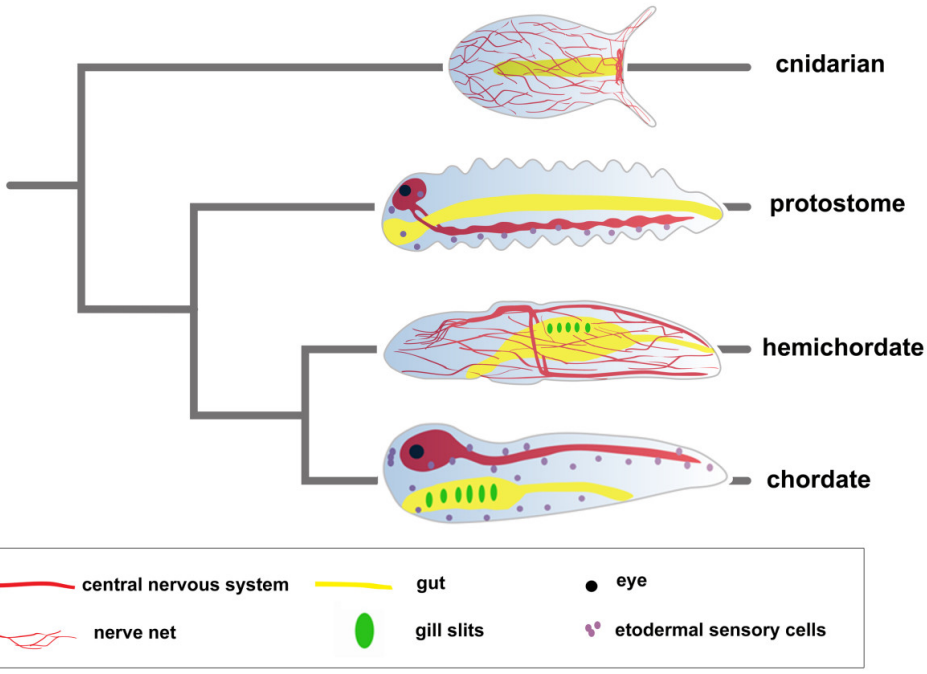


Figure 5: Body plans for metazoans. The bottom three items are all bilateria and all have nerve cords of some kinds, but only the bottommost (chordates) have a dorsal (upper) nerve cord. Vertebrates are a subphylum of the chordata. From [40, p. 3].

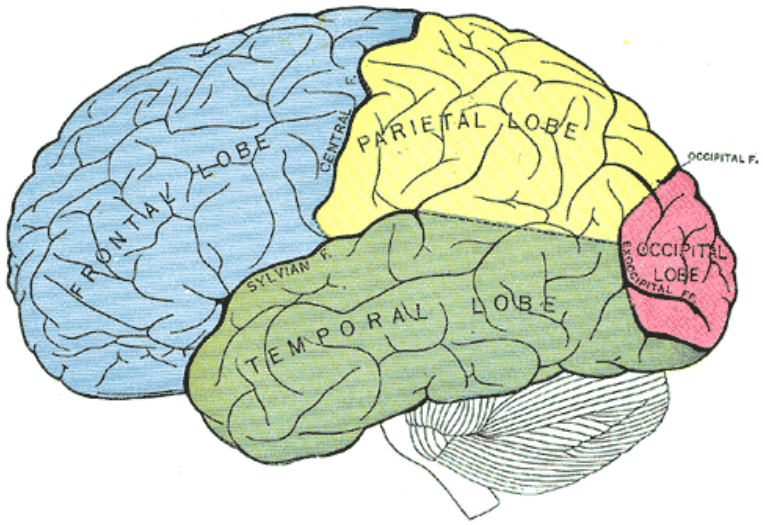


Figure 6: Illustration of the cerebrum with lobes shown. Hidden: limbic lobe, insular cortex. From [33, Fig. 728].

sense belief is that we simply have one, indivisible consciousness. Such a view would implicate the frontal lobe as an central organising unit, without which an organism, even if it could smell or see, would not consciously do so. Minsky, Sloman, and Dennett argue persuasively, but speculatively, against this view in [57, 56, 73, 22]. They differ on the details, but all agree that the unified consciousness is an illusion; that it is not a single "I am"-thing gathering raw data, but a dispersed locus of experiences that we merely perceive as immediate.[6] In this view, the frontal lobe, while still instrumental, would not be the only contributor. All other regions of at least the cerebrum would contribute in some way to the organism's conscious experience. An animal without a frontal lobe would not be conscious in the same way as we are, but it would not be utterly blind either — it would already have some dim awareness of its existence; some rudimentary "I am" that we can hardly imagine would already be present in it. To quote Sloman (emphasis mine):[7]

> It is not worth asking how to define consciousness, how to explain it, how it evolved, what its function is, etc., *because there's no one thing for which all the answers would be the same. Instead, we have many sub-capabilities, for which the answers are different*: e.g., different kinds of perception, learning, knowledge, attention control, self-monitoring, self-control, etc.

[6] Dennett criticises the idea of a CONSCIOUSNESS-THING with the concept of the "Cartesian theater" [22]. According to him, positing that there is such a thing in our brains, and that it observes all other brain functions, is fundamentally problematic: if there is such a sort of homunculus in our heads that, say, sees the result the output of visual processing in the manner in which one would see a film, then how its visual perception work? Is there yet another a homunculus inside the homunculus that interprets visual information? Such a view implies either an infinite regress, or the algorithmic inexplicability of some part of the brain.

[7] The quotation appears in [57, p. 97] and Minsky attributes it to a post made by Aaron Sloman in the `comp.ai.philosophy` newsgroup, but I have been unable to find the original.

Implications The point in all this is not to give an detailed summary of evolutionary neurobiology; it is to show that nervous systems are ancient, gradually developed things. They have been shaped by the vicissitudes of hundreds of millions of years, and they could have developed in other ways. They were not planned, as a human would understand to word. If we are to gain headway in piecing together the "big picture", we must take these facts to heart, and choose our modelling methods accordingly.

In the abstract of this work, I described the biological and the idealistic approaches as being polar opposites, and this is true as far engineering is concerned, but in terms of their assumptions, false. They are both idealistic. Neural networks, insofar as their users want to re-create human behaviour, implicitly presuppose an intelligence in neurons that is not there. The comparatively small network is taught to compute some desired function, the hope being that it might thereby come to perform some complex, real-world function like common-sense reasoning. In principle, this strategy could work, but in practice, it is unrealistic — the environment in which real organisms had to succeed was the planet's ecosphere; billions upon billions of nervous systems of all complexities were run over millions of years; nervous systems died off and were re-created from scratch by genes. It is therefore entirely unreasonable to assume that neural network, trained against an objective function over a period of hours or days could re-create the function a biological organism, unless one were to suppose that there is some inherent quality in neurons that strives for such; that groups of cells somehow *wish* to organize themselves into specific configurations in which they are able to perform activities we would call "cognition".

All this being said, we should not confuse criticism of the suitability of a method for a specific purpose with criticism of its suitability for any purpose. Neural networks have proven useful in understanding mental activity at small scales; both they and the symbolic/logic-based approaches have had a myriad of industrial applications. From this, however, it does not follow that we can build genuinely intelligent agents with them. Our only means of doing that (the only means that remain) is to laboriously unravel the developmental history of animal brains, step by step, making sense of each development in context. Where empirical data are not available, we at least have to hypothesize how things could plausibly have happened. To day, structural and genetic analyses have been done (via genetic sequencing and MRI), but they do not deliver sufficiently detailed data. Such methods are rather akin to measuring voltages and task time in a PC — they do tell us something, but an observer would never infer the existence of e.g. compilers, call stacks, or type systems from such observations. For an understanding of the brain so specific that we can re-implement it in a computer, we will need currently non-existent and not-conceived-of technology. Until that day — and this will be the main thrust of this thesis — guesswork will have to suffice.

3.2 Ways of adaptation

After the philosophical groundwork and biological basics, let us conjecture about the ways in which nervous systems can change and acquire new features. We begin with the observation that the existence of neuron bundles between parts of an organism is analogous to a loose coupling of components in a software systems. By having intermediaries that take over the task of communication, selection pressure can produce more and more complex functions, since it no longer has to act upon the body parts that send various signals, but change the nervous system that processes these signals instead. As example: pain receptors, muscles fibres, and the optical nerve have been unchanged for quite some time, long pre-dating the human species,

but more recent brain developments have given us the ability to utilise them in novel ways — by providing a rich mental experience of suffering, playing instruments, and mentally rotating objects, respectively. Manipulating the software is far easier and more quickly done than doing so with the hardware, so to speak.

Having said that, the changes still have to have occurred incrementally. Even if a nervous system can change quickly (for evolutionary timescales), it still has change in tiny steps. We shall leave the matter that for the time being, but, as we will discuss later, this simply fact profound computational consequences that are seldom thematised in discourse on this matter.

Let us return to the consideration of primitive life forms. We can imagine the malleable neuron bundles of such ancient organisms changing in a variety of ways in the face of selection pressure: when the environment required it, they could, after several generations, start to compute different or more elaborate functions. An organism which had had developed in an environment where food was abundant in bright places and which had now found itself in darkness would have benefited from a variety of plausible changes, such as

- an inversion of its light-seeking behaviour,
- switching off its metabolism in light places to conserve energy,
- accelerating its metabolism in dark places to make better use of the food there.

Of course, other changes would have also been possible, such as the metabolization of different food sources,[8] but we can see how the aforementioned three could have been effected through mutations in a simple nervous system. For a system to permit such mutations, it must be far more robust than most products of human engineering, however. If one were to take out a piston in a car or replace a cogwheel in a mechanical clock with a differently sized one, the machine would, in most cases, simply break. In all others, it would catastrophically malfunction. Machines are designed to fit together perfectly and their complexity tends to be irreducible. Even software, which is more readily changed, is easily broken by small-scale tinkering.

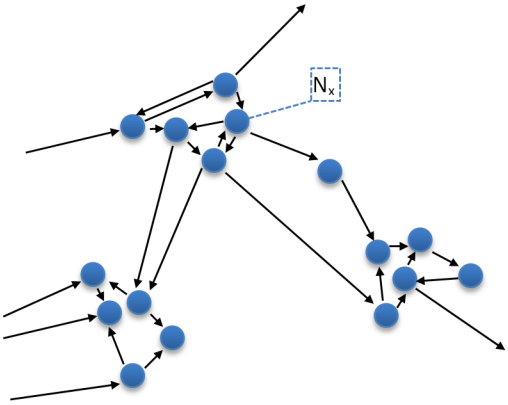
When discussing how they can evolve and, in particular, *evolve to perform new tasks* and not just variations on old ones, explanations are again constrained by two criteria: (a) the change has to be small, or at least have a small cause[9] and (b) each change must be beneficial in the short term.[10] Something that we would conventionally recognize as a program, something which has precise notions like "instruction" and "call structure" is probably not suited to this pattern of changes.[11] Instead, we ought to imagine the brain as a mesh of computation in which functions are computed cumulatively, so that small changes in neural structure only lead to small changes in output.

[8] A current-day example is given by nylon-eating bacteria, which have developed in the last century and which now have an abundant food source and no competition.

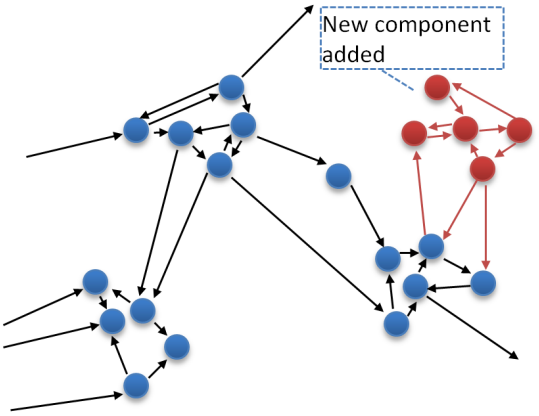
[9] The effect does not have to be small — changes in single genes can switch entire components on or off. The MYH16 gene, which is present in non-human primates but has been switched off in humans, is an example. In us, its disabling lead to a drastic reduction in the size of jaw muscles and a corresponding increase in brain size [15]. Nonetheless, such events are rare and not the main drivers of evolution.

[10] Caveats apply: if the selection pressure on a group of organisms isn't too strong, changes which may be sub-optimal but perhaps beneficial at some later point may spread, and non-selective processes like genetic drift can also play a role.

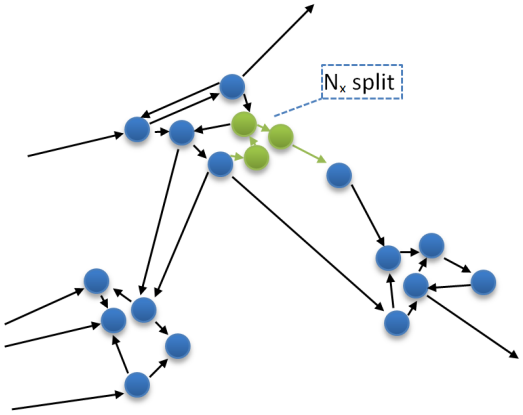
[11] Cf. evolutionary program generation, in which expression trees mutated. I charge that such algorithms are not adequate models of what happened in the evolution of our brains.



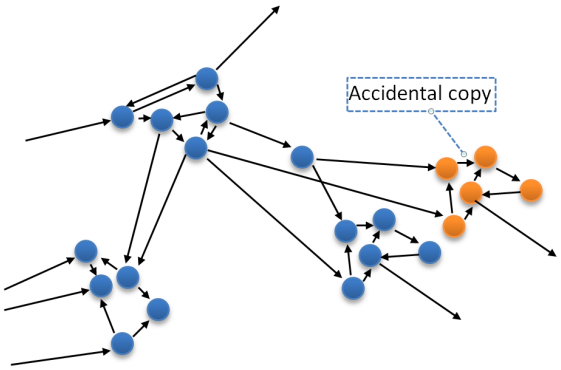
(a) A simple neural network.



(b) An unlikely change scenario in which new, discernible components are grafted on from whole cloth.



(c) A more likely change scenario in which one part is split into three but where the overall shape of the network is not appreciably altered.



(d) A second change scenario in which an entire component is accidentally copied. While the resultant change is large, a small genetic mutation can cause it.

To illustrate this, we can look at a simple neural network in Figure 7a, with a marked node N_x. Figure 7b shows an unlikely change scenario in which some new component/function is cleanly grafted onto the system. Figures 7c and 7d then show a two more likely scenarios: in the first a mutation causes N_x to be split and the new nodes take over some of its connections. In the second, a larger component is accidentally copied as-is and, over time, is moulded to do something useful.[12] In time, new functions can thus grow into the system, but never in the manner in which, say, an engineer would implement a new feature.

Sloman's brain. One might ask what the relationship between the gradual growth of neural bundles and the observed, large-scale functions in the brain is. We have now supposed at some length that the organisation isn't neat, but the question remains whether we can speak of an organisation at all (even a messy one). In [77, p. 8], Sloman illustrates the possible chaotic organisation of brain with Figure 8, conjecturing that it might be a jumble of parts that just

[12]Such copies can be caused by mutations and are known to happen with some frequency in nature.

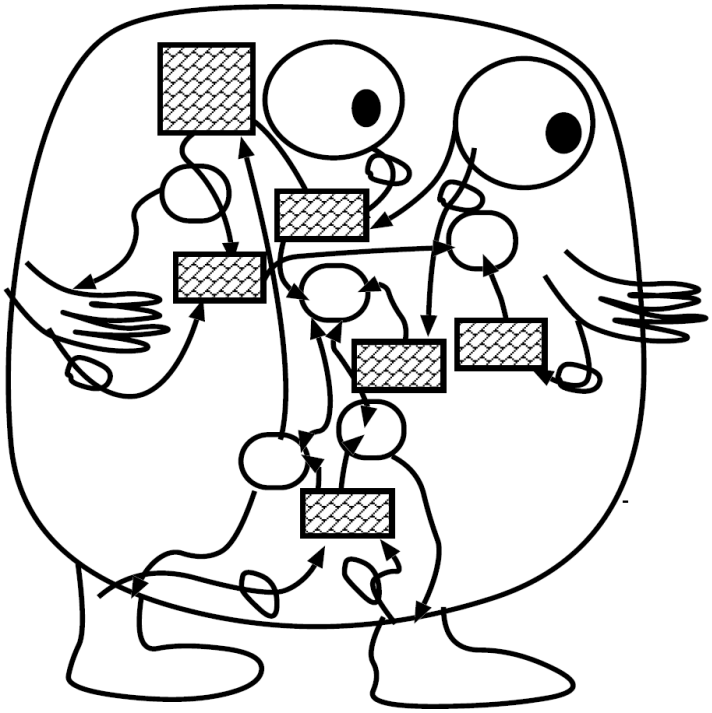


Figure 8: Sloman's illustration of the brain as an "unstructured mess".

happen to work together:

> Any observed behaviour might be produced by an unintelligibly tangled and non-modular architecture. (Rectangles represent information stores and buffers, ovals represent processing units, and arrows represent flow of information, including control signals.)

This strikes me as a cheery optimism, if anything; one that presupposes that there even are such things as information stores and control signals. The actual situation is, in all likelihood, a far worse one: it is not just different programs that are run in the brain, but entire different models of computation, with the same pattern of activity being interpreted simultaneously in more than one way. Today, we can scarcely imagine how such an "architecture" would work, let along how one would program — but if we really want to create genuinely animal intelligences, we will have to find out. Sloman himself admits to the difficulty of gaining understanding of the workings in [75, p. 6, Section 9 "Is the task too hard?"]:

> Given the enormous diversity in both design space and niche space and our limited understanding of both, one reaction is extreme pessimism regarding our ability to gain significant insights.

The following remedy is offered:

> My own attitude is cautious optimism: let us approach the study from many different directions and with many different methodologies and see what we can learn. (. . .)

> In particular, the Cognition and Affect group at Birmingham has been trying to use a combination of philosophical analysis, critical reflection on shared common sense knowledge about human capabilities, analysis of strength and especially weaknesses in current AI systems, and where appropriate hints from biology, psychology, psychiatry and brain science, to guide a combination of speculation and exploratory implementation (...).

The methods listed all have their applications, but computational analysis is missing among them. When we talk of psychology, philosophy, and critical reflection, we have already supposed too much; we want to replicate the high-level output of the brain without having explored the mechanism that produces it. In a manner speaking, we have seen the forest, but do not understand what trees are. If we are to gain the sort of knowledge of brains that we can implement into an AI, we must, empirically, find out about their method of computation. Barring that, we must at least approximate it as far as is practical, and accept that the result will necessarily be an inferior simulacrum.

What, then, is the computational model used in the brain? As yet, nobody knows, and that will stay that way for the foreseeable future. It is very much a guess, but from the concept of slowly growing neural networks (seen in Figures 7a-7d), one might infer something like the "active symbol" hypothesis in Douglas Hofstadter's *Gödel Escher Bach*: that patterns of activity form little programs and pieces of data at once; that manipulate other patterns of activation and are manipulated yet other patterns during their lifetime. These are only imperfect analogies, of course. On the coming pages, I shall outline a conceptually compatible white-box model of computation as another, imperfect analogy that will serve as the basis for the model in Sections 4-5, and for the implementation of the toy agents in Section-8.

3.3 The brain as a collection of white boxes

We now leave the realm of established fact and venture into conjecture. What has been said up to this point has been good, general fact, but it does not suffice for building actual programs. Data on the computational structure of the brain is scarce, thus I will limit myself to positing general, plausible hypotheses about what sorts of structures and loci of computation could have plausibly arisen in it over the course of its evolution.

A plausible case has been made my Minsky, Sloman, and others (especially in *The Emotion Machine* [57]) that the brain must possess components in some form. Were it not so, the organ would have long ago succumbed to the innefficiences of its design. As more and more functions are grafted onto a system, the number of interactions between its parts or regions, and therefore the bugs in it, increases. Worse yet, the system becomes brittle: even if, like in a neural network, some accidentally working configuration would have been able to be reached, small changes would surely have upset it again. The part-less system is an evolutionary dead-end from which no improvement is possible, and given how far along our cognition is, it is quite clear that we are not dealing such when we look at our brains.

If we concede that we are dealing with identifiable parts, a second question arises: how do these parts communicate? I would like to deal with with this question in some detail. In the literature, this issue is often glossed over — in diagrams, one frequently finds unannotated arrows going between functions; the accompanying texts mention concepts like "selection", "message", and "sending" under the implicit assumption that these are merely primitives in no need of further explanation. When we consider the workings of neurons, however, it is not at

all clear how groups of them could put together any sort of complex message, and, once put together, how it would travel, and how another group of neurons could receive and interpret it. Are there dedicated interpreters, akin to compilers and runtimes in computer systems? This is not known. I cautiously propose that it is not so, but we can present plausibly-sounding scenarios for both outcomes:

- On the one hand, we may image that, early one, some simple message format developed through, allowing more efficient communication between not quite differentiated regions of a nervous system. Over time, this was extended as more components came into play; these new components would have found it easier to make use of the pre-existing protocol. Larger clusters of parts might have even repeated the process and developed simple, internal message formats for communication among themselves. As an orthogonal development, newly developed components might have performed more abstract duties, using older ones as subsidiaries, if at all. To solve conflicts whenever these new and old parts proposed different solutions to whatever issue the organism faced, some other component could have received inputs from both, and adjudicated. In such way, a hierarchical and layered structure could have come into being — different layers working at different levels of abstraction, and each component only communicating on an on-basis with others. All in all, the whole system would come to resemble a human-developed program.

- On the other hand, we could imagine quite a different scenario: suppose that the basic scheme of neurons sitting as growths on the communication lines between components never changed. Their basic task was the modulation of signals, and if some new function was to be grafted into the system, then this would have been achieved by growing more neurons that modulated the signals of their fellow neurons. They would not have opened a communication channel with the existing components, but would have listened in on their activity in the manner of interlopers surreptitiously modifying messages. Since neurons would have had no reason to hide their activity (as a black box does), this would have been quite easy and straightforward to do. New bundles of functionality could have inserted themselves into the middle of the information flow (enhancing existing functionality, or adding administrative features), before it (providing pre-processing), or after it (providing post-processing, or usage of the output for higher-level tasks). In contrast to above, we concentrate on the *process* of software development instead of its product: a human programmer adds functions one at a time, here and there, extending and refining functionality where fancy strikes or necessity requires.

One could thus call the first scenario the PRODUCT-ORIENTED VIEW and the second the PROCESS-ORIENTED VIEW: the first looks at the end product of a development, the second posits that the very process of that development, fossilized, is present in the end product.

Evidence is scarce and equivocal for both. In fact, it need not even be the case that they form a dichotomy: we might just as well speculate, for instance, that the second is the low-level reality, but that the first emerged from over time due to the efficiency of its design. The components could be fuzzy, to some degree. We could also posit that the first one "degenerated" into the second one; that a formal system is emulating an informal one because of the latter's greater versatility and dynamism.

For the rest of the thesis, I will explore the second of the above two hypotheses, not necessarily because I firmly believe it to be true, but rather because the first one has been tried for some time, and has so far not produced a general AI.

Practical abstraction While such a white-box model, and the hypothesizing that preceded it, are conceptually useful, a mesh of gradually grown patterns does not lend itself to implementation in a program. We do not have the capability of faithfully pouring the structure of the human brain into a computerized mould just yet, but, for the time being, we may opt for the next best thing and take cues from it in the hopes of improving our imitations.

Therefore, I will present a simplified model which, while attempting to remain true to the conceptual view, will, pragmatically, contain discrete functions and components. The white-box nature of brain activity will be emulated by a message-passing scheme in which messages model the internal activity of components. Instead of each component blindly acting in some fashion on the activity of another, components will have explicit parsers and interpreters and later, these will be further simplified into localized message formats and tagging, for the sake of easy implementation. This effort is guided by the same thought as Sloman's cognitive architecture see in Figure 9. It is not a truly accurate representation of the brain and it does not claim to be , but it is *something like it*; something that is close, and good, enough. We will meet this cognitive architecture again later, but for now, we move on to the description of neural components.

4 White-box model of cognition

We can imagine the components of the mind as white boxes which inform other components by their very functioning — however, this does not lend itself to easy implementation. Instead, we can emulate this behaviour via a MESSAGE SPACE, from which individual components take their input and into which they put their output. A COMPONENT is then a local processing unit which continuously scans the message space, running messages through its FILTER. If the filter detects a relevant message, it is then passed to the INTERPRETER, which parses the message into the needed format and hands it over to the PROCESSOR. The processor, after having finished, puts its output back into the message space for other components to read. Figure 11 illustrates this scheme. Note the lack of explicit hierarchical structure and central organising units.

However, as I'll show in the next section, this model is generic enough to accommodate such special-purpose structures. Figure 11 shows the message-passing scheme, but it also specifies a graph in which the nodes are the components and fixed, while the edges are the accepted messages and are determined by the nodes; through their filters, components control the shape of the graph. By imposing invariants on these filters, we can have the graph take any shape we desire. In particular, we can model the kinds of structures that occur in many other cognitive models and in empirical research: central organisers, sequences of components ("pipelines"), localized messages affecting only a small part of the mind, a component reading its own messages, loops and iterative messages between two or more components et cetera.

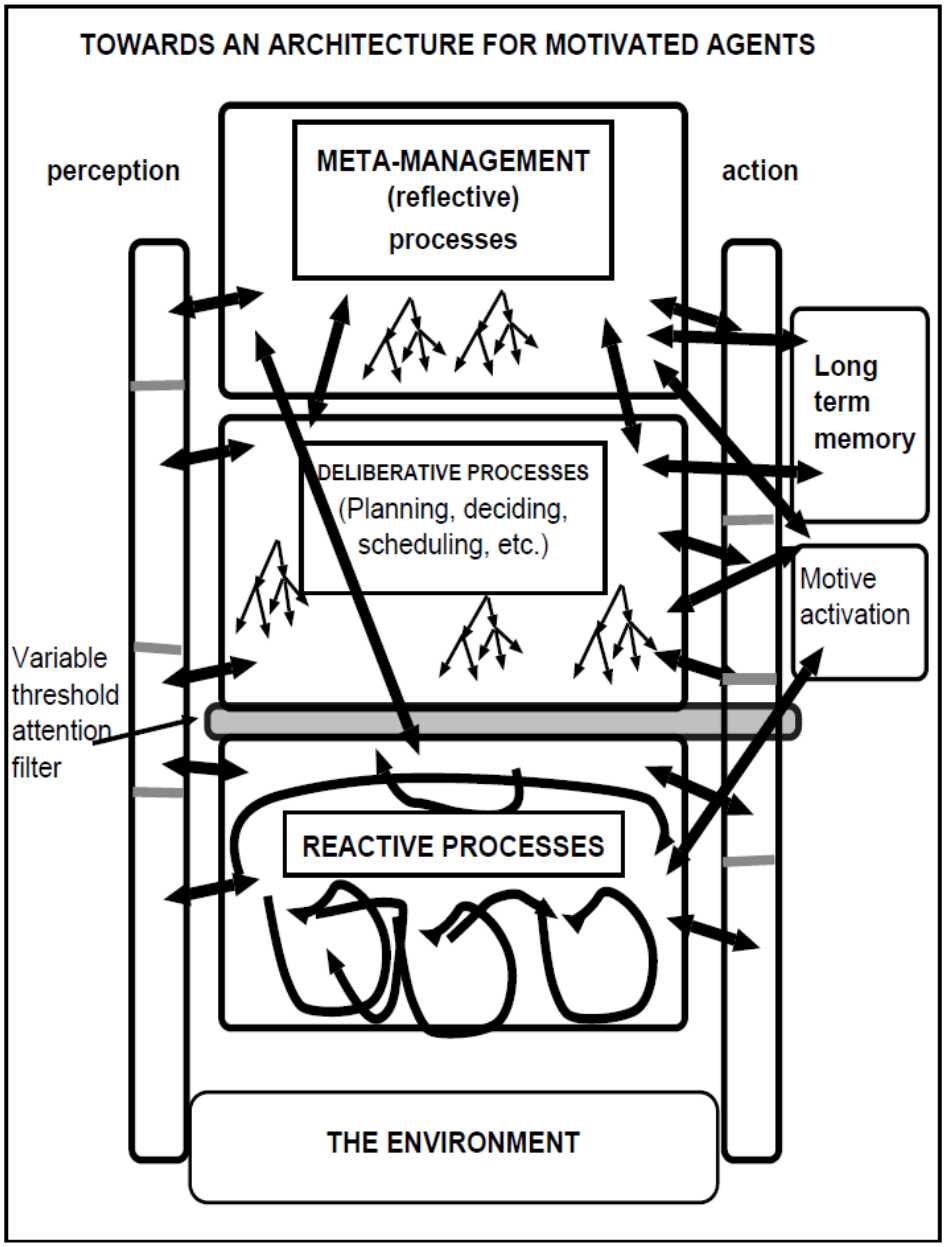


Figure 9: Architecture for motivated agents. From [75, p. 10]

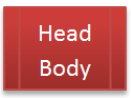
Symbol	Description
	Processing component
	Choice
	Data container (Queue, List, etc.)
	Data
	Stream generator
	Counterfactual (imaginary) data

Figure 10: Notation for the diagrams in this and the following sections.

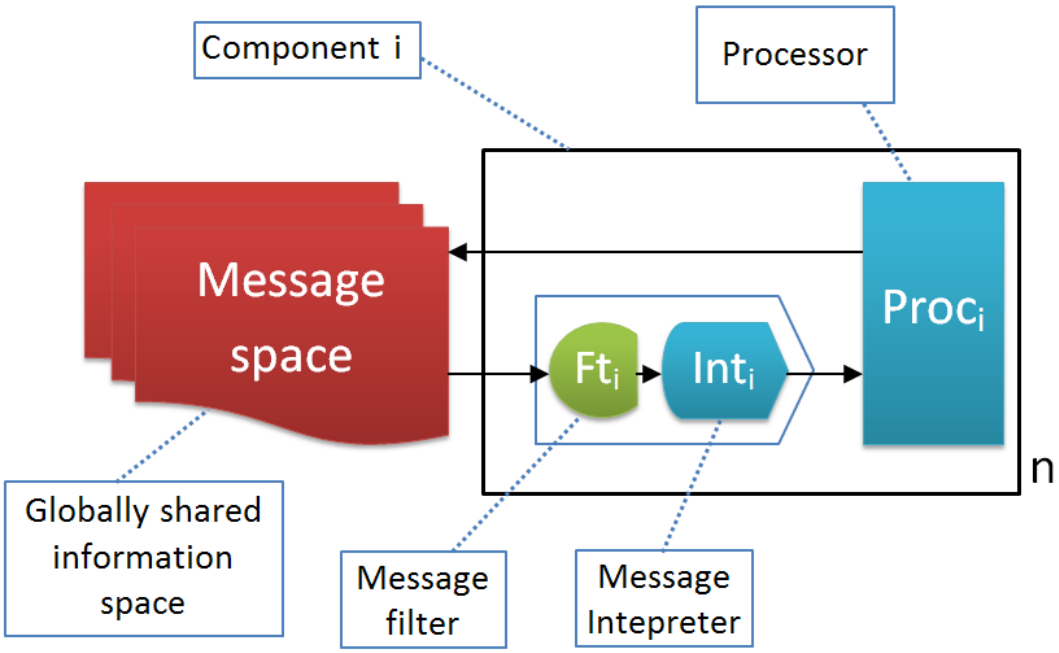


Figure 11: Global neural architecture.

Messages We may now ask how such messages between components are structured. Here, I make two empirical claims:

1. messages have a priority and
2. they are effectively unstructured.

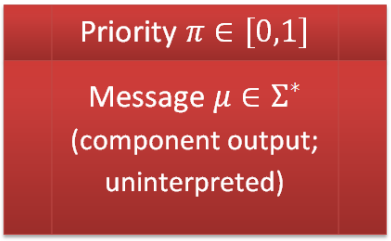


Figure 12: Structure of a neural message.

To the best of my knowledge, the veracity of either has thus far not been determined by neuroscience. For the first, Marvin Minsky's "The Emotion Machine" provides some circumstantial evidence [57, p. 222]:

> Of course, when one activates two or more Critics or Selectors, this is likely to cause some conflicts, because two different resources might try to turn on a third resource both *on* and *off*. To deal with this, we could design the system to use various policies like these:
>
> 1. Choose the resource with the highest priority.

2. Choose the one that is most strongly aroused.
3. Choose the one that gives the most specific advice.
4. Have them all compete in some "marketplace".

The selection strategies Minsky lists imply that there is some mechanism in the brain to determine the urgency of a signal. While it is possible that higher brain functions like reasoning or affect make an additional, rational evaluation, sensations like intense pain, bright lights, or great sadness can likely be communicated most easily by the appropriate components causing a flood of activity which, by its very intensity, informs other components of the urgency of their messages.

The second claim — that messages are essentially unstructured — means that there is no common, agreed-upon format in which they are stored. In addition to the evolutionary implausibility of such a format being created, an unstructured message format is in line with the white-box nature of components: since components merely "listen in" on others, and since each components will have its own pattern of activity, a listener would simply have to try and make sense of this activity as best it could. The proposed structure of messages is thus shown in Figure 12: every message comprises a priority header, together with an unstructured body which, for our purposes, is simply a string of bits.

Filters Before a component can respond to a message by another, such a message must be assessed for the presence of relevant information. Conceptually, this happens via a FILTER in each component, which pattern-matches incoming messages and, if a certain threshold is reached, signals relevance and hands the message over the INTERPRETER for parsing. Figure 13 shows such a filter: it is composed of a directed graph of nodes, and a node is activated if it detects some specific content in the message. Nodes, in turn, are connected via edges of strength $\in [0, 1]$. When a node is activated, it sends a charge proportional to the strength of its link to its neighbours, contributing to their activation as well. Some nodes are marked as *output nodes*; if enough such output nodes become activated, the message is deemed to be sufficiently relevant. This model of filters is inspired by the *spiking neural P Systems* of Georghe Paŭn et al. ([66, p. 337] and [42]), in which charges sent along directed graphs of neurons are used to compute functions.

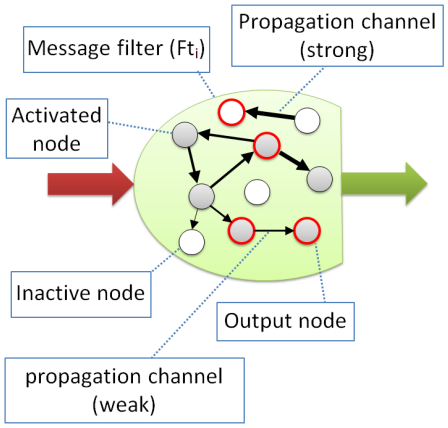


Figure 13: A pattern-matching filter for a component C_i.

5 Mathematical model

We now create a mathematical model for the description of the architecture. This model will be split into two parts: the structural and the operational semantics. The structural semantics encode the static properties of neural systems, whereas the operational semantics describe the behaviour of such a system at runtime.

5.1 Preliminaries

Since the mathematical model is built with implementation in mind, I will use some basic type theory in the coming sections. The following notions are from the λ-calculus and its attendant type systems; anyone familiar with such can therefore freely skip this section. We will introduce types, type constructors, and their relation to functions, together with a few example types which will come in handy later on. The following can be found in any introduction to type theory and was taken (with simplification) from [55], [18], and [43].

Definition 1 (Syntax: Type). *For our purposes, types are defined inductively thus:*

Basic type. $\mathbb{R}$ *and* $\emptyset$ *are types.*

Sum type. *If* $\mathtt{T}_1, \mathtt{T}_2$ *are types, the* sum type $\mathtt{T}_1 + \mathtt{T}_2$*, is a type.*

Product type. *If* $\mathtt{s}$ *is a string and* $\mathtt{T}_1, \ldots, \mathtt{T}_n$ *are types, the* product type $\mathtt{s}\ \mathtt{T}_1 \ldots \mathtt{T}_n$*, is a type. A special case is the* anonymous product type *(tuple), where* $s =$ "$\langle\rangle$"*. There, we just write* $\langle \mathtt{T}_1, \ldots, T_n \rangle$*.*

Full application. *If* $\mathtt{T}_1, \ldots, \mathtt{T}_n$ *are types and* $[\forall x_1, \ldots, x_n]\,\mathtt{C}$ *is a type constructor (see next definition), then* $\mathtt{C}\ \mathtt{T}_1 \ldots \mathtt{T}_n$*, is a type.*

μ-abstraction. *If* $\mathtt{T}, \mathtt{S}$ *are types and* $\mathtt{S}$ *occurs in* $\mathtt{T}$*, then* $[\mu\alpha]\,\mathtt{T}[\mathtt{S}\backslash\alpha]$ *(for a fresh variable name* α*) is a type.*

Definition 2 (Syntax: Type constructor). *Type constructors are the defined thus:*

Base case. *Every type* $\mathtt{T}$ *is a type constructor.*

Abstraction. *If* $\mathtt{C}$ *is a type constructor and* $\mathtt{T}$ *is a type,* $[\forall x]\,\mathtt{C}[\mathtt{T}\backslash x]$ *is a type constructor.*

Sum types. *If* $\mathtt{C}_1 \ldots, \mathtt{C}_n$ *are type constructors with* $\mathtt{C}_i = \left[\forall x_1^i \ldots, x_n^i\right] \mathtt{T}_i$ $(1 \leq i \leq n)$*, then* $[\forall x_1, \ldots, x_n]\,(\mathtt{C}_1 + \cdots + \mathtt{C}_n)$ *is a type constructor.*

Partial application. *If* $\mathtt{T}_1, \ldots, \mathtt{T}_i$ $(i < n)$ *are types and* $[\forall x_1, \ldots, x_n]\,\mathtt{T}$ *is a type constructor, then* $[\forall x_{i+1}, \ldots, x_n]\,\mathtt{T}[x_1\backslash\mathtt{T}_1, \ldots, x_i\backslash\mathtt{T}_i]$ *is a type constructor.*

Every type is interpreted as a set of values which are of that type; type constructors are interpreted as universally quantified templates for actual types. Their formal semantics are as follows:

Definition 3 (Semantics: Type). *Let* $\mathtt{T}$ *be a type. Its interpretation function* $\mathrm{int}(\mathtt{T})$ *is defined thus:.*

Basic type. $\mathbb{R}$ *is interpreted as the set of real numbers.* $\mathrm{int}(\emptyset) = \{\}$.

Sum type. *If* $\mathtt{T}_1, \mathtt{T}_2$ *are types, then* $\text{int}(\mathtt{T}_1 + \mathtt{T}_2) = \text{int}(\mathtt{T}_1) \cup \text{int}(\mathtt{T}_2)$.

Product type. *If* $\mathtt{T}_1, \ldots, \mathtt{T}_n$ *are types and* $\mathtt{s}$ *is a string, then*

$$\text{int}(\mathtt{s}\ \mathtt{T}_1 \ldots \mathtt{T}_n) = \begin{cases} \{\mathtt{s}\} & \text{if } n = 0 \\ \{\mathtt{s}\} \times \text{int}(\mathtt{T}_1) \times \cdots \times \text{int}(\mathtt{T}_n) & \text{if } n \geq 1. \end{cases}$$

Full application. *If* $\mathtt{T}_1, \ldots, \mathtt{T}_n$ *are types and* $[\forall x_1, \ldots, x_n]\,\mathtt{C}$ *is a type constructor, then*

$$\text{int}(\mathtt{C}\ \mathtt{T}_1 \ldots \mathtt{T}_n) = \bigcup_{v_1 \in \text{ int}(\mathtt{T}_1)} \cdots \bigcup_{v_n \in \text{ int}(\mathtt{T}_n)} \left(\bigcup_{C' \in \text{ cint}(C)} C'[x_1 \backslash v_1, \ldots, x_n \backslash v_n] \right).$$

μ-abstraction. *If* $[\mu\alpha]\,\mathtt{T}$ *is a type, then*

$$\text{int}([\mu\alpha]\,\mathtt{T}) = \text{int}(\mathtt{T}) \cup \text{int}(\mathtt{T}[\alpha \backslash \mathtt{T}]) \cup \text{int}(\mathtt{T}[\alpha \backslash \mathtt{T}][\alpha \backslash \mathtt{T}]) \cup \ldots$$

with $\text{int}(\alpha) = \{\}$.

Definition 4 (Semantics: Type constructor). *The partial interpretation function* cint *for type constructors is defined as follows: if* $\mathtt{C}$ *is a type constructor containing exactly the types* $\mathtt{T}_1, \ldots, \mathtt{T}_n$*, then*

$$\text{cint}(\mathtt{C}) = \bigcup_{v_1 \in \text{ int}(\mathtt{T}_1)} \cdots \bigcup_{v_n \in \text{ int}(\mathtt{T}_n)} \mathtt{C}[\mathtt{T}_1 \backslash v_1, \ldots, \mathtt{T}_n \backslash v_n].$$

Intuitively, sum types are simply unions, product types are named cartesian products, and full applications are instantiations of type constructors with all possible values. μ-abstraction represents recursive data types such as lists or trees. Type constructors themselves are just generic types.

Whenever we want to assert that an expression has a specific type, we write:

Notation 5 (Typed expressions). *Let* x *be an expression and* $\mathtt{T}$ *a type.* $x :: \mathtt{T}$ *asserts that* x *has type* $\mathtt{T}$.

Henceforth, by convention, we will write type variables in lower-case and concrete types in upper-case, omitting the explicit $\forall$-blocks. That is, a type like $[\forall x, y, z]\,\mathtt{C}\ \mathtt{x}\ (\mathbb{N} + \mathtt{T}_1)\ \mathtt{y}\ \mathtt{z}$ will simply be written as $\mathtt{C}\ \mathtt{x}\ (\mathbb{N} + \mathtt{T}_1)\ \mathtt{y}\ \mathtt{z}$ and it will be clear that $\mathtt{x}, \mathtt{y}, \mathtt{z}$ are type variables, while $\mathbb{N}, \mathtt{T}_1$ are concrete types. A special kind of type constructor is the function arrow ($\to$) which induces the function type:

Example 6 (Function arrow). *If we take, say, the type* $\to\ \mathtt{S1}\ \mathtt{S2}$ *(a product type with the product types* $\mathtt{S1}$ *and* $\mathtt{S2}$ *as arguments) and abstract twice, we get* $[\forall s, t] \to \mathtt{s}\ \mathtt{t}$. $\to\ \mathtt{s}\ \mathtt{t}$ *is the type constructor for unary functions from* $\mathtt{s}$ *to* $\mathtt{t}$*, also written infix as* $\mathtt{s} \to \mathtt{t}$*. Functions with multiple arguments, mapping* $\mathtt{t}_1, \ldots, \mathtt{t}_{n-1}$ *to* $\mathtt{t}_n$*, can be modelled in two ways: either through n-tuples, or through nested function arrows:*

$$\langle \mathtt{t}_1, \mathtt{t}_2, \ldots \mathtt{t}_{n-1} \rangle \to \mathtt{t}_n$$
$$\mathtt{t}_1 \to (\mathtt{t}_2 \to \cdots \to (\mathtt{t}_{n-1} \to \mathtt{t}_n) \cdots)$$

The first method necessitates that we supply all arguments at once, whereas the second allows them to be given one after another.

Function arrows allow the execution of functions in the obvious way:

Definition 7 (Function application). *Let* $f :: \mathtt{S} \to \mathtt{T}$ *and* x *be an expression of type* $\mathtt{S}$*. Then* $f\ x$ *is an expression of type* $\mathtt{T}$*. Function application associates to the left, that is:* $f\ x_1 \dots x_n = (\cdots((f\ x_1)\ x_2) \dots x_n)$*.*

We can combine type constructors, sum types, and product types into *algebraic data types* (ADTs).

Definition 8 (Algebraic data type (ADT)). *Let* $\mathtt{s}$ *be a string and* $\mathtt{C_1}, \dots, \mathtt{C_n}$ *be type constructors such that* $\mathtt{C_i} = [\forall x_1, \dots, x_n]\, \mathtt{T_i}$ *and* $\mathtt{T_i}$ *is a named product type with type variables* $(1 \leq i \leq n)$*. Then* $[\forall x_1, \dots, x_n]\, (\mathtt{T_i} + \cdots + \mathtt{T_n})$ *is an ADT. If we want to give a name to an ADT, we write it as* $\mathtt{s\ x_1 \dots x_n} = \mathtt{T_i} + \cdots + \mathtt{T_n}$*.*

Since an ADT is merely the sum of product types, it is itself a type constructor. If it has no type variables, it is also a type. Next, we define a couple of example ADTs, some of which we will use in the next section.

Example 9 (ℕ, 𝔹, ℚ, ℂ, Maybe, Either, List).

$$\begin{array}{lcl}
\mathbb{N} & = & [\mu\alpha]\, \mathtt{Z} + \mathtt{S}\ \alpha \\
\mathbb{B} & = & \mathtt{False} + \mathtt{True} \\
\mathbb{Q} & = & \mathtt{Rat}\ \mathbb{N}\ \mathbb{N} \\
\mathbb{C} & = & \mathtt{Complex}\ \mathbb{R}\ \mathbb{R} \\
\mathtt{Maybe\ t} & = & \mathtt{Nothing} + \mathtt{Just\ t} \\
\mathtt{Either\ l\ r} & = & \mathtt{Left\ l} + \mathtt{Right\ r} \\
\mathtt{List\ a} & = & [\mu\alpha]\ \mathtt{Nil} + (\mathtt{a} : \alpha)
\end{array}$$

ℕ is the usual Peano definition of natural numbers, with a nullary product type $\mathtt{Z}$ *representing zero, and a unary product type* $\mathtt{S}$*, which allows recursion. 𝔹, ℚ, ℂ are the sets of Boolean number and rational/complex numbers, respectively, with* $\mathtt{False}$ *and* $\mathtt{True}$ *being nullary product types, and with* $\mathtt{Rat}\ \mathbb{N}\ \mathbb{N}$ *and* $\mathtt{Complex}\ \mathbb{R}\ \mathbb{R}$ *being binary ones.* $\mathtt{Maybe}$ *represents an optional value, which may or may not be present.* $\mathtt{Either}$ *represents a choice between two values, of which either the left or the right one is present, but not both.* $\mathtt{List\ a}$ *(or just* $[\mathtt{a}]$ *as a shorthand) denotes a list of values of type* $\mathtt{a}$*. There,* $\mathtt{Nil}$ *is the nullary type constructor for an empty list and* $:$ *is an infix binary type constructor that stores the head and tail of a list.*

We also define the usual convenience functions for these types:

$$\begin{array}{rcl}
\mathtt{isJust} & :: & \mathtt{Maybe\ a} \to \mathtt{Bool} \\
\mathtt{isJust}\ m & = & \begin{cases} \mathtt{True} & \text{if } m = \mathtt{Just}\ x \\ \mathtt{False} & \text{otherwise} \end{cases}
\end{array}
\qquad
\begin{array}{rcl}
\mathtt{head} & :: & [\mathtt{a}] \to \mathtt{a} \\
\mathtt{head}\ l & = & \begin{cases} x & \text{if } l = x : xs \\ \bot & \text{otherwise} \end{cases}
\end{array}$$

$$\begin{array}{rcl}
\mathtt{fromJust} & :: & \mathtt{Maybe\ a} \to \mathtt{a} \\
\mathtt{fromJust}\ m & = & \begin{cases} x & \text{if } m = \mathtt{Just}\ x \\ \bot & \text{otherwise} \end{cases}
\end{array}
\qquad
\begin{array}{rcl}
\mathtt{tail} & :: & [\mathtt{a}] \to [\mathtt{a}] \\
\mathtt{tail}\ l & = & \begin{cases} xs & \text{if } l = x : xs \\ \bot & \text{otherwise} \end{cases}
\end{array}$$

Definitions 1–8 specify a fragment of System F_ω,[13] which is used to type expressions in the lambda calculus. Although System F_ω is strictly more powerful, our definitions are enough to provide a description language for the data types and functions in the rest of this work.

[13] Specifically, the decidable fragment of System F_ω without higher kinds and only prenex-polymorphism. That is, type constructors can only take types as arguments and are of the form $[\forall x_1, \dots, x_n]\, \mathtt{C}$ for quantifier-free $\mathtt{C}$. This is also called the Hindley-Milner type system. For details, see [6].

5.2 Neural systems

Definition 10 (Neural component). *Let I be an index set and let* `T` *be any type. Then, a neural component C with a name from I and message type* `T` *is a four-tuple*

$$\langle \texttt{name}, \texttt{ft}, \texttt{int}, \texttt{proc} \rangle$$

where

1. $\texttt{name} :: I$ *is the* name *of C,*
2. $\texttt{ft} :: \texttt{T} \to \mathbb{B}$ *is called the* filter *of C,*
3. $\texttt{int} :: \texttt{T} \to \texttt{Maybe T}$ *is called the* interpreter *of C, and*
4. $\texttt{proc} :: \texttt{T} \to \texttt{T}$ *is called the* processor *of C.*

Formally, the type of C is $\texttt{Comp}_{\texttt{T},I}$*. As a shorthand, we denote the name, filter, interpreter and processor of a given component C as* $\texttt{name}_C$, $\texttt{ft}_C$, $\texttt{int}_C$, $\texttt{proc}_C$, *respectively.*

A set of neural components, together with a set of messages, induces a *neural system*:

Definition 11 (Neural system). *Let T be any type and let I be an index set. Then, a neural system with message type T and component names from I is a tuple*

$$\langle \boldsymbol{Co}, \boldsymbol{Me} \rangle$$

where

- $\boldsymbol{Co}$ *is a set of neural components (with message type T and names from I) and*
- $\boldsymbol{Me}$ *is a set of elements of type T, called the* set of messages*.*

5.3 Sending and receiving messages

We now give a notation for the sending and receiving of messages in a system. Here, we distinguish two aspects: first, the structural, which describes how messages *can* travel in a system and the operational, which describes how they *do* travel in some given scenario.

5.3.1 Structural notation

The elements of a component statically determine which messages it can receive and send. Based on the behaviour of the filter, interpreter and processor of a component, we can express a number of properties.

Definition 12 (Message reception). *Let C be a component and m a message. C can receive m if and only if $\boldsymbol{ft}_C\ m = \texttt{True}$ and $\boldsymbol{int}_C\ m = \texttt{Just}\ m'$ for some m'. When C can receive all messages in $\{m_1, \ldots, m_n\}$, we write:*

$$\{m_1, \ldots, m_n\} \rightarrowtail C.$$

We denote the opposite statement — that C cannot receive any message in $\{m_1, \ldots, m_n\}$ — by:

$$\{m_1, \ldots, m_n\} \multimap C.$$

Definition 13 (Message sending). *Let C be a component and $m, m_1, \ldots, m_n$ messages. C can send out a message m if and only if there exists a message m_{in} s.t. $\texttt{proc}_C\ m_{\text{in}} = m$. When C can send all messages in $\{m_1, \ldots, m_n\}$, we write:*

$$C \rightarrowtail \{m_1, \ldots, m_n\}.$$

The opposite statement — that C cannot send any message in $\{m_1, \ldots, m_n\}$ — is denoted by:

$$m_1, \ldots, m_n \multimap \{C\}.$$

Definition 14 (Receiving set). *The set of components which can receive a message m is denoted by*

$$\texttt{rec}(m) \equiv \{C \in \mathbf{Co} \mid \{m\} \rightarrowtail C\}.$$

`rec` *can also be overloaded to refer to the set of components which can receive and interpret at least some message of a component C:*

$$\texttt{rec}(C) \equiv \{C_i \in \mathbf{Co} \mid \exists m :\ C \rightarrowtail \{m\} \wedge \{m\} \rightarrowtail C_i\}.$$

5.3.2 Operational notation

Whereas the structural notation pertained to the static properties of a neural system, the operational notation describes *traces*: lists of sent and received messages, and the changes they induced in the system.

Definition 15 (Message action). *When a component C_i outputs a message m_{out} that another component C_j receives and interprets as message m_{in}, we write*

$$C_i \rightarrow [m_{out}, m_{in}] \rightarrow C_j.$$

We refer to this as message action*. If it's clear that the message m does not change, we just write*

$$C_i \rightarrow [m] \rightarrow C_j.$$

Definition 16 (Trace). *Traces are defined inductively thus:*

1. *Every message action is a trace.*
2. *If T_1 and T_2 are traces, $T_1; T_2$ is a trace.*

";" denotes sequential execution and is associative. Thus, the semantics of a trace $T_1; T_2; \ldots; T_n$ are that T_1 is executed first, followed by T_2, and so forth, until T_n is reached and the execution ends. For readability, $T_1; \ldots; T_n$ will sometimes be written line-by-line as

$$\begin{array}{l} T_1 \\ \vdots \\ T_n \end{array}$$

Definition 17 (Component mutation). *Let $f_1, f_2, \ldots$ be functions $\texttt{Comp}_{T,I} \to \texttt{Comp}_{T,I}$ which preserve the names of components, m, m' messages of type T, and let C be a component of type $\texttt{Comp}_{T,I}$. When C is changed into $(f_n \circ \cdots \circ f_1)\ C$ by a message m it receives, or changed into $(f_n \circ \cdots \circ f_1)\ C$ by a message m' it sends, we write, respectively:*

$$\cdots \to [m] \to \langle f_1, \ldots, f_n \rangle C$$
$$C\langle f_1, \ldots, f_n \rangle \to [m'] \to \ldots$$

If no change occurs, that is, if

$$C\langle\rangle \to [m] \to \ldots \quad \text{or}$$
$$\cdots \to [m] \to \langle\rangle C$$

we omit the angle brackets. The semantics are as follows: after by sending or receiving a message, **Co** *is replaced by* $(\mathbf{Co} - \{C\}) \cup \{(f_n \circ \cdots \circ f_1)\ C\}$.

Definition 18 (Plastic and non-plastic neural systems). *If, for all messages m and components C, C' in a neural system, the following holds:*

$$C\langle\rangle \to [m] \to \langle\rangle C'$$

we call the system non-plastic. Otherwise, we call it plastic.

This definition intends to roughly convey the notion of neuroplasticity, as used in neuroscience: areas in the brain are changed over time through specific patterns of activity. Here, such change is modelled by the execution of functions and the replacement of C in the system by $f_n \circ \cdots \circ f_1(C)$.

5.4 Invariants

Such a model does not necessitate the existence of special structures, such as central organizers or sequences of components, one activated after another,[14] but it does not preclude them either. In fact, we can enforce certain features via first-order invariants. For example, a central organizing units for the components $C_1, \ldots, C_n$ can be emulated by a component C_{co} which accepts messages and transforms them into an appropriate format for the some other components.

Invariant 19 (Central organiser).

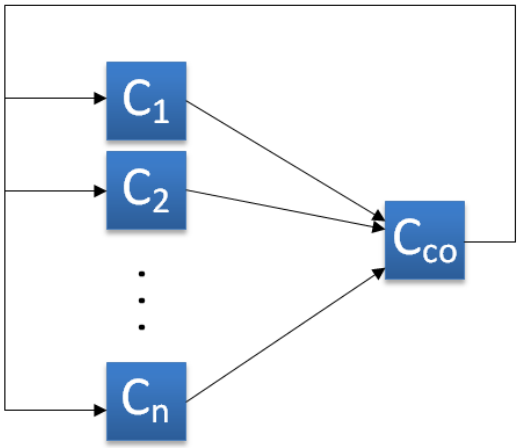
$$[\forall i \in \{1 \ldots, n\}][\forall m]:$$
$$(C_i \rightarrowtail \{m\} \Rightarrow \texttt{rec}(m) = \{C_{co}\}) \wedge \left(\left(\texttt{proc}_{C_{co}} \circ \texttt{int}_{C_{co}}(m)\right) \in \bigcup_{1 \leq j \leq n} \texttt{rec}(C_j) \right).$$

Figure 14a depicts such an organizer. Similarly, sequences can be created by components $C_1, \ldots, C_n$, where each components reads the message of the last one.

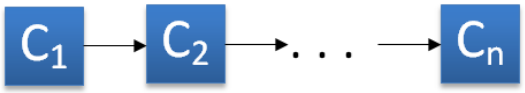
Invariant 20 (Sequence).

$$[\forall i \in \{2 \ldots, n\}] : \ \texttt{rec}(C_{i-1}) = \{C_i\}.$$

[14] An example of such a sequence is found in [70] where the authors model the emotion process as a four-step pipeline of relevance, implication, coping and normative significance.



(a) Components communicating via a central organising mechanism.



(b) A sequence of components.

6 Selected subsystems

The global architecture now specified, we will introduce three related subsystems and fit them into this global framework: sensory perception — the processing of raw sensory input into an format intelligible to other brain components —, belief generation — the imagination, which mimics the output of the senses —, and affect — broadly speaking, the emotional component of cognition.

6.1 Sensory perception

The model presented herein is inspired by Marvin Minsky's "The Emotion Machine". Therein, Minsky proposes a layered mental structure where each successive layer operates on more and more abstract representations of the world, starting with primitive sensations and proceeding all the way to self-conscious reflection and rational planning. Figure 15 shows such a layered structure.

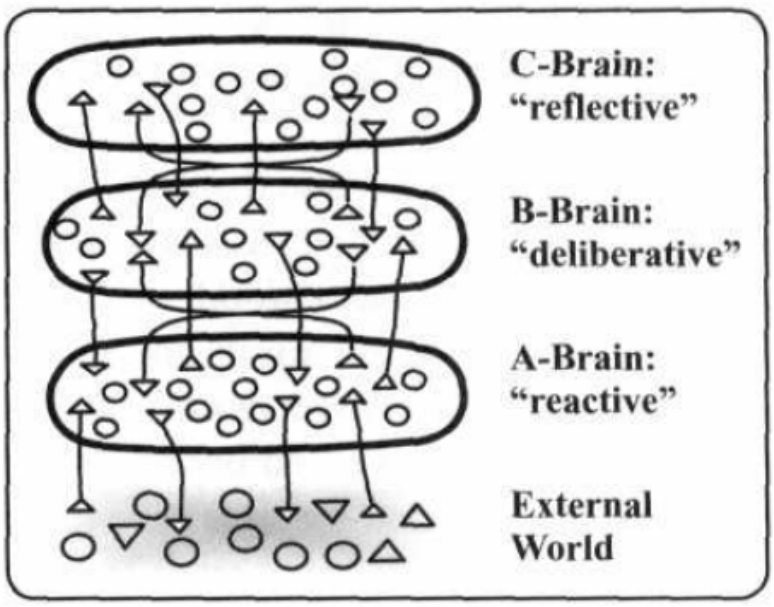


Figure 15: Layered perception of the world, from [57, p. 100].

The diagram is explained thus [57, p. 100]:

> Now suppose that your A-Brain gets some signals from the external world (via such organs as eyes, ears, nose, and skin) — and that it also can react to these by sending signals that make your muscles move. By itself, the A-Brain is a separate animal that only reacts to external events but has no sense of what they might mean. For example, when the fingertips of two lovers come into intimate physical contact, *the resulting sensations, by themselves, have no particular implications.* For there is no significance in those signals themselves: their meanings to those lovers *lie in how they prepresent and process them in the higher levels of their minds.*

If we apply this to the architecture of Section 11, we can devise a system in which each sense S has an associated component C_S which does two things:

1. Consume the raw sensory information delivered by various organs and output processed input for higher brain functions;
2. as a side a effect of this processing, cause instinctive, low-level reactions in the body, such as pulling away from pain or jumping at a sudden fright.

In Figure 16, a slice of just such a system is shown for visual, auditory, olfactory/gustatory and tactile sensation. The produced data can be of two kinds: one is more abstract than the input and facilitates deliberative action, and the other contains instructions for instinctive behaviour for the body.

6.2 Belief generation and planning

Broadly speaking, belief generation can be described as "imagination", and is closely related to sensory perception and world simulation. In examining the system, we might broadly classify

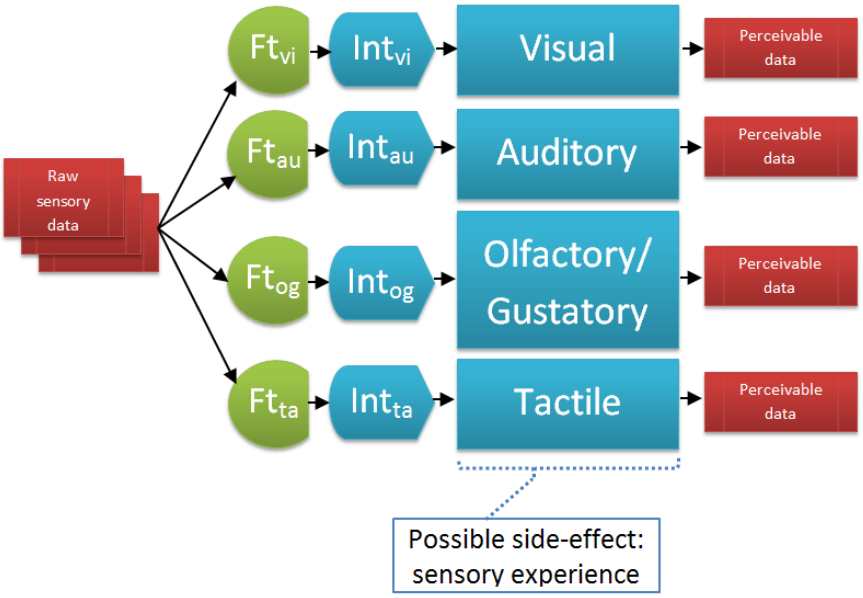


Figure 16: Partial structure of sensory perception - raw sensory data is processed and made available to higher functions such as the affective subsystem. The comment "Possible side-effect: sensory experience" signifies the fact that conscious and subconscious sensory experiences might occur as a side-effect of this processing. However, it is currently unknown to neuroscience whether this is indeed the case.

its processes into three categories:

1. Belief generation — imagining sights, sounds, etc. Such experiences have much in common with those caused by our sensory organs, yet are marked not as real. In particular, imagined experiences evoke only parts of the conscious experience that accompanies real perceptions. Research by Berthoz and Lotze et al. suggests that (a) the brain indeed uses similar circuitry for real and imagined experiences and that (b) imagined experiences are prevented from being confused with real ones via inhibitory signals. Lotze et al. write [51]:

   > The results of cortical activity support the hypothesis that motor imagery and motor performance possess similar neural substrates. The differential activation in the cerebellum during EM and IM is in accordance with the assumption that the posterior cerebellum is involved in the inhibition of movement execution during imagination.

   From the abstract of Berthoz's paper [7]:

   > (...) experimental evidence suggesting that the brain can use the same mechanisms for the imagination and the execution of movement. In particular the fact that adaptation of the vestibulo-ocular reflex can be obtained by pure mental effort and not solely by conflicting visual and vestibular cues has been suggestive of the fact that the brain could internally simulate conflicts and use the same adaptive mechanisms used when actual sensory cues were in conflict.

2. World simulation — the imagination of future states. Simulating worlds goes beyond the imagination of sensory experiences; it involves constructing models of worlds and simulating their behaviour. The details of this process are unknown, but we can assert that it is capable of a number of things:
   a) construction of non-physical worlds, such as mathematical models,
   b) extrapolation into the future and the past
   c) simulation of the minds itself and other agents.
3. Executive planning — humans can plan both both in immediate and concrete terms (such as body movement) and in the abstract. It is likely that different circuitry is used for movement planning and for planning involving abstract reasoning, in both cases it is necessary that the brain simulate the world in some way. The simulation of the consequences of body movement is likely older than humanity and distinct from the kind of world simulation described above, but both share their function: the agent proposes as series of actions to take, inserts them into some mental world and judges the utility of those actions based on the predicted consequences.

Needless to say, that this process in all its subtleties is immensely complex and thus I simply endeavour to sketch its possible structure only in extremely rough outlines. This sketch is shown in Figures 17, 18, and 19: the world simulation is an ordinary component with a filter and interpreter which outputs, for simplicity's sake, messages marked as imaginary. We can imagine such messages to be very much like ordinary sensory ones, with the exceptions that they have no accompanying sensation and, more importantly, that we are aware of their non-reality. The planning component receives instructions about desirable states and outputs hypothetical actions which the world simulator incorporates. The world simulator's output is in turn read by the planner, which then abandons the plan or decides to pursue it further.

The planner, minimally, has to perform two functions — first, it has to judge the desirability of various world states and second, it has to be able to devise possible steps for the agent based on some strategy. If these two functions and some desired goal(s) are given, the planner can do its work by issuing the following commands, as shown in Figure 18:

1. If some goals are not yet reached but appear possible, devise possible steps to take and have the world simulator predict their outcomes.
2. If the goals appear impossible the necessary steps prohibitively undesirable, command the world simulator to cease its activity.
3. If earlier proposed steps turn out to fulfil some goal, contact the agent's executive component.

6.2.1 World simulation as rationality

The way in which I just described the interaction between the world simulator and the planner suggests that they function as a pair of guesser and checker: the planner generates ideas on what to do and the world simulation tests their viability in some setting. Indeed, we can model rational thinking as embedded in the world simulator, especially if we make use of a plastic neural system. The proposed steps of the planner might be quite chaotic and irrational, but

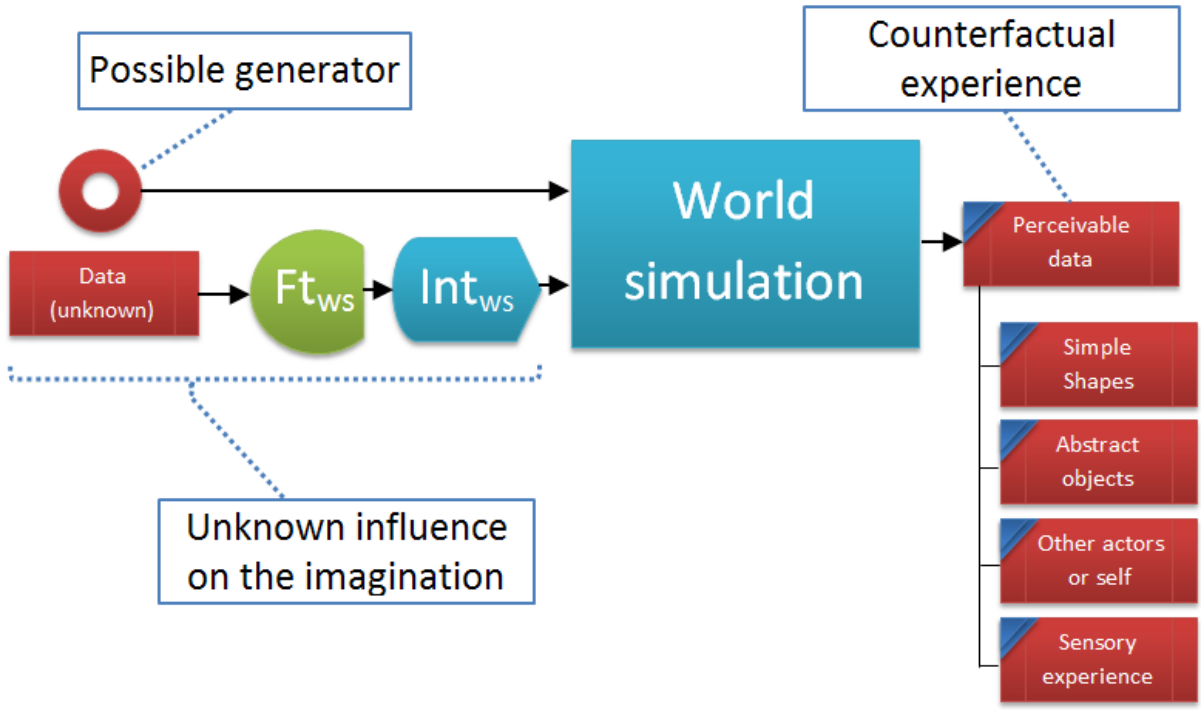


Figure 17: Structure of of belief generation & world simulation: messages emulating the output of sensory perception are generated, but are marked as imaginary by unknown means.

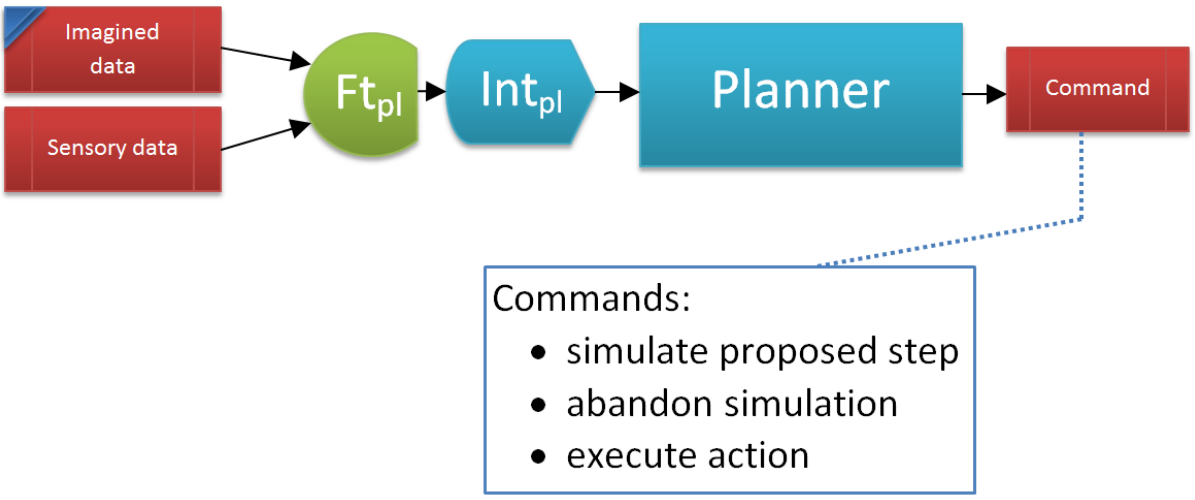


Figure 18: Planner with two kinds of inputs: (1) real sensory data and (2) imaginary data which comes from world simulation. On the basis of these inputs, possible steps are developed and sent out as commands.

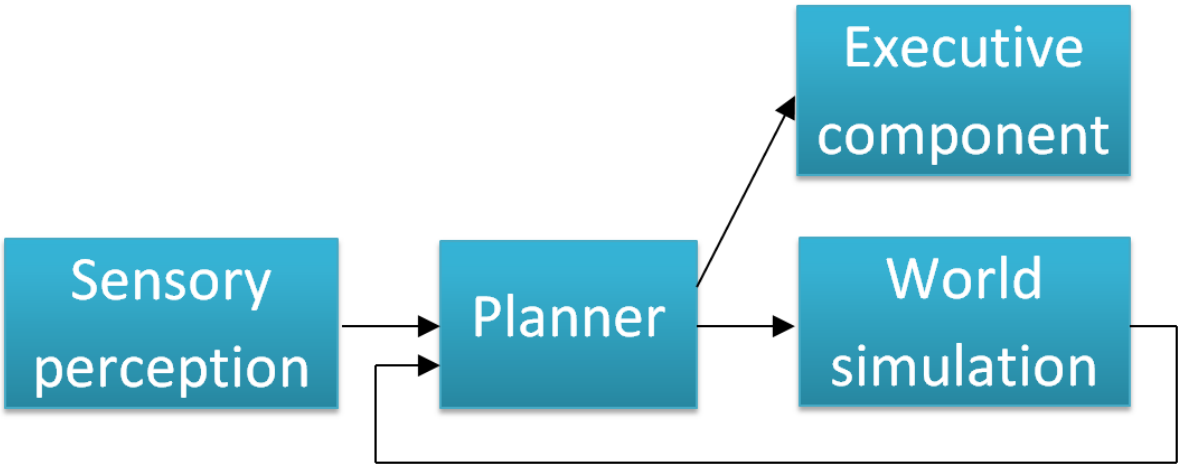


Figure 19: Interaction between world simulator and planner: the planner devises possible steps and feeds them into the world simulator, which, in turn, tries to calculate their effects. The results are fed back to the planner.

when given to the world simulator, it recognises them as such and returns a failure signal to the planner, causing it to abandon "bad" paths of cognition. A plastic planner can learn from the consistent failure of certain kinds of steps and, in time, propose them less and less often. Observed as a whole, this system of planner and simulator appears to simply deliver good plans by intuition, even though, in isolation, neither part is very clever.[15]

Model. In a simplified way, we can model the process of logical deduction in a formal system $F = (A, R)$, where A is a recursive set of axioms and R is a recursive set of production rules of the form $(r_{\text{from}}, r_{\text{to}})$ s.t. $r_{\text{from}} \to r_{\text{to}}$ is a valid production in the system. Let

1. W be a world simulator for the world of propositions $\mathcal{P}$ in (A, R),
2. P a planner,
3. $\text{St} = \{s_1, \ldots, s_p\}$ a set of messages about steps to take,
4. $\text{Cat} = \{K_1, \ldots, K_q\}$ a list of message categories,
5. $\texttt{cur} : W_S$ the current state of the world simulator,
6. $\texttt{ins} :: W_S \to \text{St} \to W \to W$, $\texttt{del} :: \text{St} \to W \to W$ functions for inserting or deleting a state change into the world simulator or the planner,
7. $t(i)$ and $b(i)$ functions which increase or decrease the likelihood of sending a message belonging to category K_i and

[15] I do not wish to idealize rationality too much; world simulation is only partly rational and, given faulty information about the world, will err considerably and in documented ways. Similarly, it is certainly possible for the planner to derange the world simulator by evaluating certain states as so desirable/undesirable that it will pursue even scenarios which the world simulator reports as highly unlikely.

8. $\bot_i, \top_i$ the failure and success signals of a message belonging to the category K_i.

One step of the interaction between W and P, in a scenario where P proposes steps $s_{i_1}, \ldots, s_{i_n}$, can then be modelled with two traces T_{guess} and T_{check}:

$$T_{\text{guess}}(\texttt{step}) \equiv P\langle \texttt{ins cur step})\rangle \to [\texttt{step}, \texttt{step}] \to \langle \texttt{ins cur step})\rangle W$$

$$\begin{aligned} T_{\text{check}}(\texttt{step}) \equiv\ & \forall K_i \in Cat : K_i(\texttt{step}) \Rightarrow \\ & \text{if } [\exists s_j]\,(\texttt{cur}, s_j) \in R \text{ then } W\langle\rangle \to [\top_i, \top_i] \to \langle t\ i\rangle P \\ & \text{else } W\langle \texttt{del step})\rangle \to [\bot_i, \bot_i] \to \langle \texttt{del step}, b\ i\rangle P \end{aligned}$$

Axioms can be selected by executing $T_{\text{guess}}(\texttt{ax})$ for all $\texttt{ax} \in A$. We can then perform deduction via $T_{\text{guess}}; T_{\text{check}}$, for a probabilistically selected $\texttt{step} \in St$.

Intuitively, T_{guess} guesses a step to take. It does so but inserting it into the planner's world-state via `ins` and then sending a message to the world simulator, which also inserts it into its world state. T_{check} then checks whether the change from `cur` to `step` was legitimate. If so, it determines to which category `step` belongs and sends the $\top$-signal for that category back to the planner. Otherwise, it sends the corresponding $\bot$-signal. The purpose of this is to make it more or less likely, respectively, that the planner should choose the same category of step in the future. The categories, we can imagine, could be things like "modus ponens", "associative reasoning", "appeal to consequences" and so forth.

If we repeat this interaction (with different proposed steps $s_1, \ldots, s_p$ in each iteration), we get an algorithm for logical deduction — that is, since A and R are recursive, the system will recursively enumerate all valid logical formulas, provided that we pursue each path and that the probability of selecting any valid step is > 0. In addition, we could add a goal function g to P s.t. it would accept certain states and stop. Thereby, P and W could be used to prove logical propositions.

6.3 Affect

When discussing human affect, one can mean various things: the causation of emotion, its internal mechanisms, the expression of emotion, social communication of emotions, etc. In this document, we restrict our attention just to the internal mechanisms — that is, to the means by which emotions are evoked in an agent and how they shape its thinking.

Furthermore, the issue will only be the causative mechanism itself; taxonomy and hierarchy of emotions are deferred to future versions of this document.

The model presented herein is adapted from Gadanho and Hallam [31], who employed it in the context of robot learning. They constructed a system of FEELINGS and SENSATIONS $\mathcal{F}$, EMOTIONS $\mathcal{E}$, and a hormone storage H.

Figure 20 shows this model: SENSATIONS enter the system and are connected to the FEELINGS. They, in turn, determine the agent's EMOTIONS. The emotions then feed into a HORMONE STORAGE, the contents of which influence, together with the SENSATIONS, the agent's FEELINGS. In the context of their paper, this model had a very restricted application. Its purpose was to merely help guide a robot through a world, and accordingly, $\mathcal{F}$ and $\mathcal{E}$ were only defined as [31, p. 47]:

$$\mathcal{F} = \{\text{Hunger}, \text{Pain}, \text{Restlessness}, \text{Temperature}, \text{Eating}, \text{Smell}, \text{Eating}, \text{Proximity}\}$$
$$\mathcal{E} = \{\text{Happiness}, \text{Sadness}, \text{Fear}, \text{Anger}\}$$

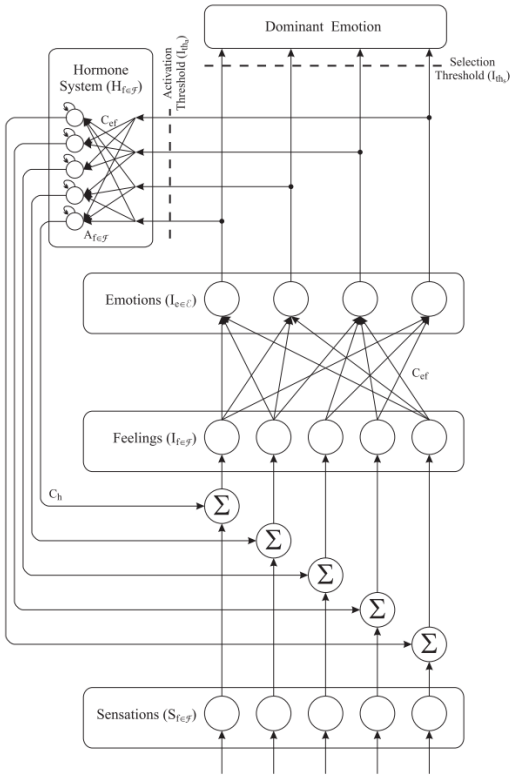


Figure 20: Emotional model of Gadanho and Hallam [31, p. 46].

The main advantage of Gadanho's and Hallam's model is that (a) it is sufficiently generic to accommodate various schemas and (b) posits an internal state (the hormone storage), giving agents a certain inertia. For example, one can imagine integrating a many-dimensional model like Brazeal's [10] detailed taxonomy of emotion like Ortony's OCC model [62]. The existence of an internal state is necessitated by the simple observation that our internal world is not solely dependent on momentary stimuli, but merely influenced by them. The idea of a hormone storage might be a simplistic approximation but it, too, can be refined as needed.[16] Figure 16 shows the adapted model. The general structure was retained, but the set of sensations was replaced by the sensory processor described in Section 6.1 and, instead of a single dominant emotion, competing emotions simply emit messages which are used by execute components and the world simulation.

6.3.1 Affective subsystems

In this section, I will develop the concept of "emotion" in greater detail. The process shown in Figure 21 might suggest we simply have a collection of emotions and that all emotions are essentially equal, but I submit that this is not so. Instead, I propose the existence of various subsystems, each responsible for a group of emotions, and each with its own history and

[16] It might be tempting to simply replace the hormone storage with the message space, but doing so would ignore the role that neurotransmitters like dopamine and serotonin play in cognition, irrespective of the purely computational activity of brain components.

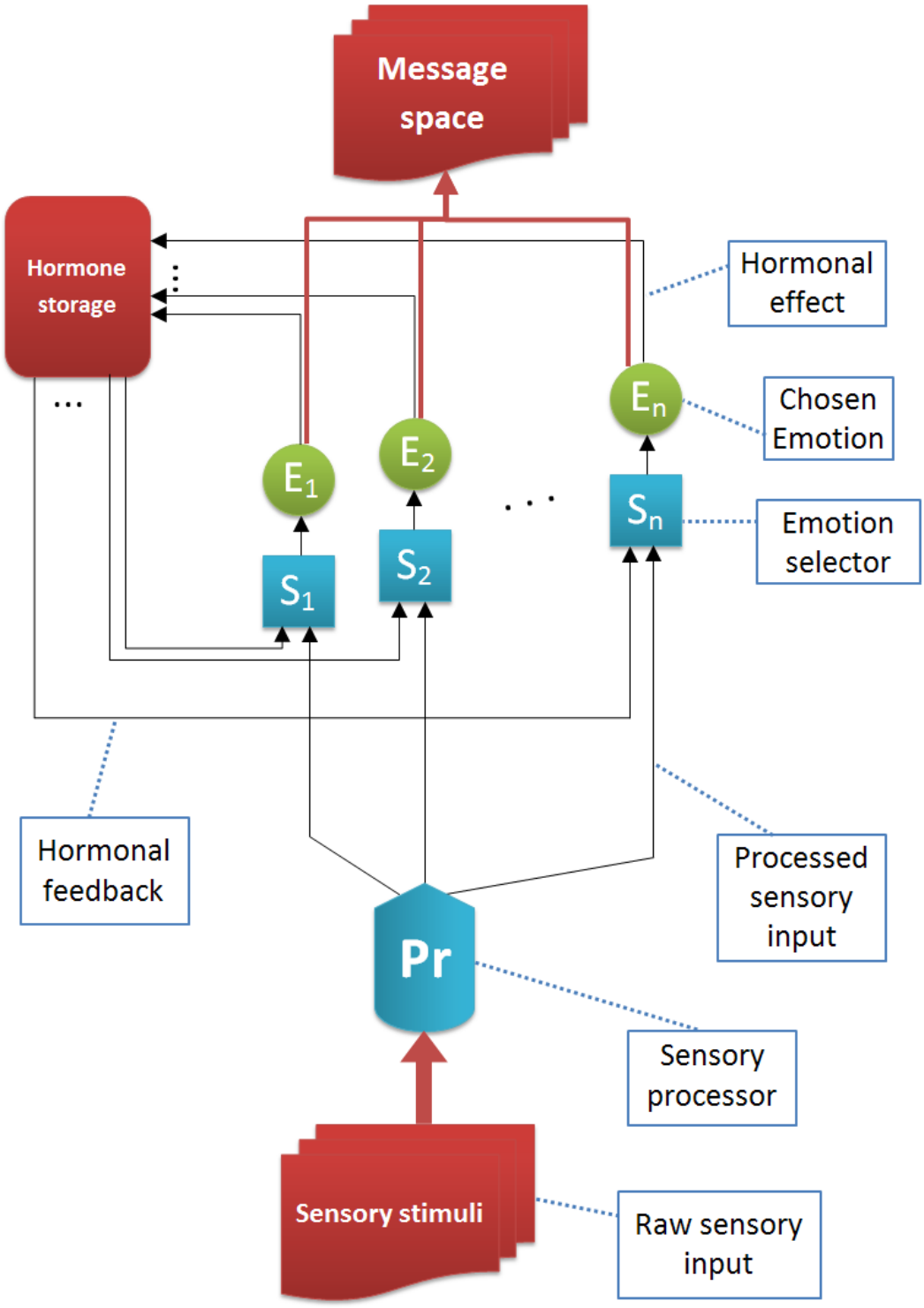


Figure 21: Affective subsystem; specialisation of the global neural architecture. In plastic neural systems, selections may change over time.

distinctive tasks. In the rest of this work, the following two assumptions will be made:

1. *"Emotion" is not a singular phenomenon.* Specifically, this is contradicts many-dimensional models of emotions which propose one, two, three or four axes and a corresponding vector space in which every emotion is a point. Such a view implies that all emotions share a neurological template which is parametrized with coordinates to result in different experiences.

2. *There exist emotions which are both different in kind and which pertain to different subsystems in the brain.* This implies that emotions cannot morally be seen as a homogeneous set $\{E_1, \ldots, E_n\}$. Instead, a number of distinct subsystems are necessitated, each responsible for the causation and processing of a group of emotions. Given this, the only substantial aspect any two emotions might have in common would be our referring to both of them as "emotion".

Both of these assumptions are rather concrete and thus deserve evidence. In 1999, Davidson and Irwin, using PET and fMRI scanning, found two different systems mediating approach- and avoidance related behaviors [21, p. 13]:

> A large body of lesion, neuroimaging and electrophysiological data supports the view that the prefrontal cortex (PFC) is an important part of the circuitry that implements both positive and negative affect. (...) A number of early studies that evaluated mood subsquent to brain damage suggested that patients with damage to the left hemisphere, particularly in PFC, were more likely to develop deppressive symptoms compared with patients having lesions in homologous regions of the right hemisphere. (...) The general finding of left dorso-lateral PFC damage increasing the likelihood of deppressive symptoms has been interpreted to reflect the contribution of this cortical territory to certain features of positive affect, which, when disrupted, increases the probability of depressive symptomatology.

In this, they echo earlies findings by Cacioppo et al. [14], Gray [34] and Lang et al. [46] that affect is lateralized, with different hemispheres being responsible for different categories of feeling. It therefore stands to reason that different emotions, being generated by different brain regions, should therefore also be different in their character.

Further, much research has been done in the area of so-called *basic emotions* — a small set of emotions are acknowledged as being both elementary and characteristically distinct from each other. The Cambridge Handbook of Affective Neuroscience provides a good overview of the basic emotion theory [3, pp. 9-10]. Matsumoto and Eckman [53], for instance, identified seven basic emotions: happiness, surprise, contempt, sadness, fear, disgust, and anger.

Damasio [20], drawing upon neuroscientific findings, sketches a model of affect mainly involving the prefrontal cortex, but also the amygdala, the hypothalamus, and the anterior cingulate cortex, as seen in Figure 22.

In the same article, he describes how different brain regions are responsible for different kinds of emotion:

> Equally problematic is the widespread view that the limbic system is the neural basis for all emotions. A rich body of evidence tells us that this is just not the

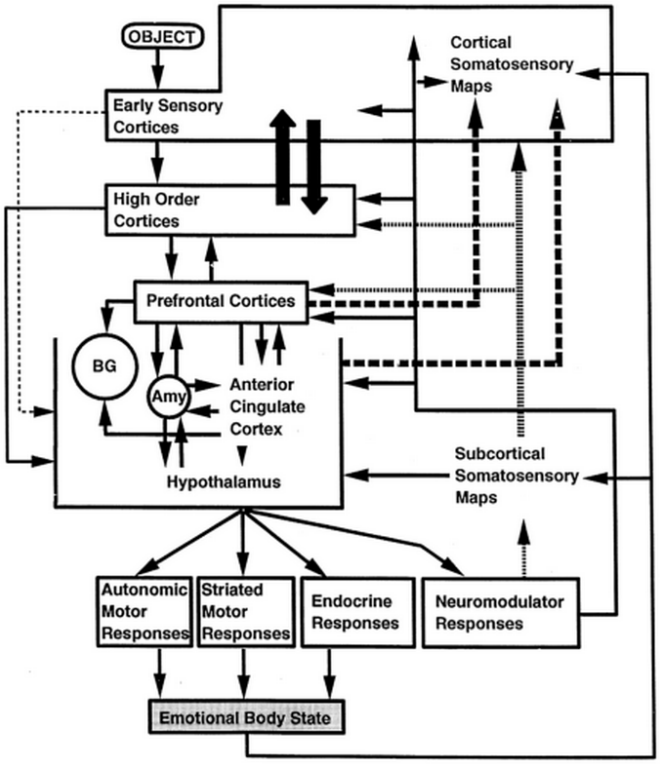


Figure 22: Neurological structure of affect, according to Damasio [20].

> case. Both within and around the limbic system, circuitry connection varied neural sites supports the operation of different emotion. For instance, work on aversive conditioning in rodents has shown that the amygdala is certainly involved in negative emotions such as fear [10,6]. *Work in humans, on the other hand, has not only confirmed the amygdala's involvement in negative emotions such as fear and anger, but also shown that the amygdala is not involved in the processing of positive emotions such as happiness, or negative emotions such as disgust.* [emphasis mine]

The last sentence of that quotation is especially revealing: it states that the neurological distinction is not simply one between positive and negative, or one between approach- or avoidance-related emotions, but that each emotion has its own profile of neurological activity and involves its own peculiar set of brain structures.

These facts make it quite clear that emotions are not simply homogeneous phenomena, being induced by a single system in the brain; rather, they are different in character and in the neural structures they involve.

Structure of affect The system depicted in Figure 21 left several parts unspecified: the sensory processor Pr, the emotion selectors $S_1, \ldots, S_n$ and the messages sent by the chosen emotions into the message space. In the following paragraphs, I will flesh out that model in greater detail, building principally on the work of Sander, Grandjean and Scherer [70]. Sander and colleagues partitioned the emotion process into four stages, as shown in Figure 23. The first is *relevance*, which functions as a filter and detects the intrinsic pleasantness and the level of

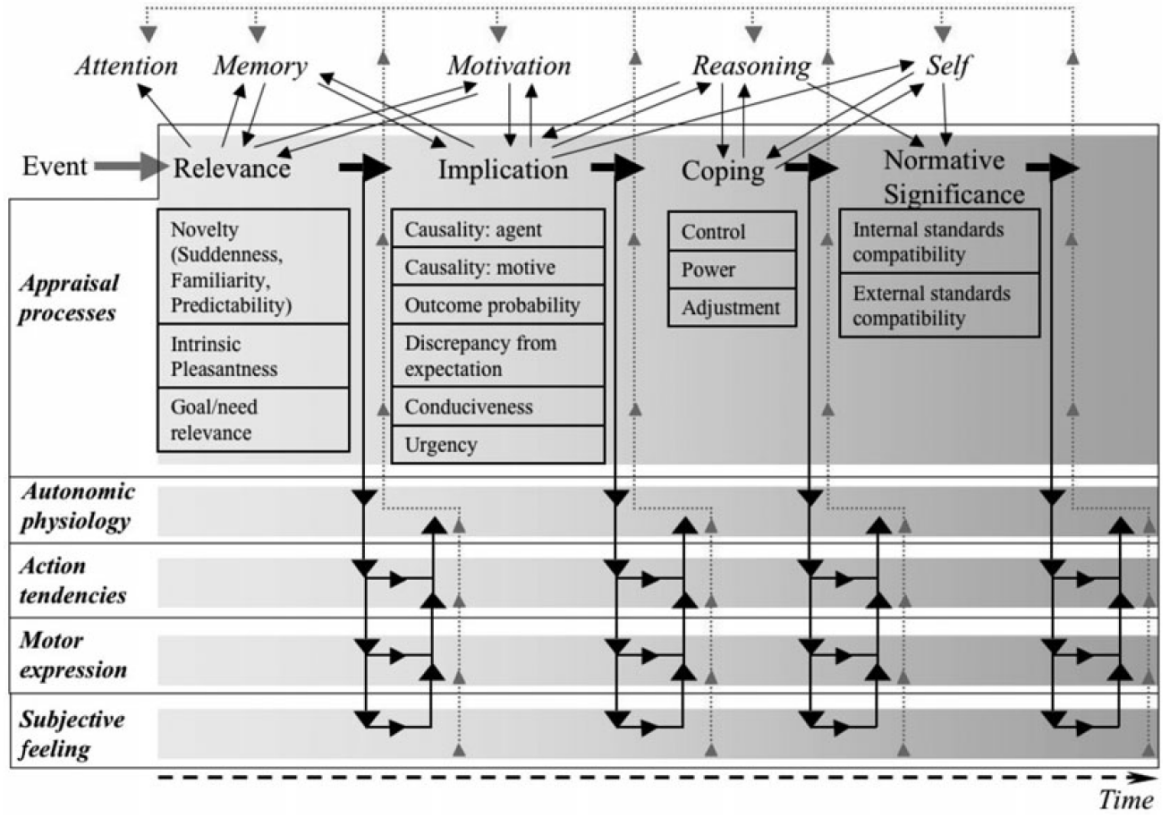


Figure 23: The four-stage emotion process according to Sander et al, consisting of relevance, implication, coping and normative significance.

(emotional) attention that a stimulus demands. The processes of this stage, roughly speaking, correspond to the work of the sensory processor Pr. The second stage is *implication*, where reasoning becomes engaged in order to determine the cause, likely outcome, and urgency of the perceived facts. At this stage, emotions like joy, anger, contentment, disgust, etc. are evoked, together with approach- and avoidance-related behaviours — this corresponds to the emotion selectors $S_1, \ldots, S_n$. Deliberate strategies come only in the next stage: *coping*. In it, reasoning and planning become fully engaged. The fourth stage is *normative significance* and deals, in essence, with moral concerns, both internal and those of other agents.

Sander et al. give a good, detailed account of the interactions of affect with other systems, although I would argue that theirs is unduly suggestive of a simple *pipeline*, rather than a mesh of systems into which the affective ones are embedded. In addition, it does not address the interactions with perception, memory, and reasoning. Based on the evidence discussed above, I shall now present a more horizontal view and construct a model of the hypothesized emotional subsystems and their interactions with other parts of the brain. Since no established vocabulary seems to exist in this specific are I shall first introduce a number of terms.

Definition 21 (Evocative system). *An evocative system is a subsystem in the brain responsible for evoking consciously experienced affect within an agent based on internal or external stimuli.*

Various such evocative systems can be imagined. For the purposes of this thesis, we will work with the following rough categorization:

Pre-social emotions. Certain behavioural mechanisms can be observed in non-social as well as social animals. The fight-or-flight instinct, for example, is nearly universal, as is the inclination to seek out food, shelter, and other resources. "Instinct" is indeed a more appropriate term in the case of most species, rather than "emotion", which connotes a certain richness of experience. Nonetheless, we can clearly see that, in more intelligent, social animals, emotions like anger, fear, and joy, have grown out of just these instincts. Hence the term "pre-social emotions": while emotion itself is quite possibly inherently social, certain emotions are rooted in instincts which are not, and an emotional animal would feel them even if it were the only one of its kind in an environment.

Social emotions. A by far richer subset of emotions are the social ones. Indeed, social situations are the ones where affect can and must truly shine: the presence of other individuals, or of the entire tribe, demand a variety of affect relating to the appraisal of the agents, sympathy/antipathy, respect/contempt, the appraisal of oneself, showing dominance or submission, influencing other group members, taking action as a group, judging the behaviour of agents against norms, etc. It is also in social emotions in which it even makes sense to *show* emotion: facial expressions and gestures provide the signalling and mechanism needed for group coherence and coordinated action.

We can identify several subsystems in the category of social emotion:

1. Reflective judgement about oneself in relation to the group or to abstract norms, primarily pride and shame [80], but possibly also jealousy and humiliation (which, in contrast to shame, is attributed to external causes) [29];
2. other-related judgement which determines whether to feel sympathy or antipathy, compassion, respect or contempt, trust or distrust for other individuals;
3. normative judgement, which determines whether others or oneself is acting in accordance with instinctive or cultural norms.

Other classifications are also possible. Haidt [35], for example, identifies those that are other-condemning (disgust, contempt), self-conscious (shame, embarrassment), other-suffering (compassion), other-praising (gratitude, awe). The picture is immensely complex and the neurological structure is presently not known. For the purposes of this thesis, we will therefore content ourselves with only this roughest of outlines.

Aesthetic emotions. This type of emotion is perhaps the least studied in neuroscience and AI. It is certainly the most subtle and the least "utilitarian" type — as such, it is philosophers, rather than AI researchers, who study it. For instance, Jenefer Robinson, in *Deeper Than Reason: Emotion and Its Role in Literature, Music, and Art* [68], writes about the affective appraisal of artwork as an unconscious process which partly reproduces the emotions of its creator. In this, she builds upon and modifies Collingwood's 1983 *The Principles of Art* [16, 45]. Since aesthetics are not the focus of this work, I shall leave it at this mention. A more thorough exploration would be interesting future work, however.

The emotions just listed can all be found in the more extensive taxonomies, chiefly among them in Ortony's OCC model [62]. The taxonomies, however, tend to neglect the underlying

neurology and the chronology of the development of these systems. Ortony's classification specifically is persuasive up to a point, but, despite it being fine-grained, one is left wondering about the underlying structure: which emotions are caused by the same brain regions, what structure, if any, do two given emotions share, to what degree is the classification scheme isomorphic to the actual neurology? This is an active area of research and while these questions are interesting, we have to leave them largely open for now.

The evoked feelings tie into and directly influence the agent's actions. This includes conscious, deliberate ones, such as avoiding an unsympathetic person, but also subconscious ones and those that are purely internal, such as the focusing one's attention to an important topic. These actions all fall under the umbrella term of *executive system*:

Definition 22 (Executive system). *An executive system is a subsystem in the brain which makes decisions about the behaviour of an agent's mind or muscular system.*

This definition leaves open what exactly a decision is. In principle, any neural activity in a part of the brain could be seen as a decision of sorts, since it influences neural activity in other parts. While we do perceive certain processes as deliberate and others as automatic, this is simply what our introspection tells us and does not reflect the underlying reality; (conscious) decision-making is as mechanical as any other process in the brain, the chief difference being that we are aware of the workings of that process and perceive the control it exerts over cognition as coming from us.[17]

Nonetheless, there are properties by which we can identify executive systems in the brain: on a sufficiently high level of abstraction, we can see that certain components are receptive to control signals. Certain other components — these are the executive systems — have as their *chief purpose* the sending of such control signals. The former accomplish some conceptually small task and essentially serve as building blocks. The latter structure the work and assemble the small building blocks into compound actions. See Section 6.2, where planner and world simulator work in tandem, with the world simulator bearing the workload and the planner having control.

We can now distinguish certain kinds of action. While those performed with the "body" (i.e. the skeletomuscular system) are the most visible ones, we, as shown, also make decisions regarding the contents of our minds — we decide *what to think about*. We then add the distinction between consciously and subconsciously made actions and get the following four categories of executive system:

Subconscious motor control: instinctive reaction, such as the jerking away from pain, jumping when startled, and turning towards interesting visual stimuli.

Conscious motor control: deliberate, planned action which the agent experiences as a choice.

[17] I should add that we are not even aware of the entirety of our decision-making. This is especially apparent when we are asked to make trivial or random choices. A person who is asked to press a left or a right button, for example, will choose one, seemingly at random, but will not be able to explain why one button was chosen over another. Moreover, there is evidence that the choice is made before the person *knows* that a choice was made: Soon et al. [72] instructed subjects to press a button and to record when they thought they made the decision to do so. Brain scanning revealed spikes in the activity of the lateral and medial frontopolar cortices and the posterior cingulate contex *before* the subjects claimed their decisions were made. In effect, they only became aware of their supposedly free decisions after they had already been made. From their conscious perspective, the decision simply "popped into their heads".

Subconscious mental control: involuntary but consciously experienced changes to the mind-state of an agent which are perceived as activity rather than mere feeling. This includes like obsessing over an issue, manias, fantasies insofar as involuntary, etc.

Conscious mental control: deliberate mental changes of an agent. This includes the making of decisions, the deliberate focusing of attention, deliberate planning, deliberate strategy selection, and so forth.

I stress that these are *categories* of systems, not systems themselves. We control our minds and our bodies in a variety of ways and there is no evidence that there is some sort of master control system anywhere in the brain responsible for these tasks. The planner from Section 6.2 only controls one other component — and it might very well be that it does not even exist in the brain as one compact component. It might be that a variety of smaller systems are tugging and vying for control and balanced against each other in such a way that the illusion of dedicated planning component is created.

6.4 Interaction between affect and world simulation

Section 6.2 outlined what could be called *deliberate action* in the from of a planner-world-simulator loop. Section 6.3 described the structure and components of affect. These systems are of course not isolated from each other; emotional states influence both the planner's chosen heuristics and the world simulator's creation of worlds. In addition, attention, also influenced by affect, controls the allocation of cognitive resources. We now explore these relationships in further detail.

Planning as search In the AI literature, search algorithms are of great importance. In this context, we can view the loop between planner and world simulator as a greedy search: the planner chooses the nodes which are to be expanded and sends them to the world simulator. It, in turn, performs the expansion by simulating the appropriate worlds. These simulated worlds are sent back to the planner for evaluation regarding desirability (i.e. cost). This presents an obvious problem: since greedy search is not complete, our planner-world-simulator loop can't be complete either. In fact, the situation is worse — greedy search computes the cost of all candidates for expansion and chooses the cheapest, whereas our planner, being heuristic, might not consider certain nodes at all.

This might seem damning, but we must also consider the interaction with attention and memory. First, planned steps are committed to memory and thus, we gain access to past costs. An agent does not plan blindly, but can recall how long its plans are and what costs past planned steps entail. Given this information, we can turn the greedy algorithm into an A* search, with the qualification that the planner might not consider certain nodes. The mechanism of attention can further be used to enhance the search: if planning along a certain path takes too long, the agent might decide to abandon it altogether and start afresh with a different strategy. This failure too is stored in memory and can influence the planner in the new planning process by making the proposing of steps of the previously pursued path unlikely.

7 Proposed architecture

8 Implementation

Having laid the theoretical framework, we come to the practical part of this thesis — a proof-of-concept implementation of multiple affective agents interacting with each other. This section contains the following parts: (1) the world in which act, (2) the architecture of these agents, and (3) the evolutionary changes in the agent pool from generation to generation.

The goal is the creation of toy AI that semi-realistically mimics animal intelligence, the operative word being "mimic". As Sloman [77] pointed out, naming a variable ANGER or LOVE does not give a program some qualitative experience. No; our much more modest goal is to EMULATE the behaviours that are associated with certain mental states — and to show how such emotional states, interacting with reasoning, can help an agent thrive in its environment. These programs will really only be soulless automata, employed to illustrate a point about living beings with brains, acting with incomplete information.

8.1 World

The choice of the world profoundly affects the implementation of the agent — its knowledge base, mechanism of perception and interaction, the required complexity of the implementation. On the one hand, the world should be simple enough to permit a reasonably small and effective agent which does not have to solve hard AI problems (like human-level sight) to deal with what we, in this context, might call details — but on the other hand, the world should be sufficiently complex to allow agents to distinguish themselves. This is especially true in the case of an affective agent whose actions should be visibly influenced in rich and subtle ways by its emotional state. I shall first lay out the design goals and then evaluate three possible worlds for agents.

Design goals The two most important criteria for prospective worlds are richness of interaction and world complexity, in that order. As said, an evaluation of affective agents is only possible if they can interact with their environment and other entities in a sufficiently complex way to allow agents with different emotional profiles to be distinguished from each other. Mechanisms of problem-solving like STRIPS [26], A* [36], ASP [49], forward-/backward-planning, etc. have been explored in the context of structurally simple worlds, generally those representable through propositional logic, cost-functions, decision tress, and the like. While these are useful, they are less appropriate in an affective scenario for the following two reasons:

1. they are geared towards finding provably optimal solutions to computationally expensive but conceptually simple problems like planning or game-playing and
2. they rely heavily on hand-crafted ontologies and domain knowledge on the part of the human programmer.

For a world to be useful to us and to avoid these pitfalls, it should be in some sense realistic: it should permit a large number of different kinds of interactions, and it should not provide agents in it with perfect knowledge about its rules.

I admit that I here stand in opposition with Marvin Minsky, who famously recommended the use of idealized micro-worlds to study artificial intelligence, in that same vein in which physics makes use of ideal, frictionless planes and perfect spheres. His argument certainly has merit, but I believe that emotion is too complex a phenomenon for such abstract scenarios. In too simple a setting, pure reasoning not only easily outperforms emotional behaviour, but avenues for exhibiting emotional behaviour are scarce to begin with. For this reason, I propose that, in this context, rich interactions should take precedence over idealization and simplicity.

It is of course still desirable to minimize complexity as far as possible. An overwhelmingly complex world has two obvious drawbacks: first, the required complexity of an agent scales with the complexity of the world; second, the more complex the world, the harder it is to reason about it. If there are a hundred ways to succeed, for instance, agent performance becomes quite difficult to measure.

8.1.1 Blocks world

Blocks worlds are the simplest type of abstract world, and many variations exist. They all have in common a number of shapes placed on top of each other in a 2-dimensional world. An agent can pick up and move a shape if and only if there are no other shapes on top of it (and if it is not already holding one). The goal generally consists of achieving some desired configuration of shapes, such as building or piecewise transporting a tower, or collecting all red triangles.

Micro-worlds like blocks worlds have been extensively studied. In this, their simplicity has been their great advantage — that very simplicity is a serious problem for us, however. Affect is inherently a subtle and social phenomenon; it is not clear how it could be believably exhibited in such an abstract and simple world. The very same properties which expedite their theoretical study make them useless for our evaluation.

8.1.2 Wumpus world

The traditional Wumpus world, as described in Russell and Norvig's *Artificial Intelligence: A Modern Approach* [69, p. 236], is a grid-based, 4x4 cave world with one agent, one monster — the Wumpus — and gold placed in random rooms. The agent starts at position $\langle 1, 1 \rangle$ and can move forward or turn 90° to the left or right. If it enters a room with a pit or a live Wumpus, it dies; its goal is to find and collect the gold and then move back to position $\langle 1, 1 \rangle$ to climb out of the cave. In addition, it has one arrow which he can fire straight ahead to defend against the Wumpus. The agent has only the following local information [69, p. 237]:

- In the square containing the Wumpus and in the directly (not diagonally) adjacent squares, the agent will perceive a *Stench*.
- In the squares directly adjacent to a pit, the agent will perceive a *Breeze*.
- In the square where the gold is, the agent will perceive a *Glitter*.
- When an agent walks into a wall, it will perceive a *Bump*.
- When the Wumpus is killed, it emits a woeful *Scream* that can be perceived anywhere in the cave.

This type of world is simple enough to be amenable to rule-based reasoning, although it can contain ambiguous situations where the agent does not have enough information to make the best choice. For example, if an agent moves to position $\langle p_x, p_y \rangle$ and experiences a breeze, 1, 2, or 3 adjacent rooms may contain pits, but it cannot be safely determined which ones these are. Thus, occasionally, the agent must choose between climbing out without the gold and risking death by pit or Wumpus.

For our purposes, this is a bit too simple, however. Caution/bravery is the only axis along which agents can be differentiated and although various complex behaviours — such as trying one dangerous cell, then going back and trying another one to explore the world — are possible, these do not have a clear relation to emotional states.

Let us, while staying true to the spirit of the original, now define a type of extended Wumpus world $\mathcal{W}_{\text{ext}}$ that allows more varied interaction between agent an environment.

Definition 23 ($\mathcal{W}_{\text{ext}}$-type world). *Let* $\mathtt{T_v}$, $\mathtt{T_e}$, $\mathtt{T_g}$ *be arbitrary types. Further, let* G *be a directed graph with vertex labels of type* $\mathtt{T_v}$ *and edge labels of type* $\mathtt{T_e}$*, and let* gl *be an object of type* $\mathtt{T_g}$*. Then the tuple* $\langle G, \text{gl} \rangle$ *is a* $\mathcal{W}_{\text{ext}}$*-type world (with type parameters* $\mathtt{T_v}$, $\mathtt{T_e}$, $\mathtt{T_g}$*). We call* G *the* world frame *and* gl *the* world data*.*

We can interpret each vertex v in the graph as a room with attached data $l(v)$ of type $\mathtt{T_v}$, and each edge e as an unidirectional connection between rooms with attached data (such as path costs) $l(e)$ of type $\mathtt{T_e}$. gl is the global world data. Next, we specify some properties of the world frame:

Definition 24 (World properties). *Let* $W = \langle G, \text{gl} \rangle$ *be a* $\mathcal{W}_{\text{ext}}$*-world. We say that* W *has property* X *iff it fulfils the first-order sentence corresponding to* X*. The following properties are of importance:*

Property name	***FO sentence***
Reflexive	$[\forall v \in V(G)]\,(v, v) \in E(G)$
Non-Euclidean	$[\forall$ *pairwise distinct* $v_1, v_2, v_3 \in V(G)]$ $\{(v_1, v_2), (v_1, v_3)\} \subseteq E(G) \Rightarrow (v_2, v_3) \notin E(G)$
Symmetrical	$[\forall v_1, v_2 \in V(G)]\,(v_1, v_2) \in E(G) \Rightarrow (v_2, v_1) \in E(G)$
Connected	$[\forall v_1, v_2 \in V(G)]$ *there exists a path from* v_1 *to* v_2 *in* G
n-dimensionally embeddable	*there exists an infinite, n-dimensional grid* S *such that* $G \subseteq S$[18]

[18] Formally, G and S must fulfil the following conditions:

1. $V(G) \subseteq V(S)$,
2. $E(G) \subseteq E(S) \cup \{(v, v) \mid (v, v) \in E(G)\}$,
3. S's drawing, embedded into $\mathbb{R}^n$, forms a regular tiling, and
4. $(v_1, v_2) \in E(S)$ iff the Euclidean distance between v_1 and v_2 in $\mathbb{R}^n$ is 1.

The first four properties speak for themselves. As for the fifth — Figure 24 shows an example of a 2-dimensionally embeddable frame. A frame G is n-dimensionally embeddable if it is a fragment of an infinite, n-dimensional, square grid of nodes S, plus any loops G might have. When we embed this infinite grid S into $\mathbb{R}^n$ through an embedding, every edge corresponds to a vector of length 1 along exactly one dimension. If we additionally take G's loops to correspond to null-vectors, this induces an *edge direction function* and a *position function*:

Definition 25 (Edge direction and position). *Let $W = \langle G, \mathrm{gl}\rangle$ be an n-dimensionally embeddable world (for some n) and ϵ an embedding of W into $\mathbb{R}^n$. Then we have an* edge direction function

$$\Delta_n^\epsilon : E(G) \to \{0, x_1^+, x_1^-, x_2^+, x_2^-, \ldots, x_n^+, x_n^-\}$$

with 0 corresponding to a loop and x_i^+/x_i^- corresponding to forward/backward movement in the ith dimension. We also have a position function

$$\pi^\epsilon :: V(G) \to \mathbb{R}^n,$$

with $pi^\epsilon(v) = r$ indicating that under ϵ, v was mapped to position r in $\mathbb{R}^n$. When the number of dimensions and the embedding are obvious, we omit n and ϵ. Since π^ϵ is injective by definition, an inverse $(\pi^\epsilon)^{-1}$ also exists. Through it, we define the indexing function *of W:*

$$[] : n\text{-dimensionally embeddable world} \to \mathbb{R}^n \to \texttt{Maybe}\ V(G)$$
$$W[p] \equiv \begin{cases} \texttt{Just}((\pi^\epsilon)^{-1}\ p) & \text{if } (\pi^\epsilon)^{-1}\ p \text{ is defined} \\ \texttt{Nothing} & \text{otherwise} \end{cases}$$

We will give agents access to Δ_n^ϵ and π^ϵ (or simply Δ and π) to allow them to determine their position and direction in the world. Providing such information might seem problematic, but we thereby free ourselves from having to insert things like landmarks, wind currents, stars, and other navigational aids into the world. Given that navigation is not the focus of this thesis, this seems an appropriate simplification. Using the above properties, we can specify a subtype of $\mathcal{W}_{\text{ext}}$-type worlds:

Definition 26 (2D grid world). *Let $W = \langle G, \mathrm{gl}\rangle$ be a $\mathcal{W}_{\text{ext}}$-type world (with type variables $\mathtt{T_v}, \mathtt{T_e}, \mathtt{T_g}$). If W is reflexive, connected, and 2-dimensionally embeddable W is a* 2D grid world. *Every 2D grid world has an associated function $\Delta_2 : E(G) \to \{0, x_1^+, x_1^-, x_2^+, x_2^-\}$ and a position function $\pi : V(G) \to \mathbb{R}^2$.*

Note: every n-dimensionally embeddable world is also symmetrical and non-Euclidean.

Grid worlds, as we have seen, are potentially infinite, n-dimensional grids, although their cells need not form a square or cube. Their shape can be irregular in that some rooms and connections may be missing, as long as the shape as a whole stays connected.

2D grid worlds are representationally the same as $\mathcal{W}_{\text{ext}}$-type worlds; they just have some structural invariants on their frames. If we additionally specialize the representation through the type parameters $\mathtt{T_v}$, $\mathtt{T_e}$, and $\mathtt{T_g}$, we arrive at the type of world which will serve as the environment for our agents: the "jungle world" $\mathcal{W}_{\text{jun}}$.

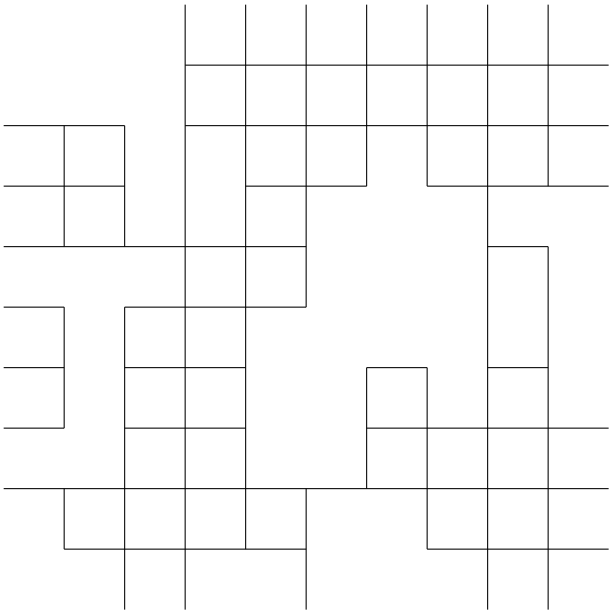


Figure 24: A segment of 2-dimensionally embeddable world. The vertices are its rooms, the edges are the connections between the rooms.

Definition 27 ($\mathcal{W}_{\text{jun}}$). *Let* $\mathtt{T_v}$, $\mathtt{T_e}$, $\mathtt{T_g}$ *be the following tuples:*

$$
\begin{aligned}
\mathtt{TV}_{\text{jun}} &= \langle \mathtt{agents} :: [\mathtt{Agent}],\\
&\quad\ \mathtt{wumpus} :: [\mathtt{Wumpus}],\\
&\quad\ \mathtt{plants} :: \mathtt{Maybe\ Plant},\\
&\quad\ \mathtt{stench} :: \mathbb{R},\\
&\quad\ \mathtt{breeze} :: \mathbb{R},\\
&\quad\ \mathtt{pit} :: \mathbb{B},\\
&\quad\ \mathtt{gold} :: \mathbb{N}\rangle\\
\\
\mathtt{TE}_{\text{jun}} &= \langle \mathtt{danger} :: \mathbb{R},\\
&\quad\ \mathtt{fatigue} :: \mathbb{R}\rangle\\
\\
\mathtt{Temp} &= \mathtt{Freezing} + \mathtt{Cold} + \mathtt{Temperate} + \mathtt{Warm} + \mathtt{Hot}\\
\\
\mathtt{TG}_{\text{jun}} &= \langle \mathtt{time} :: \mathbb{N},\\
&\quad\ \mathtt{temperature} :: \mathtt{Temp}\rangle
\end{aligned}
$$

`Agent` *and* `Wumpus` *are the following records:*

$$
\begin{aligned}
\texttt{Item} &= \texttt{Gold} + \texttt{Fruit} + \texttt{Meat} \\
\\
\texttt{Agent} &= \langle \texttt{name} :: \texttt{String}, \\
&\quad \texttt{direction} :: \texttt{X}_1^+ + \texttt{X}_1^- + \texttt{X}_2^+ + \texttt{X}_2^-, \\
&\quad \texttt{health} :: \mathbb{R}, \\
&\quad \texttt{fatigue} :: \mathbb{R}, \\
&\quad \texttt{inventory} :: [\langle \texttt{Item}, \mathbb{N} \rangle], \\
&\quad \texttt{state} :: \texttt{S} \rangle \\
\\
\texttt{Wumpus} &= \langle \texttt{health} :: \mathbb{R}, \\
&\quad \texttt{fatigue} :: \mathbb{R} \rangle
\end{aligned}
$$

The last component of `Agent`, `state` :: `S`, *is the internal state of agents which we will discuss later.*

Let gl *also be a value of type* $\texttt{TG}_{\text{jun}}$ *and let* G *be any 2D grid world with node labels of type* $\texttt{TV}_{\text{jun}}$ *and edge labels of type* $\texttt{TE}_{\text{jun}}$. *Then,* $\langle G, \text{gl} \rangle$ *is a* $\mathcal{W}_{\text{jun}}$*-type jungle world.*

The intuitive meaning of $\mathcal{W}_{\text{jun}}$is this: the two-dimensional grid world is inhabited by multiple agents and wumpuses, where the former act according to their agent function and the latter act mechanically. In addition, each cell in the world may have a plant, gold, or a deadly pit on it. Agents and wumpuses move in the world by traversing edges which have associated fatigue and danger levels, representing easy and difficult paths. Local information is available to expedite navigation: stench (emanating from wumpuses) and breeze (emanating from pits). Finally, the temperature and the time dictate global environmental conditions.

Although the field names are suggestive of the way in which a $\mathcal{W}_{\text{jun}}$-type world works, the type, strictly speaking, only specifies the data and frame properties. We can employ such worlds in any sort of scenario, with whatever semantics we wish. Notwithstanding, our implementation will use a straightforward *standard semantics*, that have the world work in the manner of a simple ecosystem in which predators hunt for prey and compete with each other. The wumpuses fulfil the role of carnivorous predators which roam the world, hunting and attempting to kill agents on sight. Agents, in turn, are hunter-gatherer omnivores who can sustain themselves either through eating plants, killing Wumpuses for their meat, or by acquiring resources from other agents. They may carry meat or fruits in their inventory, or gold, which has no intrinsic use, but which may be used as an exchange medium, provided that multiple agents have the mental ability to facilitate bartering. The term "jungle world" reflects the uncertainty under which its actors must act. They only have access to quite limited local environmental information, and they possess no communication protocol upon which they could base their cooperation. Analogously to real-world situations, agents must rely on simple gestures to infer the intentions of their peers, and they cannot know whether they are misunderstanding these, or whether they are being deceived. The aim of this mechanism is to allow the experimentations with things like social adaptation, prejudice, and trust. The goal of simulating affective agents in such a world is to see which behavioural profiles are successful, how they develop over multiple generations, and how they engage each other.

Definition 28 (Semantics and runs of $\mathcal{W}_{\text{jun}}$-type worlds). *Let* φ *be a function of type* $\mathcal{W}_{\text{jun}} \to \mathcal{W}_{\text{jun}}$. φ *is called a* semantics of $\mathcal{W}_{\text{jun}}$-type worlds. *Now let* W *be a* $\mathcal{W}_{\text{jun}}$*-type world. The iterated application of* φ *to* W, *given by the list* $[W, \varphi\ W, \varphi^2\ W, \varphi^3\ W, \dots]$, *is called a* run

of W (with semantics φ). $\varphi^n\ W$ *is referred to as the* state of W's simulation at time n (with semantics φ).

Definition 29 (Standard semantics of $\mathcal{W}_{\text{jun}}$-type worlds). *The standard semantics for $\mathcal{W}_{\text{jun}}$-type worlds are given by the function* $\text{sem} :: \mathcal{W}_{\text{jun}} \to \mathcal{W}_{\text{jun}}$. sem *is defined as*

$$\text{sem}\ \langle G, \text{gl}\rangle = \langle G', \text{gl}'\rangle,$$

where $\langle G', \text{gl}'\rangle$ is identical to $\langle G, \text{gl}\rangle$, except for the following changes:

Environment *For all $v \in V(G)$, perform the following:*

Wumpus. *If there is a Wumpus in a cell w at ≤ 3 distance from v, increase v's stench by*

$$\frac{\log_3(3 - ||v, w||) - \texttt{stench}\ l(v)}{2}$$

If there is no Wumpus within distance ≤ 3, decrease v's stench by $\frac{1}{3}$, to a minimum of 0.

Plant. *If there is a plant on v and it has no fruit, increase its growth by $\frac{1}{10}$. If its growth thereby reaches 1, add a fruit to the plant and reset the growth to 0.*

Pit *If there is a pit in a cell w at a distance ≤ 3 from v, set the breeze to*

$$\log_3(3 - ||v, w||)$$

Global data. *The* daylight function *is defined as*

$$\texttt{cycle}\ t = \begin{cases} 0 & \text{if} \quad 20 \leq |n - 25| \\ 1 & \text{if} \quad 15 \leq |n - 25| < 20 \\ 2 & \text{if} \quad 10 \leq |n - 25| < 15 \\ 3 & \text{if} \quad 5 \leq |n - 25| < 10 \\ 4 & \text{if} \quad |n - 25| < 5 \end{cases}$$

The new global data gl' *are given by*

$$\begin{aligned} \text{gl}' \quad = \quad & \langle \texttt{time}\ \text{gl} + 1 \bmod 50, \\ & (\texttt{cycle} \circ \texttt{temperature})\ \text{gl}'\rangle \end{aligned}$$

$$\texttt{temperature}\ t \quad = \quad \begin{cases} \texttt{Freezing} & \text{if}\ \ \texttt{light}(t) = 0 \\ \texttt{Cold} & \text{if}\ \ \texttt{light}(t) = 1 \\ \texttt{Temperate} & \text{if}\ \ \texttt{light}(t) = 2 \\ \texttt{Warm} & \text{if}\ \ \texttt{light}(t) = 3 \\ \texttt{Hot} & \text{if}\ \ \texttt{light}(t) = 4 \end{cases}$$

Wumpus behaviour. *Every Wumpus has three behaviors:*

1. *If the Wumpus is adjacent to a player, it performs the* attack *action on that player.*

2. *If there is a player reachable with at most* (`light` $\circ$ `time`) gl *edges, move along the edge that minimizes the distance to that player (in* $\mathbb{R}^2$*). If there are multiple players, choose one at random as target. This target choice remains until the player is no longer within range.*
3. *If there is no player within range, move in a random direction with probability*

$$0.2 \times (1 + (\texttt{light} \circ \texttt{temperature})\ \text{gl}).$$

Whenever a Wumpus travels along an edge e *with* $\Delta\ e \neq 0$*, apply* 0.1 *damage with probability* `danger` e*.*

Agent behaviour. *Agents always act after Wumpuses and, depending on their implementation, may choose one of the following actions:*

move — *move along an edge* e*. If* $\Delta\ e = 0$*, restore* 0.1 *of the agent's fatigue, otherwise reduce it by* $0.05 \times$ `fatigue` e*. Additionally (if* $\Delta\ e \neq 0$*), apply* 0.1 *damage with probability* `danger` e*.*

If an agent's fatigue is below 0.2*, it cannot choose this action.*

rotate — *the agent changes the direction into which it is facing to a value in* $x_1^+, x_1^-, x_2^+, x_2^-$*.*

attack — *move along an edge* e *to attack an agent or wumpus.*

give — *give an item* i *from the agent's inventory to another agent* a*.*

gather — *if there is a plant with a fruit on the agent's cell, take the fruit and put it in the agent's inventory.*

butcher — *if there is a dead Wumpus on the agent's cell, remove it and add an item of meat to the agent's inventory.*

collect — *if there is* n *gold on the player's cell, take an amount* m *(*$1 \leq m \leq n$*) of it an put it into the agent's inventory.*

eat — *eat a meat- or fruit-item* i *from the agent's inventory. Restore* 0.5 *health.*

gesture — *expresses a gesture in the form of a string* s*. All other agents on the same cell receive* s*.*

It ought to be said that the formulae and constants used in the above definitions are, fundamentally, judgement calls and that there is no theoretical reason for choosing them over others. Nonetheless, we can given them an intuitive meaning:

Environment.

Wumpus. Wumpuses carry around them a wafting stench, the strength of which drops off exponentially for three cells. It does not follow the wumpus with delay, however: when the wumpus comes into the vicinity of a cell, it begins to strengthen, and when it moves away, it begins to weaken, until it reaches its target value.

Plant. Fruits grow periodically on plants, although a plant can only bear one fruit at a time.

Pit. The breeze coming from pits works via the same mechanism as the stench of wumpuses, but as pits are immobile, the strength of a breeze does not change with time.

Global data. A day is segmented into 50 periods, where a time of 25 represents midday, and 0/50 represents midnight. The temperature is a function of the daytime, with midday being the hottest and midnight being the coldest.

Wumpus behaviour. Wumpuses are day-active and roam around randomly. At night, they are likely to sit still. When they sight an agent (depending on light conditions), they will invariably attempt to close the distance and attack.

Agent behaviour. Agents are free to do choose any action they wish. They may move around, attack wumpuses and other agents, gather items (fruit from plants, meat from dead wumpuses, gold lying around), consume food, give items to other agents, or communicate with them. They are limited by their health, which is depleted by traveling along dangerous paths and by fights, and by fatigue. They must thus periodically eat and rest to keep both up.

8.2 Agents

The agents of our simulation are composed of two parts: their minds and their bodies. Their minds constitute their sensors and agents functions; their bodies, make up their actuators, although they are more than that. An agent's body can be damaged and healed, perceived by others, and it can hold items. As such, the bodies are actually part of the world. From the point of view of the agent's mind, they are external objects they happen to control.

8.2.1 Body and percepts

As we saw in Definitions 27 and 29, agents (1) have a body composed of a name, health, fatigue, and an inventory of items they carry, and (2) can execute one of a fixed set of actions at each step. These data function in the obvious way: the name is publicly available information other agents can use for identification, the agent is killed when its health drops to zero, fatigue determines the effectiveness when attacking and prevents movement when low, and the inventory is used to store items which the agent can use for itself or give away to others.

What we are missing is the description of the agent's percepts in the world. As in the original Wumpus world, an agent can perceive everything on its cell:

1. the list other agents,
2. the list of (dead) Wumpuses,
3. the plant, if present,
4. the breeze,
5. the stench, and
6. the amount of gold.

In addition to this local information, the agent also has a sense of sight, modelled via an approximately $\frac{\pi}{2}$ radians cone, the length of which depends on daylight. Formally:

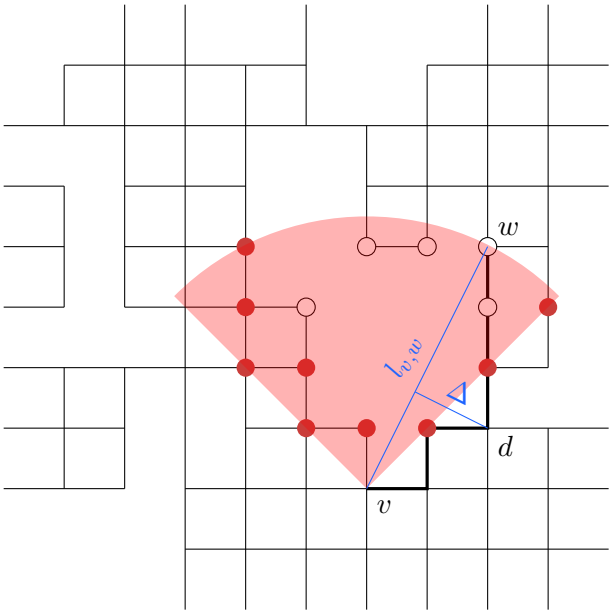


Figure 25: Sight cone of an agent at $\texttt{light}(t) = 2$. The cone with width $\frac{\pi}{4}$ signifies that agent's range of vision. Red vertices in it are perceived; the hollow black ones are not because they are blocked by holes in the world. The line $l_{v,w}$ illustrates why the vertex w is not visible from v: the shortest path from v to w runs through d, but the distance Δ between d and the closest point along $l_{v,w}$ is larger than $\frac{\sqrt{2}}{2}$.

Definition 30 (Sight cone). *Let $W = \langle G, \mathrm{gl}\rangle$ be a 2D grid world. Let an agent be on vertex $v \in V(G)$, facing into direction d. Let further l_d be the line starting at v and extending infinitely into direction d, and $l_{v,w}$ be the line from v to w. Then, any other vertex $w \in V(G)$ falls into the agent's sight cone exactly if:*

1. *the angle between $l_{v,w}$ and l_d is $\leq \frac{\pi}{4}$,*
2. *$||v, w|| \leq 1.5 \times (((\texttt{light} \circ \texttt{time})\ \mathrm{gl}) + 1)$, and*
3. *there is a path $v_1, v_2, \ldots, v_n$ from v to w in G such that the distance between v_i and the closest point along $l_{v,w}$ is $\leq \frac{\sqrt{2}}{2}$ $(1 \leq i \leq n)$.*

Criterion one restricts the sight cone to $\frac{\pi}{4}$ radians; criterion two limits its length based on light conditions; criterion three demands rough line-of-sight, saying that the path in G may never deviate more than one cell from the line in $\mathbb{R}^2$. Figure 25 illustrates the working of this mechanism. If vertex w falls into an agent's sight cone, it perceives $\pi(w)$ and the following data:

1. the list of agents on w,
2. the list of Wumpuses,
3. the plant, it present,
4. the pit, if present, and

5. the amount of gold.

The breeze and the stench, being non-visual, are not thus perceived. As we can see from criterion two in Definition 30 and the formulae for breeze and stench in Definition 29, sight reaches farther, but is directed. The non-visual cues can tell an agent that it's in danger, but not from which direction that danger comes. If that agent consequently fails to look around, it may be attacked or wander into a pit.

8.2.2 Cognition

Our goal is the design of a reasonably effective type of agent which will be able to navigate $\mathcal{W}_{\text{jun}}$-type worlds. EFFECTIVENESS, in this context, simply means SURVIVAL. There is no explicit performance measure; certain agents will survive, while others will not.

Relevant aspects. We have already seen what sort of data an agent must process if it is to perform well. It must first know or learn the geography of the world, of which it is a priori unaware. It must also be able to seek out resources in the form of plant and gold; it must be able to deal with the threat posed by Wumpuses, either by avoiding or defeating them. Most importantly, it must be able to interact with other agents in ways which avoid adverse behaviour towards the agent itself, and it must find ways to solicit beneficial behaviour from them.

In order to achieve this, three things are indispensable: (1) memory, (2) utility maximisation. If we don't impose a memory limit, it is quite easy to store everything that happens to an agent. In essence, such memories will be fragments of past states of the external which can be used to make decisions. Utility maximisation is the far more complex task: the agent must either perform individual fact synthesis or inherit certain predilections from its parents and must therewith exhibit useful behaviour. The fact synthesis can be done in a number of ways — machine learning, reasoning, heuristic —, but we must remember that knowledge, by itself, does not determine behaviour. In addition, the agent must possess a decision-making component which uses gained knowledge in whatever way it sees fit. Knowledge thus *allows* efficient decisions to be made, but fundamentally, an agent is free to disregard any fact it wants.

Design goals and dynamism. As with the world, the cognitive structure of agents is a compromise between intricacy and simplicity. Ideally, we would make every aspect of an agent's thinking dynamic and malleable under evolution, but this would necessitate a prohibitively high implementation effort. Instead, based on the description of FILTERS in Section 4, I make the following compromise: the *evocation* of an emotion will be dynamic and different from agent to agent; the effects of emotions, however, will always be the same. As an example, different agents might become angry in different situations and to different degrees, but the behavioural consequences that follow from the emotion of anger will always be the same.

Cognitive components. Based on the considerations outlines in earlier sections, I propose that agents be made out of the following six components:

Pre-social behaviour control (PSBC). This controls aspects of an agents which, in principle, can work without other agents: fear, happiness, anger. These emotions are evoked in social situations, but in principle, they would be useful in a world without any other agents present.

Social judgement system (SJS). Analogous to the PSBC, the SJS controls an agent's appraisal of other agents and thereby influences its decision-making.

Belief generation (BG). In essence, the imagination of an agent. belief generation allows reasoning and the internal simulation of parts of the world.

Attention-control (AC). Attention-control is the recognition of certain real or imaginary percepts as *important*, leading to the allocation of cognitive resources to them.

Decision-making (DM) . The executive component of an agent which includes both internal decision-making (IDM) — *what to think* — and external decision-making (EDM) — *what to do*.

Memory. Memory is a log of imagined and real events that happened to an agent. This log is utilized chiefly by the BG with the goal of providing world data.

As a side remark: these components make no claim to encompass the kind of intelligence humans have. In particular, there are no aesthetics, pure abstract reasoning, purely self-centered emotions like grief or remorse, etc. Providing such mechanisms is, however, not the goal; we merely wish to make the agents complex enough to successfully navigate the world. For this purpose, a simple, social, and animalistic sort of intelligence suffices, one that, in complexity, is actually below even that of wolves an dogs.

Pre-social behaviour control. The PSBC is responsible for evoking the kinds of emotions that non-social animals have, in some form. Here "pre-social" does not refer to the current use of this system, but to its evolutionary history: past animals were able to experience anger and fear, or something analogous to anger and fear, before they developed social lives. The fight-or-flight instinct, and deciding when to engage in activity and when to abstain from it are necessary for survival even in solitary animals. A social system, of course, does impact these emotions, but a social system is not necessary for them to be there. We categorize the experienced emotions according to approach/avoidance and positivity/negativity, based on the work of Davidson and Irwin [21]. The four combinations are:

1. Anger, which is approach-related and negative. Anger causes attack-actions against Wumpuses and other agents, and gesture-actions with parameters the agent deems to be aggressive.

2. Fear, which is avoidance-related and negative. Fear, causes flight and gesture-actions which the agent deems submissive.

3. Enthusiasm, which is approach-related and positive. Enthusiasm has a wide range of effects: gesture-actions with positive contents, fatigue-inducing activity, and the gathering and sharing of resources with other agents.

4. Contentment, which is avoidance-related and positive. Contentment is concerned primarily with the conservation of resources. Its chief effect is thus the is the cessation of action.

Figure 26 illustrates these four emotions. Each of them can be evoked with a *valence* $\in [-1, 1]$. Higher-valence emotions exert a greater pressure on decision-making and attention control. The figure, with its two axes, should not mislead us into thinking that emotions are just vectors in $\mathbb{R}^2$. There is, for example, weak/intense enthusiasm and there is weak/intense contentment, but there is no emotion halfway between contentment and enthusiasm. It *is* possible that a stimulus should activate two emotions at once, but those will actually be two emotions, not one "hybrid" emotion.

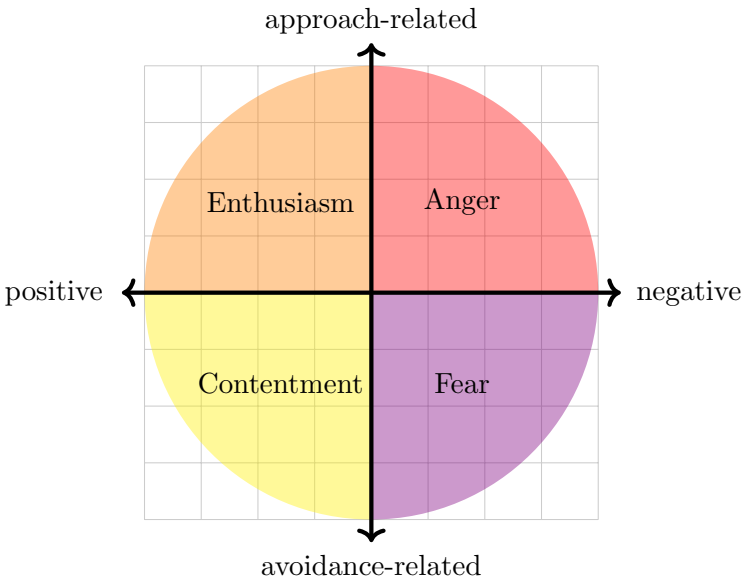


Figure 26: Emotions evoked by the PSBC.The left half contains the positive emotions of enthusiasm and contentment, whereas the right contains the negative emotions of anger and fear. Enthusiasm and anger are both approach-related, causing action, whereas contentment and fear are approach-related, causing flight or abstinence from action.

In terms of implementation, this is realized via the system we saw in Figure 21, Section 6: each of the four emotions has a SELECTOR reads percepts and the HORMONE STORAGE, using them to decide whether and how intensely to activate and emotion. Emotions, once active, flow into the HORMONE STORAGE and send messages into the global message space. The scheme is illustrated in Figure 27: the filters of each emotions continually check the agent's percepts for relevant data. If a filter is activated, the message is passed the component's interpreter (to determine its urgency), which hands it to the processor. It then puts the message "I feel emotion E with intensity π_E" into the message space. In this, it takes the HORMONE STORAGE into account: experiencing an emotion increases the corresponding hormone level, and, conversely, a high hormone level intensifies the emotion. Formally, the hormone storage is defined thus:

Definition 31 (Hormone storage). *Let $E_1, \ldots, E_n$ be the names of emotions. A hormone storage for the emotions $E_1, \ldots, E_n$ is the ADT $\mathtt{H}_n = \langle h_1 :: \mathbb{R}, \ldots, h_n :: \mathbb{R} \rangle$, together with the functions* `receive` $:: \mathtt{H}_n \to \mathbb{N} \to \mathbb{R} \to \mathtt{H}_n$ *and* `tick` $:: \mathtt{H}_n \to \mathtt{H}_n$*, given by*

$$\begin{aligned}
\texttt{receive}\ h\ e\ \pi &= 2\pi * \log_2(1 - \texttt{get}\ h\ e) \\
\\
\texttt{tick}\ h &= \langle \texttt{get}_1\ h - 2\log(\texttt{get}_1\ h), \\
&\quad \ldots \\
&\quad \texttt{get}_n\ h - 2\log(\texttt{get}_n\ h)\rangle.
\end{aligned}$$

The idea is that hormone level increases and decreases logarithmically: whenever an agent receives a message about an experienced emotion e with intensity π, the corresponding level h_e is increased proportionally to π and the logarithm of the current level. The levels also decay at each time step, returning the agent to a neutral state over time if no stimuli are experienced.

One objection might be that, while an agent can experience conflicting emotions if multiple components are activated, different emotions cannot directly interact with each other. This is true; however, they can interact indirectly, through the message space: if a component C_X reads the message of component C_Y as a percept and, because of that, begins sending negatively-valenced messages, the emotion X is effectively shutting down the emotion Y — even though the process is controlled by C_Y. I of course do not claim that this mechanism accurately reflects nature, that being an empirical question, but at the very least, it gives us a way to implement both ambivalence and quick mood changes.

Social judgement system. The SJS has the task of recognizing other agents as such and guiding friendly and hostile interactions with them. Real social behaviour is very complex and involves not only other agents as individuals, but the group itself. In the minds of tribal animals, the group exists as an entity unto itself, with its own will and mood. Our agents will not implement this group dynamic. Instead, they will appraise each other agent individually, according to three criteria:

Sympathy. This determines how much an agent likes another one. Liked agents will receive friendly gestures, assistance in the form of food and protection from Wumpuses and hostile agents, disliked agents will be denied these benefits, receive hostile gestures and, if the dislike is sufficient, might be attacked.

Trust. The trustworthiness of another agent influences the likelihood of two things: (1) the propensity to give out items in the hope of future reciprocation and (2) the aggressiveness if protection from the agent is present. The reasoning here is that the agent will be emboldened by the presence of trusted allies.

Competence. Competence judges the capabilities of another agent. Competent agents will be respected, incompetent ones will be held in contempt. Similarly to trust, the presence of friendly, competent agents emboldens the agent.

Sympathy is the primary axis of judgement, since it determines whether others are seen as friends or enemies. Trust and competence are secondary and help an agents ascertain the quality of its allies an enemies. The three criteria are illustrated in Figure 28. Figures 29 and 30 list the different antagonistic and sympathetic judgements.

The evocative mechanism is structurally similar to that of the PSBC, as we saw in Figure 27, but with two crucial differences: first, social judgements are always attached to agents; second, the SJS models each of these three categories as a single emotions which can be positive or

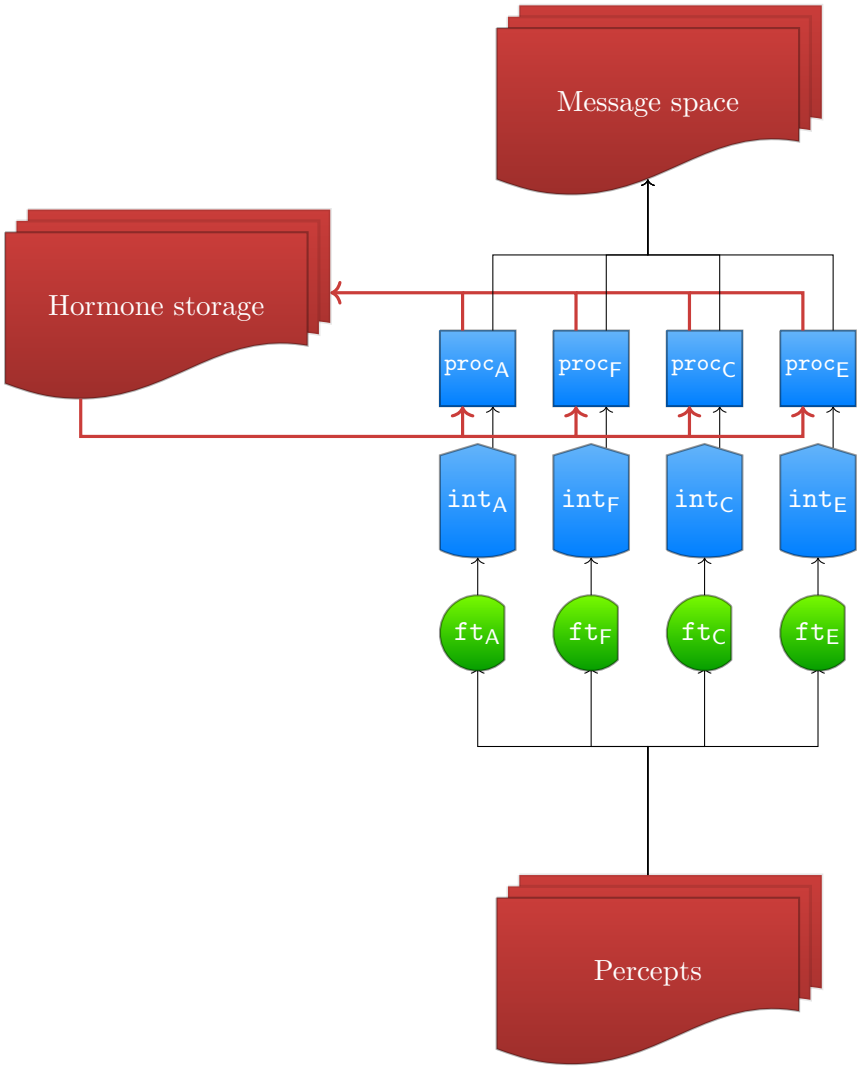


Figure 27: The PSBC as a collection of a hormone storage and four emotion selectors. The neural components shown are *anger* (A), *contentment* (C), *enthusiasm* (E), and *fear* (F).

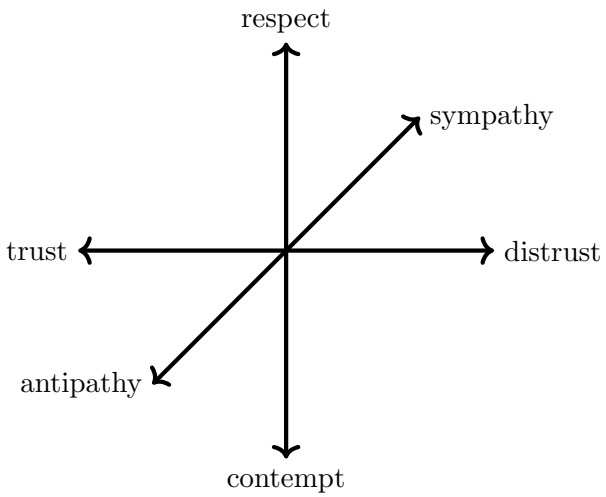


Figure 28: Emotions evoked by the SJS. The primary is axis is sympathy/antipathy, since it distinguishes friend from foe. Trust/distrust judges the loyalty/honor of another agent, whereas respect/contempt judges its competence.

negative — that is, an agent cannot simultaneously experience trust and distrust for another one, but only a single emotion (trust). We see this system illustrated in Figure 31, which shows it to be largely analogous to the PSBC in Figure 27.

This system is a quite gross simplification of the real world. In reality, one does not simply possess an emotion called "trust", the value of which can go from -1 to +1, but rather, one possesses different kinds of trust, and trust with respect to different matters. One can, for instance, have a gut feeling that someone is generally unreliable and shady, but one can, through reason, come to the conclusion that this person will keep his word in a certain situation in which punishment would ensue. This does given an assurance of loyalty, but does not change the fundamentally negative appraisal of that person. Similarly, one can have judgements which seem to lie halfway between reason and emotion, and which pertain only to certain situations, such as trusting someone with money, with completing a task on time, or with one's child.

Our agents will not implement the nuances of such concepts directly, but they won't completely neglect them either. As we will see in the sections about memory and the relationship between components, the two affective systems will make use of memory and imagination in order to deliver situational judgements. To stay with our example about trust: if an agent imagines a situation in which another was loyal, or remembers such an event, it will be able to judge that other agent as trustworthy (in that situation.)

Belief generation. World-simulation is probably the most complex identifiable part of human cognition. Our version of it, therefore, will only be a minimalistic reproduction. Instead of constructing a system which is able to extensively utilize learning and construct its own ontologies and ways of thinking from scratch, we will use a fixed ontology, and an existing reasoning tool called DLVhex [61]. DLVhex is a solver for answer-set programming. Answer-sets are a specific kind of solutions to (disjunctive) logic programs, which are reasoning schemes that take both the presence and the absence of knowledge into account. Extensive descriptions can be found in [50] and [5]. I will give the compressed definitions:

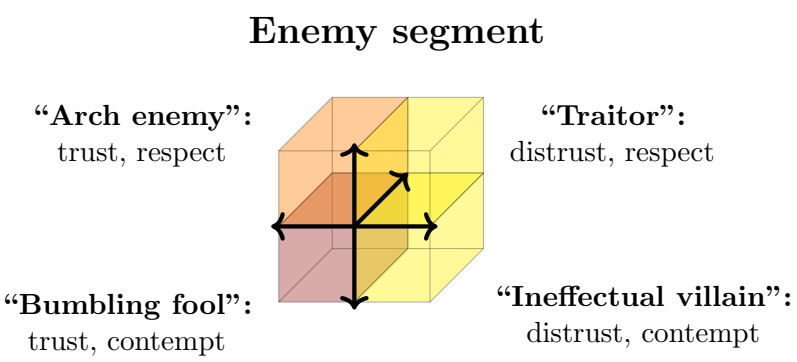


Figure 29: The four antipathic judgements. Enemies can be respected or held in contempt, and deemed trustworthy or untrustworhty. Respect for an enemy implies that an agent holds it to be competent. Trust implies that an agent knows its enemy to be basically honourable.

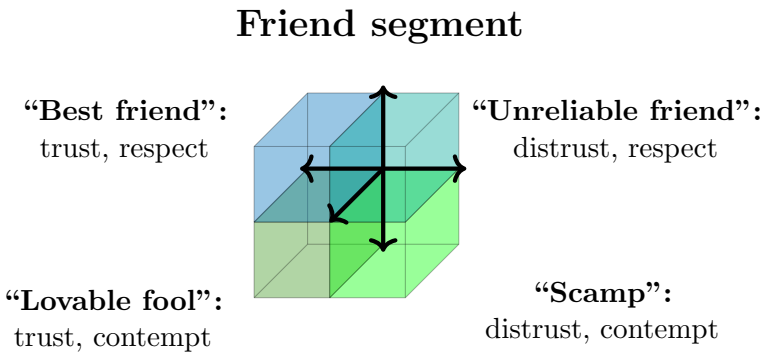


Figure 30: The four sympathic judgements. Friends, like enemies, respected or held in contempt, and deemed trustworthy or untrustworthy. Distrust renders the sympathetic judgement tentative, since the agent cannot be sure of the assistance of an untrustworthy friend. Contempt works similarly, but doubts a friend's ability, rather than loyalty.

$[\forall a : \text{agent}]\ a \overset{\text{induce}}{\Longrightarrow}$

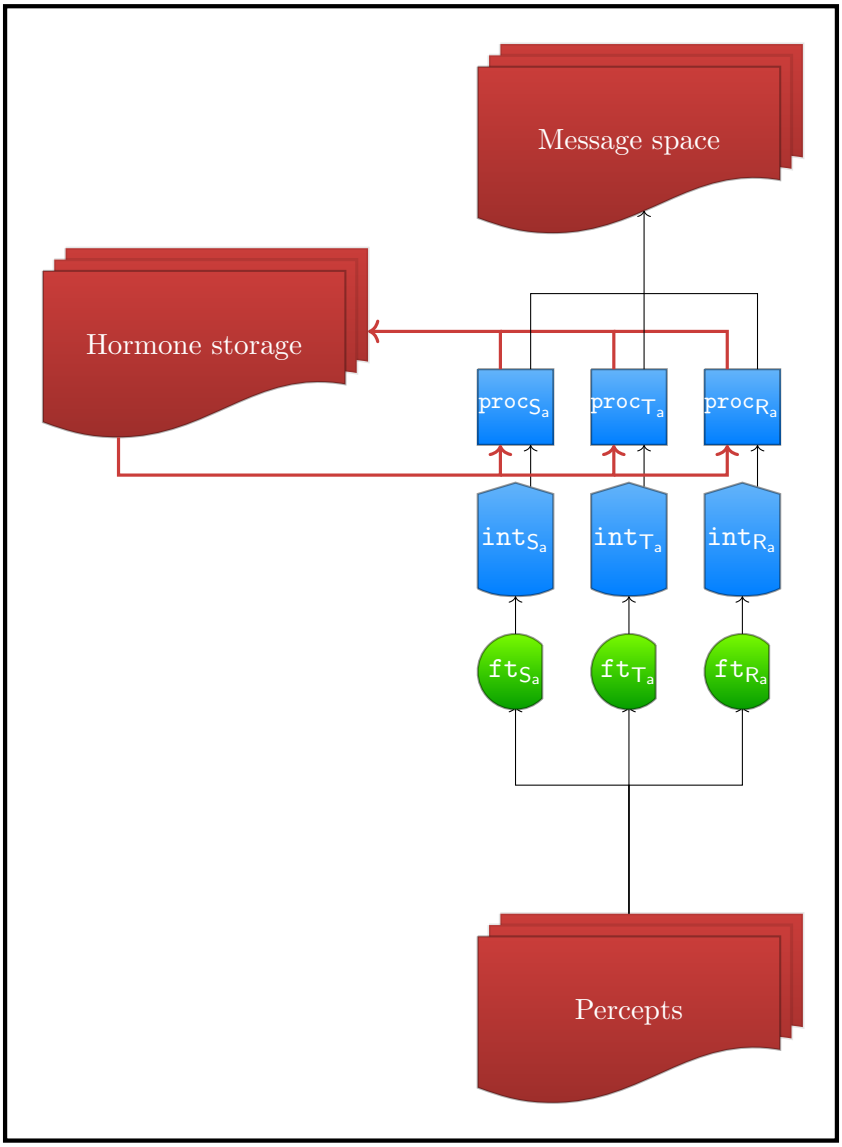


Figure 31: The SJS *for one other agent* as a collection of a hormone storage and three emotion selectors. The neural components shown are *sympathy* (S), *trust* (T), and *respect* (R). Every agent which is encountered has its own SJS instance.

Definition 32 (Syntax: Disjunctive logic program). *A finite set of rules* Π *is a disjunctive logic program exactly if every rule* r *is of the following form:*

$$\text{head}(r) \leftarrow \text{body}^+(r), \text{body}^-(r)$$

where $\text{head}(r) = \mathtt{P_1}, \ldots, \mathtt{P_h}$, $\text{body}^+(r) = \mathtt{P_{h+1}}, \ldots, \mathtt{P_k}$, *and* $\text{body}^-(r) = \mathtt{not}\ \mathtt{P_{k+1}}, \ldots, \mathtt{not}\ \mathtt{P_m}$, *with* P_i *being a first-order predicate* $(1 \leq i \leq m)$. *For all predicates* $\mathtt{P_i}$, *both* $\mathtt{P_i}$ *and its* strong negation $\neg\mathtt{P_i}$ *are* literals. *If a literal only has constant arguments, it is called a* ground literal. *If* $\text{body}^-(r) = \emptyset$ *for all* $r \in \Pi$, *we call* Π *as* positive *logic program.*

$\text{head}(r)$ *is called the head of the rule* r, $\text{body}^+(r), \text{body}^-(r)$ *its body. The predicates in* $\text{body}^+(r)$ *are asserted and the predicates in* $\text{body}^-(r)$ *are default-negated*[19]. *If the body of a rule is empty, the rule is called a* fact*; if its head is empty, the rule is called a* constraint.

Intuitively, the semantics of logic programs are that, whenever all the asserted predicates in the body of a rule are true and we do *not* know any of the default-negated predicates to be true, one of the predicates in the head of the rule must be true. We first define models for positive logic programs:

Definition 33 (Sets closed under logic programs). *Let* Π *be a positive logic program and let* X *be a set of atoms.* X *is* closed under Π *if, for all* $r \in \Pi$, $\text{body}^+(r) \subseteq X$ *implies that there exists a* $\mathtt{P} \in \text{head}(r)$ *such that* $\mathtt{P} \in X$.

The $\subseteq$*-minimal set of atoms closed under* Π *is called* $\mathsf{Cn}(\Pi)$.

Answer sets for general disjunctive logic programs are defined through the Gelfond-Lifschitz reduct:

Definition 34 (Reduct of a logic program). *Let* Π *be a (disjunctive) logic program and let* X *be a set of atoms. Then the reduct* Π^X *is*

$$\{\text{head}(r) \leftarrow \text{body}^+(r) \mid r \in \Pi \text{ and } \text{body}^-(r) \cap X = \emptyset\}$$

Definition 35 (Answer set). *A set of atoms* X *is an* answer set *of a disjunctive logic program exactly if* $\mathsf{Cn}(\Pi^X) = X$.

X being answer set thus means that it is a consistent, subset-minimal, and stable model of Π, i.e. one that, for any rule $r \in \Pi$, contains a predicate in $\text{head}(r)$ exactly if it contains all predicates in $\text{body}^+(r)$ and none of the predicates in $\text{body}^-(r)$.

Answer-sets are thus the smallest sets of knowledge that we can derive, starting from the facts of a logic program. A variety of ASP tools exist besides DLVhex, e.g. CLASP [64], GnT [60], and Platypus [65]. The main advantage of DLVhex over them is that it provides and implementation of the ACTHEX language[27], which extends logic programs with bidirectional access to the external world. Whenever DLVhexfinds a so annotated input atom, it queries an external information source; whenever it finds an action atom, it performs some specified IO action against. We will use this mechanism to implement the planner/world simulator loop between the ASP solver and the agent function.

To determine its next action in the real world, the agent function calls DLVhex with a proposed action. It, in turn, will begin simulating (i.e. imagining) the consequences of that

[19] That is, it is asserted that they cannot be derived, but not necessarily that their negation can be derived.

action. The solver will know the rules of the world, but since it won't possess any information about its state, it will query the agent's memory and perception via external atoms. After deducing the world's future state based on this information, it will call an *action atom*, sending the world state back to the agent function for evaluation, which then either proposes another step or terminates the planning process.

Memory. While real-world memory is complex phenomenon, for expediency's sake, our agents will possess only a simple analogue to it, in the form of a private database of world data which they perceived in the past. These data are of type $\texttt{TV}_{\text{jun}}$, $\texttt{TE}_{\text{jun}}$, which were given in Definition 27. We store them on a per-cell and per-edge basis and update them whenever we perceive them anew. This gives rise to the following definition:

Definition 36. *Let $\langle G, \text{gl}\rangle$ be a $\mathcal{W}_{\text{jun}}$-type world and let A be an agent. The memory database of A is given has type*

$$\texttt{Memory} = \texttt{Memory}\ (\texttt{Map}\ V(G)\ \texttt{TV}_{\text{jun}})\ (\texttt{Map}\ E(G)\ \texttt{TE}_{\text{jun}})$$

and is accessed through the functions

$$\begin{aligned}
\texttt{store} &:: \mathcal{W}_{\text{jun}} \to \texttt{Memory} \to \texttt{Memory}\\
\texttt{retrieve}_{\texttt{V}} &:: \texttt{Memory} \to \texttt{V(G)} \to \texttt{Maybe}\ \texttt{TV}_{\text{jun}}\\
\texttt{retrieve}_{\texttt{E}} &:: \texttt{Memory} \to \texttt{E(G)} \to \texttt{Maybe}\ \texttt{TE}_{\text{jun}}\\
\texttt{retreive}_{\texttt{A}} &:: \texttt{Memory} \to \texttt{String} \to [\texttt{Action}]
\end{aligned}$$

where `store` *updates the database with the edges and cells of the world which the agent can perceive.* $\texttt{receive}_{\texttt{V}}$ *and* $\texttt{receive}_{\texttt{E}}$ *return the values associated with a given cell or edge, provided that data for the given cell/edge is stored.* $\texttt{receive}_{\texttt{A}}$ *takes the name of an agent B as a key and returns the list of actions A has observed B perform.*

We should note that a number of justified criticisms can be levelled against it. For one, it does not deal with uncertain data that are either old, or were not inaccurately perceived. It only records past states, but not sequences of events. Most direly, it doesn't provide enough information to contextualise the actions of other agents. Suppose that A observes B attacking C. A may infer that B is powerful or aggressive, but the list of actions returned by $\texttt{retrieve}_{\texttt{A}}$ are not enough to construct a theory of mind for either B or C. A thus does not know whether B's attack was revenge, opportunism, betrayal, or plain hostility.

Nonetheless, this database is valuable for the agent. $\texttt{retrieve}_{\texttt{V}}$ and $\texttt{retrieve}_{\texttt{E}}$ can provide actionable information about the static aspects of the world such as the location of plants or dangerous paths. Even the information about its changing aspects, such as the location of wumpuses, will be reasonably good, since wumpuses, in the absence of agents, tend to stay in place over time[20].

Attention-control.

[20] They approximately perform 2-dimensional random walks over time. The expectation $\mathsf{E}(W)$ of a random walk is the null-vector $\langle 0, 0, \ldots, 0\rangle$. Given that they have a disproportionately high chance of just staying in place, depending on light conditions, their positions are even quite densely clustered around that.

Attention-control serves as an interrupt/prioritisation mechanism for the decision-making process. Whenever it is triggered by some percept that it deems important, it sends a message to the DM, directing it to abort its current activity and to deal with the stimulus instead. We thereby have a method of prioritising tasks that require immediate attention and to cut short, colloquially speaking, aimless deliberation.

For our agents, paying attention means to focus on a cell and a specific kind of stimulus. That is, attention-control deal with messages of the type

$$\texttt{Stimulus} = \texttt{Gold} + \texttt{Fruit} + \texttt{Meat} + \texttt{Agent String} + \texttt{Wumpus} + \texttt{Pit}$$

The AC component then emits messages of type `Stimulus` and is modelled via the functions

$$\begin{aligned} \texttt{attention} &:: \mathcal{W}_{\text{jun}} \to \texttt{Maybe}\ \langle V(G), \texttt{Stimulus}\rangle \\ \texttt{ac} &:: \texttt{s} \to \mathcal{W}_{\text{jun}} \to \texttt{s} \end{aligned}$$

where `s` is the internal state of the agent. `ac` is just a wrapper around `attention` which puts the latter's message into the agent's message space.

Three things are of note here. First, `attention` need not emit any message at all. It will indeed be a very common case that the agent perceives nothing of particular interest. During these periods, it will be inactive, leaving the agent's behaviour to the other components. As a result, decision-making will be largely dominated by the BG and the affective components, which do judge everything which the agent perceives.

Second, at most one stimulus may marked as important. This reflects the intuition of the AC being a sort of "gut reaction" which does not deliberate, but keenly and quickly focuses the agent on a single object.

Third, the `Stimulus` type does not unambiguously identify an object in the world. `Gold`, for instance, does not say how much gold there is in cell; `Wumpus` does not identify any particular wumpus; `Pit` may not mean that there is a pit on the indicated cell; only that the agent should watch out for one. This, too, is intended — the AC should only serve pointers, not perform in-depth analysis of the situation. It is the task of the belief generator and the DM to examine specific parts of the environment and to deliberate about courses of action. This mimics the cognitive mechanism in real animals quite closely: in humans, attention is a subtle and immediate feeling; one that simply causes us to notice certain objects or events in the world. No conscious thinking, or even the experience of emotions, need accompany this experience. Such more complex cognitive processes might be induced, but they are clearly separate from the mere act of noticing.

Decision-making. Decision-making is split into two components: external decision-making, which controls the agent's actions, and internal decision-making, which controls the BG and thus drives the planning process. Aside from the difference in target, both are modelled via a function

$$\texttt{choice} :: \texttt{s} \to \mathcal{W}_{\text{jun}} \to \langle \texttt{Action}, \texttt{s}\rangle$$

where `s` is the internal state of the agent. `choice` evaluates a world and the previous state of the agent and then gives a new internal state, together with a proposed action from the list in Enumeration 29 — that is, one of the following: move, rotate, attack, give, gather, butcher, collect, eat, gesture. The actions proposed by the internal decision-making component (IDM) are instructions for the BG and, in principle, can go on as long as the agent wishes to deliberate.

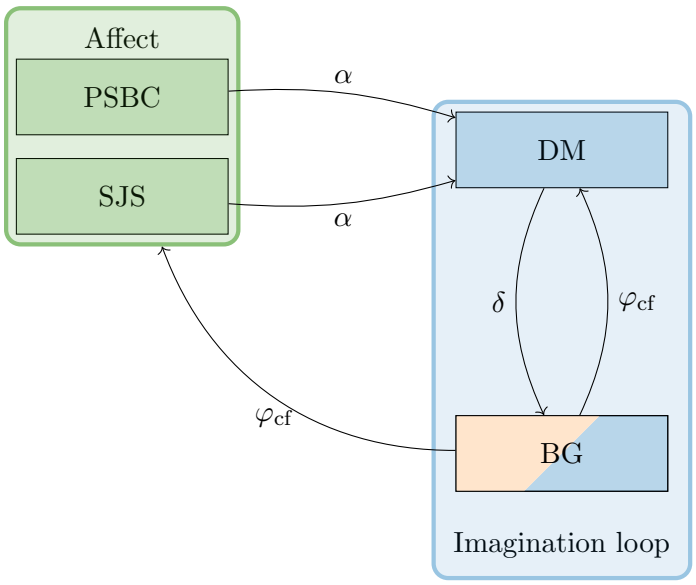


Figure 32: Imagination loop, influenced by affect. The edge labels denote the type of signal: α for affective information, δ for control signals, φ_{cf} for imagined perceptions.

Those of the external decision-maker are translated into the real world. Once the simulation program receives the return value of an agent's EDM, that agent is done, so to speak: it has performed its action for that tick no longer consulted until the next one.

TODO: Detailed description of affective/ attentional influences upon `choice`.

From the affective subsystem, the BG, and the DM, we can put together the imagination simulator loop described in previous chapters. In Figure 32, we see the DM issuing commands to the BG, which generates data for the affective systems and the IDM. The affective systems treat this data as if it were coming from the external world and generate affective messages, which are consumed by the IDM and inform its commands to the BG.

Relationship between components. Having defined the agent's components, we now put them together into a functioning whole. The core of the agent's cognition will consist of the interplay between perception and decision-making, with the affective systems and attention control influencing the latter. We see the system sketched in Figure 33.

At the very heart of the agent lies its decision-making component, which controls both the agent's actions and its belief generation. The DM and the BG form the *imagination loop* ι which develops plans by exploring the likely consequences of certain actions. In that capacity, the DM evaluate the BG's simulated worlds for desirability and chooses which imagined steps to take next. These evaluations are influenced by the second group of systems: the affective ones. The PSBC and SJS process perceptions and feed their resultant emotional states into the DM. Through this coloring of its decision-making, agents with different emotional dispositions will act and think differently from each other.

The third part of the system is the attention-control, which also evaluates real and imagined emotions and outputs its data for the DM's usage. It's only purpose is to alert the agent to important or shocking information which demands immediate action. Its alerts cause the DM

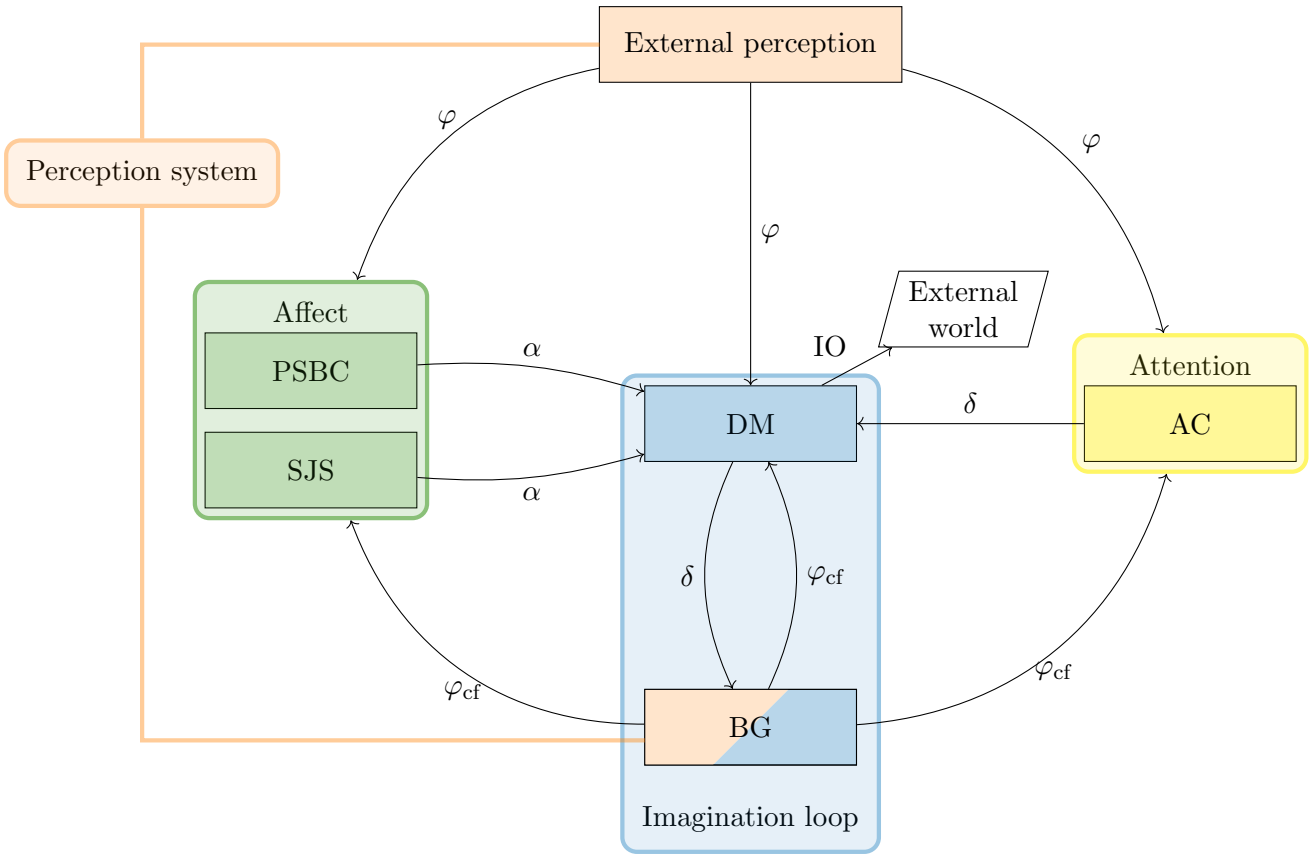


Figure 33: High-level view of the cognitive structure of agents, with groups of systems shown in colored boxes. The PSBC and the SJS comprise the affective group; the DM and BG the imagination loop responsible for planning. The BG, with External perception, makes up the perception system. As in the previous figure, the edge labels show which kind of message the system sends out: α for affective information, φ and φ_{cf} for (imagined) perceptions, δ for control signals. IO corresponds to real actions in the world.

to cease its current course of action and re-plan based on the piece of information deemed important.

We now have all the pieces we need to create the agent function `agent`:

Definition 37 (Agent function). *Let S be a type. Then an agent function with internal data of type S has type*

$$\texttt{agent} \;::\; \mathcal{W}_{\text{jun}} \to \texttt{S} \to \langle \texttt{S}, \texttt{Action} \rangle.$$

That is, `agent` *takes the current world andits current internal state, and returns its new internal state, together with the action it wishes to perform.* `agent` *is defined as:*

$$
\begin{aligned}
&\texttt{agent}\ w = \ \texttt{fromJust} \\
&\qquad \circ\ \texttt{getActionMessage} \\
&\qquad \circ\ \texttt{head} \\
&\qquad \circ\ \texttt{dropWhile noResult} \\
&\qquad \circ\ \texttt{iterate loop} \\
&\qquad \circ\ \texttt{perception}\ w \\
&\quad \text{where} \\
&\qquad \texttt{perception} :: \mathcal{W}_{\text{jun}} \to \texttt{S} \to \texttt{S} \\
&\qquad \texttt{psbc}, \texttt{sjs}, \texttt{ac}, \texttt{dm}, \texttt{bg} :: \texttt{S} \to \texttt{S} \\
&\qquad \texttt{loop} :: \texttt{S} \to \texttt{S} \\
&\qquad \texttt{loop} = \texttt{bg} \circ \texttt{dm} \circ \texttt{ac} \circ \texttt{sjs} \circ \texttt{psbc} \\
&\qquad \texttt{getActionMessage} :: \texttt{S} \to \texttt{Maybe Action} \\
&\qquad \texttt{noResult} = \texttt{not} \circ \texttt{isJust} \circ \texttt{getActionMessage} \\
&\qquad \texttt{iterate} :: (\texttt{a} \to \texttt{a}) \to \texttt{a} \to [\texttt{a}] \\
&\qquad \texttt{iterate}\ f\ x = x : \texttt{iterate}(f x, x) \\
&\qquad \texttt{dropWhile} :: (\texttt{a} \to \texttt{Bool}) \to [\texttt{a}] \to [\texttt{a}] \\
&\qquad \texttt{dropWhile}\ p\ xs = \begin{cases} h : \texttt{dropWhile}\ p\ t & \text{if } xs = (h : t) \wedge (p\ h = \texttt{True}) \\ xs & \text{otherwise} \end{cases}
\end{aligned}
$$

Note: ∘ is function concatenation; the list of functions in `agent` *has to be read bottom-to-top.*

This agent function can now be plugged into the standard semantics we defined back in Definition 29: the function sem calls every agent with the world and its last internal state and receives a new internal agent state, together with the action the agent has chosen to perform at that time step.

9 Evaluation

Results of the implementation.

References

[1] James S. Albus. A reference model architecture for intelligent systems design. In *An Introduction to Intelligent and Autonomous Control*, pages 27–56. Kluwer Academic Publishers, 1993.

[2] James S. Albus. The engineering of mind. In *Information Sciences*, pages 23–32. John Wiley & Sons, Inc, 1996.

[3] Jorge Amory and Patrik Vuilleumier. *The Cambridge Handbook of Human Affective Neuroscience*. Cambridge University Press, 2013.

[4] František Baluška and Stefano Mancuso. Deep evolutionary origins of neurobiology: Turning the essence of "neural"; upside-down. *Communicative & Integrative Biology*, 2(1):60–65, 2009.

[5] Chitta Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, New York, NY, USA, 2003.

[6] Henk Barendregt. Introduction to generalized type systems. *Journal of Functional Programming*, 1:125–154, 1991.

[7] Alain Berthoz. The role of inhibition in the hierarchical gating of executed and imagined movements. *Cognitive Brain Reseach*, 3(2):101–13, 1996.

[8] K. Birman and T. Joseph. Exploiting virtual synchrony in distributed systems. *SIGOPS Oper. Syst. Rev.*, 21(5):123–138, November 1987.

[9] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, November 1987.

[10] Cynthia Breazeal. Emotion and sociable humanoid robots. *Internation Journal of Human-Computer Studies*, 59:119–155, 2003.

[11] Rodney A. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, Mar 1986.

[12] Rodney A. Brooks. Intelligence without reason. In *Computers and Thought, IJCAI-91*, pages 569–595. Morgan Kaufmann, 1991.

[13] Bruce G. Buchanan and Edward H. Shortliffe. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley Series in Artificial Intelligence)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[14] John T. Cacioppo and Wendi L. Gardner. Emotion. *Annual Review of Psychology*, 50(1):191–214, 1999. PMID: 10074678.

[15] Sean Carroll. *Endless forms most beautiful: the new science of evo-devo and the making of the animal kingdom*. Norton, New York, 2005.

[16] Robin George Collingwood. *The Principles of Art*. Oxford University Press, London, 2005.

[17] Jack Copeland. What is Artificial Intelligence? http://www.alanturing.net/turing_archive/pages/Reference%20Articles/what_is_AI/What%20is%20AI11.html, 5 2000.

[18] Thierry Coquand and Peter Dybjer. Inductive definitions and type theory an introduction. In P.S. Thiagarajan, editor, *Foundation of Software Technology and Theoretical Computer Science*, volume 880 of *Lecture Notes in Computer Science*, pages 60–76. Springer Berlin Heidelberg, 1994.

[19] Daniel Crevier. *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books, Inc., New York, NY, USA, 1993.

[20] Antonio R. Damasio. Emotion in the perspective of an integrated nervous system. *Brain research reviews*, 26:83–86, 1998.

[21] Richard J. Davidson and William Irwin. The functional neuroanatomy of emotion and affective style. *Trends in Cognitive Sciences*, 3(1):11–21, 1999.

[22] Daniel C. Dennett. *Consciousness Explained*. Penguin, 1991.

[23] Thomas Eiter and V. S. Subrahmanian. Heterogeneous active agents, ii: Algorithms and complexity. *Artif. Intell.*, 108(1-2):257–307, March 1999.

[24] Thomas Eiter, V. S. Subrahmanian, and George Pick. Heterogeneous active agents, i: Semantics. *Artif. Intell.*, 108(1-2):179–255, March 1999.

[25] Thomas Eiter, V. S. Subrahmanian, and T. J. Rogers. Heterogeneous active agents, iii: Polynomially implementable agents. *Artif. Intell.*, 117(1):107–167, February 2000.

[26] Richard E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.

[27] Michael Fink, Stefano Germano, Giovambattista Ianni, Christoph Redl, and Peter Schüller. Acthex: Implementing hex programs with action atoms. In Pedro Cabalar and TranCao Son, editors, *Logic Programming and Nonmonotonic Reasoning*, volume 8148 of *Lecture Notes in Computer Science*, pages 317–322. Springer Berlin Heidelberg, 2013.

[28] Adam Fisher. Inside google's quest to popularize self-driving cars. http://www.popsci.com/cars/article/2013-09/google-self-driving-car, 9 2013.

[29] Johnny Fontaine. Self-reflexive emotions. In *The Oxford companion to emotion and the affective sciences*, pages 357–359. Oxford University Press, New York, 2009.

[30] Jörg Fromm and Silke Lautner. Electrical signals and their physiological significance in plants. *Plant, Cell & Environment*, 30(3):249–257, 2007.

[31] Sandra Clara Gadanho and John Hallam. Robot learning driven by emotions. *Adaptive Behaviour*, 9(1):42–64, 2001.

[32] Gerd Gigerenzer and R. Selten. *Bounded Rationality: The adaptive toolbox*. Cambridge: The MIT Press, 2001.

[33] Henry Gray. *Anatomy of the human body*. Lea & Febinger, Philadelphia, twentieth edition, 1918.

[34] J. A. Gray. Three fundamental emotion systems. In *The Nature of Emotion: Fundamental Questions*, pages 243–247. Oxford University Press, 1994.

[35] Jonathan Haidth. The moral emotions. In *Handbook of affective sciences*, pages 852–870. Oxford University Press, New York, 2003.

[36] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, July 1968.

[37] Carl Hewitt, Peter Bishop, and Richard Steiger. A universal modular actor formalism for artificial intelligence. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, IJCAI'73, pages 235–245, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc.

[38] Douglas R. Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid*. Basic Books, Inc., New York, NY, USA, 1979.

[39] Douglas R. Hofstadter. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, Inc., New York, NY, USA, 1996.

[40] Linda Z. Holland, João E. Carvalho, Hector Escriva, Vincent Laudet, Michael Schubert, Sebastian Shimeld, and Jr-Kai Yu. Evolution of bilaterian central nervous systems: a single origin? *EvoDevo*, 4, 2013.

[41] Francois F. Ingrand, Michael P. Georgeff, and Anand S. Rao. An architecture for real-time reasoning and system control. *IEEE Expert: Intelligent Systems and Their Applications*, 7(6):34–44, December 1992.

[42] Mihai Ionescu, Gheorghe Păun, and Takashi Yokomori. Spiking neural p systems. *Fundam. Inf.*, 71(2,3):279–308, February 2006.

[43] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:62–222, 1997.

[44] Dave K. Jacobs, Nagayasu Nakanishi, David Yuan, Anthony Camara, Scott A. Nichols, and Volker Hartenstein. Evolution of sensory structures in basal metazoa. *Integrative and Comparative Biology*, 47(5):712–723, 2007.

[45] Gary Kemp. Collingwood's aesthetics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2012 edition, 2012.

[46] P. J. Lang, M. M. Bradley, and B. N. Cuthbert. Emotion, attention and the startle reflex. *Psychological Review*, 97:377–398, 1990.

[47] S.P. Leys, G.O. Mackie, and R.W. Meech. Impulse conduction in a sponge. *Journal of Experimental Biology*, 202(9):1139–1150, 1999.

[48] Benjamin J. Liebeskind, David M. Hillis, and Harold H. Zakon. Evolution of sodium channels predates the origin of nervous systems in animals. *Proceedings of the National Academy of Sciences*, 108(22):9154–9159, 2011.

[49] Vladimir Lifschitz. Action languages, answer sets and planning. In *In The Logic Programming Paradigm: a 25-Year Perspective*, pages 357–373. Springer Verlag, 1999.

[50] Vladimir Lifschitz. What Is Answer Set Programming? In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 1594–1597, 2008.

[51] Martin Lotze, Pedro Montoya, Michael Erb, Ernst Hülsmann, Herta Flor, Uwe Klose, Niels Birbaumer, and Wolfgang Grodd. Activation of cortical and cerebellar motor areas during executed and imagined hand movements: And fmri study. *Journal of Cognitive Neuroscience*, 11(5):491–501, 1999.

[52] Paul MacLean. *The Triune Brain in Evolution: Role in Paleocerebral Functions*. Plenum Press, New York, 1990.

[53] D. Matsumoto and P. Ekman. Basic emotions. In *The Oxford companion to emotion and the affective sciences*, pages 69–73. Oxford University Press, New York, 2009.

[54] Pamela McCorduck. *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*. AK Peters Ltd, 2004.

[55] Paul Francis Mendler. *Inductive Definition in Type Theory*. PhD thesis, Ithaca, NY, USA, 1988. AAI8804634.

[56] Marvin Minsky. *The Society of Mind*. Simon & Schuster, New York, 1988.

[57] Marvin Minsky. *The Emotion Machine*. Simon & Schuster, New York, 2006.

[58] Marvin Lee Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge Mass., expanded ed. edition, 1988.

[59] L. L. Moroz, K. M. Kocot, M. R. Citarella, S. Dosung, T. P. Norekian, I. S. Povolotskaya, A. P. Grigorenko, C. Dailey, E. Berezikov, K. M. Buckley, A. Ptitsyn, D. Reshetov, K. Mukherjee, T. P. Moroz, Y. Bobkova, F. Yu, V. V. Kapitonov, J. Jurka, Y. V. Bobkov, J. J. Swore, D. O. Girardo, A. Fodor, F. Gusev, R. Sanford, R. Bruders, E. Kittler, C. E. Mills, J. P. Rast, R. Derelle, V. V. Solovyev, F. A. Kondrashov, B. J. Swalla, J. V. Sweedler, E. I. Rogaev, K. M. Halanych, and A. B. Kohn. The ctenophore genome and the evolutionary origins of neural systems. *Nature*, 510(7503):109–114, 6 2014.

[60] Helsinki University of Technology. GnT (Generate'n'Test). http://www.tcs.hut.fi/Software/gnt/, 8 2014.

[61] Vienna University of Technology. DLVhex solver. http://www.kr.tuwien.ac.at/research/systems/dlvhex/, 8 2014.

[62] Andress Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge University Press, Cambridge, 1988.

[63] Bruce Pandolfini. *Kasparov and Deep Blue: The Historic Chess Match Between Man and Machine*. Touchstone, 1997.

[64] Universität Potsdam. clasp. http://www.cs.uni-potsdam.de/clasp/, 8 2014.

[65] Universität Potsdam. Platypus. http://www.cs.uni-potsdam.de/platypus/, 8 2014.

[66] Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa. *The Oxford Handbook of Membrane Computing*. Oxford University Press, Inc., New York, NY, USA, 2010.

[67] Anand S. Rao and Michael P. Michael P. Georgeff. Bdi agents: From theory to practice. In *ICMAS-95*, pages 312–319, 1995.

[68] Jenefer Robinson. *Deeper Than Reason: Emotion and Its Role in Literature, Music, and Art*. Oxford University Press, New York, 2005.

[69] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, New Jersey, 2010.

[70] David Sander, Didier Grandjean, and Klaus R. Scherer. A systems approach to appraisal mechanisms in emotion. *Neural Networks*, 18(4):317–352, 2005.

[71] David K. Simmons, Kevin Pang, and Mark Q. Martindale. Lim homeobox genes in the ctenophore mnemiopsis leidyi: the evolution of neural cell type specification. *EvoDevo*, 3(1), 2012.

[72] Chun Siong Soon, Marcel Brass, Hans-Jochen Heinze, and John-Dylan Haynes. Unconscious determinants of free decisions in the human brain. *Nature Neuroscience*, 11:543–545, 2008.

[73] Aaron Sloman. Developing concepts of consciousness. *Behavioral and Brain Sciences*, 14:694–695, 12 1991.

[74] Aaron Sloman. The mind as a control system. *Royal Institute of Philosophy Supplement*, 34:69–110, 3 1993.

[75] Aaron Sloman. What sort of control system is able to have a personality? In *Creating Personalities for Synthetic Actors, Towards Autonomous Personality Agents*, pages 166–208, London, UK, UK, 1997. Springer-Verlag.

[76] Aaron Sloman. What sort of architecture is required for a human-like agent? In Michael Wooldridge and Anand Rao, editors, *Foundations of Rational Agency*, volume 14 of *Applied Logic Series*, pages 35–52. Springer Netherlands, 1999.

[77] Aaron Sloman. Beyond shallow models of emotion. In *Cognitive Processing: International Quarterly of Cognitive Science*, pages 177–198, 2001.

[78] Aaron Sloman. The SimAgent TOOLKIT. http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html, 11 2014.

[79] V. S. Subrahmanian and Carlo Zaniolo. Relating Stable Models and AI Planning Domains. In *In Proc. ICLP-95*, pages 233–247. MIT Press, 1995.

[80] F. Teroni and J. Deonna. Differentiating shame from guilt. *Consciousness and Cognition*, 17(3):725–740, 2008.

[81] Carnegie-Mellon University. 4CAPS Cognitive Neuroarchitecture. http://www.ccbi.cmu.edu/4CAPS/index.html, 11 2014.

[82] Denise Woodward. Animals I – An Overview of Phylogeny and Diversity. https://wikispaces.psu.edu/display/bio110/Animals+I+-+An+Overview+of+Phylogeny+and+Diversity, 10 2012.