An Evolutionary, Affective Model of Cognition

Janos Tapolczai

August 19, 2014

Modern AI generally falls into two categories, separated by the employed level of abstraction: low-level modelling on one hand, which employs neural networks to simulate individual neurons, and the symbolic or logic-based approach, which first posits an ideal, rational intelligence and then tries to capture it search algorithms and calculi. In this thesis, I describe a model of human cognition based on an evolutionary approach. The basis of this endeavour is the idea that one must understand the developmental history of a complex system if one wished to understand how it works. Starting with a white-box model of computation, we construct a semi-organized mesh of neural components, and then describe plausible models of emotion and reasoning, without invoking rule-based or provably correct methods. Lastly, we implement a number of agents based on these models and employ them in a competitive, moderately complex scenario as a proof of concept.

Contents

1 Introduction

In this document, I will sketch a possible architecture of the human brain and a select few of its subsystems. The descriptions presented are supported by some empirical evidence, but I do not claim that they are straightforward transcriptions of neurological realities. The model is grounded substantially in evolutionary considerations, which provide the backdrop and the plausibility check for the claims presented herein.

Section 3 outlines the basic considerations that lead to the model. Section 5 sketches the proposed model of the mind. Section 6 presents the mathematical model. In Section 7, we look at three concrete subsystems: sensory perception, counterfactual perception (imagination) and affect.

It should also be understood that everything in this document is, at best, a *rough* outline; it may be likened to a hexagon which approximates a circle: though (conjectured to be) basically correct, and useful, it is marred by significant incongruities with the object of its approximation.

2 Related work

3 Preliminary considerations and justification

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 engineering, but as a historical artefact, and as one which was not produced "in one step", but gradually, where each stage of its evolution had to be viable on its own. What, one might ask, is the consequence of such a view? Most importantly, 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 components 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* [24].

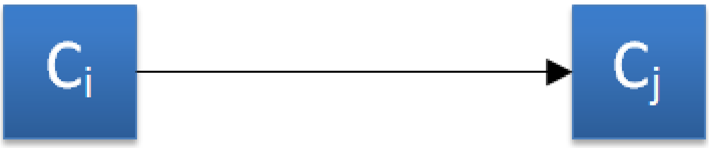


Figure 1: Relationship between the components of an organism without a nervous system.

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.[1] The benefits of such mechanisms were obvious: let us imagine a microscopic organism without any sort of nervous system — all of its behaviour is hard-coded and mechanical. It can take

[1] See any textbook on evolutionary biology.

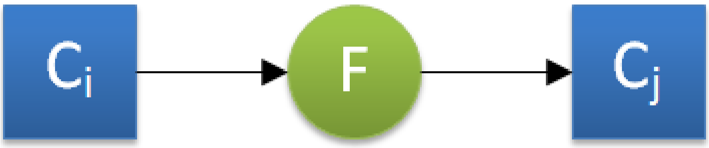


Figure 2: Relationship between the components of an organism possessing a nervous system. F can be understood as a simple signal transformer or a central coordinating mechanism.

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 being sent between C_i and C_j (the case $i = j$ is of course possible). Such an organism suffers from two disadvantages: (a) the behaviour is necessarily simple and (b) it is not very adaptable.

Let us now imagine that such an organism develops a bundle of cells which transmit the signals from various parts of its body, modulate them in some way, and then send them to various parts, inducing changes. Schematically, this is shown in Figure 2, where a function F is interposed between two components. The first such nervous systems were likely little more than signal transformers or magnifiers that expedited communication between parts: with a few neurons, an organism would have had the ability to coordinate movements or rely on sensing parts induce, say, movement.

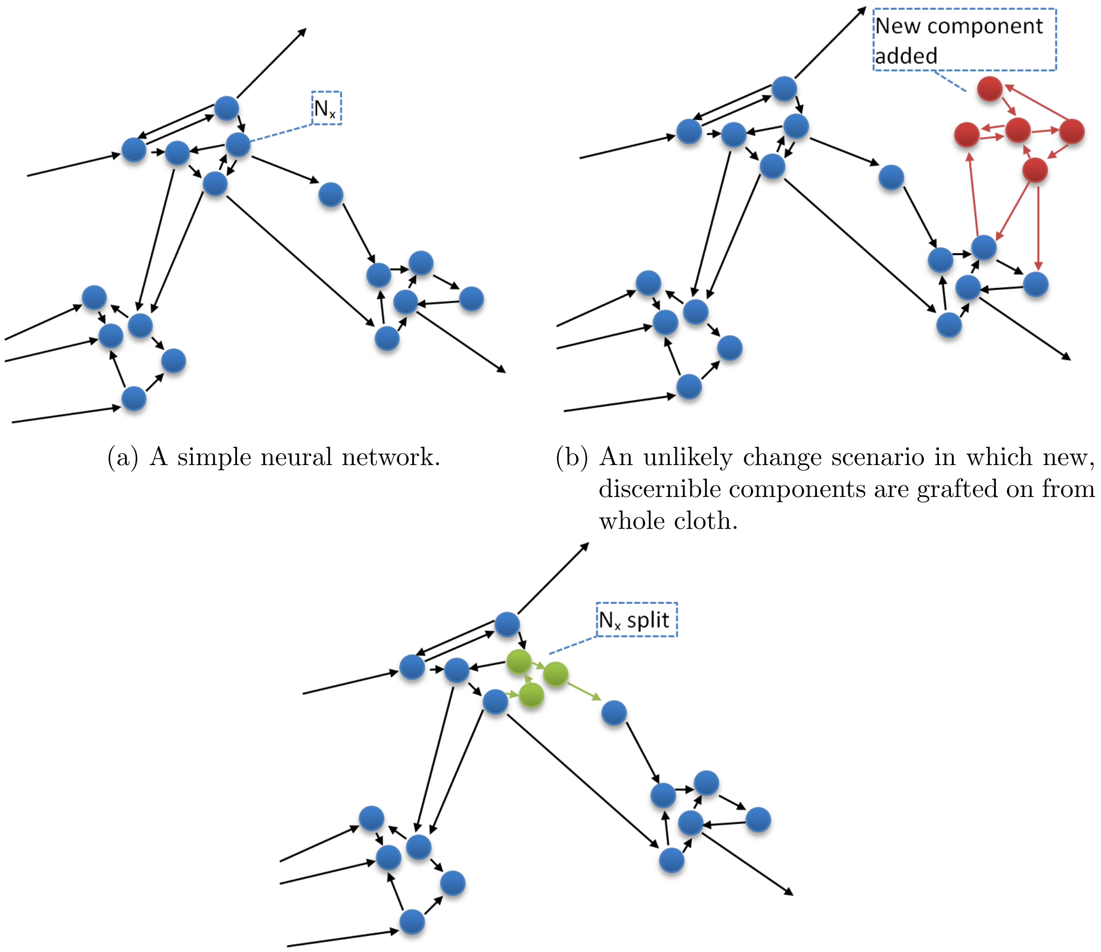
The neuron bundles would have been quite malleable in the face of selection pressure: when the environment required it, they could, after several generations, start to compute different or more elaborate functions. For instance, an organism which had had developed in an environment where food was abundant 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,[2] but we can see how the aforementioned three could have been effected through changes in a simple nervous system alone. Let us recall the beginning of this section and contrast such a malleable computational mesh with most products of human engineering: one cannot simply take out a piston in a car or replace a cogwheel in a mechanical clock with a differently sized one. Machines are designed to fit together perfectly and their complexity tends to be irreducible. Even programs, which are more open to mutation and which are often evolved in evolutionary algorithms, are easily broken by small changes.

Adaptation of nervous systems When discussing how an organism's nervous system can evolve and, in particular, *evolve to perform new tasks* and not just variations on old ones, explanations

[2] 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.



(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.

are again constrained by two criteria: (a) the change has to be small, or at least have a small cause[3] and (b) each change must be beneficial in the short term.[4]

To illustrate this, we can look at a simple neural network in Figure 3a, with a marked node N_x. Figure 3b shows an unlikely change scenario in which some new component/function is cleanly grafted onto the system. Figure 3c then shows a much more likely scenario: a mutation causes N_x to be split and the new nodes take over some of its connections. 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.

[3] 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 [7].

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

The brain as a collection of functions The processes hitherto described are quite uncontroversial and can be found reiterated in any textbook on the evolution of nervous systems. The functional structure and the model of computation used in the brain, however, are not well understood. FMRI and similar brain imagining techniques, while invaluable, give only rough impressions about the neural correlates of certain forms of cognition and do not give fine-grained insight into its structure. As such, the model I shall describe in the following paragraphs is a conjecture. The implication of such an evolutionary viewpoint, I conjecture in this document, is that brain functions don't "just appear", but are rather the result of small changes and the recombination of pre-existing parts. This, in turn, informs the plausibility of various possible brain architectures. In becomes unlikely that the brain should be a collection of neatly delineated functions, or that it should have certain coordinating units or universal message formats for communication between components. The reason for this is that administrative mechanism confer little evolutionary benefit on their owen, and do not confer it gradually: the imposition of a central coordinating mechanism on a pre-existing mesh of neurons would necessitate the complete reorganisation of such, and the abandonment of the previous communication channels on favor centralized coordination. The same objections can be raised against a universal or even a local message format. Moreover, such mechanisms require substantial changes in the organism with no obvious or immediate advantage.

Such objections do not contradict the existence of macroscopic structures in the brain, dedicated to certain tasks. The development and adaptation of such remains entirely plausible. They do, however, give insight into the pattern of processing inside such structures, which is often simply regarded as atomic or replicated in computers as if it were a conventional engineering product.

Instead of a rigidly ordered brain with central organisation and large, discrete, and highly complex features like "sight" or "reason" which function like pluggable black boxes, I propose a decentralized white-box architecture composed of simple parts: first, every component, while perhaps sophisticated, is conceptually simple. Second, communication between different components is not performed in the function-call pattern of computer programs, but rather by one component listening in on the activity of another. Since there is, inherently, no mechanism of function abstraction in neural systems, it stands to reason that the most likely way for new functions to develop is for additional neurons to modulate the activity of others. In such a scheme, a visual perception component does not have to know which other components will consume its output (or rather, listen on its activity); changes which affect agent activity in useful ways based on the visual data can occur gradually and, over time, become large enough to count as components in their own right.

Practical abstraction While such a white-box model is conceptually useful, a mesh of gradually grown patterns does not lend itself to implementation in a program. 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.

4 Diagram notation

In the rest of this document, a number of diagrams appear. These will use the following notation:

Symbol	Description
C P	Processing component
Ft Ch	Choice
Messages	Data container (Queue, List, etc.)
Head Body	Data
○	Stream generator
Head Body	Counterfactual (imaginary) data

5 Schema 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 other components to read. Figure 4 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 4 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.

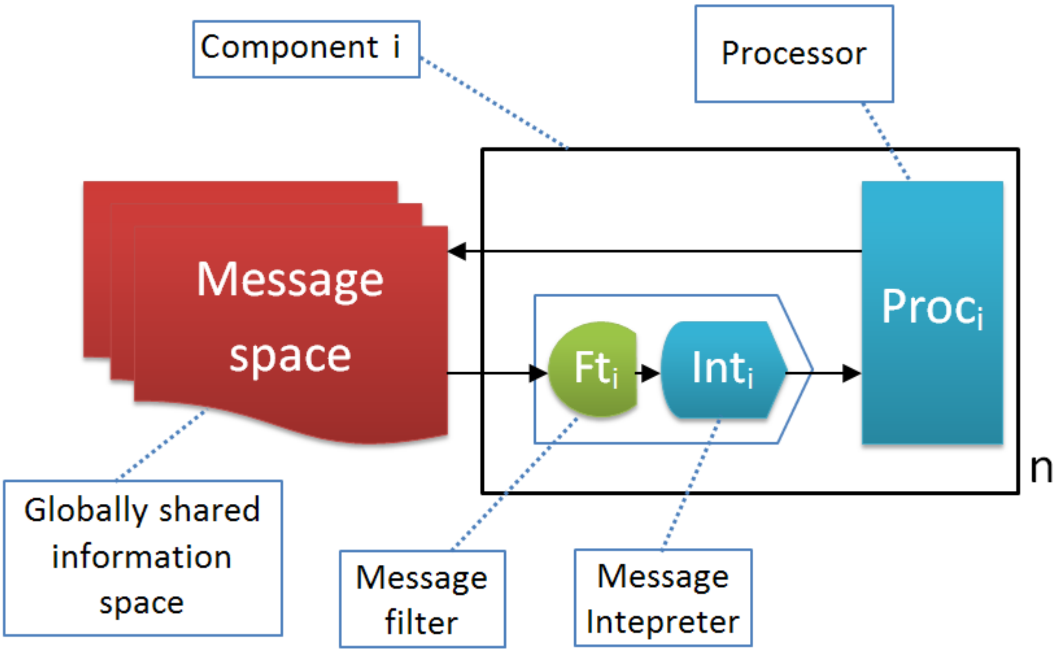


Figure 4: Global neural architecture.

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.

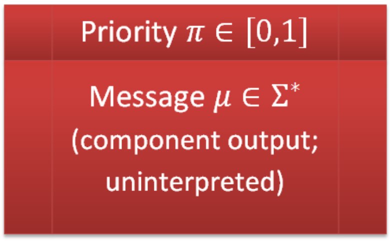


Figure 5: Structure of a neural message.

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 [26, 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 5: 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 6 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. ([32, p. 337] and [17]), in which charges sent along directed graphs of neurons are used to compute functions.

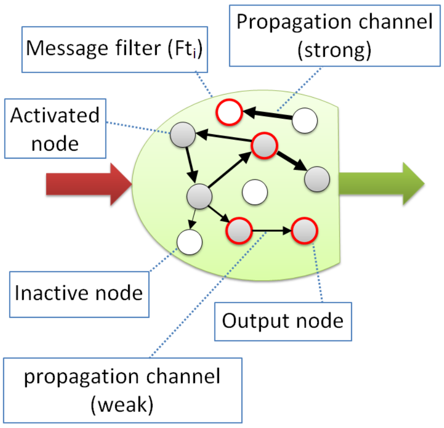


Figure 6: A pattern-matching filter for a component C_i.

6 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.

6.1 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 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.

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

Basic type. $\mathbb{N}$, $\mathbb{R}$, *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.*

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 simply 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. $\mathbb{N}$ *and* $\mathbb{R}$ *are interpreted as the natural and real numbers, respectively.* $\mathrm{int}(\emptyset) = \{\}$.

Sum type. *If* $\mathtt{T_1}, \mathtt{T_2}$ *are types, then* $\mathrm{int}(\mathtt{T_1} + \mathtt{T_2}) = \mathrm{int}(\mathtt{T_1}) \cup \mathrm{int}(\mathtt{T_2})$.

Product type. *If* $\mathtt{T_1}, \dots, \mathtt{T_n}$ *are types and* $\mathtt{s}$ *is a string, then*

$$\text{int}(\mathtt{s}\ \mathtt{T_1} \dots \mathtt{T_n}) = \begin{cases} \{\mathtt{s}\} & \text{if } n = 0 \\ \{\mathtt{s}\} \times \text{int}(\mathtt{T_1}) \times \dots \times \text{int}(\mathtt{T_n}) & \text{if } n \geq 1. \end{cases}$$

Full application. *If* $\mathtt{T_1}, \dots, \mathtt{T_n}$ *are types and* $[\forall x_1, \dots, x_n]\, \mathtt{C}$ *is a type constructor, then*

$$\text{int}(\mathtt{C}\ \mathtt{T_1} \dots \mathtt{T_n}) = \bigcup_{v_1 \in \text{ int}(\mathtt{T_1})} \cdots \bigcup_{v_n \in \text{ int}(\mathtt{T_n})} \left(\bigcup_{C' \in \text{ cint}(C)} C'[x_1 \backslash v_1, \dots, x_n \backslash v_n] \right).$$

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* $\mathtt{T_1}, \dots, \mathtt{T_n}$*, then*

$$\text{cint}(\mathtt{C}) = \bigcup_{v_1 \in \text{ int}(\mathtt{T_1})} \cdots \bigcup_{v_n \in \text{ int}(\mathtt{T_n})} \mathtt{C}[\mathtt{T_1} \backslash v_1, \dots, \mathtt{T_n} \backslash v_n].$$

Intuitively, sum types are simply unions, product types are named cartesian products, and full applications are instantiations of type constructors with all possible values. 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 $(\rightarrow)$ which induces the function type:

Example 6 (Function arrow). *If we take, say, the type* $\rightarrow \mathtt{S1}\ \mathtt{S2}$ *(a product type with the product types* $\mathtt{S1}$ *and* $\mathtt{S2}$ *as arguments) and abstract twice, we get* $[\forall s, t] \rightarrow \mathtt{s}\ \mathtt{t}$. $\rightarrow \mathtt{s}\ \mathtt{t}$ *is the type constructor for unary functions from* $\mathtt{s}$ *to* $\mathtt{t}$*, also written infix as* $\mathtt{s} \rightarrow \mathtt{t}$*. Functions with multiple arguments, mapping* $\mathtt{t_1}, \dots, \mathtt{t_{n-1}}$ *to* $\mathtt{t_n}$*, can be modelled in two ways: either through n-tuples, or through nested function arrows:*

$$\langle \mathtt{t_1}, \mathtt{t_2}, \dots \mathtt{t_{n-1}} \rangle \rightarrow \mathtt{t_n}$$
$$\mathtt{t_1} \rightarrow (\mathtt{t_2} \rightarrow \dots \rightarrow (\mathtt{t_{n-1}} \rightarrow \mathtt{t_n}) \cdots)$$

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* $f :: \mathtt{S} \rightarrow \mathtt{T}$ *and* $x :: \mathtt{T}$*. Then* $f\ x :: \mathtt{T}$*. 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* $\mathtt{C_1},\ldots,\mathtt{C_n}$ *be type constructors such that* $\mathtt{C_i} = [\forall x_1,\ldots,x_n]\,\mathtt{T_i}$ *and* $\mathtt{T_i}$ *is a named product type with type variables* $(1 \le i \le n)$*. Then* $[\forall x_1,\ldots,x_n]\,(\mathtt{T_i}+\cdots+\mathtt{T_n})$ *is an ADT. If we want to give a name to an ADT, we write it as* $\mathtt{s}\ \mathtt{x_1}\ldots\mathtt{x_n} = \mathtt{T_i}+\cdots+\mathtt{T_n}$*.*

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 ($\mathbb{B}$, $\mathbb{Q}$, $\mathbb{C}$, Maybe, Either, List).

$$\begin{aligned}
\mathbb{B} &= \mathtt{False} + \mathtt{True}\\
\mathbb{Q} &= \mathtt{Rat}\ \mathbb{N}\ \mathbb{N}\\
\mathbb{C} &= \mathtt{Complex}\ \mathbb{R}\ \mathbb{R}\\
\mathtt{Maybe\ t} &= \mathtt{Nothing} + \mathtt{Just\ t}\\
\mathtt{Either\ l\ r} &= \mathtt{Left\ l} + \mathtt{Right\ r}\\
[]\ \mathtt{a} &= [] + (\mathtt{a} : [\mathtt{a}])
\end{aligned}$$

Here, False*,* True*,* Nothing*, and* [] *are nullary product types;* Left*,* Right *are unary type constructors (which can be instantiated into unary product types). ":" is a binary type constructor, written infix.* Rat $\mathbb{N}$ $\mathbb{N}$ *and* Complex $\mathbb{R}$ $\mathbb{R}$ *are binary product types.*

$\mathbb{B}$, $\mathbb{Q}$, $\mathbb{C}$ *are the sets of Boolean number and rational/complex numbers, respectively.* Maybe *represents an optional value, which may or may not be present.* Either *represents a choice between two values, of which either the left or the right one is present, but not both.* [a] *denotes a list of values of type* a*.*

We also define the usual convenience functions for these types:

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

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

Definitions 1–8 specify a fragment of System F_ω,[5] 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.

6.2 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*

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

where

[5] 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 $[\forall x_1,\ldots,x_n]\,\mathtt{C}$ for quantifier-free $\mathtt{C}$. This is also called the Hindley-Milner type system. For details, see Barendregt [3].

1. `name` $::$ *I is the* name *of C,*

2. `ft` $::$ `T` $\rightarrow \mathbb{B}$ *is called the* filter *of C,*

3. `int` $::$ `T` $\rightarrow$ `Maybe T` *is called the* interpreter *of C, and*

4. `proc` $::$ `T` $\rightarrow$ `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.

6.3 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.

6.3.1 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 $\mathtt{ft}_C\ m = \texttt{True}$ and $\mathtt{int}_C\ m = \texttt{Just}\ m'$ 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. $\mathtt{proc}_C\ m_{\text{in}} = m$. 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 $\{m_1, \dots, m_n\}$ — is denoted by:

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

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

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

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

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

6.3.2 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*

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

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

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

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 $T_1; T_2; \dots; T_n$ 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, $T_1; \dots; T_n$ will sometimes be written line-by-line as

$$\begin{matrix} T_1 \\ \vdots \\ T_n \end{matrix}$$

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

$$\begin{matrix} \dots \to [m] \to \langle f_1, \dots, f_n \rangle C \\ C\langle f_1, \dots, f_n \rangle \to [m'] \to \dots \end{matrix}$$

If no change occurs, that is, if

$$\begin{array}{l} C\langle\rangle \to [m] \to \dots \quad \text{or} \\ \dots \to [m] \to \langle\rangle C \end{array}$$

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

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 $f_n \circ \dots \circ f_1(C)$.

6.4 Invariants

Such a model does not necessitate the existence of special structures, such as central organizers or sequences of components, one activated after another,[6] 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 $C_1, \dots, C_n$ 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).

$$\begin{array}{l} [\forall i \in \{1 \dots, n\}][\forall m] : \\ \qquad (C_i \rightarrowtail \{m\} \Rightarrow \texttt{rec}(m) = \{C_{co}\}) \wedge \left(\left(\mathit{proc}_{C_{co}} \circ \mathit{int}_{C_{co}}(m)\right) \in \bigcup_{1 \le j \le n} \texttt{rec}(C_j) \right). \end{array}$$

Figure 7a depicts such an organizer. Similarly, sequences can be created by components $C_1, \dots, C_n$, where each components reads the message of the last one.

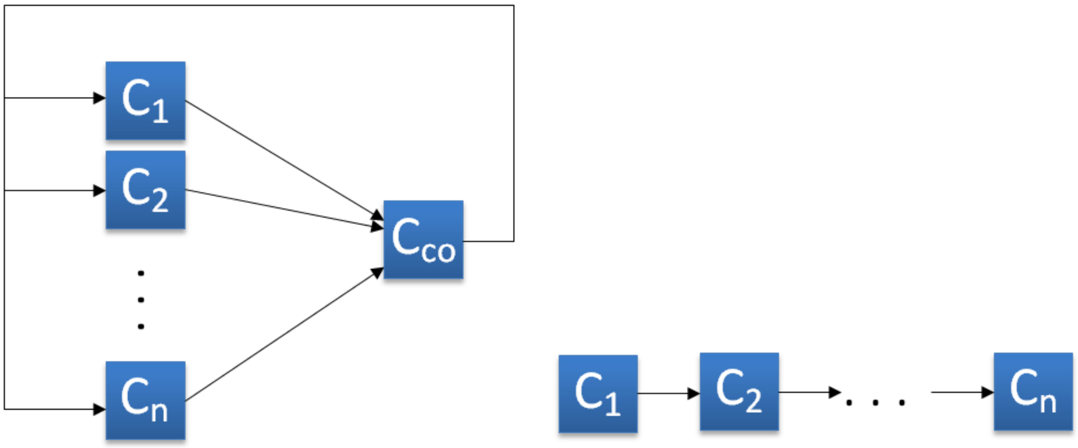
Invariant 20 (Sequence).

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

7 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 —, counterfactual perception — the imagination, which mimics the output of the senses —, and affect — broadly speaking, the emotional component of cognition.

[6] An example of such a sequence is found in [34] 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.

7.1 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 8 shows such a layered structure.

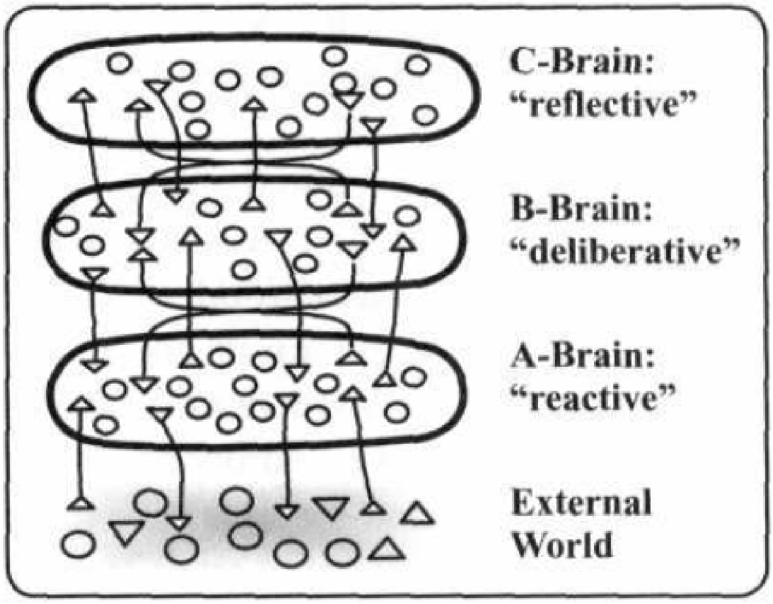


Figure 8: Layered perception of the world, from [26, p. 100].

The diagram is explained thus [26, 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 4, 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 9, 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.

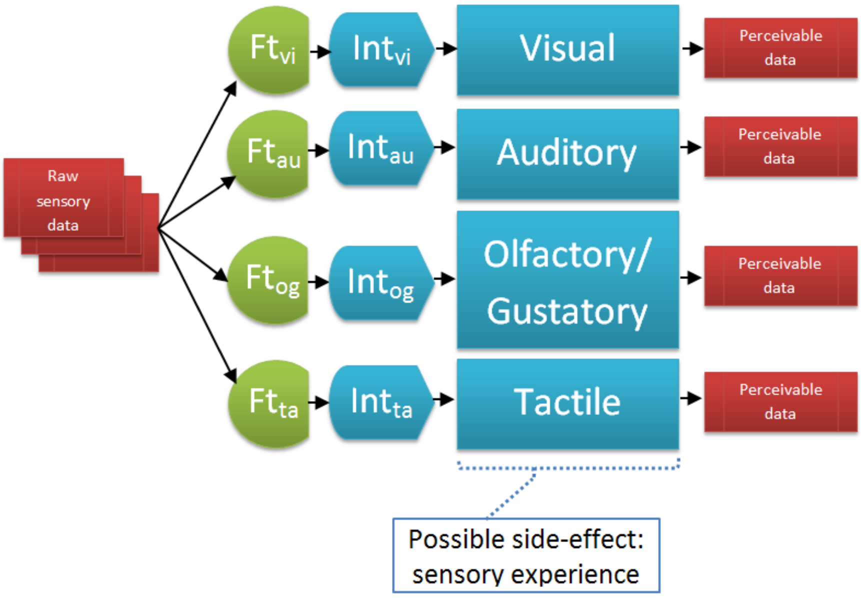


Figure 9: Partial structure of sensory perception - raw sensory data is processed and made available to higher functions such as the affective subsystem. The comment "Possible side-effect: sensory experience" signifies the fact that conscious and sub-conscious sensory experiences might occur as a side-effect of this processing. However, it is currently unknown to neuroscience whether this is indeed the case.

7.2 Counterfactual perception and planning

Broadly speaking, counterfactual perception can be described as "imagination", and is closely related to sensory perception and world simulation. In examining the system, we might broadly classify its processes into three categories:

1. Counterfactual perception — 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 [23]:

   > 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 [4]:

   > (...) 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

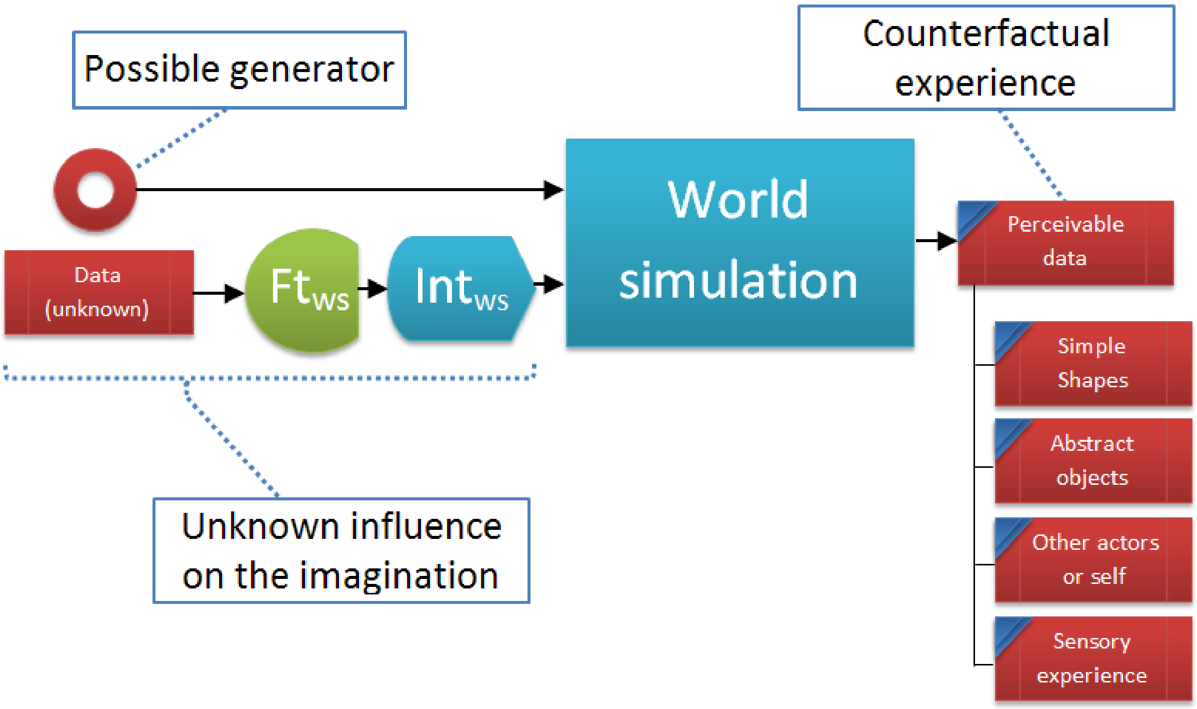


Figure 10: Structure of of counterfactual perception & world simulation: messages emulating the output of sensory perception are generated, but are marked as counterfactual by unknown means.

shown in Figures 10, 11, and 12: the world simulation is an ordinary component with a filter and interpreter which outputs, for simplicity's sake, messages marked as counterfactual. 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 11:

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.

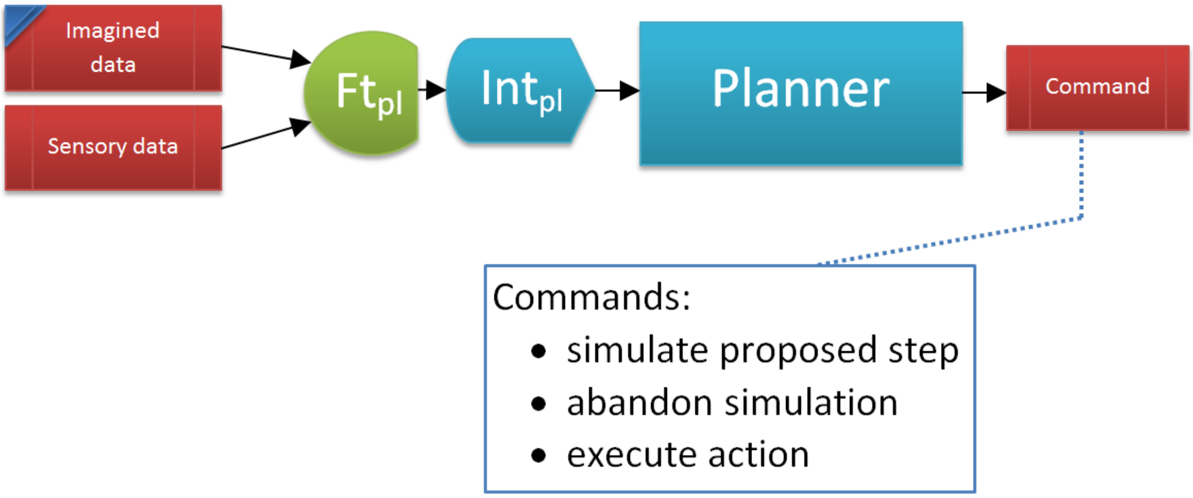


Figure 11: Planner with two kinds of inputs: (1) real sensory data and (2) counterfactual data which comes from world simulation. On the basis of these inputs, possible steps are developed and sent out as commands.

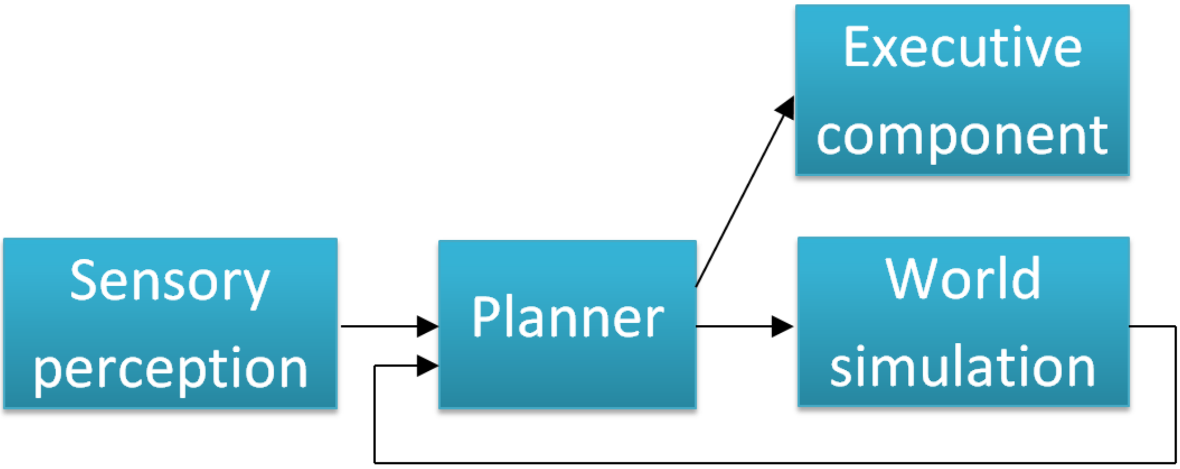


Figure 12: Interaction between world simulator and planner: the planner devises possible steps and feeds them into the world simulator, which, in turn, tries to calculate their effects. The results are fed back to the planner.

7.2.1 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 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.[7]

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
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}:

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

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 $\texttt{step} \in St$.

[7] 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.

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.

7.3 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 [13], 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 13 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 [13, p. 47]:

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

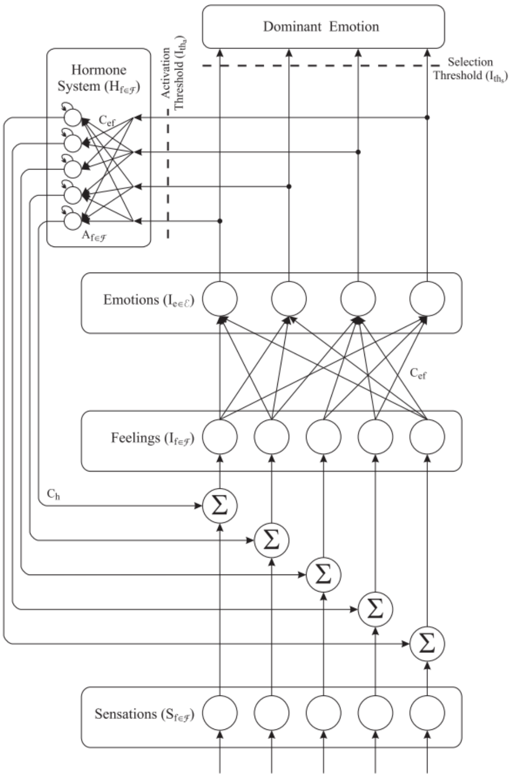

Dominant Emotion
Selection Threshold (I_{th_s})
Activation Threshold (I_{th_a})
Hormone System ($H_{f\in\mathcal{F}}$)
C_{ef}
$A_{f\in\mathcal{F}}$
Emotions ($I_{e\in\mathcal{E}}$)
C_{ef}
Feelings ($I_{f\in\mathcal{F}}$)
C_h
Σ
Σ
Σ
Σ
Σ
Sensations ($S_{f\in\mathcal{F}}$)

Figure 13: Emotional model of Gadanho and Hallam [13, 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 [5] detailed taxonomy of emotion like Ortony's OCC model [29]. 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.[8] Figure 9 shows the adapted model. The general structure was retained, but the set of sensations was replaced by the sensory processor described in Section 7.1 and, instead of a single dominant emotion, competing emotions simply emit messages which are used by execute components and the world simulation.

7.3.1 Affective subsystems

In this section, I will develop the concept of "emotion" in greater detail. The process shown in Figure 14 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

[8] 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.

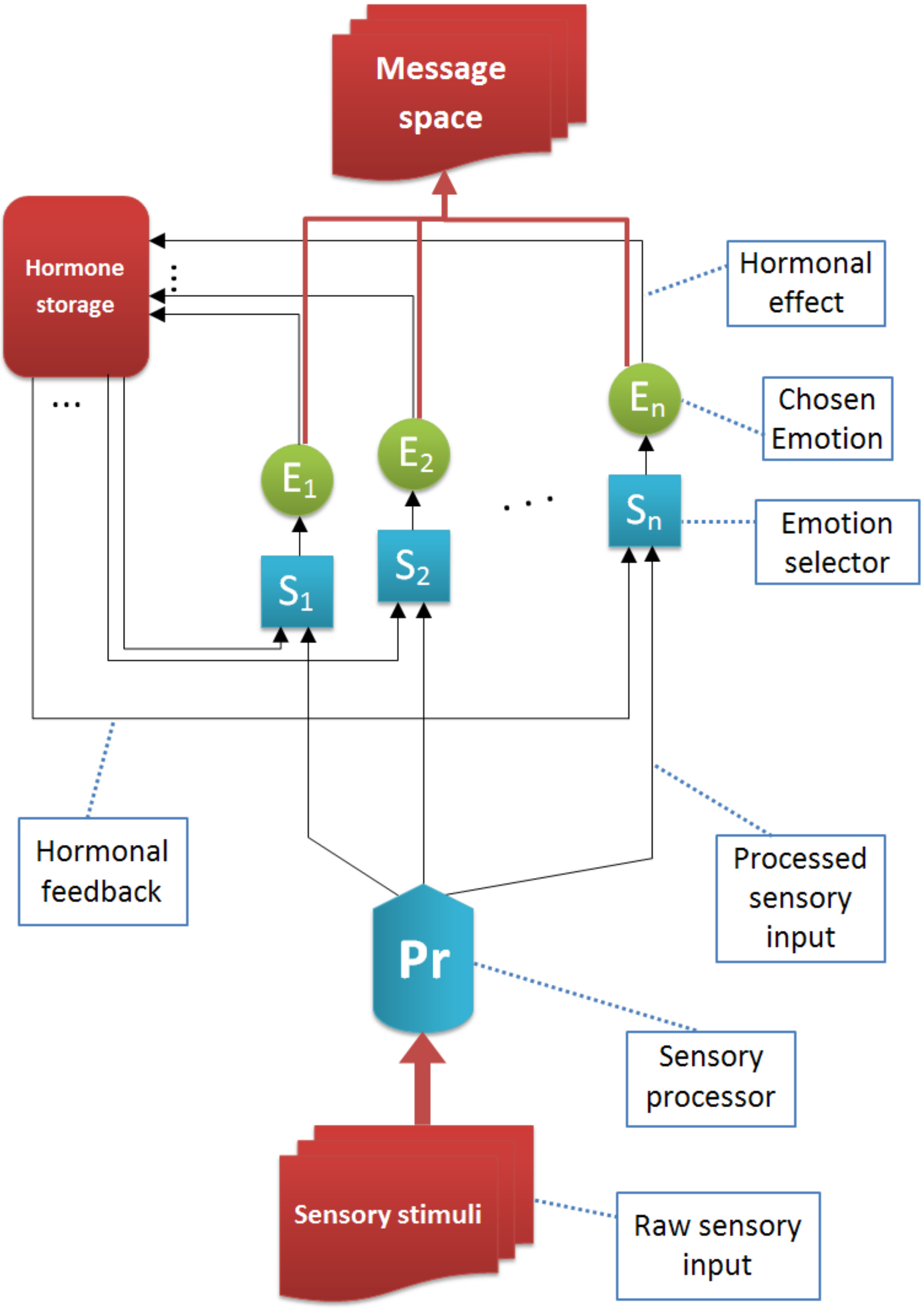


Figure 14: Affective subsystem; specialisation of the global neural architecture. In plastic neural systems, selections may change over time.

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 [10, 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. [6], Gray [14] and Lang et al. [20] 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 [1, pp. 9-10]. Matsumoto and Eckman [25], for instance, identified seven basic emotions: happiness, surprise, contempt, sadness, fear, disgust, and anger.

Damasio [9], 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 15.

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

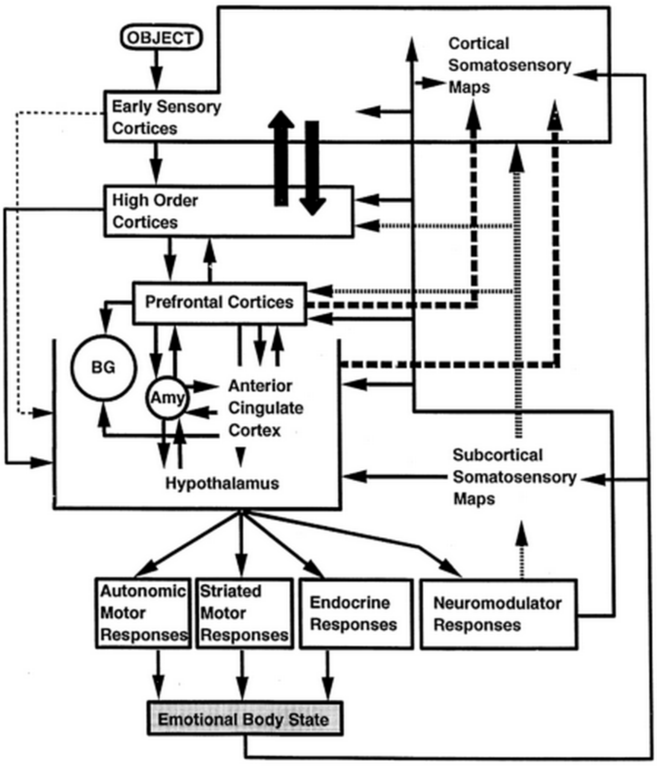


Figure 15: Neurological structure of affect, according to Damasio [9].

> 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 14 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 [34]. Sander and colleagues partitioned the emotion process into four stages, as shown in Figure 16. The first is *relevance*, which functions as a filter and detects the intrinsic pleasantness and the level of

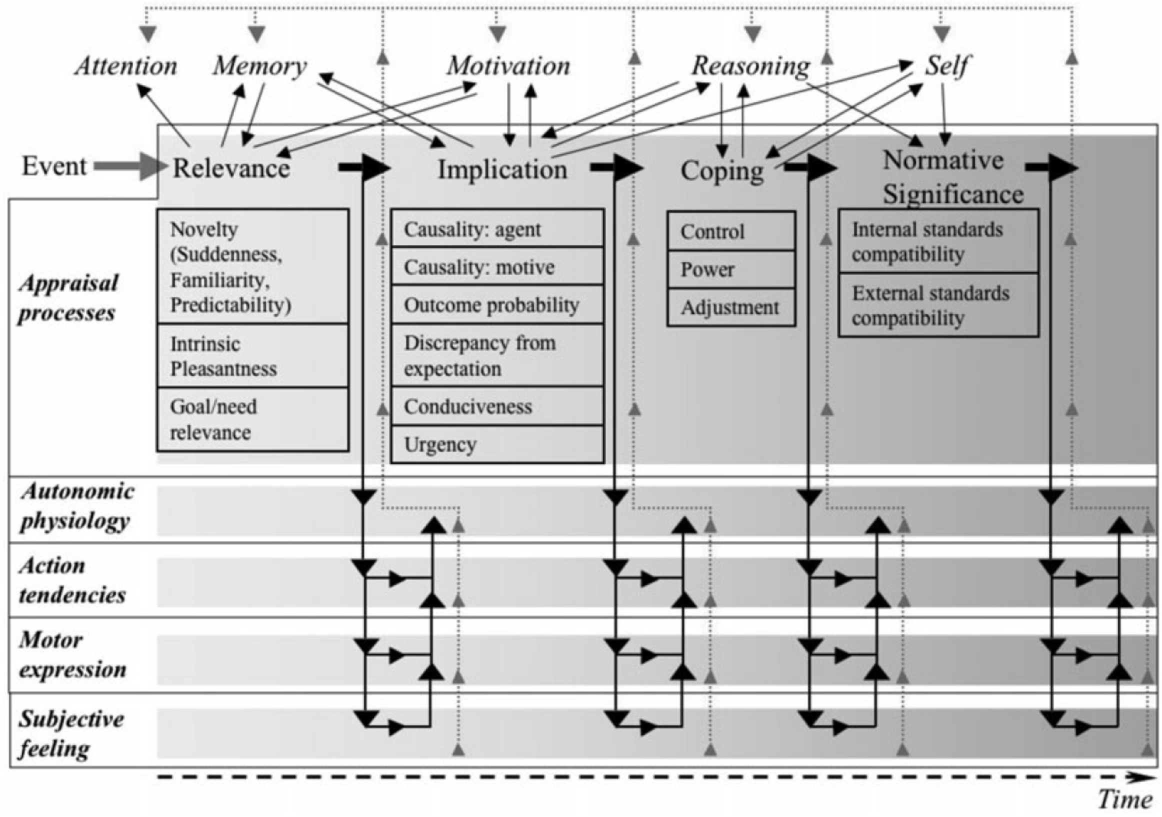


Figure 16: The four-stage emotion process according to Sander et al, consisting of relevance, implication, coping and normative significance.

(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. I propose the following rough categorization:

Pre-social emotions. Certain behavioural mechanisms can be observed in non-social as well as social animals. The flight-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 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 [36], but possibly also jealousy and humiliation (which, in contrast to shame, is attributed to external causes) [12];
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 [15], 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* [18], 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* [8, 19]. 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 [29]. 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 sub-conscious 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.[9]

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 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 7.2, 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 sub-consciously made actions and get the following four categories of executive system:

Sub-conscious 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;

[9] 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 at seemingly random, but will not be 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. [35] instructed subjects to press a button and to record when they through 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 simple "popped into their heads".

Sub-conscious 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 7.2 only controls one other component — and it might very well be that there it doesn't even exist in the brain. 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.

7.4 Interaction between affect and world simulation

Section 7.2 outlined what could be called *deliberate action* in the from of a planner-world-simulator loop. Section 7.3 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.

8 Proposed architecture

9 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.

9.1 World

The choice of world profoundly affects the implementation of the agent: its knowledge base, mechanism of perception and interaction, the required complexity of the implementation, etc. On 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, the world should be sufficiently complex to allow the agent to shine. 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 [11], A* [16], ASP [21], 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.

9.1.1 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 extensively studied. In this, their simplicity has been their great advantage — That very simplicity is 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.

9.1.2 Wumpus world

The traditional Wumpus world, as described in Russell and Norvig's *Artificial Intelligence: A Modern Approach* [33, 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 [33, 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* $\langle G, \mathrm{gl}\rangle$ *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* $W = \langle G, \mathrm{gl}\rangle$ *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 graph* S *such that* *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 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 17 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

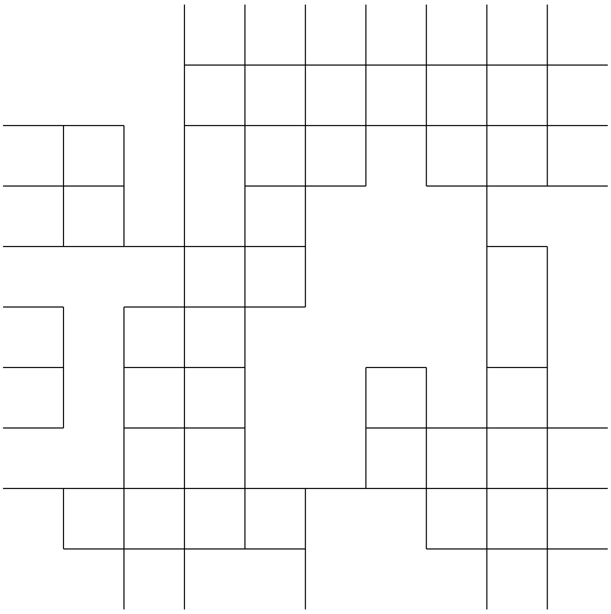


Figure 17: A segment of 2-dimensionally embeddable world. The vertices are its rooms, the edges are the connections between the rooms.

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

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

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

When the number of dimensions and the embedding are obvious, we omit n and ϵ. Since π^ϵ is injective, an inverse $(\pi^\epsilon)^{-1}$ also exists. Through it, we define the indexing function *of W:*

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

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 $W = \langle G, \text{gl}\rangle$ 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 $\Delta_2 : E(G) \to \{0, x_1^+, x_1^-, x_2^+, x_2^-\}$.*
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" $\mathcal{W}_{\text{jun}}$.

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

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

Agent and Wumpus are the following records:

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

The last component of Agent, $\mathtt{state} :: \mathtt{S}$, *is the internal state of agents which we will discuss later.*

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

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*, defined below.

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 $[W, \varphi\ W, \varphi^2\ W, \varphi^3\ W, \ldots]$, 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 (Standards 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*

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

where W' is identical to W, 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*

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

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

Wumpus behavior *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* $(\texttt{light} \circ \texttt{time})\ \mathrm{gl}$ *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*

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

Whenever a Wumpus travels along an edge e *with* $\Delta(e) \neq 0$*, apply* 0.1 *damage with probability* $\texttt{danger}(e)$.

Agent behavior *Agents always move 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* $0.05 \times \texttt{fatigue}\ e$*. Additionally (if* $\Delta\ e \neq 0$*), apply* 0.1 *damage with probability* $\texttt{danger}\ e$*.*

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 should now be clear why $\mathcal{W}_{\mathrm{jun}}$ is called a jungle world: it is a social hunter-gatherer scenario in which uncoordinated agents act and interact without any explicit performance measure. They can gather food or gold, rest, hunt wumpuses, communicate via gestures, and even develop friendships, but fundamentally, everyone is out for himself. 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.

9.2 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.

9.2.1 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:

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 $v_1, v_2, \dots, v_n$ 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 18 illustrates the working of this mechanism. If vertex w falls into an agent's sight cone, it perceives $\pi(w)$ and the following data:

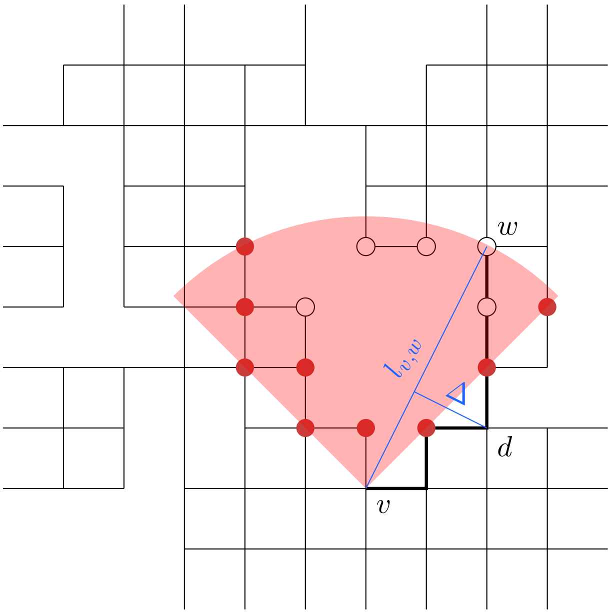


Figure 18: Sight cone of an agent at `light`$(t) = 2$. The cone with width $\frac{\pi}{4}$ signifies that agent's range of vision. Red vertices in it are perceived; the hollow black ones are not because they are blocked by holes in the world. The line $l_{v,w}$ illustrates why the vertex w is not visible 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 $\frac{\sqrt{2}}{2}$.

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.

9.2.2 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 5, 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.

Counterfactual perception (CFP). In essence, the imagination of an agent. Counterfactual perception allows reasoning and the internal simulation of parts of the world.

Attention-control (AC). Attention-control is the recognition of certain real or counterfactual 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 counterfactual and real events that happened to an agent. This log is utilized chiefly by the CFP 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 [10]. 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 19 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.

In terms of implementation, this is realized via the system we saw in Figure 14, Section 7: 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 20: 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

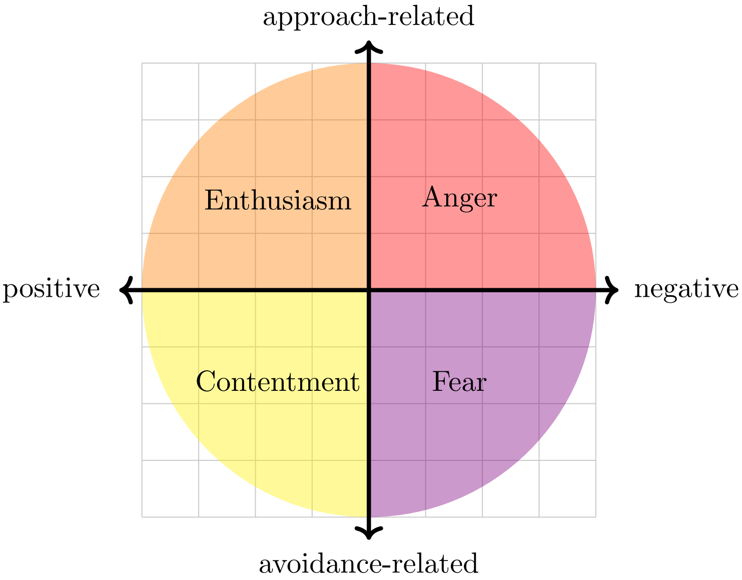


Figure 19: Emotions evoked by the PSBC.The left half contains the positive emotions of enthusiasm and contentment, whereas the right contains the negative emotions of anger and fear. Enthusiasm and anger are both approach-related, causing action, whereas contentment and fear are approach-related, causing flight or abstinence from action.

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 $\mathtt{receive} :: \mathtt{H}_n \to \mathbb{N} \to \mathbb{R} \to \mathtt{H}_n$ and $\mathtt{tick} :: \mathtt{H}_n \to \mathtt{H}_n$, given by*

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

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.

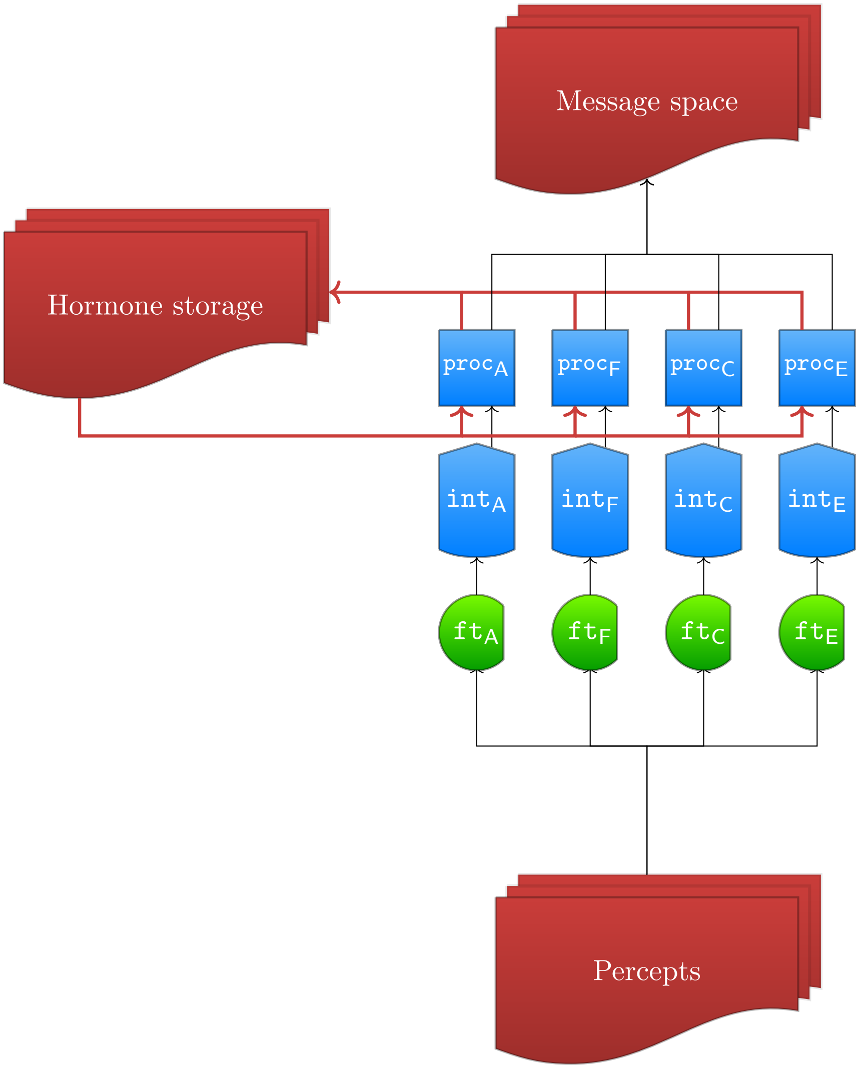

Message space
Hormone storage
$\mathtt{proc}_{\mathsf{A}}$
$\mathtt{proc}_{\mathsf{F}}$
$\mathtt{proc}_{\mathsf{C}}$
$\mathtt{proc}_{\mathsf{E}}$
$\mathtt{int}_{\mathsf{A}}$
$\mathtt{int}_{\mathsf{F}}$
$\mathtt{int}_{\mathsf{C}}$
$\mathtt{int}_{\mathsf{E}}$
$\mathtt{ft}_{\mathsf{A}}$
$\mathtt{ft}_{\mathsf{F}}$
$\mathtt{ft}_{\mathsf{C}}$
$\mathtt{ft}_{\mathsf{E}}$
Percepts

Figure 20: 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).

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 21. Figures 22 and 23 list the different antagonistic and sympathetic judgements.

The evocative mechanism is structurally similar to that of the PSBC, as we saw in Figure 20, 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 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 24, which shows it to be largely analogous to the PSBC in Figure 20.

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.)

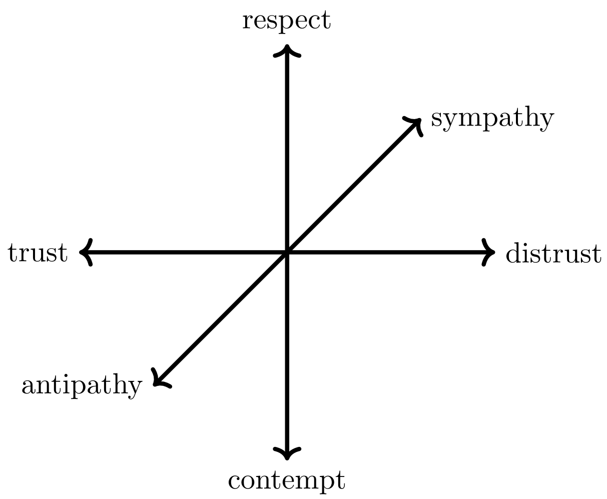


Figure 21: Emotions evoked by the SJS. The primary is axis is sympathy/antipathy, since it distinguishes friend from foe. Trust/distrust judges the loyalty/honor of another agent, whereas respect/contempt judges its competence.

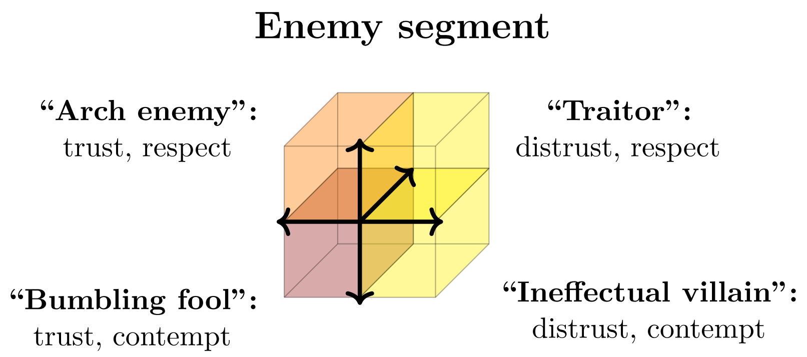


Figure 22: The four antipathic judgements. Enemies can be respected or held in contempt, and deemed trustworthy or untrustworhty. Respect for an enemy implies that an agent holds it to be competent. Trust implies that an agent knows its enemy to be basically honourable.

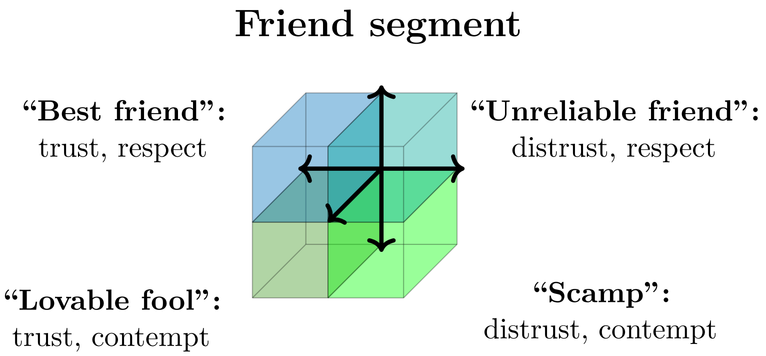


Figure 23: The four sympathic judgements. Friends, like enemies, respected or held in contempt, and deemed trustworthy or untrustworhty. Distrust renders the sympathetic judgement tentative, since the agent cannot be sure of the assistance of an untrustworthy friend. Contempt works similarly, but doubts a friend's ability, rather than loyalty.

Counterfactual perception. 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 extensively utilize learning and construct its own ontologies and ways of thinking from scratch, we will use a fixed ontology, and a existing reasoning tool called DLVhex [28]. 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 [22] and [2]. I will give the compressed definitions:

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

$$\mathtt{P}_1 \vee \cdots \vee \mathtt{P}_\mathtt{h} \mathrel{:-} \mathtt{P}_{\mathtt{h}+1}, \ldots, \mathtt{P}_\mathtt{k},\ \mathtt{not}\ \mathtt{P}_{\mathtt{k}+1}, \ldots,\ \mathtt{not}\ \mathtt{P}_\mathtt{m}.$$

where $\mathtt{P}_\mathtt{i} = \mathtt{p}_\mathtt{i}(\mathtt{a}_{\mathtt{i},1}, \ldots, \mathtt{a}_{\mathtt{i},\mathtt{n}_\mathtt{i}})$ *such that* p_i *is a predicate symbol and every* $\mathtt{a}_{\mathtt{i},1}, \ldots, \mathtt{a}_{\mathtt{i},\mathtt{n}_\mathtt{i}}$ *is either a variable or a constant* $(1 \leq i \leq m)$*. We also call predicates* literals*. If a predicate only has constant arguments, it is called a* ground literal*. For every literal* c*, the literal* $\neg c$ *is called the* strong negation *of* c*.*

$\mathtt{P}_1 \vee \cdots \vee \mathtt{P}_\mathtt{h}$ *is called the head of the rule,* $\mathtt{P}_{\mathtt{h}+1}, \ldots, \mathtt{P}_\mathtt{k},\ \mathtt{not}\ \mathtt{P}_{\mathtt{k}+1}, \ldots,\ \mathtt{not}\ \mathtt{B}_{\mathtt{h}+\mathtt{k}+\mathtt{m}}$ *the body. The predicates* $\mathtt{P}_{\mathtt{h}+1}, \ldots, \mathtt{P}_{\mathtt{h}+\mathtt{k}}$ *are asserted, the predicates* $\mathtt{P}_{\mathtt{k}+1}, \ldots, \mathtt{P}_\mathtt{m}$ *are default-negated. 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. Note that not knowing a predicate to be true is different from knowing it to be false. Answer-set solver computer so-called answer-sets of logic programs. An answer-set is defined as:

Definition 33 (Answer-set). *Let* Π *be a disjunctive logic program. A set of ground literals* $\mathcal{A}$ *is an answer-set of* Π *if the following holds:*

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

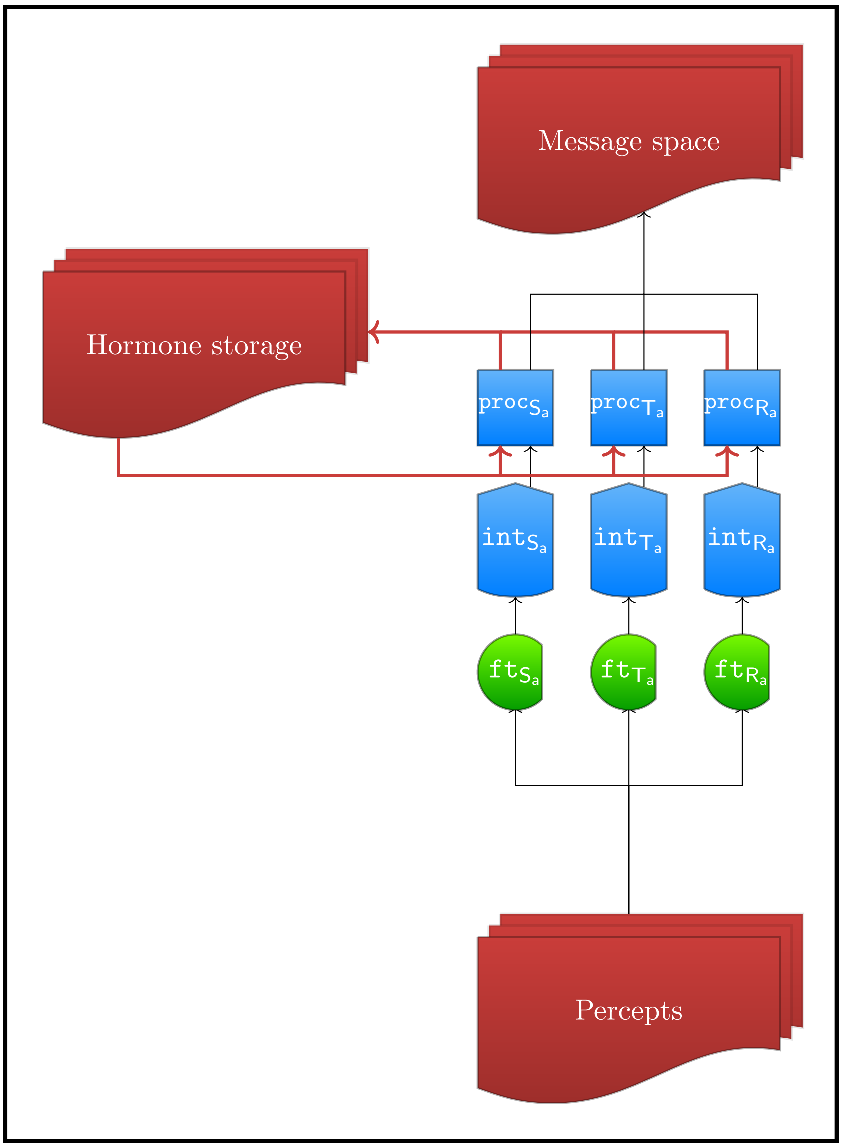

Message space
Hormone storage
$\texttt{proc}_{\mathsf{S}_a}$
$\texttt{proc}_{\mathsf{T}_a}$
$\texttt{proc}_{\mathsf{R}_a}$
$\texttt{int}_{\mathsf{S}_a}$
$\texttt{int}_{\mathsf{T}_a}$
$\texttt{int}_{\mathsf{R}_a}$
$\texttt{ft}_{\mathsf{S}_a}$
$\texttt{ft}_{\mathsf{T}_a}$
$\texttt{ft}_{\mathsf{R}_a}$
Percepts

Figure 24: 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.

Consistent. *There is no ground literal c such that $\{c, \neg c\} \subseteq \mathcal{A}$.*

Complete. *For every rule* $(\mathtt{P_1} \vee \cdots \vee \mathtt{P_h} :\!- \mathtt{P_{h+1}}, \ldots, \mathtt{P_k},\ \mathtt{not}\ \mathtt{P_{k+1}}, \ldots,\ \mathtt{not}\ \mathtt{P_m}) \in \Pi$*:* $\{P_{h+1}, \ldots, P_k\} \subseteq \mathcal{A}$ *and* $\{P_{k+1}, \ldots, P_m\} \cap \mathcal{A} = \emptyset$ *together imply* $\{P_1, \ldots, P_h\} \cap \mathcal{A} \neq \emptyset$*.*

No violated constraints. *There is no constraint* $(:\!-\ \mathtt{P_1}, \ldots, \mathtt{P_k},\ \mathtt{not}\ \mathtt{P_{k+1}}, \ldots,\ \mathtt{not}\ \mathtt{P_m}) \in \Pi$ *such that* $\{P_1, \ldots, P_k\} \subseteq \mathcal{A}$ *and* $\{P_{k+1}, \ldots, P_m\} \cap \mathcal{A} = \emptyset$*.*

Minimal. $\mathcal{A}$ *is minimal with respect to set inclusion, i.e. if any* $\mathcal{A}'$ *of* Π *is consistent, complete, and doesn't violate any constraints, then* $\mathcal{A}' \not\subseteq \mathcal{A}$*.*

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 [30], GnT [27], and Platypus [31]. The main advantage of DLVhex over them is that it allows the outsourcing of literals: whenever the solver finds a so annotated literal, it consults an external program for its value instead of deriving its value internally. We will use this interface to implement the loop between the CFP and the DM:

TODO: Describe usage of DLV. At each step, the atom `choice` is outsourced by DLVhex.

Attention-control.

TODO: Describe attention-control.

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 CFP and thus drives the planning process. Aside from the difference in target, both are modelled via a function

$$\mathtt{choice} :: \mathtt{s} \to \mathtt{World} \to \langle \mathtt{Action}, \mathtt{s} \rangle$$

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 CFP and, in principle, can go on as long as the agent wishes to deliberate. 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`.

Memory.

TODO: Describe memory (knowledge-base).

Relationship between components. Having defined the agents 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 25.

At the very heart of the agent lies its decision-making component, which controls both the agent's actions and its counterfactual perception. The DM and the CFP form the *imagination loop* ι which develops plans by exploring the likely consequences of certain actions. In that capacity, the DM evaluate the CFP'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 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 34 (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:*

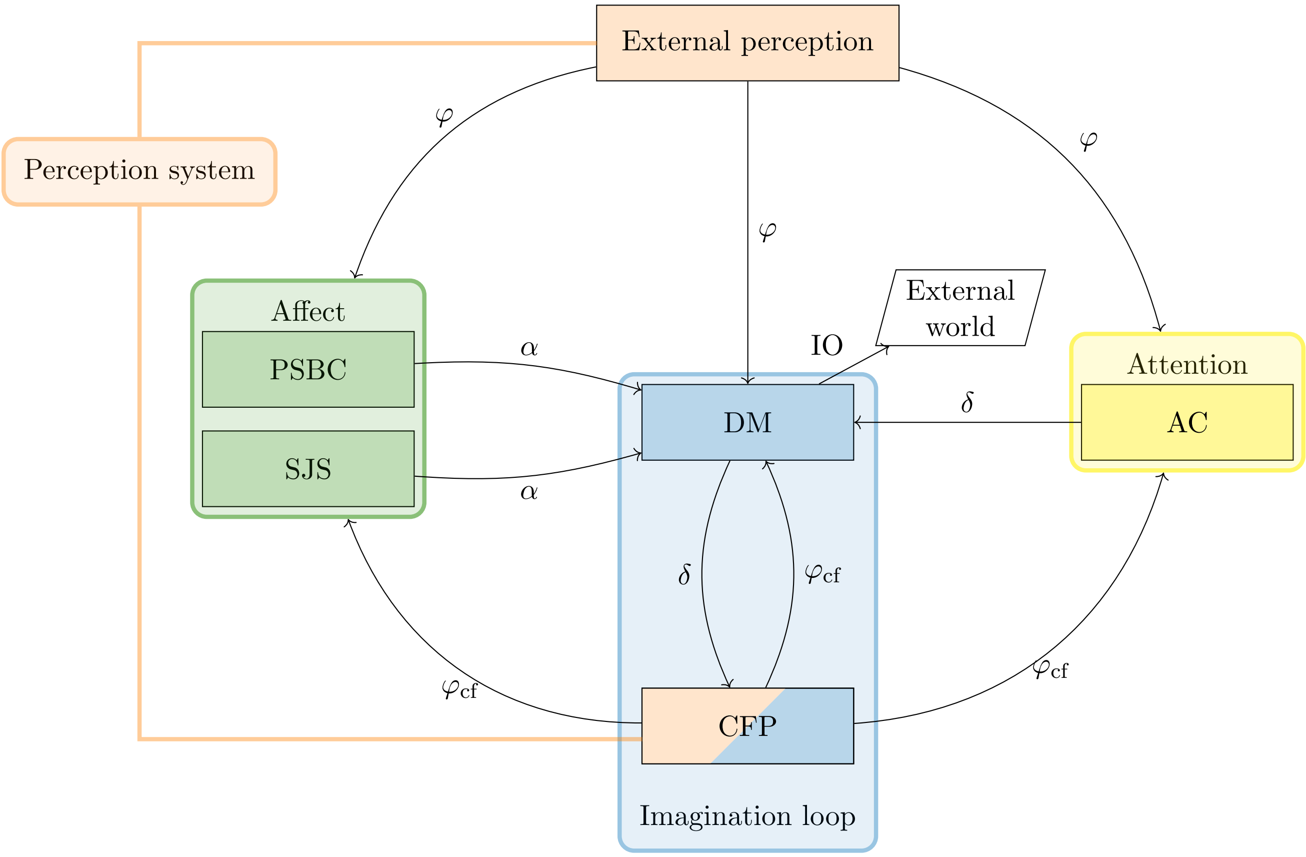


Figure 25: High-level view of the cognitive structure of agents, with groups of systems shown in colored boxes. The PSBC and the SJS comprise the affective group; the DM and CFP the imagination loop responsible for planning. The CFP, with External perception, makes up the perception system. The edge labels show which kind of message the system sends out: α for affective information, φ and φ_{cf} for (counterfactual) perceptions, δ for control signals. IO corresponds to real actions in the world.

$$
\begin{aligned}
\mathtt{agent}\ w = {} & \mathtt{fromJust} \\
& \circ\ \mathtt{getActionMessage} \\
& \circ\ \mathtt{head} \\
& \circ\ \mathtt{dropWhile\ noResult} \\
& \circ\ \mathtt{iterate\ loop} \\
& \circ\ \mathtt{perception}\ w
\end{aligned}
$$

$$
\begin{aligned}
&\text{where} \\
&\quad \mathtt{perception} :: \mathcal{W}_{\text{jun}} \to \mathtt{S} \to \mathtt{S} \\
&\quad \mathtt{psbc}, \mathtt{sjs}, \mathtt{ac}, \mathtt{dm}, \mathtt{cfp} :: \mathtt{S} \to \mathtt{S} \\
&\\
&\quad \mathtt{loop} :: \mathtt{S} \to \mathtt{S} \\
&\quad \mathtt{loop} = \mathtt{cfp} \circ \mathtt{dm} \circ \mathtt{ac} \circ \mathtt{sjs} \circ \mathtt{psbc} \\
&\\
&\quad \mathtt{getActionMessage} :: \mathtt{S} \to \mathtt{MaybeAction} \\
&\quad \mathtt{noResult} = \mathtt{not} \circ \mathtt{isJust} \circ \mathtt{getActionMessage} \\
&\\
&\quad \mathtt{iterate} :: (\mathtt{a} \to \mathtt{a}) \to \mathtt{a} \to [\mathtt{a}] \\
&\quad \mathtt{iterate}\ f\ x = x : \mathtt{iterate}(fx, x) \\
&\\
&\quad \mathtt{dropWhile} :: (\mathtt{a} \to \mathtt{Bool}) \to [\mathtt{a}] \to [\mathtt{a}] \\
&\quad \mathtt{dropWhile}\ p\ xs = \begin{cases} h : \mathtt{dropWhile}\ p\ t & \text{if } xs = (h : t) \wedge (p\ h = \mathtt{True}) \\ xs & \text{otherwise} \end{cases}
\end{aligned}
$$

Note: $\circ$ *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.

10 Evaluation

Results of the implementation.

References

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

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

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

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

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

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

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

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

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

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

[11] 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.

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

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

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

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

[16] 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.

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

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

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

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

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

[22] Vladimir Lifschitz. What is answer set programming? In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 1594–1597, 2008.

[23] 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.

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

[25] 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.

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

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

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

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

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

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

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

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

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

[35] 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.

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