Perception, Logic and Action through Structured Motivated Associative Pandemonium

John V. Jackson

jj2598@tutor.open.ac.uk

Abstract. The Pandemonium Association Engine (PAE) [5,2,6] is proffered in an attempt to fulfil the cognitive necessities of an agent embodying true intelligence and common sense, and hopefully helping unify the main cognitive paradigms. The operation of the basic model and the performance of the second stage of development introducing imaginary concepts and goals are outlined, along with the principles and design of the third stage, which implements both schemata and a second predictive link network to accompany the original gain links. The Prediction links help implement planning and memory, helping define and recognise processes. A box-free style of architecture and design of intelligent systems, "epiprogramming", is preached.

Keywords: PAE, Pandemonium Association Engine, PLASMAP, epiprogramming

1 Introduction

The “Pandemonium Association Engine” (PAE) cognitive paradigm [5,2,6] arose from a consideration of the conscious components of mental problem solving. Although much mental activity may be unconscious, even a record of just the conscious part of anyone's stream of thought reveals both transfer of method between one “compartment” and another (e.g. Minsky's “analogy” [7]), and use of recently learned short cuts, to an extent not matched in any A.I. system of the 1980's. The conviction was formed that massive flexibility and learning ability at all levels might allow a system to achieve an unprecedented degree of both intelligence and common sense. In the 1970's, “blackboard” concepts recapitulated an idea first discussed at least as early as the Teddington conference in 1958 that has never stopped being advocated, where much, most or all communication between certain types of subsystem was channelled through a limited capacity exchange such as some kind of working memory [1,9].

The PAE was intended to avoid perennial problems besetting the design of minds, such as inflexibility, complexity, discordance with subjective experience, the frame problem (do you update everything whenever you learn anything?) and the "A-brain/B-brain" problem [7] (which threatens an infinite regression of supervisory modules). In addition to [5,6] it has had explicit implementations e.g. [3,8].

2 Design Principles

An introduction to the design principles can be found in [5,6] but three of the most significant are as follows:

- Choosing a system that carries out simplest operations as part of its basic cycle (in the PAE by learning and exploiting simple associations) but builds more sophisticated cognitive structures and strategies as appropriate; this useful feature is demonstrated when programming in Prolog;
- Staying as faithful as possible to the operation of natural minds, in particular to what has subsequently become known as the "Global Workspace Theory" [1] (GWT) interpretation, where much communication and interaction between aspects of the mind are channelled through a restricted global workspace such as the working memory (the latter is represented in the PAE by the "stage").
- "Epiprogramming", where functions the system is expected to carry out are not coded explicitly, but, by cunningly twiddling its parameters, are planted as potentialities so they arise as patterns in its data or operation.

Epiprogramming is so vital a component of genuine AI, and so seldom appreciated, that its elucidation in this paper has a higher priority than that of recounting advanced developments of the PAE. Two essential benefits spring from epiprogramming:

Diagrams of the functions of a mind and the links between those functions are familiar to all in the field. However it is not widely recognised that these diagrams are functional specifications, not system specifications. In AI, the very first attempt to implement subcomponents explicitly can risk dooming the venture.

Explicit modularisation is so familiar in human organisations, conventional computer programming, and most deceptively even in the physical brain, that it is easy to assume this is at least one aspect we've got right.

But consider the requirement that every piece of the material of thought, the concepts, must be associable with every other, even though they may comprise emotions, sensations, memories or complex plans or perceptions, and many of which may be constructed en courant. If any pair needs an arrow linking them, every pair at least needs a potential link. Although neuronal tracts of varying breadth link different parts of the brain, in the democracy of the mind, all thoughts are potentially equal at birth.

What principles should govern this total potential flexibility? Clearly it is impossible for everything to communicate significantly with everything else at once, but when should two components be linked?

If the requirement is universal, maybe the solution is simple. The working memory/STM seems so central, maybe its rule of temporal association isn't so much an obvious constraint but an essential feature. You can go a very long way with a machine that links according to similarity of structure, content or context etc., but also involves temporal association.

We identify concepts composing the mind, but although we know at some level these are not all separate objects, it is all too easy to forget it. Diagrams of the mind typically include boxes such as "Meta-management" and "Long-term associative memory", but although the designer is obviously aware that the former is built out of

the latter, when implementing the architecture, the temptation to code them as separate software objects is hard to resist. A chessboard contains rectangles, crosses, even triangles, but they are not all separate entities; they often compose each other.

Things like meta-management, and even higher levels of perception and action, are epi-phenomena, and we must always be wary of coding them as explicit procedures if we do so at all.

Those privileged to have experienced the queasy mind shift required with logic programming, will sympathise with the weirdness of another programming approach such as epiphenomenal programming - designing a system of some basic simplicity but causing more complex patterns of behaviour to arise indirectly through the design of the operational parameters of the basic system. The new design of PAE does combine and use demons in a way which spontaneously develops more complex facilities.

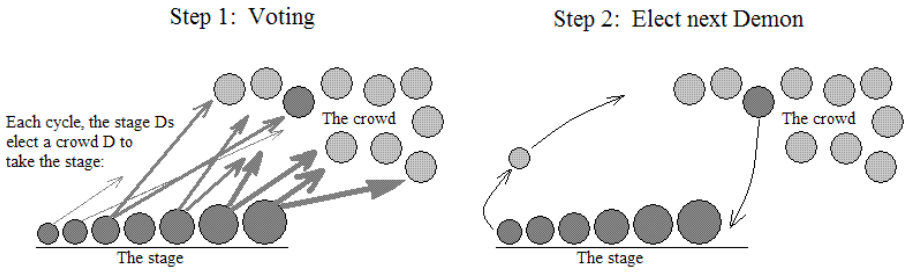
The following points are of the essence:

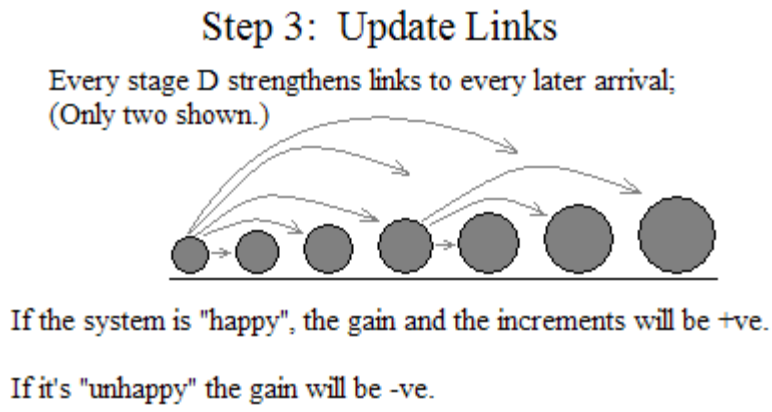
- Many things such as aspects of the mind don't have to communicate since they are different aspects of the same thing;
- Many entities, even packets of the material the system is designed to process as "data", all communicate using the same simple scheme;
- Step one should be the construction of this universal system of communication, which will implement the inductive learning needed by operations from the most basic up;
- Step two is arranging for high level structures to arise as epiphenomena;
- The attempt to implement directly any but the most basic aspects of an intelligent system immediately restricts its capacity for creative thought.
- The universal system of communication mentioned in the third point is the use of inter-demon links (see below) to forge and use associations between any pair of demons under successful circumstances, and by using them, to seek to exploit regularities in the external and internal universe to repeat the success.

These links never need be explicitly defined by the human inventor between any components, even, or especially, between those created during the course of negotiating the cognitive landscape, whether they be bottom-level demons or compound structures of any size or complexity. Components may be elicited not just in any sequence, but perhaps subsuming each other in turn in hierarchical structures at different times.

3 Outline of "Vanilla" version

The elements of the simplest version of PAE [5,6]: the “Vanilla model” [11], will be revisited, but later models employ the same basic design. Although developed under a philosophy of box-free design, some diagrams are useful:





As in any implementation of GWT, actors embodying discrete components of processing - elements of thought of various kinds - strut and fret their time upon some sort of stage before departing for a while to a holding area. In PAE the actors are called Demons (Ds), and represent actions, perceptions, compounds of other Ds, or merely links. The stage is now called the “stage”. (Its original term “arena”, intended to refer to the area of sand in the middle, the playing field where the main action took place, was very commonly taken to refer to something else.) The holding area where the Ds wait is called the crowd.

The currently active demons vote (using weighted links developed in step 3 below) to select the next to be activated:

The next demon takes the stage with its strongest strength, the other stage Ds decay in strength, and the currently weakest one returns to the crowd of inactive demons, but with its links to and from other stage Ds amended:

If the new demon represents an action, it is executed (via some "non-A.I." mechanism notionally considered to be beneath the stage, in the "Sub Stage Mechanism" (SSM)). However if the new demon represents a stimulus in the outside world, it will have already carried out its main duty simply by appearing. The SSM will have encouraged its election by strongly supplementing its vote. ("Grist" demons, neither action nor stimulus, may vote and be elected, and they confer a useful complexity, taking on interesting intermediate roles.)

Links are strengthened by increments proportional to the "current state of well-being" (as calculated by the SSM, and which may cause negative increments and voting) that are applied each cycle from each demon on the stage to each later arrival.

This may be compared with Selfridge's original model [10], which was essentially a neural net, structured like a multi-layer perceptron, each layer producing varying arousals in demons in the next layer. The PAE differs from the Selfridge model in that the end feeds back to the beginning, there is full connectivity, stimuli can occur at any point, action Ds may occur, and although all Ds on stage can vote simultaneously, only one fires at once.

Another distinction between Selfridge's original and PAE is that PAE is more a model of the mind as a whole, alternating between parallel and sequential behaviour, whereas the original is a closer representation of parallel processing in the early stages of the visual system, at least in the bottom-up sense.

For a brief comparison with classifier systems see [11].

The performance of the "vanilla" version of PAE is gratifyingly similar to a rat or pigeon under behaviouristic investigation. The first experiment (see [6] for further details) had a simulated "rat" in a "room" with two sides. From time to time one side of the room became unpleasant (suddenly becoming very cold) but the "rat" could, by electing the "jump" demon, move instantly to the other side of the room. It could learn to jump precisely as the change occurred if it repeated after a fixed number of cycles and with a repeat period (up to a dozen cycles) for which the number of grist demons was adequate (e.g. two or three dozen). It could also learn to jump at the right moment if the change occurred after varying numbers of cycles but signalled by an external stimulus represented by the appearance on the stage of a characteristic demon, and perhaps with an intervening delay though this was harder to learn. Analysis of the sequences of demons elected in episodes on the way to finding the right sequence gave a strong impression of a search process [6]. Using a "pain" demon greatly helped learn these tasks since it formed a reliable point on which to start attaching avoidance behaviours which otherwise would be extremely unlikely to form under conditions of negative gain. Negative gain tends to disperse fixed sequences and leads to exploration, while positive gain leads to solidification of regularity. A good mix of these two with occasionally a little negative gain works best, and is reminiscent of structures characteristic of life, on the borderline between order and chaos.

4 Constructing sub-goal sequences

To help deal with sequences of sub-goals efficiently, the first enhancement [6] allowed the creature to imagine something that was currently unreal/untrue (for example an unachieved sub-goal), and work towards realising it.

An experiment was run where the rat was subjected to drives, e.g. hunger and thirst. When each of these was high, an action D designed to lower the drive (“expiation D”) e.g. “Eat” or “Drink” was encouraged onto the stage by supplementing its vote, and if successful, gain was raised for that stage cycle. The expiation D only succeeded when its object (i.e. food/drink) was present. If the object

were not present, the expiation D was elected in the “unreal” sense; however, a sequence of the right actions under the right circumstances could make it “real”. Designing a special form of behaviour and treatment of unreal Ds led to sequences of subgoals being negotiated more efficiently. These special features are a simplified version of more sophisticated enhancements envisioned for the future.

The experiment took place in a "house" on a single floor, with four rooms arranged in a square. By electing the demon for "go to room x" the creature could go to a room where further achievements became possible. The "kitchen" room contained items of food and a "fridge", which could be opened so that when the loose food around it had run out, more food could be revealed; the "bar" room contained items of loose drink and a "bottle" which operated in a similar if slightly less realistic way.

But assembling sequences randomly becomes increasingly unlikely with length, and until they do occur they can't be reinforced. It's ok if the creature has run a sequence of sub-goals in the past, and it happens upon the most junior sub-goal, because the links formed in the past will tend to drive the useful sequence of demons. But how can we make the creature work backwards from the final goal to the initial goal, and then in the reverse sequence perform the series of subgoals in the right order? This becomes particularly significant if more than one pressing end goal, such as eating vs. drinking, are offered, and just tagging onto any old sequence won't do.

By marking action Ds elected to the stage but which cannot be executed, as "unreal", and making their voting work backwards, in the sense of voting for Ds that have strong gain links to the unreal D instead of from it, we can assemble a sequence of unreal subgoals in the correct but reverse order, prior to execution in the proper sequence. For this to work, gain links need to be generated between the sub-goals, and so the process of realisation needs to be accompanied by a gain spike. This was an automatic aspect of the realisation of an expiation D, but when added as a feature of the realisation of any D, the sub-goal sequencing behaviour emerged. (In later versions, this slightly artificial mechanism is to be enhanced in a number of ways, so that for example, the gain is only provided if the new reality status had been desired, and it might represent a change either from unreal to real or vice versa.)

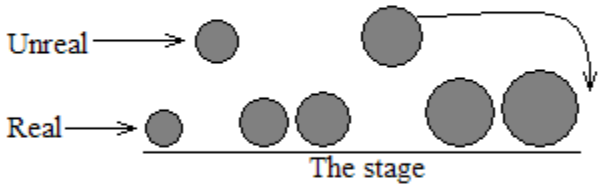
Eventually if the initial subgoal can be executed successfully causing each next one in turn to realise, the sequence will not only work but will have been caused to assemble by the end goal.

In the experiment, the creature used its "unreal" facility to move from room to room. When hungry it went to the kitchen, where it learned to use the "open" D to open the fridge, so it could then eat. It also moved to the bar and opened a bottle so it could drink. Happily it seemed to be able to converge on the "open" D but then continue along the correct food/drink path as appropriate. This showed it could use the "Unreal" facility to handle two different drive systems without confusion, even when they shared a sub-goal.

As well as the "important" Ds (the ones that actually do or signal things) others, described earlier as "Grist Ds", acquire useful roles. If there are say 20-25 Ds in total, and say 7 or 8 actually correspond to direct actions or percepts, the rest start out by just swilling about amongst the more important ones, but eventually playing a useful part in directing operations, as a kind of "middle men".

Realisation

A "realising" D re-enters the stage at full strength, typically accompanied by a "Gain Spike":

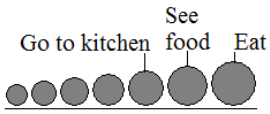


Each cycle, the unreal Ds are checked to see which may be realised. One, e.g. the strongest, is encouraged onto the stage by supplementing its vote.

Sub-goal sequences:

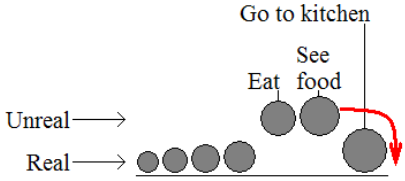
Execution mode:

The normal execution sequence of two sub-goals followed by an expiation goal:



On expiation (Eat) gain rises and links are formed in this direction ⟶

Initial sequence assembly:



If the sub-goals have been called in reverse order, "See food" then "Eat" may realise.

The relative strengths of links between the Ds of all kinds, formed after the creature has mastered a task, can be rather surprising [6]. This should be borne in mind when trying to instill behaviour patterns into the system by directly manipulating the links instead of by training.

A classic analysis of sub-goals was made by Holland [4] who rewarded each sub-goal with a fragment of the utility of the final goal. For this project, consideration was given to equating "PAE gain" with utility, and while this could have been attempted, the demands of information processing will often clash with the demands of the economics of energy or survival. For example, where the creature might be undergoing "sustained punishment", some kind of artificial gain must be added to produce an overall climate of gain, otherwise the conditions of constant negativity would prevent complex strategies from developing. The gain spikes awarded for realisation of each sub-goal along a sequence cannot be expected to sum to the value of the end goal. For the system to work the sum will often exceed the end goal, and anyway, subjective experience suggests that an end goal isn't less pleasurable the more you've worked for it! Also, as the creature's routines for dealing with all

situations approach perfection, link changes will need to tend to zero, and thus the gain schedule will tend to disappear, though utility considerations will still apply. It seems inevitable that multiple networks depending on different kinds of gain allow natural systems to "learn by useful play" while calculating separately whether a strategy is "worth it", and also to hold complex strategies/systems of D sequences under development under conditions of negative utility, but to stop developing them in environments fully mastered yet still of substantial utility.

5 P-links (Prediction)

The first additional network confers considerable extra potential and is a simplified version of the basic linking mechanism. In the original network, link increments were proportional to the gain applying at that cycle. The new links are calculated in the same way, just leaving out the gain factor. Increments of these links would never be negative, and would embody the number of cycles for which any two Ds had shared the stage in that sequence (although inevitably "close but less frequent" could be equivalent to "less close but more frequent").

This second system of links would also predict what Ds will be elected based on past experience, irrespective of the value to the creature of the outcome. By considering the rising and falling profile of the predictedness of the Ds as they are elected, stretches of higher and lower prediction can be identified, and Ds sharing a stretch of enhanced mutual predictedness can be bundled together into a compound demon (CD). (It turns out that these stretches of higher predictedness form ramps rather than plateaux.)

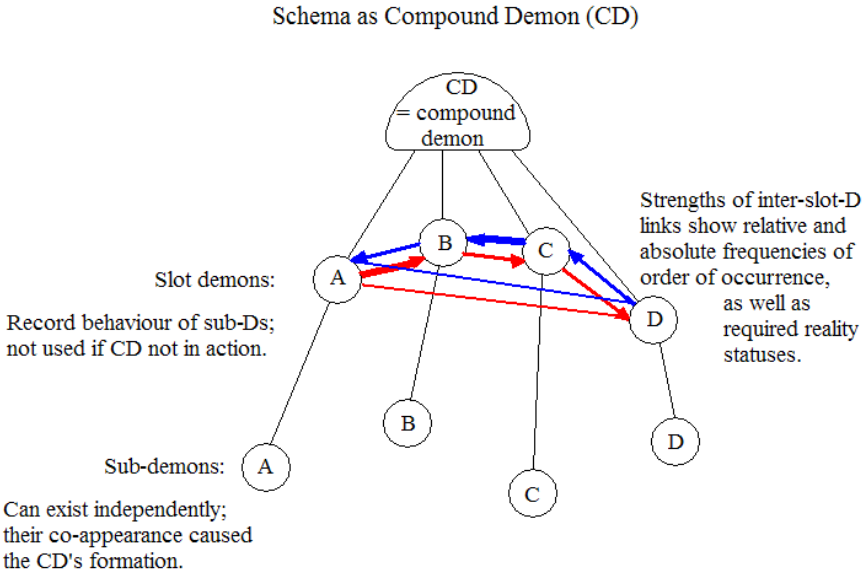
Demons, simple or compound, involved in high levels of prediction can be rewarded by high gain, especially if the prediction is higher than...predicted, thereby implementing the observed and useful preference for familiarity.

Links do not simply go from D to D; sometimes the creature may want something to be untrue, in other words the D in "unreal" state would be preferred, in which case it will vote for the unreal sense of that D. There are four "gain" links from any D “A” to any D “B”: to and from the real and unreal senses, and a similar four in the "predictive" system. (And another eight from “B” to “A”.) When a D is elected, the overall "reality" desired of it is noted, and efforts are made to make it so. Gain due to "realisation" can in fact arise from a D elected as real being changed to a preferred unreality. The gain granted to improving the reality of a sub-goal D will be commensurate with the gain expected of it by its electors (closely related to the average strength to it of their G/P links – see below).

However a very straightforward benefit arising from the predictive (P) link system is to vote, not according to the strength of the raw gain (G) links, but by G/P. This would give an expected gain as opposed to a cumulative gain, and as a result a long-established habit could easily be overcome by a new option with only one prior example, if that example gave more gain than the average of the old habit.

6 Schemata as Compound Demons (CDs)

The most important development is the implementation of "schemata". This allows branched sub-goal structures to be implemented, as well as compound percepts (and combinations of both), and logical operations. (The generation of a CD corresponds to induction, and the firing of a CD to deduction, which can incorporate Ands, Ors, Nors and Nots.) In the PAE, schemata are represented by Compound Demons (CDs).



A CD, at the head of the structure, is a proper stage-capable demon, as is a “sub-demon” (sub-D), and they tend to encourage each other to share occupancy of the stage. A sub-D can exist independently of the CD, may in fact be older than it, and even be a sub-D to other CDs, and a CD in its own right. However a “slot D” is an administrative ghost for storing, on its links, details of the performance of its sub-D when it is being a sub-D for this CD.

Any pair of slots could be required to execute in an exact order, or have no sequence restriction, or some intermediate status. The CD can only fire if enough pairs of slots match their sequence requirements adequately, and if enough slots match their own reality requirements adequately. (Recall that there is a link to each reality sense of a D; if the links to the unreal sense of a slot D are stronger than to the real, then the CD prefers the sub-D to be unreal). If all the links were of the same strength, it would mean their order was unimportant. If a sequence had to be in one direction, the slot strength in the other would be zero. If the slots weren't realised in a good enough order, their CD would not realise.

This provides a mechanism for expressing "both/all" or "either/some" or even "neither/none" and a host of grey-scale relationships, and allows us to represent sequentially constrained logical operations

6.1 Matching/triggering and firing

When prediction success is high enough, the sequence of Ds having recently crossed the stage is compared against each sequence defining a CD, and the best matching CDs will be given a vote supplement to encourage them onto the stage. Implicit in this process is the rewarding of CDs that have predicted the most recent stage-D sequence from earlier sections of the sequence. However, predictive reward must be mixed with goal ambitions for selecting the best next D, simple or CD.

The core calculation for the goodness of match might be expressed as follows: in each episode, only some of the possible sequences will occur, e.g. If sub-D “A” preceded sub-D “B”, that will be only links from A to B, and not from B to A. These are “episode links”. The total weight of those inter-slot D links currently defining the CD that are episode links is summed, and then divided by the total weight of all inter-slot D links currently defining the CD. In the diagram, the width of the blue (towards the left) and red (towards the right) arrows shows how often one of a pair of slot Ds has appeared before the other.

This metric of goodness of match will need to be complemented by other considerations, which we will never finish exploring.

From the matching CDs, one may be elected, but it won't necessarily have the desired reality on election. Achieving this is to "fire".

6.2 Tuning CDs

Each episode, differences between the slots' behaviour and the usual are noted, and considered against how well the CD is doing this time compared to usual. Updates to the standard definition of the CD (the inter-slot D links) will be made in the light of these comparisons.

6.3 Creating CDs

When a sequence of Ds has a profile of prediction strengths both high and relatively solid, the system checks for matches with existing CDs. If nothing good enough is found, a new CD is formed starting with Ds in the prediction ramp noticed so far and any further Ds in any relatively unbroken continuation of the prediction ramp that might follow. On completion of the initiating episode, the new CD is stored with inter-D and intra-D link strengths as the increments from that episode. Thus, concepts in the PAE will be born at a time and from material influenced by multiple past experiences, while case-based on a single episode, but with tuning will progress towards an idealised version, ideal in terms not entirely of “average” but largely of “wished for”.

6.4 Invocation and Passage of CDs

How should CDs execute? In the "bottom-up" mode, its sub-Ds will be elected first, and then through a combination of matching and gain/prediction voting, the CD. It will then tend to occupy the stage, allowing completed sub-Ds to leave but assisting the election of further slots. The idea is for components of a CD to sustain each other, and for particularly the CD itself to resist considerably the tendency to dwindle in strength with each stage cycle. (It will be necessary to maintain separately the age and the strength of Ds on the stage.)

Sometimes, particularly if deep hierarchies of CDs are being processed, a high-level CD body D will have left the stage, but the successful activity of its slots will tend to re-elect it. The system will know the sub-Ds belonged to that CD because they will also be flagged as currently active slot Ds.

As well as bottom-up, top-down invocation is possible. The CD itself can be elected, and will work towards achieving the reality level intended by those that elected it initially. In the analogy to an automated basketball player, it would recognise a moment when the "shoot for a basket" subroutine – i.e. CD - should be invoked, and then unfold it.

One way in which a CD might be elected top-down is through the use of the reverse voting principle mentioned already, with "unreals", where an answer to the wish "Oh for something that will do X" is provided. The system will vote for Ds with strong links to X in the real or unreal sense as required (which could lead to a solution as part of a chain). This principle really comes into its own with the CD as "class demon" instead of just "compound demon". When a problem can be expressed in terms of general principles, a solution could appear, also in general principles. The elected CD would specify the solution...in terms of slots again of a generalised nature, which would need to be instantiated. This would allow solving problems by analogy.

7 Conclusion

The many paradigms used in the cognitive sciences are all valid but they badly need to be integrated. This version of the PAE proposes to merge connectivist and symbolic approaches in a way that also embodies behaviourism and logic processing. In addition, the communication between and creation of new structures will be highly automated and flexible, and the system will also embody the perception, action and intermediate processes such as goal seeking that an agent needs, and provide the basis on which further enhancements may be readily implemented. Anticipation plays an essential role and in this system is first subdivided into predicted goals and predicted events irrespective of goal benefit, which in fact comprise the vast majority of the system – it's “G” and “P” links. These two combine to form an expectation of benefit for better comparison and choice of possible actions. Predictions of both kinds arise from memories of past sequences, and this allows current episodes to be compared with those in the past, stored in this system as summaries. All structures: data, process, perceptions, actions, goals, memories, predictions, complex objects and strategies are part of the same single sea of anticipatory links.

Acknowledgments. I am very grateful to Stan Franklin and Giovanni Pezzulo for advice and support in various ways, and to two anonymous reviewers.

References

1. Baars, B.J.: A Cognitive Theory of Consciousness. Cambridge University Press, NY (1988)
2. Franklin, S. Artificial Minds. MIT Press, Cambridge, MA (1995)
3. Franklin, S., A. Kelemen, L. McCauley.: IDA: A Cognitive Agent Architecture. In: IEEE Conf on Systems, Man and Cybernetics. IEEE Press. (1998); http://csrg.cs.memphis.edu/csrg/assets/papers/IDA - A Cognitive Agent Architecture.doc
4. Holland, J. H.: Properties of the bucket brigade. In: Proceedings of an International Conference on Genetic Algorithms. Hillsdale, NJ. (1985)
5. Jackson, J.V.: Idea For A Mind. SIGART Newsletter Vol. 181 23-26 (1987); http://www.jiscmail.ac.uk/cgi-bin/filearea.cgi?LMGT1=MINDMECHANISMS&X=&Y=&a=get&f=/PAE/IdeaMind_for_MM.html
6. Jackson, J.V.: The Pandemonium Association Engine: Explorations (1998) unpublished; http://www.jiscmail.ac.uk/cgi-bin/filearea.cgi?LMGT1=MINDMECHANISMS&X=&Y=&a=get&f=/PAE/paeksecondpaper_for_MM.htm
7. Minsky, M.: The Society of Mind. Simon and Schuster, New York (1986)
8. Pezzulo, G., Calvi, G.: A Pandemonium Can Have Goals. In: Proceedings of ICCM, Pittsburgh (2004) ?-?; http://www.istc.cnr.it/doc/1a_800p_ICCM2004.pdf
9. Shanahan, M.P.: Global Access, Embodiment, and the Conscious Subject. Journal of Consciousness Studies 12 (12) (2005) 46-66
10. Selfridge, O., Neisser, U.: Pattern Recognition by Machine. Scientific American, Vol. 203(2) (1960) 60-68
11. Whitledge, J.: Pandamat: Controlling an Animat with Pandemonium (1998); http://www.cswnet.com/~jwhitled/thesis.htm