A Neural Cognitive Architecture

Christian R. Huyck[a,*]

[a]*Middlesex University, London NW4 4BT, UK*

Abstract

It is difficult to study the mind, but cognitive architectures are one tool. As the mind emerges from the behaviour of the brain, neuropsychological methods are another method to study the mind, though a rather indirect method. A cognitive architecture that is implemented in spiking neurons is a method of studying the mind that can use neuropsychological evidence directly. A neural cognitive architecture, based on rule based systems and associative memory, can be readily implemented, and would provide a good bridge between standard cognitive architectures, such as SOAR, and neuropsychology. This architecture could be implemented in spiking neurons, and made available via the Human Brain Project, which provides a good collaborative environment. The architecture could be readily extended to use spiking neurons for subsystems, such as spatial reasoning, and could evolve over time toward a complete architecture. The theory behind this architecture could evolve over time. Simplifying assumptions, made explicit, such as those behind the rule based system, could gradually be replaced by more neuropsychologically accurate behaviour. The overall task of collaborative architecture development would be eased by direct evidence of the actual neural cognitive architectures in human brains. While the initial architecture is biologically inspired, the ultimate goal is a biological cognitive architecture.

Keywords: cognitive architecture, intelligent agents, learning, computational neuroscience

1. Introduction

The mind is extraordinarily complex, and to understand it sophisticated tools are needed. One set of tools for studying the human mind, which emerges from the behaviour of the brain, is cognitive architectures. A cognitive architecture is the fixed or slowly varying structure that forms the framework for the immediate processes of cognitive performance and learning [1]. This paraphrase is a cornerstone of research in cognitive architectures, and is consistent with many architectures (see section 2.1).

Human cognition is based on the behaviour of neurons. While Newell [1] based his architecture on a rule based system, he also partitioned cognition into several bands including the neural band. The coarse topology of neurons is slowly varying, so the coarse human neural topology is a cognitive architecture, a neural cognitive architecture.

One of the main goals of the Human Brain Project (HBP) is to devise a complete simulation of the human brain [2]. Simulation can happen using a variety of primitives, but the HBP is

*Corresponding author
Email address: `C.Huyck@mdx.ac.uk` (Christian R. Huyck)

particularly interested in spiking point neuron models. This simulation will then be used to help decode the function of the human brain [3].

Cognitive architectures are theories of how the mind works, but as typically used they are formal languages that can be used to develop new models that can be executed on a standard computer. Systems written in the architecture's language are then run and this behaviour is a model of cognitive behaviour. Each of these written systems instantiate the architecture, though no instance to date (or in the foreseeable future) instantiates a full model of a human. As these instances are executable programs, they are verifiable.

The architecture develops over time to become more effective and more closely approximate cognitive behaviour with new versions extending capabilities and adding constraints. Systems developed in these new frameworks more accurately reflect human cognitive behaviour.

This paper will propose a neural cognitive architecture. The proposal will include a sketch of an initial version of the architecture. This will include a development language, making use of spiking point neurons, that could be used in the near future in simulated spiking neurons on, for instance, the HBP's EBrains platforms. The paper will then propose ways to move forward from the initial architecture to develop more effective versions that more closely approximate neural and cognitive behaviour.

2. Literature Review

Cognitive architectures have been a research area for at least thirty years, and have been productive for furthering understanding of the mind, and for practical applications of agents, task analysis, and cognitive modelling. More recently, neural cognitive architectures have been proposed (see section 2.2); these show that neurons are capable flexible processing and memory devices. The HBP is a large interdisciplinary project with a primary goal of simulating the entire brain using spiking neurons (see section 2.3). It has been proposed [4] that Cell Assemblies (CAs) are a good intermediate level of granularity to study cognition, with neurons being a finer grain, and symbolic and statistical architectures being a coarser grain. CAs (see section 2.4) provide some support to bridge the gap in understanding between neural and psychological behaviour.

2.1. Some Standard Cognitive Architectures

There are many long standing cognitive architectures. The most popular are ACT-R[5], which is widely used for cognitive modelling, and SOAR[6], which is widely used for agents. A third popular architecture, EPIC [7], is used for task analysis and cognitive modelling. The sketch of these architectures below is necessarily far from complete.

ACT-R[5] is probably the most widely used cognitive architecture, and is particularly widely used for cognitive modelling. It is based around production rules and declarative memory chunks. There are restrictions and timing associated with production rules. The memory chunks can be embedded in an associative network and have varying activation that is related to the environment and the current task. The chunks are used to instantiate rules and give them activation values that can be used to select the supported rule to apply. ACT-Rhas sensory and motor buffers with limited capacity to provide input from and output to the environment. This can all be used to make specific predictions about performance, timing and failure on quite specific tasks. For example, one system has been used to model human driving behaviour [8], and these models have been used to show increased driving errors while, for example, driving and telephoning.

SOAR[6] has been widely used for agents in simulations (e.g. [9]). It again is based around rules, but further structures its operation around problem spaces. A particular problem can be broken down into subspaces, and the system will search for an operator that can make progress. If it cannot find an operator, it will search a subspace to find the answer. Once the answer is found, a new rule can be generated by 'chunking' the operation. Thus, the system learns new rules and becomes more efficient with practice.

EPIC [10] also makes extensive use of rule based systems, but unlike SOARand ACT-R, EPIC rules may fire in parallel. Like ACT-R, EPIC makes use of buffers, but the timing and capacity of the visual [7], motor and auditory buffers have been carefully measured.

EPIC is often used for task analysis, showing how a particular task is performed in a variety of conditions. For example, it has recently been used to demonstrate visual search behaviour [7]. Parallel rule firing enables the system to perform multiple tasks simultaneously, so that tasks can be analysed in isolation or in combination.

These architectures are supported by a large academic community. This community contributes to the development of the underlying theory, implementations of that theory, and implementations of instantiations. This community is, perhaps, the greatest advantage of these architectures.

It is interesting to note that each of these architectures, and many others, make use of rules for cognitive actions and other systems for other behaviour such as sensing. Indeed, as all are meant to model the mind, this community is developing a common framework [11]. However, unlike these architectures, the brain makes no distinction between the cognitive and subcognitive actions; they are all performed in neurons.

2.2. Some Neural Cognitive Architectures

It has been proposed that ACT-Ris a neural cognitive architecture because its structures can be cached out to a connectionist system, a Hopfield net [12]. ACT-Rmodels can be executed in standard computer machine code, or it can be compiled out to a Hopfield net, which can also be executed in machine code. As Hopfield nets can also be implemented in spiking neurons, this is a reasonable claim. None the less, the neurons do not learn in this system, and nodes in Hopfield nets are well connected and bidirectionally connected unlike synapses in the brain.

The most popular neural cognitive architecture is probably Spaun [13]. This makes extensive use of vector processing, simulating the vectors with groups of spiking neurons. Each vector item is made up of a group of neurons, and the more of those neurons fire, the higher the value of the specific vector field. It is likely that this is Turing complete. Spaun is instantiated in a single system that manages several complex tasks all in a single robot driven by spiking neurons.

Computational models of cognition developed by Howard and colleagues (e.g. [14]) make extensive use of the Laplace transformation. These transformations can approximate any function to an arbitrary degree of precision. As the transformations can be simulated by spiking neurons, they can perform any computation, and are thus also Turing complete. While the focus of cognitive simulation is on temporal behaviour and integrating evidence, it has been noted that this is a cognitive architecture.

There are many ways to calculate with neurons. Many are Turing complete [15], meaning they can perform any function that can be calculated. The question is not whether simulated neurons can perform a function, but how biological neurons perform a function. A brief version of the question for a neural cognitive architecture is what is the coarse neural topology and how does the dynamic executing version of that topology perform cognitive function.

A wide range of systems are Turing complete, for instance, Java is Turing complete. However, being Turing complete is insufficient for being a good cognitive architecture. What is important about a cognitive architecture is that it performs the calculations in the way that people do, at some band.

One benefit of a neural cognitive architecture is that it uses primitives, simulated neurons, that are close approximations of one Newell's bands of human processing, biological neurons [1]. However, a good neural cognitive architecture will use those neurons in the way that humans use them.

Neuroscientists make extensive use of animals. So, it is entirely reasonable to study animal cognitive architectures. Much of the standard architecture work is based on symbol processing, and makes a distinction between cognitive and subcognitive processing. This boundary seems ill founded, and the understanding of animal neural system behaviour is a very valuable contribution to understanding human cognition.

2.3. The Human Brain Project

The HBP is multifaceted, but a large portion of it is devoted to neural simulations. There are many neural models and neural simulators that can be used, and are used in the HBP, but for reuse and interoperability, there is middleware for specifying neural topologies; PyNN [16] is a Python API for specifying neural networks. PyNN provides several neural models, synapse models, input and output mechanisms, methods for specifying neural connections, and runtime control; new models can be added. The system thus specified can then be run through a backend simulator like NEST [17] or Neuron [18], or on neuromorphic hardware like SpiNNaker [19] or BrainScaleS [20].

NEST [17] can be run both on standard desk top computers and on supercomputers, and is particularly important for the HBP as a neural simulator because it is used on the project's high performance computers. When interacting with the Neurorobotics Platform (NRP see below) this, at least theoretically, provides for support for simulating a full human brain interacting with an environment. NEST has been used for simulations of millions of neurons [21], benefitting from the ease of parallelisation that are natural to neurons.

The HBP also develops neuromorphic computers, in particular there is work on SpiNNaker [19] and BrainScales [20]. SpiNNaker works on ARM chips, with a toroidal topology between chips that supports efficient message passing. This enables the system to, theoretically, support a billion neurons running in real time with biologically realistic connectivity. Practically, a million processor machine currently exists, with each processor able to simulate a thousand neurons. While SpiNNaker makes use of commonly available digital hardware to simulate neurons, BrainScaleS has neurons in hardware. This enables the neurons to be emulated at accelerated time, so network simulations can be run ten thousand times faster than real time. While SpiNNaker can be used to simulate large numbers of neurons in real time, it is practically being used for physical robots. BrainScaleS can theoretically simulate the life of a rat in a few hours.

The NRP [22] provides virtual environments and virtual robots to support interaction with brain models. Environments, robots, and physics can be specified on the platform. These can be linked to brain models, specified in, for instance, PyNN and NEST, so that an embodied neural agent can interact with an environment. The model can be run either faster or slower than real time. So, large scale neural simulations run on supercomputers can run much slower than real time, and still react as if they were in real time. Moreover, at least theoretically, the agent can run much faster than real time. So, with BrainScaleS accelerated hardware, it is feasible to run a year long experiment in a few hours. The NRP can be run locally, but there is extensive support for running over the Internet. The NRP is available to the public (as are all of the HBP's platforms), so anyone can use it for developing simulated neural robots.

The NRP is particularly important as brains are embodied [23]. That is, brains are not used for some arbitrary computation, but to move the body about the environment, avoid predators, eat, and reproduce. The NRP provides a mechanism to simulate environments, to have virtual bodies move about in those environments, and to have those bodies governed by brains. Currently those brains can be simulated by NEST or on SpiNNaker. Moreover, the NRP can be run over the Internet, where tools can be used to modify the environment, body, and brain. These modified experiments can then be run in a reliably repeatable form. The HBP platforms are available to almost all interested parties and can be found on https://www.humanbrainproject.eu/.

The standard cognitive architectures have input to the mind and output from the mind in the form of separate programs that fill buffers or read them. The NRP has input to the brain in the form of specific spikes to neurons. It then uses spikes from (typically different) neurons to perform motor behaviour.

2.4. Cell Assemblies

Unlike standard Von Neumann architectures [24], neurons do not separate memory and processing. The neurons act as memory items and together perform calculations. CAs are one widely agreed neuropsychological mechanism [25] that combines both memory and processing. Ongoing neuropsychological research confirms the importance of CAs (e.g. [26]), though work in simulation has been sporadic, and they have yet to be used in any formal cognitive architecture.

CAs are groups of neurons that have a relatively high synaptic connectivity. When sufficient neurons in the CA fire, they ignite a reverberating circuit of neurons within the CA. This persistently firing circuit acts as a short term memory item. The formation of the CA, via the creation of that connectivity typically by Hebbian learning, is a long term memory item.

While the neurons are firing, they perform some sort of calculation [27]. When the CA is the neural basis of a symbol, this is an active symbol [28], providing context sensitive processing.

The author is unaware of simulated models of CAs in spiking neurons that are biologically realistic, perform useful calculations and are robust. Fortunately, there are models of binary CAs that perform useful calculations and are robust. A binary CA can be created in a variety of ways, the simplest being a well connected set of neurons. If the neurons all fire in an instance, they will cause each other to fire again, and this will be repeated indefinitely. The synaptic delay enables the activation to propagate slowly enough to cause the neurons to fire again slightly later. The CA is binary because it is largely in one of two states on (firing) or off (not firing).

This means that the binary CA is neither continuously valued nor does it stop on its own. (Neurons that have adaptation can use oscillating pairs of neurons for a binary CA to overcome the adaptation.) Binary CAs are robust programming devices, so that systems built from them are reliable. For now, they are a useful placeholder for performing many computations in spiking neurons.

3. The Proposed Architecture

The long term goal of the proposed architecture is to build a neural network that closely approximates a human architecture in both neural detail and cognitive function. This raises at least three major questions. 1. What is the basic neural unit? 2. How are those units connected? 3. How does that neural topology generate cognitive function?

There are many neural models including simple rate coded neurons, point models and complex compartmental models [29]. What is the best model for an architecture? It is not at all clear at the moment, but a good place to start is with spiking point models; they can effectively approximate biological firing behaviour and are sufficient for the most popular current synaptic modification model [30].

How are those neurons connected? It is pretty clear that biological neurons are connected via synapses, but it is again not entirely clear the best way to model synapses. They are typically modelled with weights, and with point models, these weights are used to change the activity of the post-synaptic neuron after the pre-synaptic neuron has fired. However, biologically, these weights change at both long and short time scales in response to firing behaviour. The model of the connection is important, but so are the actual connections; what is the brain's topology? How are the 65 billion neurons in a typical human brain connected? There is work on decoding that connectivity [31], but the connectivity differs from person to person and over time, though slowly varying. A basic rule is that connectivity is sparse with neurons connected to at most thousands of other neurons. Connectivity also varies throughout the brain, and connectivity is largely local, with most synapses from a neuron going to nearby neurons.

How does the network generate cognitive function? This is the big question. While there are tentative answers about parts of the system, the full answer is not known. This is where neural cognitive architectures can be extremely helpful. These architectures can be used to explore neural and cognitive behaviour, while constraining it by known limitations. A basic theory can be developed and neural systems and subsystems can be developed within that theory. Once developed the systems can be used, reused, and compared.

While these three questions, and many others, remain unanswered, it is still possible to make progress. Processing with simulated neurons is sufficiently well understood, that an initial architecture can be developed. Simulations of neurons developed within this theory can be run, interacting with an environment.

Using point neural models, simulated neurons can implement rule based systems [32] and associative memories [33]; code can be found at http://www.cwa.mdx.ac.uk/NEAL/NEAL.html. A high level language is used to specify the rules and initial facts, and the associative memory. That is, generic rule based systems can be cached out to neurons. Similarly, simple associative memories can be cached out to neurons; both systems currently run on the HBP platforms.

A simple addition to these two high level specification languages, enabling the two to communicate, will create a simple cognitive architecture. Caching them out to neurons will make it a simple neural cognitive architecture.

This will be similar to existing standard cognitive architectures, such as ACT-R. These are based around rule based systems and associative memories. Standard architectures are of course much more elaborate, with for example buffers and rule complexity limitations.

It will be useable on the HBP platforms, so agents in virtual environments can be readily developed. Using the NRP, others will be able to modify the rules and associative memory to explore particular models, and, indeed, to find flaws with the model. Note that others will not need to understand neural processing at this stage; cognitive architecture users familiar with rule based systems and associative memory should be able to work with these models. Of course, those interested in exploring the neural behaviour will be able to work at that band.

The rule based and associative memory systems are based on binary CAs (see section 2.4). While these are poor models of CAs, they provide a place holder for better models of CAs.

Other systems (like vision) can be easily integrated. Like EPIC, sensory and motor systems can be readily integrated. For instance, a neural model of active vision could be plugged into the architecture, and used by models. Moreover, one neural active vision model could be compared with another.

4. Architecture Evolution and Exploration

The initial architecture is obviously far from a complete architecture. It will need to develop including more refined neural systems that more closely approximate cognitive function. The author does not want to prejudge this evolution, but would like to propose some plausible future steps. Moreover, others are encouraged in participating in the development of this architecture.

One simple neural mechanism to improve is to have better CAs. CAs have several behaviours, implied by biological, psychological and theoretical evidence, that binary CAs do not have. Unlike the binary version, CAs do not persist indefinitely; they become less active over time (seconds) and eventually stop firing, as this firing is crucial to short term memory. They can also be variably active with more firing from more active memories. They also perform a calculation [27]. One proposal is that semantic CAs perform the calculation of spreading activation, supporting priming, conflict resolution, and default reasoning. CAs from other domains (e.g. spatial CAs) will perform different calculations. It is hoped that these improved semantic CAs will readily replace binary CAs in the architecture in the form of associative memory, providing more cognitively accurate behaviour.

Perhaps the most important evolution of the architecture is to make instances of it learn from the environment. Neural learning is typically modelled by synaptic weight change. It is usually limited to Hebbian learning rules, so that the weight changes are based on the firing behaviour of the pre and post-synaptic neurons. Other options include synaptic growth and death, and neuronal growth and death. It is relatively simple to set up a neural topology and training regime to have the overall system learn a particular task. However, what is needed is not a simple one off task, but an ongoing change of the topology in response to the environment that enables the overall brain body system to perform better in the environment.

One example of learning is to have the system learn new CAs for objects in the environment. These CAs could then be co-presented with words to ground the symbols in the environment [34]. Learning is performed in standard cognitive architectures, but symbol grounding is not. Moreover, this is not a one off task, but should continue with the system modifying its topology, perhaps forgetting, and requires the brain body to exist for quite some time, perhaps months.

Hebbian learning is typically modelled as a long term phenomenon, which changes synaptic weight permanently. Short term synaptic learning is often neglected in simulations, but widely occurs in the brain. This may support the long term change, but can also be used for binding [35]. One relatively simple modification of the architecture is to use short term synaptic changes to support binding in rules. Learning is essential to the architecture, and these proposals are just example initial steps. It is likely that learning will be an ongoing concern.

Rule chunking is perhaps the major form of learning in standard architectures. This could be implemented in neurons by some calculation combined with Hebbian learning. The author, however, is unaware of any neuroscientific evidence of this chunking.

The basic premise of the architecture is simple, use the brain's neural structure. As it is not currently practical to explore the full neural topology, there is a great deal of latitude to explore within the architecture. This latitude includes several options, for example: neural models, synaptic models, working subsystems, cognitive models and different agents. Many current mechanisms work with simple spiking point neural models, but there is no reason to believe that these models are sufficiently accurate, so further exploration is possible. Fortunately, at least over short time periods (days), these models can be directly compared with behaving neurons. Of course, it is likely that different models will be needed for different types of neurons, and in different stages of the evolution of the architecture.

Different research groups are currently working on the behaviour of different systems in the brain. For example, a spatial reasoning and imagery model [36] could be developed that closely modelled biological behaviour. This would do a different type of neural calculation than rules, but would still be neural. This could then be integrated into the larger neural cognitive architecture. Other example subsystems include vision, episodic reasoning, and motor planning. There is also reason to believe that circuits through different brain areas perform complex operations [37]. When areas have multiple circuits passing through them, the problem becomes particularly complex. This is consistent with evidence that particular neurons perform multiple functions [38]

In general, these new subsystems will come along with cognitive models that fit neural and cognitive data. However, the existing architecture could be used to develop new neural cognitive models. For example, there is interest in developing neural cognitive models to account for linguistic data like back tracking during sentence processing.

Agents can be developed that function in virtual environments, including those on the NRP. While systems can be developed without bodies, agents provide a useful mechanism for interaction. The agent can learn from the environment, and interact with others, including humans, in that domain. This need not be a physical environment, but physical robots are a possibility.

The early stage of architecture evolution will probably focus on individual subsystems. Jackendoff's tripartite theory [39] provides some support for getting subsystems to work together. The tripartite theory was originally developed with language processing in mind, with three large language subsystems, the lexical subsystem, the syntax subsystem, and the semantics subsystem, working together. They each have their individual neurons, but communicate via shared neurons. This mechanism could be used for combining other subsystems. Learning can function within a subsystem.

Eventually subsystems will have to learn across subsystem boundaries. Eventually, versions of the architecture may support development from infancy. As the body brain system develops, it will develop new subsystems. These must be learned in collaboration with other subsystems. The lexical subsystem is learned at largely the same time as the syntax subsystem. The two influence each other making for a more robust overall system. This will be influenced by research in developmental neuropsychology.

Another division within the architecture could be between a slow, serial, explicit system, and a fast, parallel, implicit system [40]. This could easily work with the initial rule based system, and might work well with translating a rule based learning mechanisms. The slow system could work with one rule firing at a time, and this seriality could be enforced by a sort of program counter. As the system becomes more practiced with the rules, it may be able to transfer them to the parallel system by removing the program counter.

5. Conclusions

It is difficult to develop systems in neurons that perform tasks. An instantiated complete neural cognitive architecture is an enormous task. There are, however, known features of neural behaviour that are important that can simplify the search. This paper advocates that the neural cognitive architecture make explicit assumptions and note the evidence behind these assumptions. Almost necessarily, systems developed in the near future will not be complete brains as it is difficult to simulate all 65 billion neurons, and it is not clear how the neurons are connected. Five principles to guide the development of this architecture are proposed:

1. Use Spiking Neurons as much as Possible
2. Make Learning a First Order Problem
3. Make Short Comings Explicit
4. Share Code and Neural Systems
5. There is a Ground Truth

Currently, there are many widely used models of spiking neurons. There is a place for continuously valued models, and non-neural models in the developing architecture, but the community is technically sophisticated enough to work with spiking neurons. Moreover, spiking neurons are sufficient for very good models of synaptic modification (e.g. [30]), they are more biologically accurate than continuous or non-neural models, and there may be efficiency gains from neuromorphic computation. Continuous and non-neural models can be used for functionality as a place holder, but should eventually be replaced by spiking models. Eventually, point models may be replaced by more biologically accurate models, though it seems wise to have a good reason to do so.

Though it is not typical to program things in simulated neurons, it is reasonably straight forward. What gives neural systems their strength is to learn, so learning is an essential field of study in the architecture. There is biological evidence of learning, which is mostly Hebbian in nature, though it is both of the long term synaptic change variety and the short term variety. However, it is also important to note that learning is continuous. What gives the overall neural system stability is the overall dynamic of the system. Stable circuits continue to fire in a way that maintains their stability despite plasticity. The brain, and even relatively small groups of neurons are extremely complex dynamic systems.

When creating and modifying the architecture, and running systems based on the architecture, short comings should be made explicit. It is difficult to make systems in neurons, and as these systems become more complex, it will become more difficult. If simplifying assumptions are made, they should be made explicit. The initial prototype (see section 3) will be made using binary CAs. These are not good models, but are a reasonable place holder. The neurons that make up these CAs will not be plastic. Clearly, all things being equal, a model of binary CAs that has plasticity is better. Making assumptions explicit is not to say there is no place for argument and conjecture, but it is important to acknowledge weaknesses.

Code for the architecture and for examples will be made available for others to use. This provides a degree of verifiability, but it also helps others understand the systems, and enables them to use and modify it. Moreover, the code will be in a standard language, PyNN, working on at least one commonly available backend (initially NEST and SpiNNaker). People like to work with their own systems, but some degree of standardisation is needed for collaboration.

The final principal is that there is a ground truth, or ground truths. Brains really do work. While each brain is different, and even particular brains change over time, they do perform and can be measured. While developing the neural cognitive architecture, disagreements are bound to arise. Eventually, these disagreements can be resolved by referring to the actual biological systems that are being modelled.

Newell noted the problems of the divide and conquer approach to cognition, and proposed unified theories of cognition to address those problems. Cognitive architectures arose as unified theories of cognition. Neurons are a unified theory of cognition, but a divide and conquer approach will encounter the same problems Newell noted. What is needed is a neural cognitive architecture.

Acknowledgements

This work was supported by This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 720270 (the Human Brain Project).

- [1] A. Newell, Unified Theories of Cognition, Harvard University Press, 1990.
- [2] H. Markram, The human brain project, Scientific American 306(6) (2012) 40–55.
- [3] K. Amunts, C. Ebell, J. Muller, M. Telefont, A. Knoll, T. Lippert, The human brain project: creating a european research infrastructure to decode the human brain, Neuron 92(3) (2016) 574–581.
- [4] C. Huyck, Cell assemblies as an intermediate level model of cognition, in: S. Wermter, J. Austin, D. Willshaw (Eds.), Emerging Neural Architectures based on Neuroscience, Springer, 2001, pp. 383–397.
- [5] J. Anderson, C. Lebiere, The Atomic Components of Thought, Lawrence Erlbaum, 1998.
- [6] J. Laird, A. Newell, P. Rosenbloom, Soar: An architecture for general cognition, Artificial Intelligence 33,1 (1987) 1–64.
- [7] D. Kieras, Visual search without selective attention: A cognitive architecture account, Topics in cognitive science 11(1) (2019) 222–239.
- [8] D. Salvucci, Modeling driver behavior in a cognitive architecture, Human Factors 48:2 (2006) 362–380.
- [9] R. Wray, J. Laird, A. Nuxoll, D. Stokes, A. Kerfoot, Synthetic adversarites for urban combat training, AI Magazine 26:3 (2005) 82–93.
- [10] D. Kieras, S. Wood, D. Meyer, Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task, ACM Transactions on Computer-Human Interaction 4:3 (1997) 230–275.
- [11] J. Laird, C. Lebiere, P. Rosenbloom, A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics, AI Magazine 38(4).
- [12] J. Hopfield, Neural nets and physical systems with emergent collective computational abilities, PNAS 79 (1982) 2554–2558.
- [13] C. Eliasmith, T. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, D. Rasmussen, A large-scale model of the functioning brain, Science 338(6111) (2012) 1202–1205.
- [14] M. Howard, A. Luzardo, Z. Tiganj, Evidence accumulation in a laplace domain decision space, Computational Brain and Behavior 1(3-4) (2018) 237–251.
- [15] E. Byrne, C. Huyck, Processing with cell assemblies, Neurocomputing 74 (2010) 76–83.
- [16] A. Davison, D. Brüderle, J. Eppler, E. Muller, D. Pecevski, L. Perrinet, P. Yqer, PyNN: a common interface for neuronal network simulators, Frontiers in neuroinformatics 2.
- [17] M. Gewaltig, M. Diesmann, NEST (neural simulation tool), Scholarpedia 2 (4) (2007) 1430.
- [18] N. Carnevale, M. Hines, The NEURON Book, Cambridge University Press, 2006.
- [19] S. Furber, D. Lester, L. Plana, J. Garside, E. Painkras, S. Temple, A. Brown, Overview of the SpiNNaker system architecture, IEEE Transactions on Computers 62 (12) (2013) 2454–2467.
- [20] J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, S. Millner, A wafer-scale neuromorphic hardware system for large-scale neural modeling, in: In Circuits and systems (ISCAS), proceedings of 2010 IEEE international symposium, 2010, pp. 1947–1950.
- [21] M. Helias, S. Kunkel, G. Masumoto, J. Igarashi, J. Eppler, S. Ishii, T. Fukai, A. Morrison, M. Diesmann, Supercomputers ready for use as discovery machines for neuroscience, Frontiers in neuroinformatics 6 (2012) 26.
- [22] F. Roehrbein, M. Gewaltig, C. Laschi, G. Klinker, P. Levi, A. Knoll, The NeuroRobotic Platform: A simulation environment for brain-inspired robotics, in: In ISR 2016: 47st International Symposium on Robotics; Proceedings of, 2016, pp. 1–6.
- [23] R. Brooks, Intelligence without representation, Artificial Intelligence 47:1 (1991) 139–159.
- [24] J. Von Neumann, The computer and the brain, Yale University Press, 1958.
- [25] D. O. Hebb, The Organization of Behavior: A Neuropsychological Theory, J. Wiley & Sons, 1949.
- [26] E. Miller, M. Lundqvist, A. Bastos, Working memory 2.0, Neuron 100(2) (2018) 463–475.
- [27] C. Tetzlaff, S. Dasgupta, T. Kulvicius, W. Florentin, The use of hebbian cell assemblies for nonlinear computation, Scientific reports 5 (2015) 12866.

[28] S. Kaplan, M. Weaver, R. French, Active symbols and internal models: Towards a cognitive connectionism, Connection Science 4 (1990) 51–71.

[29] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. Bower, M. Diesmann, A. Morrison, P. Goodman, F. Harris, M. Zirpe, T. Natschalager, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A. Dafison, S. ElBoustani, A. Destexhe, Simulation of networks of spiking neurons: A review of tools and strategies, Journal of Computational Neuroscience 23 (2007) 349–398.

[30] G. Bi, M. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type, Journal of Neuroscience 18:24 (1998) 10464–10472.

[31] O. Sporns, G. Tononi, R. Kotter, The human connectome: a structural description of the human brain, PLoS computational biology 1(4) (2005) e428.

[32] C. Huyck, D. Kreivenas, Implementing rules in neurons, in: Thirty Eighth SGAI International Conference on Artificial Intelligence, 2018, pp. 21–33.

[33] C. Huyck, Y. Ji, Two simple neurocognitive associative memory models, in: Sixteenth International Conference on Cognitive Modelling, 2018, pp. 31–36.

[34] S. Harnad, The symbol grounding problem, Physica D 42 (1990) 335–346.

[35] J. Feldman, The neural binding problem (s), Cognitive neurodynamics 7(1) (2013) 1–11.

[36] A. Bicanski, N. Burgess, A neural-level model of spatial memory and imagery, ELife 7 (2018) e33752.

[37] R. Granger, Engines of the brain: The computational instruction set of human cognition, AI Magazine 27:2 (2006) 15–32.

[38] F. Pulvermuller, Neural reuse of action perception circuits for language, concepts and communication, Progress in neurobiology 160 (2018) 1–44.

[39] R. Jackendoff, Foundations of Language: Brain, Meaning, Grammar, Evolution, Oxford University Press, 2002.

[40] D. Kahneman, Thinking, fast and slow, Macmillan, 2011.