MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Hybrid Computational Models of Cognition as a Tool for Cross-Pollination Between Cognitive Science Paradigms"

verfasst von / submitted by

Christina Light, B.A.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 013
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Joint Degree Programme MEi:CogSci Cognitive Science
Betreut von / Supervisor:	Dipl. Ing. Dr., Paolo Petta

Acknowledgments:

First, I want to thank Paolo for giving me the space to challenge myself with this thesis, for providing the necessary guidance, and for being an exceptionally dedicated supervisor. Special thanks also belong to my lovely husband Colby and our wonderful flatmate Bianca, both of which have fed and emotionally supported me, especially in the last intense episode of writing. Another thanks to Colby for generously providing his copy-editing skills. And last but not least, thanks belong to my mother Rosemarie, who has enabled this general life trajectory for me.

Contents

Chapter 1

Introduction

1.1 Context

Since the upcoming of the field there has been serious debate whether there is the need for a unified cognitive *science*, or whether a loose assembly of disciplines that deal with cognition (cognitive *sciences*) is sufficient (Stephan & Walter, 2013). Within the Middle European interdisciplinary Cognitive Science (MEi:CogSci) Masters program I familiarized myself with many of the core disciplines feeding into cognitive science, wanting to soak up all that has been explored from different angles with respective methodologies that are interesting and important. However, the body of knowledge about cognition that was available in the early 2020s was confusingly copious and highly differentiated. Suspecting that this is anything but a trivial task, I began to wonder: How can an emerging cognitive scientist successfully integrate what is learned from research in the sub-disciplines into a cohesive picture? Or: “How could we possibly hope to understand how billions of neurons interacting with [tens] of thousands of other neurons produce complex human cognition, just by talking in vague verbal terms, or simple paper diagrams?” (O’Reilly, Munakata, Frank, Hazy, & Contributors, 2020, p. 9). The system in question is too complex to keep all the components and their interactions actively in mind. Some researchers explored having those interactions realized in an object, into which knowledge could b offloaded. This object, existing outside one’s own body and brain, would allow one to not lose track of the overwhelming mass of moving parts. The pioneer work regarding these questions has been performed by Allen Newell (computer scientist, cognitive psychologist), who opened the William James Lecture in 1987 with the statement, "Psychology has arrived at the possibility of unified theories of cognition – theories that gain their power by positing a single system of mechanisms that operate together to produce the full range of human cognition. I do not say they are here. But they are within reach and we should strive to attain them," (Newell, 1990, p. 1). Deeply rooted in the community of early AI research, the means he suggested for constructing such unified theories of cognition was the *computer* and its ability to host models that *behave*. The program implementing a system of mechanisms he called a *cognitive architecture*, and many such structures have been developed since.

1.2 Motivation

My specific fascination for *hybrid* cognitive architectures is rooted in my own lived experience. Hybrid architectures stress that human beings unite 'low-level' and 'high-level' cognitive functions, which aligns with my personal observations. Having a background in construction work and philosophy, I view the social strata as constructed around those professions critically. While labeling people as *doers* or *thinkers* because of their profession is a useful description of goal-related habits, it does not reflect their cognitive abilities. Being a *good* doer requires both levels of cognitive functions, and being a *good* thinker does so, as well. Philosophers *do* navigate spaces and use material objects (e.g., when they go to the library to find a book that they assume aids their thinking) and construction workers have complex thoughts (e.g. when they are planning which materials and tools are needed for a particular step in construction). By observing their behavior it becomes really difficult to asses where the boundary between 'low-level' and the 'high-level' functions lies. Thus, the systems I want to investigate are the ones that represent the integration of these cognitive functions.

1.3 Research Questions

On the theoretical level, this polarity is more evident. The *cognitivist* framework is (to this date) very suitable for describing high-level cognitive functions, while the *emergent* framework has more success describing low-level cognitive functions (Sun, 2023b, p. 15). In response, a third class (hybrid systems) emerged, one which strives to combine their explanatory strengths to maximize the reach and power of a model (Vernon, 2014, p. 21). What sounds like the ideal win-win combination turns out to be practically complicated and theoretically complex, because the paradigms partially make incompatible assertions about the fundamental nature of cognition (Vernon, 2022, p. 192). Further, cognitive architectures themselves being high-dimensional constructs means the options for combination are plentiful. This brought about the following research questions:

RQ1 How is this integration of seemingly competing theoretical standpoints realized?

RG2 Which theoretical problems may arise from such realizations?

1.4 Methodology

To develop my answers, I conducted a structured literature review inspired by the description of library research by sociologist Andrew Abbott (2014). As research on modeling paradigms and desiderata for cognitive architectures progressed, it appeared that, despite the large number of implemented hybrid cognitive architectures, descriptions of a hybrid modeling paradigm are short and unspecific, compared to the descriptions of emergent and cognitivist paradigm (Vernon, 2014, 2022). I collected and summarized materials that make it possible to understand the assumptions on which computational models of cognition have been generally built. For that, I consulted introductory literature on computational modeling of cognition, as well as technical reports describing specific architectures, reviews, and methodological comments. In doing so, I could describe some of the degrees of freedom available in designing cognitive architectures. This enabled me to lay out the diversity of approaches summarized under the label of hybridity. To make an informed choice about which architectures to analyze as exemplars, I consulted introductory works, as well as surveys on the development of the field of cognitive architectures. Despite the large number of hybrid cognitive architectures, only a few related publications discuss the aspect of integration in detail (Kotseruba & Tsotsos, 2020, p. 9). The explicitness of documentation and the focus on integration became key features for selection. This led to *Sigma* by P. S. Rosenbloom, Demski, and Ustun, and *SAL* (*Synthsis ACT-R Leabra*) by Jilk, Lebiere, O'Reilly, and Anderson being selected as exemplarsfor the present work. The order in which the methodological steps and the results are presented here does not resemble the actual flow of research work. It contained a lot of switching back and forth and 'reverse engeneering' what I needed to know in order to understand what I was reading. Even so, this thesis tries to attain progressive deepening of domain-specific knowledge and, in that, summarize my learning progress.

The rest of the thesis is structured as follows: chapter 2 gives the technical and theoretical background knowledge about the tool *computational modeling* in general. Chapter 3 is introducing cognitive architectures, and their relation to modeling paradigms and provides further knowledge for understanding the upcoming chapters. Chapter 4 narrows down the scope further toward *hybrid* cognitive architectures, to provide answers to the research questions. Chapter 5 synthesizes the findings of chapters 3 and 4 discusses the upsides and downsides of hybridization and points to future perspectives for researching hybrid cognitive architectures.

Chapter 2

Background

2.1 Models

Before diving into the description of models as scientific tools, it must be mentioned that, regardless of the method used, experiments and the data that arise from them are always theory-laden. What will be observed at any level is determined by the instrument utilized for inspection (Bogen, 2020). One can ‘see’ fairly different things though a microscope, a telescope, on an X-Ray, a voltage meter or through ‘normal’ human eyesight. However, being mindful of the mode of inspection and the constraints it puts on the observation is not the same as projecting biases onto the data provided. What one can see and how it is interpreted are two different things. That is the mental exercise that I want to engage in: keeping in mind what is produced (which degrees of freedom are constrained by the mode of observation), yet remaining open towards what still is freely moving. Furthermore, one must keep in mind that the method of choice for any one experiment only constrains this particular set of data, it does not rule out the possibility of other methods, yielding interesting results by fixing different degrees of freedom.

2.1.1 Models as Scientific Tools

The use of models is crucial in a variety of scientific contexts. A lot of journal space is devoted to analyzing and discussing the implications of models, and scientists invest much time developing, testing, comparing, and amending models. Next to their enthusiastic use there is a lively debate about the existing types of models, their respective ontological status, semantics, epistemological value and relation to theory (Frigg & Hartmann, 2020). The *type* of model central to this thesis is the *computational* model, whose semantics, i.e., relation to the target phenomenon, will be explained in the chapter 2.2.2. Hence, this section will focus on questions regarding epistemology, (relation to theory, i.e., the discipline of cognitive science). Being tangible abstractions of target phenomena, models perform a number of cognitive functions. With proper understanding of a model and how it relates to the target, models become vehicles for learning about the world (Frigg & Hartmann, 2020). In the next section the relation to theory is emphasized: models as tools for exploring the implications of ideas (McClelland, 2009, p. 12).

2.1.2 Why are Models Important for Cognitive Science?

The fact that models serve for learning about phenomena in the world, as well as exploring the implications of theoretical assumptions, makes them essential for the research program of cognitive science, which consists of bringing together all the sciences directly relevant to human thought, perception and experience (Stephan & Walter, 2013, p. 2). By communal effort toward the integration of those 'sub-disciplines' (see Figure 2.1) – the cognitive sciences – the new field was able to emerge. Yet, it soon became apparent that the comprehensive and multi-layered explanatory claims that drove the research program from the very beginning, cannot be fulfilled solely by cooperation of individual cognitive sciences. Many researchers have expressed the need for a *uniform* set of methodologies and theory. Certainly, recognizing the *need* for something like that is much easier than finding a suitable candidate and garnering discipline-wide support for it. Consequentially, a number of researchers reject the necessity or possibility of it, while others are, to this day, still striving toward unification (Stephan & Walter, 2013, p. 3).

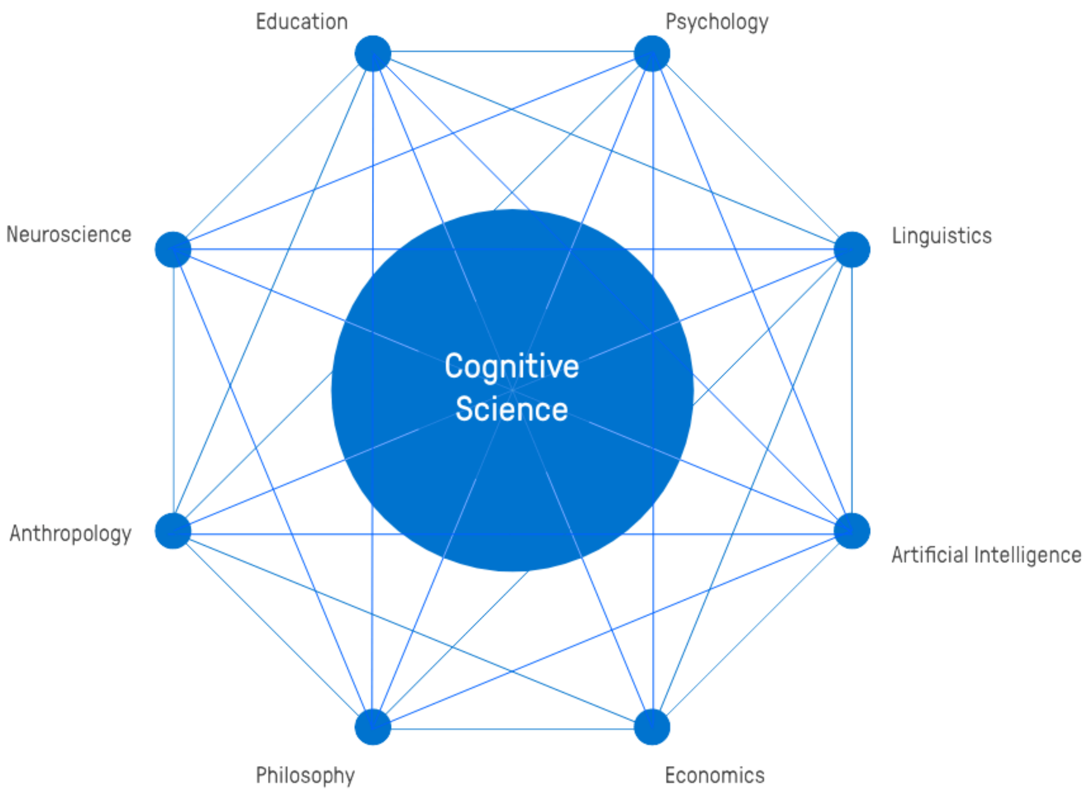

Figure 2.1: Logo of the Mei:CogSci study program depicting the scientific disciplines contributing to cognitive science. Source: (MEi:Cogsci, 2017).

For the latter, the computer became an important tool, with its ability to produce mechanistically detailed and 'behaving' models. As McClelland puts it, "The essential purpose of cognitive modeling is to allow investigation of the implications of ideas, beyond the limits of human thinking," (McClelland, 2009, p. 16). To do so is important, as researchers often rely on intuitive (or verbal-conceptual) theories when designing experiments and generating hypotheses: "However, without precise, mechanistic, process-oriented theories, most of the details of an intuitive theory are left out of consideration and the theory might be somehow vacuous, internally inconsistent, or otherwise invalid. These problems of an intuitive theory may not be discovered until a more detailed model/theory is developed," (Sun, 2023b, p. 8). Such a theory forces the modeler to specify the proclaimed mechanism explicitly and in a formalized way, producing transparency and tractability through mathematical simplification (McClelland, 2009, p. 24). Once developed, the model can, like any other theory, be tested, falsified, and refined in the light of experimental data (O'Reilly et al., 2020, p. 10). For the researcher this provides multiple learning opportunities; depending on the stage, the model fulfills different roles in the research process. As learning takes place in the construction, modification, validation and evaluation, (Frigg & Hartmann, 2020) computational models can be either pre-theoretical, post-theoretical, or theoretical (Sun, 2023b, p. 9). They can stimulate new thought, aid in hypothesis generation and exploration of possibilities regarding details of cognitive processes (Sun, 2023b, p. 8). In this way, computational models may serve as a 'multi-tool' for theoretical work in cognitive science.

The subset of computational models of cognition central to this master-thesis are the ones implementing unified theories of cognition. Those models are built on cognitive architectures, which function as a substrate for integrating knowledge from contributing disciplines of cognitive science (Newell, 1990).

2.1.3 Quality Criteria of Models

General usefulness of computational models does not imply their usefulness in every single instance; thus, evaluating a model is crucial. This needs to be done in the light of a model's *purpose*: The reason *why* a certain phenomenon (*what*) is being modeled can have a significant impact on *whether* and *how* a model turns out to be appropriate or not. This dependency will be discussed further in the ultimate-proximate distinction (see 3.8). Important questions to answer in that regard are (Blaha & Gluck, 2023, p. 1167):

- What is the scope of the phenomena of interest?
- How much is understood about those phenomena?
- Is there a theory of why those phenomena occur?
- Is there a model as an instantiation of that theory?

Further steps in determining model quality involve *verification*, (as in, the exclusion of internal processing errors/bugs). This must always precede the step of *validation*. Most of the methods for assessing validity of cognitive models have their origin in psychometrics and map adequately onto many computational cognitive models. Different dimensions include: *face validity, criterion validity, content* *validity, construct validity.* For models implementing unified theories of cognition, however, formal and comprehensive validity approaches are often not suitable, as they can become intractable or prohibitively expensive (Blaha & Gluck, 2023, p. 1167–1169). For guidance in model selection, researchers can draw on mathematical and statistical methods, such as Akaike's information criterion, Bayesian information criterion, likelihood ratio test, minimum description length, and cross-validation; for cognitive architectures, better methods are still needed (Sun, 2023b, p. 20).

The *sufficiency* of a model can be assessed through testing whether its predictions match the empirical data of the target phenomenon. To determine whether a sufficient model is also *necessary*, requires changing the *assumptions* of the model and evaluating whether its predictions still match the empirical data: if the modified model's predictions no longer match the empirical data, this is evidence for the model being necessary in addition to being sufficient. It can then be called an *appropriate* model (Blaha & Gluck, 2023, p. 1168).

2.1.4 Caveats in Model Construction and Evaluation

Still, one needs to understand that models do not *per se* have a meaningful relationship with the 'real system' they are depicting. This is why O'Reilly et al. suggest that models be constrained by data on as many levels as possible. Yet, it is important to not transfer the mystery of the original system into the model by adding too much detail. The goal must be to find (1) the simplest possible model that accounts for (2) the broadest range of data, which are the two dimensions of Ockham's razor (Lieto, 2021; O'Reilly et al., 2020, p. 23-24). Another aspect to bear in mind is raised by Lewandowsky and Farrell: "There are always several alternative models that vie for explanation of data, and we must select among them. [...] Model selection rests on both quantitative evaluation and intellectual and scholarly judgment," (Lewandowsky & Farrell, 2011, p. 5).

2.1.5 Multiplicity of Models that Fit

To this aspect I want to dedicate some special attention, as it is essential for the direction of impact this thesis is attempting. This work focuses on a subset of computational models of cognition (CMoC), ones that are built from cognitive architectures. These models are intended to serve as unified theories of cognition (UToC). More on what distinguishes UToC from other CMoC will be provided in chapter 3. For now it is sufficient to state that they embody a larger set of assumptions, try to account for a very broad range of data, and integrate knowledge from several sub-fields of cognitive science. Newell, when coining the terms *cognitive architecture* and *UToC* and suggesting the corresponding research agenda, was not oblivious to the multiplicity of fitting models. He explicitly called out the need for multiple UToC that potentially converge in the future. For reaching the vast scope he aims for, any given UToC is taken to grow towards it, setting out from some particular starting point. The starting point is arbitrary, but there are good reasons for him to start at a particular location, (i.e., his being rooted in particular types of research, and his noticing technological development that made this perspective seem fruitful at the time), but he never ruled out the usefulness of other starting points and focuses when building a UToC. Nevertheless, one must recognize that the chosen starting point heavily influences the direction in which the theory can develop: "Since not all things can be done first, a particular theoretical orientation gets some of its flavor from what it puts first" (Newell & Simon, 1972, p. 13).

Gelman and Loken also reflect on this type of path dependency. They describe the degrees of freedom researchers have in the production and statistical analysis of their data, using the picture of a labyrinth with multiple paths originating at every new turn to describe this process of making sense of data. "In this garden of forking paths, whatever route you take seems predetermined, but that [i]s because the choices are done implicitly. The researchers are not trying multiple tests to see which has the best p-value; rather, they are using their scientific common sense to formulate their hypotheses in reasonable way, given the data they have" (Gelman & Loken, 2013, p. 10). Despite the obvious differences in statistical models and CMoC (Farrell & Lewandowsky, 2018), I borrowed this metaphor for the thesis, as it is a vivid picture to help keep in mind the path-dependency of choices made in the process of designing a CMoC.

After this many caveats, problems, and notices of contingency, one might ask if that makes modeling a futile endeavor. However, "[t]he implication of this is not that models are useless, but that care in interpreting their successes and failures, as well as further investigation, is usually required. This makes modeling an ongoing process, just like other aspects of science," (McClelland, 2009, p. 16).

2.1.6 Unification

It has already been mentioned that integrating contributions from various disciplines in a consistent framework is important for cognitive science. Also, the subset of computational models that deals with this particular task has been touched upon in section 2.1.2. Before they are explained in more detail in section 3, this part preempts the characteristic of *unification*.

Bélanger, Potvin, Horst, Shtulman, and Mortimer – researching representational pluralism in human cognition – describe that humans are repeatedly confronted with bits of knowledge or theories about processes in the world throughout their lives, knowledge that contrasts their previous beliefs or experiences. The learning and acceptance of a new concept, however, does not usually replace the old concept completely, nor does a convergence or integration of sorts usually happen. Rather, the concepts exist next to each other. This is what the authors call *representational plurality*, which, according to them, is a normal feature of cognition (Bélanger et al., 2022, p. 6). They argue that the process of trying to make sense of these dissonances is cognitively effortful, so a more dissipative approach may be both justifiable on explanatory grounds and more cognitively efficient (Bélanger et al., 2022, p. 251). While this is a valid point about human cognition, and it can explain the reservations of some people taking on the demanding task of integration, it is not an argument good enough to direct what science should strive for.

Newell provided a list of reasons for wanting to achieve unified theories (Newell, 1990, p. 18):

1. Unification is always an aim of science
2. Bringing all parts to bear on a behavior
3. Bringing multiple constraints to bear
4. Increased rate of cumulation
5. Increased identifiability
6. Amortization of theoretical constructs
7. Open the way to applications
8. Change from discriminative to approximative style
9. Solve the irrelevant-specification problem

He further stressed that, despite the dissociations and modularities that have been found, it is *one* 'mind' that produces the whole range of behavior. Knowing those parts is important, but does not substitute for the necessity of having an encompassing theory that explains the function of these parts and why they exist (Newell, 1990, p. 18). Thus, "[a] unified theory becomes a repository of knowledge that fits together," (Newell, 1990, p. 21). While the process of achieving a UToC clearly is effortful, the result may be easier to comprehend, as it would rely on a single set of mechanisms for all of cognitive behavior (Newell, 1990, p. 15).

2.2 Computational Methods

The previous section described the importance of models to (cognitive) science, quality criteria, caveats and general remarks. What has not been covered yet are the theoretical issues regarding *computational* models, specifically. The first of these issues is the levels of analysis in computational systems; the second one is computational implementation. After those, the technical dimension of building computational models is discussed. It clarifies some notions that come up in chapter 4 and introduces dimensions like: input data, logic, problems, learning types, algorithms.

2.2.1 Levels of Analysis

First and foremost, levels of analysis are not synonymous with levels or dimensions of design. However, it may be that the way a researcher typically analyzes cognitive systems has an influence on the type of system they tend to design (see 3.8) According to David Marr's influential distinction, cognitive systems can be analyzed on at least three levels – the computational level, the algorithmic level and the implementational level. The computational level (3) is the most abstract one, which specifies the problem being solved (what and why). The algorithmic level (2) specifies the distinct rules of reasoning processes to achieve the problem solving (how). This includes which representations (data types of input and output) are used and how/by which algorithm they are manipulated. The implementational level (1) deals with the physical organization of the system that enables the implementation of the cognitive process (Sprevak & Colombo, 2019, p. 106). He further states that computation is multiply realizable, meaning that there is only a very loose coupling between the different levels (Stephan & Walter, 2013, p. 203). On the other hand, "Marr stressed, the appropriateness of any given algorithm depends crucially on the nature of the computational problem to be solved" (Sprevak & Colombo, 2019, p. 106).

On the first of Marr's levels a significant difficulty consists of finding a mathematical description that adequately reflects the complexity of the system under consideration, while being mathematically manageable and abstract enough to generate understanding. This is analogous to finding the balance in statistics between overgeneralisation and overfitting. Furthermore, the implementational level faces the difficulty of ascribing functional meaning to mechanistic biophysical processes (Stephan & Walter, 2013, p. 83). This leads to the problem of implementation, which I will discuss only briefly to not exceed the scope of this thesis in section 2.2.2.

Nevertheless, some researchers disagree with Marr's stance and advocate for a stronger coupling between the layers. Stephen, for example, declares that the central task of cognitive neuroscience is to clarify how abstract computational theories (at the highest level) and specific algorithmic-representational models for their solution (at the intermediate level) are implemented concretely in the biological system with its neuronal substrate (on of the lowest level). Adhering to the levels suggested by Marr, Stephen's overarching question is, how the understanding of these three individual levels can be brought together, suggesting a stronger interaction (Stephan & Walter, 2013, p. 78). Additionally, Sun, Coward and Zenzen (2005) argue in response to Marr's levels that separation of these computational details did not produce significant insights in relation to cognition (Sun, 2023b, p. 11).

Another approach that claims a strong interaction comes from Schöner and Kelso, who frame cognition as a dynamical process that can be analyzed on different levels of abstraction. This implies that the physical substrate (low level) of a given cognitive system crucially influences what is happening at higher levels. They also specify three levels that are bound together by mutual influence. The most abstract one is referred to as *boundary constraint level*, which represents tasks, goals, and initial /nonspecific conditions. Below it follows the level of *collective variables*, which represents coordinated states. Lowest is the *component level*, specifying how the system is realized (e.g. with nonlinear coupled oscillators or neural networks) (Schöner & Kelso, 1988). The major difficulty with this approach is the identification of order parameters (collective variables) and their dynamics (Schöner & Kelso, 1988, p. 1518), as it is still unknown how complex biological systems are coordinated to produce functionally specific ordered behaviors or spatiotemporal patterns. Even if the behavior of every degree of freedom would be known, there would still be the need to select only the relevant parameters to represent the system as a dynamical system (Schöner & Kelso, 1988, p. 1513).

2.2.2 Theories of Implementation

As indicated earlier, the process of attributing computational meaning to physical mechanisms is by no means straightforward and thus requires theoretical deliberation. The problem springs from the implementationist agenda of wanting to use the theory of mathematical computation to express the behavior of physical systems (Curtis-Trudel, 2022, p. 2). In order to 'bridge' the gap between mathematical and physical reality, a theory of implementation is required, Chalmers (1994) stated, as cited by (Curtis-Trudel, 2022, p. 3). It defines the conditions a physical system must satisfy in order to carry out a computation. By that it can provide the truth value for the (metaphysical) relation between abstract mathematical entities (e.g., Turing machines) and concrete physical systems (electronic PCs, brains,...). A theory of implementation is desired to be adequate in metaphysical, explanatory and extensional qualities (Ritchie & Piccinini, 2019, p. 194). Sprevak adds the qualities of being non-circular and naturalistic. He further states that a theory of implementation needs to answer the following two questions (Sprevak & Colombo, 2019, p. 176):

1. "Under which conditions is it true/false that a physical system implements a computation?"

2. "Under which conditions is it true/false that a physical system implements one computation rather than another?"

Sprevak provides the mapping account to implementation as concept to discuss, as it is intuitive and the basis of more sophisticated theories of implementation. "A physical system X implements a formal computation Y if there is a mapping f that maps physical states of X to abstract states of the formal computation Y, such that: for every step- wise evolution S→S´ of the formalism Y, the following conditional holds: if X is in physical state s where f(s)=S then X will enter physical state s´ such that f(s´)=S´" (Sprevak & Colombo, 2019, p. 177).

2.2.3 Triviality Arguments

Sprevak sheds light on different triviality arguments that have been brought forward by theoreticians since the 1980s. Such arguments attack the theories of implementation by asserting that they either attribute computation to too many physical systems, or that too many computations are attributed to a given system, or both. If such a loose mapping from physical system to computation is diagnosed, the computation is said to be implemented trivially. For example, Hinkfuss as cited by (Sprevak & Colombo, 2019, p. 183) refers to the state-changes on the micro-level on a pail of water being complex enough to potentially implement cognitive processes. Many of the arguments can be refuted, as they rely heavily on chance, make fuzzy assumptions about which systems are vulnerable to trivial implementation, or overlook that their account of triviality only applies to a small subset of abstract computations (Sprevak & Colombo, 2019, p. 183). However, Sprevak argues that the triviality does generalize to all abstract computers with finite storage. He does so based on three premises:

1. There is an isomorphism between the physical activity of any open physical system, A, (with a clock and dial) and any FSA, B.

2. If there is an isomorphism between A and B, and an isomorphism between B and C, then there is an isomorphism between A and C.

3. There is an isomorphism between any computer with finite storage, C, and an FSA, B (Sprevak & Colombo, 2019, p. 184).

This led him to conclude that, "triviality arguments are effectively unrestricted in this respect. If an abstract computation is physically implementable, it is trivially implemented" (Sprevak & Colombo, 2019, p. 185). This poses a threat to "computational functionalism (the metaphysical claim that implementing a certain computation is sufficient for having a certain mental state or process) and for computational explanation in science (the scientific practice of explaining mental and behavioral phenomena in terms of physical computations)" (Sprevak & Colombo, 2019, p. 175).

Although formulated slightly differently, Curtis-Trudel poses essentially the same questions about computational implementation, yet discusses different strategies to answer them and questions whether a theory of implementation is actually necessary. He proposes exploring two alternatives to the implementationist view. Instead of using the mathematical theory to develop the physical theory, the order could be reversed to construct an abstract computational theory in terms of physical computers (like theory of computability has done). Another option is to claim there should not be any deep relationship assumed between the two, meaning there is resemblance that might be informative but no one-to-one mapping or necessary grounding of features in one direction or the other (Curtis-Trudel, 2022, p. 29).

These two approaches I will call the grounding approach and the modeling approach. In both cases there is no metaphysically binding relationship between physical and abstract computation. Although deriving abstract computations from the behavior of a physical system avoids triviality, as it clearly attributes a specific computation to a specific (type of) physical computer, the choice of a fitting abstraction does not rule out the possibility of finding another useful abstraction that could describe the same system. For the modeling approach it similarly holds, as described earlier in section 2.1 and indicated by Curtis-Trudel, "models are useful tools for investigation in certain circumstances and certain respects, but bear no more theoretical weight than this," (Curtis-Trudel, 2022, p. 29). Framing the relation this way, both approaches essentially avoid a theory of implementation, as they do not intend to make metaphysical claims. These two approaches have been and are still used in the research displayed in this work, e.g., constructing computational systems according to think-aloud protocols of humans and general models of cognition. Does this make the construction of unified theories of cognition a futile endeavor? According to Newell, it does not. Next to his admitting that theories are for approximating, he stands against the Popperian belief that a theory once falsified has to be discarded. "How a theory is wrong is carried along as part of the theory itself," (Newell, 1990, p. 14). So will be the reminder that unified theories are approximations and epistemological tools.

2.2.4 Clarifications

In 3.6 I discussed the entanglement of AI and cognitive science, and also touched upon the flourishing and diversification of the field. This has brought about some other terms that need disentangled before I continue introducing the technical and mathematical components such as algorithms, logics and data types. I will begin with the differences and overlaps between AI, machine learning and statistics:

Artificial Intelligence (AI)

Despite its evolution, this early definition of Newell is still a fitting one to describe the breadth of the field: "AI is the study of the mechanisms of intelligence and all supporting technologies, extending to perception and motion, namely robotics. AI is itself part of computer science, which is the study of information and its processing in all its variety. AI provides the theoretical infrastructure for the study of human cognition," (Newell, 1990, p. 40). Naturally, not all researchers of the field make use of the full breath, as intelligence in AI can be defined either as closeness to human performance, or the formal definition of rationality. Further, the focus can be on internal thought process and reasoning, or behavior (Russell & Norvig, 2021, p. 1). These distinctions have lead to the formation of four possible goals, designing systems that *think humanly, act humanly, think rationally* or *act rationally* (Russell & Norvig, 2021, p. 2–4).

Machine Learning (ML)

Machine learning is commonly defined as follows: "A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (Murphy, 2022, p. 34). As such, machine learning is part of the field of AI. What separates them is that GOFAI operated under the presumption that intelligence could be manually created. As it turned out to be challenging and expensive to directly encode all the knowledge complex intelligent systems require, interest sprawled in using ML to speed up an AI system's ability to acquire new information. In fact, although being arguably incorrect, ML and AI are sometimes used interchangeably due to the close connections (Murphy, 2022, p. 62). The acquisition of knowledge in machine learning is based on extraction of statistical regularities in the data presented to a system. This begs the question: in which way are statistics and ML different?

Statistics:

First and foremost, statistics revolves around drawing population inferences from a sample, while machine learning aims at finding generalizable predictive patterns within data (Bzdok, Altman, & Krzywinski, 2018, p. 233). Inference is about building a mathematical model for the process that generates the data to foster understanding or test a hypothesis. Prediction aims at forecasting unobserved events or behavior, and without explaining the underlying mechanisms, aiding in optimal action selection (Bzdok et al., 2018, p. 233). Irrespective of researchers' goals, how effective one or the other can be done, depend on the type of data available. ML methods are particularly helpful when the data available are *wide data*, i.e., the number of input variables exceeds the number of subjects. Classical statistical modeling, on the other hand, was created for data with a low number of input variables and modest to moderate sample sizes. In case of increasingly growing number of input variables, the model that represents these linkages grows more complex and less tractable. As a result, the line separating statistical and ML techniques blurs (Bzdok et al., 2018, p. 233). Despite building on the same mathematical foundations, statistics and machine learning present themselves mostly as two disjointed research communities, which also shows in their terminology: "In statistics, the w parameters are usually called regression coefficients (and are typically denoted by β) and b is called the intercept. In ML, the parameters w are called the weights and b is called the bias," (Murphy, 2022, p. 41).
Summing up, despite the differences listed, AI, ML and statistics do have overlapping methods and use-cases. The term AI was used to include ML and GOFAI.

2.2.5 Problems in AI

The realm of AI is also structured according to the problems that are solved by a model. Classical problems in AI comprise:

Classification

Classification involves matching input data to an output space consisting of unordered and mutually exclusive labels, also called *classes*. Predicting to which class a particular input belongs is also called pattern recognition. Classification is part of the supervised learning techniques, as in, it requires a training set consisting of already 'truthfully' labeled pairs of input and label to start the learning process (Murphy, 2022, p. 35). A classic example is predicting whether a picture shows a dog, a cat, a (specific) human face, or specific objects. This can be achieved via neural nets, but if the input data are very structured (tabular data), it can also be achieved via decision tree (Murphy, 2022, p. 39). In any case, the prediction of the label does come with uncertainty – the possibility of misclassification; this so-called loss function is calculated and minimized during model fitting (Murphy, 2022, p. 40).

Regression

Regression is similar to classification, except that the output data are a continuous variable. This means the expected error needs to be computed differently; the most common way to compute the loss function is via mean squared error, as it penalizes large residuals more than small ones (Murphy, 2022, p. 41). Minimising this error can be achieved by finding the minimum of the loss function via *gradient descent* (see section 2.2.6). Depending on the complexity of the data, the regression can be linear or polynomial (Murphy, 2022, p. 44).

Clustering

Clustering is an unsupervised learning method. There the task is to partition a dataset into regions of similar points. The number of clusters emerging from the analysis is subject to a trade-off between model complexity and data fit (Murphy, 2022, p. 48). The difficulty of unsupervised learning is, however, to evaluate the quality of the output. This is often done via a density estimation, calculating the likelihood that the model assigns to unseen test examples. If high probability is assigned to actual data and low probability to a region data do not come from, this indicates the model has learned typical patterns of the data (Murphy, 2022, p. 50). Apart from the criticism that the model just learned to memorize data points, density estimations are also difficult to perform, especially as the dimensionality of data increases. Given that, the use of downstream supervised learning is a common method for evaluating the quality of clusters (Murphy, 2022, p. 51).

Standardized Problems

Standardized problems are created to demonstrate or test different approaches to problem solving. They are used as benchmarks for academics to compare the performance of algorithms, since it can be described in a clear, precise manner. Examples are grit world problems, or sliding tiles puzzles.

Real-World Problems

Contrary to standardized problems, real-world problems are those that are of pragmatic meaning to humans, such as robot navigation, route finding, or automatic assembly sequencing (Russell & Norvig, 2021, p. 70).

2.2.6 Types of Models in AI

To get an idea through which structures the problems can be addressed, here is a non-exhaustive list of different models used in AI.

1. **Neural Networks** are assemblies of multiple simple processing units. The algorithms employed, the number of units and layers, as well the way units are connected, all have a substantial impact on the capabilities of the network.

2. **Deep Learning** refers to a subgroup of neural networks with nonlinear activation functions and multiple hidden layers. They perform well at, e.g., feature extraction, by automatically decomposing the feature extractor into simpler functions (Murphy, 2022, p. 45).

3. **Graphical Models** work by decomposing complex multivariate formulas into products of simpler functions. These functions are then translated into graphs. From this representation the *marginals* of the individual variables as well as the function's gobal *mode* can be computed efficiently. The *meaning* of the graph is determined by the function it represents (P. S. Rosenbloom et al., 2016, p. 43).

4. **Expert Systems** are computer programs for solving real-world problems that would normally be done by a human expert. Classical expert systems encode human knowledge via symbolic representation and heuristics (Russell & Norvig, 2021, p. 17). However, neural network expert systems have also been explored (Gallant, 1993).

5. **Generative Adversarial Networks** In contrast to discriminative models like the SVM, generative models do not try to find the decision lines between predefined classes, but instead try to depict the actual distribution across a class. The practice of Generative Adversarial Networks is pairing those properties to train the networks. The generator network attempts to create fake data that mimics actual data so closely that the discriminator network cannot tell it apart from training data. When actual and artificial data cannot be distinguished by the discriminant function, this min-max game is finished (Schinagl, 2016, p. 3).

6. **Support Vector Machines** Classification problems can also be solved by Support Vector Machines if the data are being represented as vectors. This works through finding an optimal separating hyper-plane, which is determined by the so-called support vectors. They are the set of vectors from the two classes that achieve the minimum distance to the optimal separating hyper-plane.

7. **Search Trees** For navigating the state space of a given problem to the current to a goal state, search algorithms build search trees (also called decision trees), where the root corresponds to the initial state of the problem. By considering the actions possible from that state, the node (representing a state) is expanded (Russell & Norvig, 2021, p. 71). Depending on the

algorithm chosen, the expansion of nodes follows different patterns, e.g., follows the cost attached to certain actions, or strictly explores the depth /breadth of the tree first (Russell & Norvig, 2021, p. 72–74).

8. **Random Forest** Decision trees can also be used for classification. It becomes especially powerful as an ensemble method, meaning when it is combined with the method of *bootstrapping*. In doing so, many decision trees are randomly grown to each give their choice of class. Summing over all trees, the class most mentioned is selected (Bzdok et al., 2018, p. 234).

9. **k-Nearest Neighbour** The k-nearest neighbors algorithm (k-NN) is a non-parametric supervised learning method, which is sensitive to the local structure of the data. It can be used for classification and regression. For both a test point will be set in the dataspace, which specifies the input – the k closest training examples in a data set. In a classification the algorithm will return the labels of the assessed neighbours (Murphy, 2022, p. 594).

10. **Gradient Descent** A class of optimization algorithms for finding a local or global optimum in differentiable functions, which is used for minimizing the cost- or loss function of a machine learning model (Murphy, 2022, p. 135). It works through iterative computing of the functions derivative at a random starting point and following the gradient for determining the next point until a minimum is found. However, choosing an appropriate value for the learning rate is tricky, as there is the risk to jump over desirable points or get stuck in local minima. Further, it is very unlikely to hit a minimum directly, so a stopping point needs to be set, e.g., number of iterations. Gradient descent can be used as a first-order or stochastic method (Murphy, 2022, p. 323, 333).

11. **Backpropagation** is applied in Artificial Neural Networks and "makes it possible to determine, for each connection weight in the network, what effect a change in its value would have on the overall network error," (Thomas & McClelland, 2023, p. 35).

12. **Marker Passing** is a parallel, non-deductive, spreading activation algorithm. By propagating information through a semantic network, relationships between objects can be determined (Hendler, 2014).

2.2.7 Learning Types

Supervised Learning

The task is to learn the mapping from inputs (called features, covariates, predictors) to output (called label, target, response). The experience is given from a training set consisting of input-output pairs (Murphy, 2022, p. 35). Problems tackled by supervised learning are classification and regression. In classification the output space contains mutually exclusive labels called classes, which are not ordered in a particular way. It is also called pattern recognition (Murphy, 2022, p. 35). Learning the mapping from colored pixels to labels, is difficult but can e.g. be done by convolutional neural networks.

Unsupervised Learning

In unsupervised learning there are no pre-known target values and also no reward from the environment. The machine tries to recognize patterns in the input data that deviate from the structureless noise. From a probabilistic perspective, it means fitting an unconditional model of the form p (x), which can generate new data x (Murphy, 2022, p. 47).

Reinforcement Learning

In reinforcement learning a system or agent learns through interaction with its environment. This can be realized via policies that map actions to inputs derived from the environment (Murphy, 2022, p. 51). But the agent is not directly told which action to select, it needs to test out freely and gets rewarded or punished by the rule. A problem for the agent is that reward is only occasionally given, and once given, the reward cannot safely be attributed to a single action, as often many actions lead to a reward (Murphy, 2022, p. 51).

Learning in GOFAI

While the aforementioned types of learning can occur in GOFAI agents and systems, learning is not necessarily part of their intelligence and problem-solving. Problem-solving is achieved through combining symbolic representations (knowledge bases) with search algorithms to, e.g., build production systems or semantic networks (Sun, 2023b, p. 15). Thus, learning appears in the form of modification of prior knowledge (Russell & Norvig, 2021, p. 41). If the system or agent should be capable of learning, a few mechanisms are required: a learning element – responsible for making improvements, a performance element – responsible for selecting external actions, a critic – providing feedback to the learning element on how the agent is doing, and a problem generator – suggesting actions that yield new experiences and thus potential for learning. An external performance standard helps the critic to assess the agent's success, because through it, part of the incoming percepts can be processed like a reward (or penalty), as they provide direct feedback on the quality of the agent's recent actions (Russell & Norvig, 2021, p. 56–57).

2.2.8 Types of Logic

What justifies calling a classical AI system intelligent, even if it does not learn, is the use of reasoning strategies. Formal logic, a sub-discipline of philosophy, classifies different types of reasoning (induction, deduction, abduction) and makes assertions about how knowledge is represented and the rules to reason about it. The so-called *classical logic* that has been very influential for the field of computer science includes propositional (covering simple propositions = sentences as a whole) and first-order predicate logic (covering predicates and quantification) (d'Avila Garcez, Lamb, & Gabbay, 2009, p. 11). However, not the whole spectrum of human reasoning can be represented with classical logic, yet the programming of a computer requires a formal way of reasoning. Considering dimensions such as time, space, uncertainty, beliefs, knowledge and probability, multiple non-classical logics have been developed to tackle these issues (d'Avila Garcez et al., 2009, p. 14). Commonly used logics include modal, temporal, intuitiuonistic, nonmonotonic, conditional, epistemic, linear and fuzzy logic. The development of nonmonotonic logics, for example, is strongly connected with advances in modeling phenomena such as common-sense reasoning or belief revision (d'Avila Garcez et al., 2009, p. 14). Many of these logics rely on the notion of possible worlds for modeling an agent's beliefs or knowledge about the world, by adding modes like *possibly* true, or *necessarily* true, to the binary true-false distinction of boolean logic (d'Avila Garcez et al., 2009, p. 15).

The example of time is well suited to elaborate the difference between classical logic and non-classical logics. In classical logic the truth value does not and cannot change over time, which is suitable for fields like mathematics. In *temporal logic* the truth value of a proposition can vary with the flow of time, which is well suited to express problems handled in AI. As an example, the statement, 'I am writing my master's thesis' is true in the moment I am writing this, but was not true 5 years ago, and will also not be true at all times in the future (d'Avila Garcez et al., 2009, p. 15). Another way to extend the bivariate character of boolean logic is *probability theory*, which expresses the uncertainty inherent in things in the future, or past events that were not directly observed. To do so, set theory needs to be extended by the assumption that the probability over the set of possible variables sums up to one. However, in many cases, it is not possible to perfectly predict the exact output given the input. This can be due to lack of knowledge of the input-output mapping, which is called *epistemic uncertainty* or *model uncertainty*, or it can be due to intrinsic (irreducible) stochasticity in the mapping, which is called *aleatoric uncertainty* or *data uncertainty* (Murphy, 2022, p. 40).

2.2.9 Types and Origin of Input Data

The kind of processing suitable and possible also depends on the type of input data. Those can be sets of pictures, where size and available color are defined, or language inputs, e.g., sets of comments that are labeled a certain way, or sentences in one language paired with equivalents in another language (Murphy, 2022, p. 56). Additionally, the data can come directly from the environment through sensors of the systems (Russell & Norvig, 2021, p. 36).

Tabular data refers to low-dimensional data consisting of fixed-length vectors usually stored in an design matrix, where rows represent examples, and columns represent features. Other data are often reduced to fit into design matrices through a process called *featurization* (Murphy, 2022, p. 36). This is one popular way of pre-processing data before analysis. Another one deals with heterogeneous data types of variables in the matrix. This is the case in datasets where some, but not all, variables are vectors. Via *dummy encoding* the discrete variables can be converted to vectors (Murphy, 2022, p. 57). Language inputs, however, need different types of pre-processing. These include, bagging the words, removing filler words, dropping punctuation, stemming the words, and others (Murphy, 2022, p. 58). If the number of input variables exceeds the number of subjects in a given dataset it is called *wide data*, as opposed to *long data*, where it is the other way around (Bzdok et al., 2018, p. 233). Apart from crafted datasets, the data can also continuously be gathered from the environment in which an agent or system is embedded. This functions through *sensors* like cameras, sonar, speedometers, GPS, microphones, radar, touchscreens or else (Russell & Norvig, 2021, p. 43). While the use of sensory data has many upsides, especially for intelligent embodied agents, it also implies the risk of sensors being faulty. To make useful inferences from potentially faulty data available, it is helpful to rely on probabilistic reasoning, as it allows the extraction of useful information if the sensor is wrong in less than 50% of the cases (Russell & Norvig, 2021, p. 134).

2.3 Summary of Background

Section 2.1 displayed that computational *models* may serve as an important tool for theoretical work in cognitive science. However, there needs to be proper understanding of a model, for it to aid in learning about the 'world'. Evaluating a model is a crucial step of the modeling process. Nevertheless, there are always multiple different models that can be fit to the data, making selection among them necessary. The starting point from which a model is developed was identified as a key issue, because it heavily influences the direction in which the model can develop. The comprehensive and multi-layered explanatory claims that drove the field cognitive science from the beginning call for models built on cognitive architectures. Section 2.2 introduced theoretical issues that arise from using *computational* models. It listed differences and overlaps in methodology and use-cases regarding AI, ML and statistics. The term AI was established to include ML and GOFAI. For the key technical dimensions of computational model construction, a selection of possible options was presented and important vocabulary was established.

Chapter 3

Cognitive Architectures

Having established some basics about computational models, I can proceed to a subset of those and give an introduction to the field of cognitive architectures. The section covers how they are commonly defined, how the connection to paradigms in cognitive science is used for their classification, and how the field has developed historically.

3.1 What are Cognitive Architectures?

So far, the term *cognitive architecture* has been introduced through preempting Newell's definition. For him, a cognitive architecture is the structure intended to serve as a UToC. Such a theory shall provide a single set of mechanisms for all of cognitive behavior. It does not gain its unification though 'vagueness and platitudes;' rather, it shall unify the existing understanding of cognition, in a way that all parts *work* together (Newell, 1990, p. 15-16). This definition, however, was not referenced by the whole community researching CMoC. Rather, there were multiple overlapping conceptions of the term and ways to include or exclude certain systems in one's definition. At times, these descriptions have significant differences, but somehow point in a similar direction. For example, Polk and Seifert curated a roughly 1200-page book in 2002 without giving a definition, instead reprinting influential articles from the cognitive modeling community. However, many other texts, from journal articles to books, do come without explicit idiomatic definitions (Chella, Cossentino, Gaglio, & Seidita, 2012; Marcus, 2001). I started with a similar approach, by listing different ‘definitions’ – without claiming to fully represent the field – and then *operationalizing* them for this thesis. Ways by which operationalizations can be obtained are, e.g., goal-based, capability-based, and label-based.

Label-based The easiest way to decide whether a certain system is a cognitive architecture, would be to count everything that is - by trusted sources - called/labeled a cognitive architecture. This raises the issue of how to assess trust in sources and a general issue of labelling is touched upon in 3.6. Because of its circularity (a cognitive architecture is a cognitive architecture) this approach is of marginal explanatory value. Further, Iuliia Kotseruba and John K. Tsotsos (computer scientists) state that, despite some very prominent examples, the term is also often used in a less restrictive way, rendering this approach unreliable (Kotseruba & Tsotsos, 2020, p. 22).

Capability-based “A cognitive architecture is [defined as] a software framework that integrates all the elements required for a system to exhibit the attributes considered to be characteristic of a cognitive agent,” (Vernon, 2022, p. 193). The task, then, lies in identifying this set of core capabilities. Such lists are provided by many researchers, especially when producing literature that enables an overview of the field. The list given by Newell when he introduces the notions of unified theories of cognition and cognitive architectures includes: “Problem solving, decision making, routine action, [m]emory, learning, skill, [p]erception, motor behavior, [l]anguage, [m]otivation, emotion, [i]magining, dreaming, daydreaming,” (Newell, 1990, p. 15). A publication where the authors of every featured architecture directly provided their descriptions is published by Samsonovich (2010). It covers 26 implemented architectures, and the capabilities assessed are *learning, goal and value generation, and cognitive development* (Samsonovich, 2010). Kotseruba and Tsotsos focus on *perception, attention, memory, learning, action selection (practical applications)* in their review from 2020, which features 84 cognitive architectures. This review does not entail individual profiles (Kotseruba & Tsotsos, 2020). Vernon, citing this publication and all named capabilities, further adds *action, learning, adaptation, anticipation, motivation, autonomy, internal simu-*

lation, reasoning, and *metareasoning* (Vernon, 2022, p. 193). Cognitive scientist Ron Sun adds: *categorization and concepts, decision-making, planning, problem-solving, motor control, emotion, language and communication, and others* (Sun, 2023b, p. 18). Despite considerable overlap, what counts as a core capability (or characteristic attribute of a cognitive agent) is one of the degrees of freedom researchers have in both building and analyzing cognitive architectures; allowing for a multitude of different subsets to be construed. This roots in the fact that what is viewed as cognition strongly depends on the context and the tradition that informs the conception of it (Vernon, 2014, p. 7), but also springs from the sheer need to start with a particular capability (see 2.1.5). A discussion of the relevant theoretical traditions is held in the next sections; nevertheless, an approach like this raises more questions for defining, even generally, cognitive architectures. How many of these capabilities should be implemented for a system to count? Is there a hierarchy within the core capabilities? Is implementation a necessary condition? Or should it be seen as spectrum where architectures move toward fuller realization of core-capabilities as research on them progresses?

Goal-based The question posed at the end of the previous item implies a goal towards which research endeavors are headed. Indeed, this is another prominent operationalization, as in:

1. "[A] cognitive architecture is the overall, essential structure and process of a broadly-scoped domain-generic computational cognitive model, used for a broad, multiple-level, multiple-domain analysis of cognition and behavior," (Sun, 2004, p. 341).

2. "A cognitive architecture is a comprehensive computational cognitive model, which is aimed to capture the essential structure and process of the mind, and can be used for a broad, multiple-level, multiple-domain analysis of behavior. A cognitive architecture also carries with it theories of cognition and understanding of intelligence gained from studying human cognition," (Sun & Bringsjord, 2007).

3. "[T]his work is usually concerned with the goal of capturing the essence of human cognitive abilities. Even so, it is important to consider both the

sufficiency of a model as well as its ability to explain the patterns of data obtained in experiments," (McClelland, 2009, p. 20).

4. "In particular, a cognitive architecture represents any attempt to create what is referred to as a unified theory of cognition. This is a theory that covers a broad range of cognitive issues, such as attention, memory, problem solving, decision making, and learning. Furthermore, a unified theory of cognition should cover these issues from several aspects including psychology, neuroscience, and computer science," (Vernon, 2014, p. 64).

5. "Most of the surveys define cognitive architectures as a blueprint for intelligence, or, more specifically, a proposal about the mental representations and computational procedures that operate on these representations enabling a range of intelligent behaviors," (Kotseruba & Tsotsos, 2020, p. 22).

6. "[T]he notion of computational cognitive architecture denotes a comprehensive, domain-generic computational cognitive model that captures the essential structures, mechanisms, and processes of cognition [...] It can be used for broad, multiple-level, multiple-domain analysis of cognition and behavior. It addresses cognition in a structurally and mechanistically well defined way and provides an essential framework to facilitate more detailed modeling and exploration of various components of the mind," (Sun, 2023b, p. 9).

Comparing the differences in these descriptions, the following emphases occur: stressing mechanistic detail, the epistemiological value, the constraint *human* cognition, interdisciplinarity, quality measures/desiderata, extension of processes to be accounted for, or the behavior aspect. Not every description constrains their goal to *human* cognition, allowing the construction of fairly different systems. Other ways to conceptualize cognition may include biological (animal) cognition, or refer to an abstract concept i.e. *generic* cognition (see. 4.3). Stuart J. Russell and Peter Norvig (computer scientists) summarize a classification scheme for types of cognition as researched in current artificial intelligence (AI); it divides systems that: think rationally, think human-like, act rationally, and act human-like. To address this issue, Kotseruba and Tsotsos suggest using the term 'cognitive architecture' when aiming at modeling human cognition, and using the term 'agent architecture' otherwise (Kotseruba & Tsotsos, 2020, p. 67).

Despite the blurry boundaries, there are a number of systems which are central to the discourse and endeavor of striving for UToC, for those, there is little doubt about their being cognitive architectures in line with Newells proposal (Kotseruba & Tsotsos, 2020). Most of these systems self-reference as cognitive architectures and aim at implementing general human intelligence (with varying definitions of what that entails). Naturally, these systems lie at the complex end of a spectrum of computational modeling frameworks (O'Reilly, Hazy, & Herd, 2017, p. 2); from a cognitive science perspective, these systems can serve as epistemological tools for exploring unified theories of cognition.

My Operationalization The working definition I used builds on Newell's proposal of UToC; however, it also incorporates technical terms from recent definitions in order to aid my understanding: A cognitive architecture is a programming environment that embodies an initial set of assumptions which qualifies it to serve as a UToC. Such a theory shall provide a single set of mechanisms for all of cognitive behavior, by unifying the existing understanding of cognition, in a way that all parts *work* together (Newell, 1990, p. 15-16). To date, no architecture has reached this goal, although some already include a range of cognitive functions and still continue to encompass more as they are constantly developed.

Now that classification was discussed on the level of what counts as a cognitive architecture and what does not, the models can be further divided to get an overview. After all, in a study from 2020, Kotseruba and Tsotsos list 84 different cognitive architectures, 49 of which were still being actively developed (Kotseruba & Tsotsos, 2020). Characteristics used for grouping of cognitive architectures range from technique of implementation, modeling objective, application domain, scale and scope of the architecture, granularity closeness to biology, to single vs. multi-agent, nature- vs. machine-inspired, input data, and modeling paradigm (Vernon, 2014, p. 19).

When it comes to the theoretical assumptions about what cognition ultimately is, and their guiding influence in the process of model-building, the literature consistently classifies architectures according to their modeling paradigm, distinguishing between cognitivist, emergent and hybrid architectures. These paradigms are summarized with regard to their postulates, research focus, implications, strengths, as well as their problems and shortcomings in the subsequent sections.

3.2 The Cognitivist Paradigm

The cognitivist tradition can be summarized as being rooted in Good Old Fashioned AI (GOFAI), which entailed a focus on high-level cognition and stressed the interdisciplinarity and exploratory nature of cognitive science. Representation and manipulation of symbols combined with heuristics as a method for problem-solving were postulated to be the building blocks for intelligent behavior (Lieto, 2021, p. 8). The historic origin is covered in more detail in section 3.6.

3.2.1 Postulates

Two of the core postulates influencing cognitivist thought – the Physical Symbol System Hypothesis (PSSH) and the Heuristic Search Hypothesis (HSH) – date back to Newell and Herbert A. Simon (political scientist) in their ACM Turing Award Lecture from 1975 titled, 'Computer Science as Empirical Enquiry: Symbol and Search' as cited by (Vernon, 2014, p. 26):

1. "The Physical Symbol System Hypothesis: A physical symbol system has the necessary and sufficient means for general intelligent action."

2. "The Heuristic Search Hypothesis: The solutions to problems are represented as symbol structures. A physical-symbol system exercises its intelligence in problem-solving by search, that is, by generating and progressively modifying symbol structures until it produces a solution structure."

The first one states that systems displaying general intelligence are necessarily physical symbol systems PSS, and that every PSS of appropriate size has the necessary conditions to display general intelligence. The second hypothesis is fixating the means by which intelligence is realized, i.e., that PSS exploit heuristics to avoid explosion of search (Vernon, 2014, p. 27). Here the closeness of early AI and early cognitive science (especially cognitivist cognitive science) is obvious, as the two hypotheses, coming from a very technical motivation, are seamlessly woven into the cognitivist paradigm: "It is evident that, to a very great extent, cognitivist systems and physical symbol systems are effectively identical with one another. Both share the same assumptions, and view cognition or intelligence in exactly the same way," (Vernon, 2014, p. 29).Utilizing the analogy between physical computers and (human) minds is called *classical computational theory of mind*. With some subtle differences in formulation, the mind is viewed as a classical computational system in a very literal sense — as an interpretable, formal, symbol manipulator — of some sort. That means that all cognitive processes, like reasoning or visual perception, are classical computational processes (Samuels, 2019, p. 106).

Such a stance entails that cognition is representational in a particular sense: it requires the manipulation of explicit *symbols*. Those are localized abstract encapsulations of information that denote the state of the world, which is external to the cognitive agent (Vernon, 2014, p. 22). The manipulation of such symbols is rule-based, meaning it is implemented with if-then (production) rules (Kotseruba & Tsotsos, 2020, p. 24). Additionally, PSS also comprise processes and patterns, which like symbols, can be produced, destroyed and modified. That means the system can build increasingly abstract representations and reasoning about them, and, as a function of its processing, it can *modify itself* (Vernon, 2014, p. 28).

3.2.2 Research Focus

Based on the aforementioned postulates, the cognitivist tradition focuses on high-level cognitive functions, including the following (Langley, Laird, & Rogers, 2009, p. 145–149):

1. Recognition and categorization

2. Decision-making and choice

3. Perception and situation assessment

4. Prediction and monitoring
5. Problem-solving and planning
6. Reasoning and belief maintenance
7. Execution and action
8. Interaction and communication
9. Remembering, reflection, and learning

Although it has not been one of the main focuses of cognitivist research, PSS are *in principle* capable of a form of development through self-modification (Vernon, 2014, p. 28). After all, there was no clear cut-off criterion that would distinguish learning and development from each other (Newell, 1990, p. 20).

3.2.3 Implications

In section 3.1 the necessary distinction between architecture and model was mentioned implicitly. The way this distinction is made depends on the paradigm (see 3.3.3). In the cognitivist paradigm, the architecture is a developed system that defines how the agent manages the resources at its disposal. Once this is combined with knowledge and a task, the entity is called a model (Vernon, Metta, & Sandini, 2007, p.162). Typically, the task and the "symbolic knowledge [are] the descriptive product of a human designer," (Vernon, 2014, p. 24). If this denotional relation is true, knowledge can be easily transferred between agents.

3.2.4 Strength

As knowledge is usually produced independent of the model, it is very easy transfer knowledge from one system to another. This transferability is a big strength from the engineering point of view. It means that knowledge possessed by one agent is inherently compatible with the reasoning mechanisms of another agents, which allows such systems to exchange and share acquired knowledge (Vernon, 2014, p. 191). Also, cognitivist systems perform well in many higher-level cognitive function, like reasoning and planning, or tasks that involve serial order, propositional inference or hierarchical behavior (Boden, 2023, p. 1232).

3.2.5 Problems and Shortcomings

A downside of symbolic processing is that it runs into the *reference problem*. This problem appears when the system is in contact with the external world, because it then needs an interpretation function that anchors the internal symbols in the external world. This function, however, encounters an infinite regression when designed purely symbolic. Thus, symbolic systems have difficulties creating reliable sensorimotor interactions if the surroundings are complex, dynamic and noisy (Vernon et al., 2007, p. 161). The reference problem can be alleviate if the primitives of this interpretation function are designed sub-symbolic (Chella et al., 2012, p. 101), which is, one form of hybridization (see 4). The theoretical side of this is the *symbol grounding problem*, which questions the denotional relationship between the world and the symbols that supposedly represent 'things' in it.

3.3 The Emergent Paradigm

One of the motivations to work on emergent systems was disaffection with the ideas that drive cognitivism, especially the sequential, atemporal, and localized character of symbol-manipulation. In order to depart from that, three modeling approaches surfaced, which were subsumed under the emergent paradigm: *connectionist models, dynamical system models, and enactive system models*. The first two can be seen as complementary, in the sense that dynamical systems describe *macroscopic* behavior at an *emergent* level, while connectionist systems model *microscopic* behavior at a *mechanistic* level. This perspective enables the investigation of the emergent dynamical properties of connectionist systems in terms of attractors, meta-stability, and state transition, which arise from the underlying mechanistic dynamics. Besides, it opens the possibility of implementing dynamical systems models with connectionist architectures (Vernon et al., 2007, p. 157–159).

3.3.1 Postulates

What the three types of emergent systems share are their strong assertions about the *nature* of cognition. Contrasting the cognitivist approach, their definition does not need to refer to concepts such as representation and/or capabilities like problem-solving. The uniting factor of emergent architectures is the rejection of the claim which equates cognition to rule-based shuffling of symbols. More so, cognition should be seen as a *behavior* that is the consequence of self-organization (Vernon, 2014, p. 42). Ultimately, cognition is the process through which an *autonomous* system achieves viability and effective behavior in its environment (Vernon et al., 2007, p.155). This is reached through "continual self-organization whereby the agent interacts with the world around it but only in such a way as not to threaten its autonomy." (Vernon, 2014, p. 32). The agent's body plays a pivotal role in this process, as it is enabling the interaction with the environment and determines many aspects of what viability means (Vernon, 2014, p. 33). Thereby, cognition is viewed as a process that happens in the real-time interaction, yet enables the agent to "break[...] free of the present in a way that allows the system to act effectively, to adapt, and to improve," (Vernon et al., 2007, p. 151).

3.3.2 Research Focus

Similar to the definition, the research focus does not consist of a list of separate capabilities, but lies in the construction of systems that undergo development and manage to maintain their own autonomy. This includes continuous interaction with the environment; thus, instead of knowledge acquisition, most emergent approaches stress anticipatory skill construction (Vernon et al., 2007, p. 156). Vernon, von Hofsten, and Fadiga provided a list of desiderata (for more see 3.7) which specifies some lower-level characteristics which are necessary for emergent and developmental systems (Vernon et al., 2016, pp. 120–123):

- value system
- physical embodiment
- sensorimotor contingencies
- perception
- attention
- prospective action
- declarative and procedural memory
- multiple modes of learning
- internal simulation
- constitutive autonomy

3.3.3 Implications

In contrast to the designer-specified knowledge of cognitivist systems, emergent agents produce their ‘knowledge’ by induction in the constant interaction with the world. The result is an agent-specific ontology that depends on the history and experiences of the system and thus is determined by the richness of the action interface and which experiences the environment offers. The goal of this ontology is not to build a mental model of the world that is as close to 'the real world' as possible, but one that enables the agent to stay viable (Vernon et al., 2007, p. 155). The fact that emergent systems produce their own ontology by induction has multiple implications:

1. Emergent systems do not exhibit a semantic gap; in other words, they do not suffer from the symbol grounding problem (Vernon et al., 2007, p. 160). These systems do usually do not refer to their internal representations as symbols, but nomenclature is not what enables them to evade the problem of symbol grounding. The origin of their representations makes the difference, as emergent systems are acquiring their own representations, through the process of interaction with the external world. What is more, some emergent systems abstain completely from making use of representations for their conception of cognition (Lieto, 2021, p. 16).

2. For emergent systems, cognition is inseparable from 'bodily action:' without physically embodied exploration, a cognitive system has no basis for learning and development. "Emergent systems, by definition, must be embodied and embedded in their environment in a situated historical developmental context," (Vernon et al., 2007, p. 173).

3. Thus, there is a close linkage between a models 'body' and the knowledge and skill a model acquires. The knowledge that emergent system acquires is dependent on what is useful for it to stay viable; this in turn depends on the agent's body and the environment it was trained in. So its ontology is both agent-specific and body-specific (Vernon, 2014, p. 116).

4. That is why the differentiation between architecture and model for emergent systems has to be made differently than in cognitivist systems (see **??**). The architecture in this case is an undeveloped minimal set of mechanisms. When it is connected to a 'body' and exposed to a 'world' that can be explored, it becomes a model. However, this model needs to undergo an *ontogenetic* development to become adapted to the world it is coupled with (Vernon, 2014, p. 115). As a result, emergent systems are 'easier' to design, but

require a lot of training to produce useful behavior (Kotseruba & Tsotsos, 2020, p. 24–25).

3.3.4 Strength

Emergent systems do well with low-level cognitive functions like perception and motor-control, as they are less vulnerable to noise. Their flexible, efficient and robust processing enables the necessary reactivity for these tasks (Sun, 2023b, p. 15). Specific strength of connectionist models, dynamical system models, and enactive system models are presented in the respective upcoming sections.

3.3.5 Problems and Shortcomings

In emergent systems it is very complicated to transfer 'knowledge' from one model to the other. This is due to their agent- and body-specific ontology, e.g., learned sensorimotor contingencies of one model may be of no value for another model. Additionally, the representations in emergent systems are not localized, but are distributed over the system, which makes it complicated and time consuming to move them (Thomas & McClelland, 2023, p. 57). As the interaction with the environment is continuously shaping the internal structure of an emergent system, there is the chance that the acquisition of knowledge may include the deterioration of previous knowledge and skill (Kotseruba & Tsotsos, 2020, p. 25). Another shortcoming was the current lack of implemented systems. To date, only a few systems have been built. One prominent is the iCub humanoid robot by Giulio Sandini (bio-engineer), Giorgio Metta(electronic engineer), and their group at the Italian Institute of Technology (Lieto, 2021, p. 60).

3.3.6 Connectionist Models

The first type subsumed under emergent models are connectionist models, which get their inspiration from neuronal systems. They consist of a set of simple processing units, operating in parallel that are affecting each other's activation states via a network of weighted connections. Such a model generally consists of seven features (Thomas & McClelland, 2023, p. 32-37):

1. A set of simple processing units (neurons),
2. A state of activation at a given time for each processing unit,
3. A pre-specified pattern of connectivity between the units,
4. A rule for propagating activation states through the network,
5. An activation rule, specifying how incoming signals are affecting the activation of a receiving unit,
6. An algorithm for modifying the patterns of connectivity as a function of experience,
7. A representation of environment with respect to the system.

The representation described in item 7 is what is being crafted over the course of training. The non-symbolic distributed activation patterns arising from input of a given domain get ingrained into the network via change of its internal numerical weights. Through this learning mechanism, the network mirrors the statistical regularities found in the training set to produce effective behavior, instead of applying logical rules. This provides flexible, robust processing in an efficient manner (Sun, 2023b; Vernon, 2014).

The assembly of such simple processing units can take many forms; the *architecture* of a network has an influence on its properties, i.e., varying the number of input and output units, the number of hidden layers or the direction of signal propagation, affects how well certain problems can be solved. Still, all connectionist systems "are unified on the one hand by the proposal that cognition comprises processes of constraint satisfaction, energy minimization and pattern recognition, and on the other that adaptive processes construct the microstructure of these systems, primarily by adjusting the strengths of connections among the neuron-like processing units involved in a computation," (Thomas & McClelland, 2023, p. 32).

Implications

The difficulty of transferring knowledge is inherent in the principles of connectionism. Through learning knowledge becomes ingrained in the structure of a network. This results in four consequences for the architectures (Thomas & McClelland, 2023, p. 57–59):

1. There necessarily needs to be a distinction made between encoding *how things are at the moment* and *how things tend to be*. Active states contain a trace of recent events, while latent knowledge represents a history of experience.

2. The process of moving knowledge within the system takes time and is very costly.

3. As information processing and storage takes place in the same substrate, there is the need for an external strategic control to activate regions in long-term memory.

4. This changes the idea of domain generality in processing systems, because: "in contrast to traditional cognitive theory, there is no such thing as working memory as a general mechanism; rather it is a content-specific activity carried out in multiple systems," (Thomas & McClelland, 2023, p. 59).

Strength

The strong learning abilities of connectionist models makes them a useful framework for studying different trajectories of cognitive development, like developmental disorders, intelligence, or giftedness (Thomas & McClelland, 2023, p. 62). Also, they have improved the understanding of what modularity in a neuro-computational system might look like. Being framed as a partial property modularity could arise from a developmental process, where structure-function correspondences enable specialization of parts of a network (Thomas & McClelland, 2023, p. 61). A big success of the paradigm was the *interactive activation model* of McClelland and Rumelhardt, which showed that massively parallel constraint satisfaction processes can behave *as if* it contained rules. This model raised the question whether it is necessary to define rules as processing structures in order to explain the observable regularities in human behavior. For the interactive activation model, the connections were hardwired by the designers; its successor however, the past tense model, acquired the weights based on statistical regularities in the data it was exposed to (Thomas & McClelland, 2023, p. 45).

Problems and Shortcomings

While the occurrence of *deep* neural networks has greatly improved the range of tasks connectionist models can perform, it has also brought back into discussion some of the earlier reservations about parallel distributed processing. Since the processing inside of the model is not comprehensible for humans (so called black box model), it is questionable how building such models can enhance the *understanding* of cognition (Thomas & McClelland, 2023, p. 65). Deep neural networks also suffered from the vanishing gradient problem, which was caused by the structure of that back-propagation algorithm. It consists of the continuous multiplication of numbers below 1 with each other, which makes them increasingly smaller the deeper they lie in the network. This significantly slows down the adjustment of weights – thus learning is very time-consuming. A similar problem exists in recurrent networks (Thomas & McClelland, 2023, p. 36).

Further, through their way of processing, connectionist systems performed worse at precise calculation, which is elegantly described through symbol manipulation. For example, when presented the question *2 + 2 = ?*, a connectionist system would provide an approximation *3,9999* or *probably 4* as an answer (Boden, 2023, p. 1232).

3.3.7 Dynamical Systems Models

Dynamical systems are a very general type of model used in a number of scientific disciplines such as physics, biology, astronomy, ecology and many others. Dynamical systems describe patterns of behavior through a temporal dimesion (Vernon, 2014, p. 39).

Postulates

The systems are characterized by the following traits (Vernon, 2014, p. 40):

1. Open: things from outside the system can enter it and change its behavior (pertubation).
2. Dissipative: the system is losing energy over time.
3. Nonlinear: the relationship between variables cannot be described through weighting and addition of the components, but needs to be described through multiplication.
4. Non-equilibrium: the system never comes to rest.
5. Being a system: having multiple degrees of freedom.

Their ability to capture the behavior of high-dimensional systems with low-dimensional models is what distinguishes dynamical systems models from connectionist models (Vernon et al., 2007, p. 158). Dynamical systems models describe behavior by using differential equations which capture, how a distinct set of variables (also called *order parameters* or *collective variables*) that characterizes the state of the system, changes with time (Vernon, 2014, p. 39). All possible states of the system are expressed by the *phase space*, yet the *dissipative* nature of dynamical systems makes the phase space decrease in volume as time passes. Though the system never comes to rest, its behavior is not characterized by constant chaos, rather by a succession of relatively stable states (meta-stability). The regions surrounding these *meta-stable* states in the state space are called *attractors*, because once achieved they pull the system towards meta-stability, until a sufficiently strong perturbation changes the behavior (Vernon, 2014, p. 41). Learning in a dynamical system is characterized as a change in layout of attractors and phase space (Vernon et al., 2007, p. 158).

Vernon further lists two conditions that a dynamical system must satisfy, which are very close to the requirements for enactive systems. They must be interactive and self contained, meaning “any change in one component or aspect of the system must be dependent on and only on the states of the other components,” (Vernon et al., 2007, p. 158). Lastly, they also share situatedness, embodiment and real-time interaction with enactive systems, the last of which seems to be in contradiction with the what the emergent paradigm states about the nature of cognition, namely that it ‘breaks free from the present.’ But, all anticipatory capabilities – like self regulating, internal modeling and self-perturbation – are learned from real-time interaction with the world, yet work internally and separate from current experience (Vernon et al., 2007, p. 158).

Research Focus

The research focus of dynamical systems describes *macroscopic* behavior of high dimensional systems though using only few variables. This set of variables are called order parameters (or collective variables); they determine the emergent properties of the system in terms of attractors, meta-stability, and state transition. The major difficulty with this approach is the identification of these order parameters and their dynamics (Schöner & Kelso, 1988, p. 1518). Even if the behavior of every degree of freedom would be known, there would still be the need to select only the relevant parameters to represent the system as a dynamical system (Schöner & Kelso, 1988, p. 1513), (see also 2.2.1).

Strength

According to its proponents, dynamical systems exhibit many of the same qualities as natural cognitive systems including: multi-stability, adaptability, pattern creation and recognition, intentionality, and learning (Vernon et al., 2007, p. 158). All these qualities are achieved through self organization and do not make use of human-designed representations (Vernon, 2014, p. 46). Also, the property of nonlinearity provides the basis for modeling complex behavior, which is very suitable for modeling cognition. Not least the fact that high-dimensional phenomena can be represented with a very small number of relevant variables makes modeling with dynamical systems practical and attractive (Vernon, 2014, p. 40–41).

Problems and Shortcomings

Though dynamical systems models can account for a number of nontrivial behaviors that require the integration of visual stimuli and motor control, for example the perception of affordances, perception of time to contact, and figure-ground bistability, the viability of higher-order cognitive faculties has not yet been validated as a matter of principle (Vernon, 2014, p. 46). Further, Clark, as cited by (Vernon et al., 2007, p. 158), has pointed out that their departure from symbol manipulation has thus far only proven successful in cognitive abilities, which are not particularly 'hungry' for representations and the manipulation thereof.

Another weak point lies in the coupling of processing to the real-time interaction rate, which limits the speed of ontogenetic development. Because of the need to be embodied, ontogenetic development cannot be skipped or curtailed. In terms of epistemological value, Vernon et al. pointed out that the state-space dynamic, through it being an abstraction from the physical organization of the underlying system, does not help to identify organisational structures, that naturally occur in cognitive systems (Vernon et al., 2007, p. 158, 161).

3.3.8 Enactive Systems Models

Five crucial traits that characterize emergent systems according to Vernon are sense-making, embodiment, autonomy, emergence, experience (Vernon, 2014, p. 47). Especially the emphasis on *sense-making* and *experience*, take the concept of an agent-specific ontology, utilized by all emergent systems, even a step further. Instead of assuming a relatively stable mapping between representations and the external world, for enactive systems a constant real-time selection of relevance is claimed. The relevance is based on context and influenced by the actions of the system. The purpose of cognition is to uncover unspecified regularity and order that can be *construed* to be meaningful, thus, facilitate ongoing operation and development. That means enactive systems are based on a fundamentally phenomenological position, meaning that reality and relevance are dependent on the nature of the system (Vernon et al., 2007, p. 159). This assumption rests upon on findings that biological organisms can learn about their body an the environment they are embedded in through analyzing the dependencies between motor commands and consequent sensory data. Their interaction with the external world is not bound to any *a priori* model of sorts (Vernon et al., 2007, p. 160). Consequently, knowledge is particular to the history of interaction, like in other emergent systems. The architecture of an enactive system corresponds to its phylogeny: the innate capabilities of an autonomous system with which it is equipped at the beginning of its development (Vernon, 2014, p. 48). Vernon stated in 2014 that formal mechanistic, computational, or mathematical models of enaction remain goals for future research (Vernon, 2014, p. 34); as my research did not turn up any implemented cognitive architectures which are explicitly enacivist, this section remains as short as it is.

3.4 The Hybrid 'Paradigm'

Through a direct comparison of the respective strength and problems of cognitivist and emergent systems, one can see that some of them are complementary; e.g., while cognitivist systems are suffering from the reference problem and emergent systems do not, or, the transfer of knowledge being complicated and costly in emergent systems, while easy in cognitivist ones. This has motivated researchers to pursue the design of hybrid systems. At first glance the definition of hybridity seems easily made: a system that exhibits aspects of cognitivist and emergent systems. Ideally, it could manage to combine them in a way that gets the benefits of both without the disadvantages of either (Vernon, 2014, p. 53). While being an accurate description of the goal and direction of research – an umbrella term – it is lacking the conciseness found in description of cognitivism and emergentism. Hybrid systems were a growing and successful proportion of the systems implemented, however a "general solution to the problem of the ad-hoc integration of such heterogeneous components does not yet exist," (Lieto, 2021, p. 17). One reason for this lack of a red thread is the fact that there are no general restrictions on how the hybridization is to be done. The high dimensionality of cognitive architectures leads to a huge number of possibilities for hybridization – for example, Vernon lists 14 characteristics on which to compare the cognitivist and the emergent paradigm (Vernon, 2014, p. 58). Many of these options have been explored already, thus multiple taxonomies of the hybridization types have been proposed (Kotseruba & Tsotsos, 2020, p. 25). Those are featured in section 4.1.

Paradigmatic descriptions presented a less diverse picture. In a publication from 2014, Vernon devoted not even one page to summarizing the hybrid paradigm (p. 53–54), while the cognitivist stretched roughly ten pages (p. 22–32) and the description of emergent systems took up roughly 20 pages (p. 2–52) (Vernon, 2014). There he described only *one* type of hybrid, which still use symbolic representations. These representations, however, are not fabricated by a designer, but are constructed by the system itself through its interaction with and exploration of the world (Vernon, 2014). In an article from 2007 Vernon et al. described a slightly different type of hybrid, namely an embodied one. A main goal of embodied systems is to achieve behavioral plasticity, i.e., the ability to learn how to perform a task they were not explicitly designed for. Such hybrid systems were in many ways consistent with emergent systems, but may also exploit programmer-generated representations (Vernon et al., 2007, p. 161). Given this, the sections *postulates, research focus* and *implications* were collapsed into the following paragraph. Meanwhile, a deeper analysis of hybrid systems was deferred to section 4.1.

The encompassing *postulate* regarding hybrid systems is that it is possible and useful to integrate aspects from the cognitivist agenda with aspects from the emergent agenda. Therefore, the *research focus* is designing systems that realize such integration, exploiting the benefits of both paradigms and avoiding the disadvantages of either (Vernon, 2014, p. 53). As *implications*, Vernon listed their dependency on interaction with an environment, as well their "ignoring [of] some of the incompatible assumptions that the cognitivist and the emergent paradigms make about the fundamental nature of cognition," (Vernon, 2022, p. 192). More detail on how this integration can be realized is provided in section 4.1.

3.4.1 Strength

Successes for hybrid models have been shown regarding categorization, as well as memory and content issues (Lewandowsky & Farrell, 2011, p. 280). Lieto stated that hybrid *representations* and *processing* were main points of convergence. The strength of attaching relevant meta-data like frequency, recency, similarity or activation to representations of any type being processed, has led to wide acceptance of such integration (Lieto, 2021, p. 103). Sun, having developed a notable example of a hybrid cognitive architecture, argues that this approach is suitable to cover existing cognitive dichotomies such as: implicit versus explicit learning and memory, automatic versus controlled processing, and unconscious versus conscious perception. "Hybrid models have been used to address a broad range of issues, including human memory, concept learning, skill learning, reasoning, creativity, motivation, and even human consciousness," (Sun, 2023b, p. 16).

3.4.2 Problems and Shortcomings

While it seems promising to outsmart their shortcomings of either paradigm through hybridization, it is yet unclear whether its actually possible to combine these theoretically highly antagonistic stances (Vernon et al., 2007, p. 161). If so, the question would still remain, whether the integration is beneficial. In the proceedings of a workshop on hybrid connectionist symbolic models in 1996, Sun – who is a big proponent of hybrid systems – admitted that many hybrids inherit the difficulty in autonomous learning from their symbolic part, instead of leveraging the strength in learning of the connectionist part (Sun, 1996, p. 100).

3.5 Modeling Paradigms as Orientation Within the Field

Summing up, there are a couple characteristics that are essential to certain modeling paradigms, but one should not be fooled into believing that there is a simple checklist for identifying an architecture as belonging to this, that or both of the paradigms. In fact, there are frequently inconsistencies within the literature when it comes to assigning a paradigm to a certain architecture or the other way round. One reason some authors refer to the same architecture as hybrid, while others claim them to be cognitivist or neo-cognitivist, is a difference in classification criteria. For example, not every assumption is necessarily exploited to the same degree by each architecture. Further, cognitive architectures that are still actively researched are progressively evolving. Regarding the early architectures, researchers have had more than 40 years of time to question and readjust some of their assumptions. This can lead reviewers at one point in time to a justified assignment of a label, which at some later point in time does not hold anymore. For instance, Vernon et al. described Soar and ACT-R as cognitivist (in a paper from 2007 both are listed as cognitivist, while in a book from 2014 Soar is mentioned as quintessentially cognitivist (Vernon, 2014, p. 76)); meanwhile, a survey from Kotseruba and Tsotsos lists both of them as hybrids (Kotseruba & Tsotsos, 2020; Vernon et al., 2007).

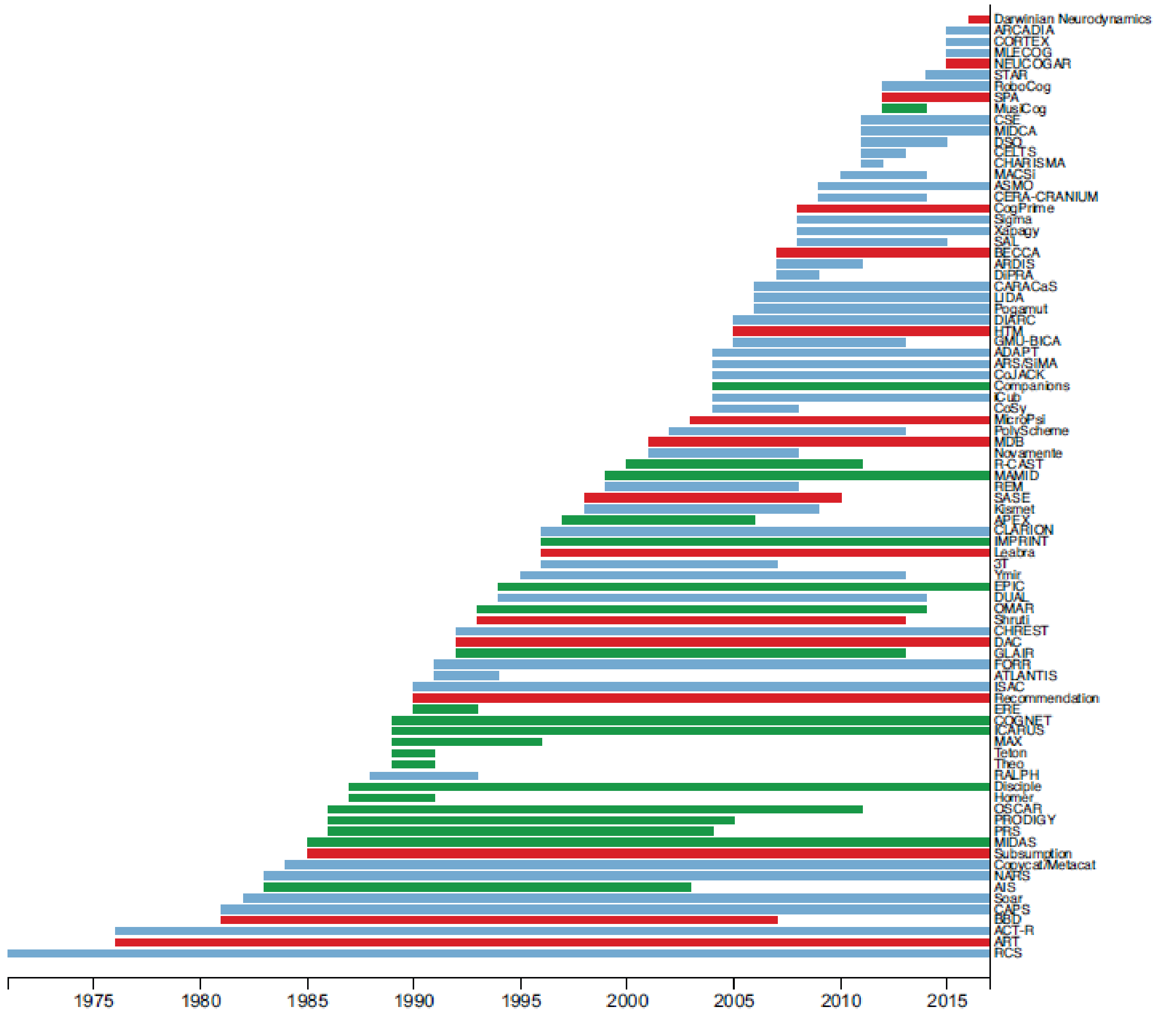

Figure 3.1: The graphic depicts the years of active development of all cognitive architectures reviewed in the survey of Kotseruba and Tsotsos. The colors represent the modeling paradigm as categorized by the authors. Red = symbolic, green = emergent, blue = hybrid. Source: (Kotseruba & Tsotsos, 2020, p. 20)

.

Consulting Vernon regarding the history of cognitive architectures, the origin of the term is commonly attributed to Newell and his publication 'Unified theories of cognition' from 1990. Vernon additionally suggests Newell and colleagues' pioneering work and their being rooted in cognitivism implies that, "it means something very specific in cognitivism," (Vernon, 2014, p. 64). Further, he states: "Cognitivism has undoubtedly been the predominant approach to cognition to date and is still prevalent.[...]However, as we will see, it is by no means the only paradigm," (Vernon et al., 2007, p. 153). To contrast this stance and recite some of the earlier complications, Figure 3.1 shall serve as a comparison. It is taken from a publication titled, ‘A Review of 40 Years in Cognitive Architecture Research: Core Cognitive Abilities and Practical Applications’ by Kotseruba and Tsotsos and depicts a timeline of work on cognitive architecture. Newer work of Vernon extensively cites this review, and the review does also cite some of Vernon’s articles. The reasons for this mismatch can be explained by the following factors:

1. **Classification/Exclusion of Data:** What is considered a cognitive architecture? Does the classification simply adhere to certain systems self-identifying as such, or are research endeavors counted that strive for particular goals, or with particular methods? The graphic shows cognitive systems already being developed before 1990 – the year the term supposedly was coined. This can be explained by the research objective origination in the early overlap of cognitive science and general AI in the 1950s – long before it was gifted the label ‘cognitive architectures’ that is now used to refer to such systems. If one, such as the authors of (Kotseruba & Tsotsos, 2020) opts for crafting a class based around goals, they are likely to include systems that do not carry the label, but might even exclude some that do. For example, the Global Workspace Theory (GWT) is not included in the review, despite being recognized by the authors as a not implemented cognitive architecture (Kotseruba & Tsotsos, 2020, p. 18).

2. **Representation:** In the graphic three dimensions are plotted; there are discrete steps on the vertical axis, where each level represents a single architecture; a continuous measure on the horizontal axis representing the timeline for publication and a color schema representing the paradigm the architecture. So each architecture is represented by a horizontal bar of color z = paradigm, and length x = start until end of development.

   According to Figure 3.1, the first six architectures that have been developed were either hybrid or emergent, which runs counter to the narrative of a cognitivist origin of the research field (which is not explicitly stated, but might be inferred from Vernon’s description). A closer look at the graphic reveals that these six earliest architectures include Soar and ACT-R that, by Vernon are classified as strictly cognitivist, but both are represented by a blue line, standing for hybrid. Contextualizing helps resolve part of the counterfactual. For both ACT-R and Soar, the story of a cognitivist origin is adequate, but over the years of development the architectures have strayed to include sub-symbolic processing, and thus, now self-identify as hybrid. Nevertheless, Figure 3.1 does suggest multiple modeling approaches being present from the origin of cognitive systems research, which relates back to (1) as in, what is considered as an architecture by reviewers. RCS as an example, being a multi-agent architecture, might not have been included in the review under altered selection criteria.

3. **Reasoning:** building on choices from before: i.e., which systems count, and how they are represented, Kotseruba and Tsotsos interpreted the plot as follows: "According to this data there was a particular interest in symbolic architectures since mid-1980s until early 1990s, however after 2000s most of the newly developed architectures are hybrid. Emergent architectures, many of which are biologically inspired, are distributed fairly evenly in the timeline, but remain a relatively small group," (Kotseruba & Tsotsos, 2020, p. 20). Given the plot this is mostly a fair judgment, but keeping in mind earlier choices, e.g., GWT not being listed (Vernon, however, lists it as an emergent architecture), the multi-agent architecture RCS (Albus & Barbera, 2005) being listed, choosing a representation that allows only one paradigm (color) for each architecture, and assigns that to the complete time of their development, may help in not religiously defending this interpretation.

3.6 The Origin of Research on Cognitive Architectures

Now that the historical development has been gently touched upon, the following section will continue to investigate the emergence of the research direction, as well as the development of the contrasting paradigms.

3.6.1 Demise of Behaviorism

Simon and Newell (broadly referred to as forefather cognitivists) traced their inspiration for their research back to Selz who, among other European psychologists of the time (pre-World War II), had focused on internal representation, thus departing from the dominating stimulus response doctrine of behaviorism (Newell & Simon, 1972, p. 874). Vital to this demise of behaviorism was the emergence of the computer as a tool, as it inspired researchers to think about the internal and thus unobservable mechanisms producing its behavior (McClelland, 2009, p. 12). A different approach for filling the vacuum in behaviorist agents was brought forth by the tradition of the Gestaltists and a publication of Boring (titled: The physical dimension of consciousness) from 1933. The explanatory mechanism they chose are the workings of the neurological substrate found within humans and other animals (which had the majority of American Psychologists wrongly identify stimulus-response correlations with neurophysiology) (Newell & Simon, 1972, p. 875).

3.6.2 Cybernetics

These developments culminated in a new stream that Norbert Wiener would coin *Cybernetics* at the end of WW II; it encompassed elements like information theory, theory of feedback systems and the electronic computer, and even more topics (game theory etc.) behind the Iron Curtain.

In these early days of cybernetics Newell and Simon state there being a close tie to physiology and biological feedback mechanisms and the concept of homeostasis (1972). The war also produced the field of human factors, researching performance and skills of humans especially in man-machine systems (gunners, pilots, radar personnel) who actively exchanged and overlapped with research in concept formation, thus installing the bridge between psychology and the newly emerging field of computer science (Newell & Simon, 1972, p. 881). In the 1940s scholars of Cybernetics proposed a unified study of organisms and machines. "In this perspective, the computational simulation of biological processes was assumed to play a central epistemological role in the development and refinement of theories," (Lieto, 2021, p. 9). Despite the fact that most of the prominent figures of this development were indulged into modern symbolic logic early in their careers, the symbolic computer was barely mentioned in the early days of cybernetics (Newell & Simon, 1972, p. 879). However, the simulation approach was inherited by early AI research dating back to the 1950s. Human cognition equally served as a reference point to, e.g., identify satisfactory solutions to a given problem, which would make complicated state spaces manageable. These were later called heuristics.

3.6.3 Early AI

In fact, the borrowing of concepts between psychology and research that would later be termed *AI* remained so pervasive that the field of cognitive science was born with it. Pivotal to the common origin was the 1956 summer-school at Dartmouth College, where Newell and Simon presented the logic theorist, a program that could automatically prove logic theorems (Newell & Simon, 1972, p. 885). Describing these early entanglements, Vernon states: "It is impossible to overestimate the influence that AI has had on cognitivist cognitive science and, to a slightly lesser extent, that of cognitivism on AI. This is not surprising when you consider that they were both born as disciplines at the same time by more or less the same people and that their goals were identical: to develop a comprehensive theory — inspired by the computational model of information processing," (Vernon, 2014, p.31-32). Although over time a distinction was made between systems that – in a mechanistically detailed way – model human intelligent behavior (as Information Processing Psychology), and systems that seek the most efficient way to solve particular problems inspired by human heuristics, both endeavors are recited as part of the cognitivist tradition. And although they continue to be researched according roughly to the field of *cognitively inspired AI /cognitive systems* and *cognitive modeling /computational cognitive science* (Lieto, 2021, p. 9), it is now not accurate to view either of the fields exclusively in the light of cognitivism.

3.6.4 Parallel Distributed Processing

Foreshadowed by Boring's assumption of the importance of physiology, Grossberg in the 1970s, as well as McClelland and Rumelhardt and their PDP (Parallel Distributed Processing) Research Group in the 1980s and 1990s, put forward the approach later dubbed connectionisim. Contrary to the continuity of the cognitivist school of thought, connectionism – which is now taken to be a strand of the emergent paradigm – was shaped by phases of intense research and hibernation. McClelland and Thomas locate the first summer of connectionisim in the 1930s and 1940s, where it was part of the genesis of the first formal theories of computation. The second arose in the 1980s and 1990s with PDP models of cognition (McClelland and Rumelhardt), and the third emerged in the mid-2000s with advances in 'deep' neural networks (Thomas & McClelland, 2023, p. 29).

3.6.5 Shared Inspiration

While the cognitivist paradigm was strong and influential in the beginning of the research, there has always been diversity within the approaches and focuses. Though grounded on different assumptions, the methodological apparatus developed by scholars in cybernetics was borrowed by both cognitivism and emergentism (Lieto, 2021, p. 9). Further, there is a curious convertibility between the two, as Vernon summarizes: "McCulloch and Pitts [1943] showed that any statement within propositional logic could be represented by a network of simple processing units and, further-more, that such nets have, in principle, the computational power of a Universal Turing Machine. Depending on how you read this equivalence, McCulloch and Pitts contributed to the foundation of both cognitivism and connectionism," (Vernon et al., 2007, p. 156).

Another curious case of mutual inspiration was that of Ross Quilian – PhD student of Simon – who invented the Semantic Networks in 1968. Those frame human memory as associative, meaning concepts are represented as sort of nodes in graphs, that can be reached through a mechanism of 'spreading activation.' This research equally paved the way for the development of the first graph-like, knowledge-based systems and formalisms, and led to improvement of connectionist, sub-symbolic systems (Lieto, 2021, p. 3).

3.6.6 Antagonizing of cognitivism and emergentism

It still remains a question though, what has led to the antagonizing of the two, which is presented so prominently in discourse nowadays. Boden, reflecting on the development of the field in an article from 2023, identifies two divides that have been heavily discussed. Next to the already discussed connectionist vs. cognitivist/symbolic dispute, she also introduces the dichotomy between situationist (explaining behavior in reflexes) and representationalist (explaining behavior with planning) approaches (Boden, 2023, p. 1229). At the time, these different understandings first appeared they did not result in outbursts of infighting, as all figures believed they were pioneering toward a common goal, just with different approaches. As a turning point she suggests a paper from 1969 by Minsky and Papert as cited by (Boden, 2023, p. 1228), which proves that Perceptrons can only solve linearly separable problems, which she argues, "is often blamed – to some extent, unjustly [...]– for the twenty-year connectionist 'winter,' in which virtually all the DARPA funds for AI were devoted to symbolic approaches," (Boden, 2023, p. 1228). She goes on to sketch Hofstadter's counter attacks on symbolism (1979, 1983/1985) and the vigorous defense of it by Fodor and Pylyshyn (1988) and a confident reissuing of Minsky and Papert's book (1988) as cited by (Boden, 2023, p. 1228). Nevertheless the Parallel Distributed Processing research group was celebrating successes at that time, proving that Minsky and Paperts argument could be levered out by introducing the backpropagation algorithm in their networks of simple processing units. And John Hopfield showed in 1982 that symmetrical neural nets modeling the function of associative memory necessarily evolve towards steady states, which was interpreted as attractors in the dynamic system theory as cited by (Lieto, 2021, p. 14).

The successes of emergent systems have also led to a movement that is completely abandoning the use of representations. It has been led by the roboticist Rodney Brooks, who proposed the 'Subsumption Architecture' in 1986, which relies on using 'the world as a model' for its cognitive functions (Lieto, 2021, p. 16). Yet, on the other hand, the severe limitations of such radical stances, but also the contrasting strength of the paradigms that were described earlier, have led researcher to explore hybrid systems.

3.6.7 Diversification and Unification

The mid-eighties were also the time in which the pioneering collaborations between AI and cognitive science became fewer, and the differentiation into several sub-fields, each with its own goals, methods, and evaluation criteria, emerged (Lieto, 2021, p. 18). Meanwhile, in giving the William James Lectures, Newell stated that "Psychology has arrived at the possibility of unified theories of cognition - theories that gain their power by positing a single system of mechanisms that operate together to produce the full range of human cognition. I do not say they are here. But they are within reach and we should strive to attain them," (Newell, 1990, p. 1). Through the lecture he developed the general thesis about unified theories of cognition, which curiously has become a sub-group of research trying to provide the counterweight to the fragmentation. Many such systems have been developed or fit to the research goal since then. However, this was achieved to some extent by "disjointed communities that speak different languages and frequently ignore each other's existence," (Samsonovich, 2010, p. 2).

3.7 Desiderata

Section 3.1 thematized what cognitive architectures are and section 2.2 thematized some techniques used to implement them. This section collects an additional set of criteria that should be taken into consideration, in order to build a *good* cognitive architecture. These *desiderata* take different forms depending on the researchers that propose them. Sun, for example, proposed a set of high-level desiderata, including: *realism* in a *cognitive, ecological* and *bio-evolutionary* sense, *eclecticism* of methods, and *minimalism* (Sun, 2004, p. 344-345). The latter, he described, can either mean a system that starts with minimal initial learning capabilities that enable the system to bootstrap itself, or one that starts with minimal internal structures and representations. Further, he urged researchers to distinguish between architectural features and innate capabilities (Sun, 2004, p. 343).
Many of these desiderata overlap with Newell's list of 'constraints that shape the mind,' which can also be understood as a list of desiderata. Yet, it is extended through clear features like: use of symbols, abstractions and language, learning from the environment and from experience, or having a sense of self. For the full list of 13 constraints see (Newell, 1990, p. 19). Appealing to that and the capability-based definitions in 3.1, researchers also list certain capabilities and characteristics as desiderata. Here, Sun's structures suggested the subcategory of human everyday activities: Modularity, bottom-up and top-down learning, synergistic interaction, dichotomy of implicit and explicit processes; as well as the subcategory of behavioral characteristics: reactivity, sequentiality, routineness and trial-and-error-adaptation (Sun, 2004).

Langley, Laird, and Rogers, suggested the following capabilities to be required for covering the full-fledged human-level intelligent activities (Langley et al., 2009, p. 145–149):

- Recognition and categorization
- Decision-making and choice
- Perception and situation assessment
- Prediction and monitoring

- Problem solving and planning
- Reasoning and belief maintenance
- Execution and action
- Interaction and communication
- Remembering, reflection, and learning

Another list of desiderata was given by Vernon et al.. Extending Sun's high-level desideratum of bio-evolutionary realism, they specified 10 lower-level characteristics necessary for emergent and developmental systems (Vernon et al., 2016, p. 120–123):

- value system
- physical embodiment
- sensorimotor contingencies
- perception
- attention
- prospective action
- declarative and procedural memory
- multiple modes of learning
- internal simulation
- constitutive autonomy

Apart from these criteria formulated for cognitive architectures more generally, certain authors also named key desiderata for the architectures they were currently developing. For example, P. S. Rosenbloom et al. selected *grand unification, functional elegance, generic cognition* and *sufficient efficiency* for Sigma (4.3). This leads to the next section, as the desiderata that P. S. Rosenbloom et al. set themselves for Sigma are explicit high-level design choices.

3.8 Design Choices

That cognitive architectures and the models built from them are high-dimensional structures was laid out by the preceding sections. The following section summarizes a broad list of choices that have to be made (implicitly or explicitly) in the process of designing such a system. Keeping in mind the quote mentioned in 2.1.5: "Since not all things can be done first, a particular theoretical orientation gets some of its flavor from what it puts first," (Newell & Simon, 1972, p. 13).

3.8.1 High-level Design Choices

There are two dualities that need to be addressed which are, to some extent, covered in the cognitivist vs. emergent paradigm. The first duality is *idiographic* vs. *nomothetic* systems. The distinction between them is rooted in differences between clinical efforts that are concerned with individuals in their full uniqueness (idiographic) and experimental psychology, where the individual only appears as a data point in a statistically defined population (nomothetic) (Newell & Simon, 1972, p. 10).

The second duality concerns the stance toward modularity – *compositional* vs. *holistic* systems – which will be addressed in section 4.1.1. It refers to the distinction between systems that consist of multiple heuristically separable components, and those that are built as a homogeneous unit.

The fact that those dualities are high-level distinctions, does not imply that those have to come first. If a researcher has a particular goal, e.g., to build a holistic, nomothetic system, it makes sense to define these first and let them infringe the options available on lower levels. But, if it is not in the particular interest of a researcher to fall under a specific label, they can also start building elsewhere. Nevertheless, decisions made on lower levels will influence whether the system in question is a compositional or a holistic one. Similarly, one can decide to make the the trade-off between *biological inspiration* and *computational theory*, a high-level distinction that guides development from the very beginning. However, it is not necessary to give this dimension priority, nor does it have to be answered in an all-or-nothing way. Especially in modular systems, it can be revisited for each single module (Vernon, 2014, p.19). Additionally, Vernon lists three aspects that must be borne in mind when modeling a cognitive system: the *level of abstraction* of the model, the *mutual dependence of brain, body, and environment*, and the *ultimate-proximate distinction*, referring to what cognition is for and how it is achieved (Vernon, 2014, p. 19). Each of these distinctions can be a possible starting point, or decided upon implicitly or explicitly further down the line. Thus, the first decision implicitly or explicitly is always the one of *choosing a starting point*. Before I proceed, it needs to be recognized that what I call *high-level design choices* could also be formulated as desiderata for specific architectures. I chose to do so, because describing them as design choices means spelling out trade-offs and dichotomies which are relevant in the construction process. Additionally, the section on desiderata (3.7) only contains characteristics that the authors of cited works name *desiderata*.

3.8.2 Inter-level Design Choices

So far I have listed seven high-level decisions. Vernon also provides 14 characteristics that can be used to understand the different stances cognitivist and emergent paradigm take on them. This list nicely serves to display lower level design choices. It includes: computational operation, representational framework, semantic grounding, temporal constraints, inter-agent epistemology, role of physical instantiation, perception, action, anticipation, adaptation, motivation, embodiment, and autonomy (Vernon et al., 2007, p. 163), and will be revisited in section 4.1.2. However, scale and scope of the architecture, single vs. multi-agent – and dimensions specific to hybridization (see 4.1) were not yet mentioned.

3.8.3 Low-level Design Choices

On the lower level, design choices refer to the direct technical decisions of hardware and software, as well as the type of input data; the explications in 2.2 show only a fraction of the possibilities hidden behind the characteristics *computational operation* and *representational framework*.

3.9 Summary of Cognitive Architectures

Section 3.1 summarized different emphases that researchers make when operationalizing the term *cognitive architecture* and gave my working definition of the term for this body of work. Furthe, the key stances of the cognitivist paradigm (3.2) and the emergent paradigm (3.3) were presented. Section 3.4 detected a relative shortness of description concerning the hybrid 'paradigm' in the literature consulted. The complication arising form labeling architectures regarding their modeling paradigm were assessed in 3.5. A historical perspective on the origin and development of the 'opposing' paradigms, revealed several points of mutual inspiration (3.6). The desiderata summarized in 3.7 are, similar to the goal(s) described in section 3.1, purposes to strive for. They describe an additional set of criteria that should be taken into consideration, in order to build a *good* cognitive architecture. Like a modeling paradigm, desiderata can also guide and constrain the process of model building, thus they may have an influence on which design choices are made. The design choices (3.8) describe trade-offs and dichotomies which are relevant in the construction process.

Chapter 4

How is Hybridization Realized?

As a lack in paradigmatic descriptions of the hybrid ‘paradigm’ was discerned in section 3.4, this chapter provides a more detailed discussion of hybrid cognitive architectures. Before diving into the analysis of specific examples, it gives an overview of different ways in which this hybridization can be achieved.

4.1 General Notes on Hybridization

In section 3.4 a relative shortness of paradigmatic descriptions concerning the hybrid ‘paradigm’ was detected in the literature that was consulted. This and the following recent quote by Vernon motivate the deeper analysis in this section: “Hybrid systems attempt to combine the cognitivist and emergent paradigms to varying degrees, quite often ignoring some of the incompatible assumptions that the cognitivist and the emergent paradigms make about the fundamental nature of cognition,” (Vernon, 2022, p. 192).

Recapitulating, section 3.4 listed a type of hybrid that still uses symbolic representations constructed by the system itself, through its interaction with and exploration of the world (Vernon, 2014). The second hybrid – focusing on embodiment – was described to be consistent with emergent systems in many ways, but may also exploit programmer-generated representations (Vernon et al., 2007, p. 161). What these descriptions share is that they leave the original paradigms relatively intact and do not engage in a structural analysis of the possibilities for different systems to be construed.

4.1.1 Taxonomy

In the spirit of combining advantages of cognitivist and emergent approaches, many possibilities of hybridization have been explored. Apart from representation, a taxonomy of hybridization can look at granularity, modularity, heterogeneity, coupling between symbolic and sub-symbolic components, as well as types of learning and memory (Kotseruba & Tsotsos, 2020, p. 8). In their review, Kotseruba and Tsotsos differentiated simply between hybrids that are *fully integrated* and those exhibiting *symbolic sub-processing* (Kotseruba & Tsotsos, 2020, p. 25). Within hybrids there are single-module and multi-module models, the latter of which can be homogenous or heterogeneous (Sun, 1996, p. 100).

Hybrid Representation

Generally, representation can be symbolic, localist or distributed, and the type of representation facilitates certain types of knowledge. Symbolic knowledge is usually more easily represented through symbolic or localist models; the representations produced in connectionist learning, on the other hand, are usually distributed (Sun, 1996, p. 100). When tackling the combination or the mapping of the representation types onto another, two routes can be chosen. Each one relies on a (1) *static topology*, or on (2) *network dynamics*, which in turn can both be translational or transformational. In the *direct translational approach* for (1) a network structure is created that directly corresponds to the symbolic structure. Usually this works via a localist network. In the *transformational approach*, however, the equivalent of symbolic structures is created in connectionist networks without actually embedding the structures directly in networks. This can work through the encoding of trees, for example (Sun, 1996, p. 100). A different route was chosen for Sigma (see 4.3), as it is built on factor graphs; the representation in factor graphs is continuous at its core, but also supports discrete and symbolic processing (P. S. Rosenbloom et al., 2016, p. 47).

Hybrid Learning

Possibilities for hybrid learning include combining of several back-propagation networks to achieve instructability through simple symbolic prompts, or learning symbolic knowledge in a bottom-up fashion with connectionist networks through reinforcement learning (Sun, 1996, p. 102).

Another way to achieve hybrid learning is through building neural-symbolic networks. This can happen via carefully engineered network ensembles, where each network is responsible for a specific task or type of logic. By that, the overall model can become very expressive despite its relatively simple components. Although it is not necessary, the modularity has advantages for comprehensibility and maintenance (Besold et al., 2017, p. 6). Further, neuro-symbolic integration can be reached through fibring, which is one of the standard methodologies of neural-symbolic integration. Fibred networks consist of classically interconnected neurons that are connected to other networks, in order to form a recursive structure (Besold et al., 2017, p. 6).

Mao, Gan, Kohli, Tenenbaum, and Wu, in their neuro-symbolic concept learner, take the path of joint reasoning through language. They use a symbolic reasoning process to connect the learning of visual concepts, words, and semantic parsing of sentences that all come without explicit annotations. The neural-based visual perception module treats attributes such as *Shape* and *Color* as neural operators. Those operators map object representations into a visual-semantic space, readying them for similarity-based classification (Mao et al., 2019, p. 3–4). The model consists of three modules. The neural-based perception module extracts object-level representations from the input pictures. The visually-grounded semantic parser translates questions about the visual input into executable programs. The outputs of both modules are read by a symbolic program executor that classifies their attributes and the relations among them and executes the program, which gives an answer to the questions posed (Mao et al., 2019, p. 1).

Modularity

Hybridity can either be reached within a single module or through multiple modules. *Homogeneous* multi-module models are essentially single-module models that encompass replica for redundancy, or consist of structurally identical modules that each incorporate knowledge from specialized domains (Sun, 1996, p. 100). *Heterogenous* multi-module models, on the other hand, can be classified by which types of representation they use in the different modules, and how those are connected among each other. A loose connection exists when certain models perform the pre- and post-processing, while the main processing lies in the scope of another module. The following relations can either be executed through loose or tight coupling (meaning that modules have more connection points): In a *master-slave-relationship* one module has full control overt the task to perform, and can, when necessary, hand parts of the task to other modules. In a *processor-monitor* combination one module is performing the task, while the monitoring module waits for certain events to occur and maybe alters the working of the processor. An *equal partnership* consists of complementary processors or functionally equivalent but representationally different processors, or of differently specialized and heterogeneously represented processors (Sun, 1996, p. 101).

Granularity

Multi-module models also vary regarding their granularity. A very coarse-grained system consists of a small number of modules, while fine-grained models realize *micro-level integration* in plenty of tiny and simple modules. This actually closes the circle back to homogeneous systems, as a system consisting solely of alike hybrid processing units qualifies as being *homogeneous* (Sun, 1996, p. 101).

Direction of Integration

Lieto identified a type, which is signified by the combination of low-level systems for reactive capacities and higher-level modules dealing with planning and reasoning. He stated that this confluence was unavoidable because of the respective strengths of the cognitivist and emergent systems (Lieto, 2021, p. 89). Such an integration can be referred to as *vertical hybridization.* Contrastly, in *horizontal hybridization*, symbolic and sub-symbolic components are realized on the same level (Wermter & Sun, 2000, p. 5).

4.1.2 Possible Hybrids

The different types of hybridization described above are by no means an exhaustive list. To get an idea of how many possible hybrids can be constructed (assuming that hybridity just means combining some traits from the emergent paradigm with some traits from the cognitivist paradigm) I made use of the 14 characteristics that Vernon uses to compare the two paradigms (Vernon, 2014, p. 54-55).

The Cognitivist Paradigm vs. the Emergent Paradigm

Characteristic	Cognitivist	Emergent
Computational Operation	Syntactic manipulation of symbols	Concurrent self-organization of a network
Representational Framework	Patterns of symbol tokens	Global system states
Semantic Grounding	Percept-symbol association	Skill construction
Temporal Constraints	Atemporal	Synchronous real-time entrainment
Inter-agent epistemology	Agent-independent	Agent-dependent
Embodiment	No role implied: functionalist	Direct constitutive role: non-functionalist
Perception	Abstract symbolic representations	Perturbation by the environment
Action	Causal result of symbol manipulation	Perturbation by the system
Anticipation	Procedural or probabilistic reasoning	Traverse of perception-action state space
Adaptation	Learn new knowledge	Develop new dynamics
Motivation	Criteria for goal selection	Increase space of interaction
Autonomy	Not entailed	Cognition entails autonomy
Role of Cognition	Rational goal-achievement	Self-maintenance and self-development
Philosophical Foundation	Positivism	Phenomenology

Figure 4.1: A table that lists 14 characteristics Vernon uses to compare the stance of cognitivism and emergentism. Source: (Vernon, 2014, p. 58).

If we would assume, building on this graphic, that in constructing a hybrid architecture, one had only those 14 choices to make, and could only answer them in an all-or-nothing way, that would lead to 16,382 ($2^{14} - 2$) options of designing a hybrid. Obviously, the assumptions are questionable. For example, one could argue that it is not enough to have a 1:13 ratio to count as a hybrid. Enforcing more balance between the parameters would substantially limit the number of possibilities. On the other hand, all the categories listed here are quite high-level and leave out a lot of technical detail that opens further options. Moreover, I would challenge the all-or-nothing assumption. In fact, many hybrid systems often opt for having varying degrees of both, again increasing the space of possibilities. Finally, there are certain path dependencies, i.e., If the choice in the category *Anticipation* fall to *Traverse of perception-action state space*, that influences the representational framework, as well as choices regarding perception and action, because at the end there is the need for such a state space to traverse. This is where the picture of a *garden of forking paths* comes in handy, as it can serve as a reminder for the path dependencies of the decisions that come up in the process of building a cognitive architectures. One example for architectures starting from very different theoretical stances, but still developed a surprising convergence, is described in the following section.

4.2 Synthesis of ACT-R and Leabra

The SAL cognitive architecture, which stands for Synthesis of ACT-R and Leabra, was the combination of two architectures that have been developed by disjunct research teams starting from very different theoretical standpoints. That means it was possible to get a distinct picture of how the 'more' cognitivist and the emergent part were realized and constructed, and from there assess their convergence. Further, SAL served as a fitting subject for the present thesis because of the authors' explicitly pluralistic stance in pursuing computational cognitive modeling. The coming sections 4.2.1 and 4.2.2 were designated to explaining SAL's components before their integration was discussed.

4.2.1 ACT-R

ACT-R (Adaptive Control of Thought–Rational) was one of the first cognitive architectures and was still actively developed. The central figure in the development leading to ACT-R is psychologist John R. Anderson (Newell, 1990, p. 26), though over the decades many researchers contributed to its success. In line with Newell's proposal of unified theories of cognition, ACT-R was striving for modeling and understanding the full range of human cognition (Taatgen, Lebiere, & Anderson, 2005, p. 29).

Having started as a symbolic system, a combination of HAM (Human Associative Memory) and a production system for procedural memory, ACT-R has taken a stance for pluralism in its evolution. A neural-like calculus for activating declarative memory was included in the 1980s to represent the graded, probabilistic nature of cognition (Jilk et al., 2008, p. 201). Inspired by Marr's information processing levels, the *rational analysis* approach was added to the procedural memory in form of a utility calculus that would mitigate the production selection (Jilk et al., 2008, p. 202). Anderson proposed viewing cognition as a set of *optimal solutions*, which correspond to the problems an agent has to face as cited by (Lieto, 2021, p. 40). To act rationally, a system has to:

1. Specify precisely the goals of the cognitive system.

2. Develop a formal model of the environment to which the system is adapted.

3. Make minimal assumptions about computational limitations (of the system).
4. Derive the optimal behavioral function given 1–3 above.
5. Examine the empirical evidence to see whether the predictions of the behavioral function are confirmed.
6. Repeat; Iteratively refine the theory.

This *rational analysis* is a main source of constraint for ACT-R. Due to the focus on interaction with the environment, the rationality does not come with strong assertions about the *truth* of certain representations; rather, knowledge is important because of its usefulness (Taatgen et al., 2005, p. 29).

Modularization

Overall, ACT-R chooses the path of modular decomposition of cognition and offers a theory of how to integrate these modules to produce coherent cognition. It features a vision module for obtaining the identity and position of objects in the visual field, a manual module for controlling hands, a declarative module for retrieving information about objects in the environment, and a goal module for keeping track of the internal state during problem-solving. Those modules are orchestrated via a production system that is connected to them through individual perceptual buffers. The authors claim that four of ACT-R's modules and all perceptual buffers correspond to areas in the human brain. The goal module is the only one not connected to a specific area; its buffer however, is said to correspond to the dorsolateral prefrontal cortex. The declarative module matches the temporal hippocampus and its buffer corresponds to the ventrolateral prefrontal cortex. The visual buffer is associated with the parietal area, the visual module with the occipital area. The manual buffer corresponds to the motor system, and the manual module to the motor system and cerebellum. Lastly, the production system matches the function of the basal ganglia (Vernon et al., 2007, p. 165). The processing of the individual modules in the architecture is asynchronously, and in parallel; within the modules, however, the processing is mostly serial.

Memory

A central aspect brought about by the modular structure is the division between procedural memory for rules and declarative memory for facts. While the declarative memory has its own module, the procedural memory is incorporated in the *production system*, which functions like a switchboard, connecting information patterns in the buffers to changes in associated buffer content. Thereby it triggers operations in the corresponding modules. Items stored in declarative memory are called *chunks*. Each chunk is associated with a level of activation, influencing if and how fast it will be retrieved. The value of activation is based on how often or how recent a chunk has been activated and whether it relates to the current context. This level decays over time and contains a noise component, introducing a probabilistic element into the retrieval process. As ACT-R is a goal-driven theory, chunks are usually retrieved to achieve a goal (Taatgen et al., 2005, p. 30–31).

Learning

The selection of both chunks and production rules depends on the utility or activation associated with them. ACT-R's learning mechanism is constantly updating these parameters to estimate utilities based on experience. The noise in this selection process can lead to a suboptimal selection, which enables the system to explore new strategies. Additional learning strategies exist in converting completed goals or motor events into chunks, and in fusing common sequences of production rules into single rules (Taatgen et al., 2005, p. 32).

Cycle

ACT-R also operates on a $\tilde{5}0$msec cycle in which the perceptual buffer is updated and one production can fire (Vernon et al., 2007, p. 164). Procedural memory is located in the production system, which selects the production with the highest utility, thereby encoding observed transformations. Through the combination of declarative and procedural memory, complex cognition can arise. As the declarative module is influenced by many numeric parameters, ACT-R by itself is a hybrid architecture (Vernon et al., 2007, p. 165). The integration with Leabra, however, takes it to another level.

4.2.2 Leabra

The Acronym Leabra stands for Learning in an Error-driven and Associative, Biologically Realistic Algorithm (O'Reilly et al., 2020, p. 79). Kotseruba and Tsotsos list Leabra as an emergent neuronal model. The goal is to produce an internally consistent theory of the neural basis of human cognition (O'Reilly et al., 2017). The theory strives to isolate those biological factors essential to the computational result at both a neural and network level (Jilk et al., 2008, p. 204). Thus, Leabra has been described as a biologically plausible version of AI mechanisms such as deep learning (Lebiere, Cranford, Martin, Morrison, & Stocco, 2022).

In SAL, as well as its components, the authors always stress that cognition needs both top-down and bottom-up processing. While it is quite clear that bottom-up means everything that arises from the biological/matter level, it is not so clear what top-down processing means. More precisely, it is unclear what to think of the causal mechanism enabling top-down processing. O'Reilly et al. argue for Leabra that bidirectional excitation, which takes the place of the biologically implausible back-propagation, is responsible for top-down processing and is also likely the mechanism that produces consciousness.

Guiding Principles

Leabra is described at two distinct levels – the micro- and macro-levels – constrained by each other. These two levels together are constrained by a set of meta-level guiding principles. On the macro-level, the architectures resemble specific brain areas according to their biological function; the micro-level, however, holds the neural mechanisms facilitating these functions (Lebiere, O'Reilly, Jilk, Taatgen, & Anderson, 2008, p. 4). O'Reilly et al. identify 20 principles that guide and constrain Leabra. Principles 1–5 are associated with the meta-level, principles 6–9 belong to the micro-level, principles 10–13 regard the macro-level, and the rest are not associated with the levels defined beforehand (O'Reilly et al., 2017):

1. **Balance:** Recognizing the trade-offs between different approaches, objectives and computational solutions by finding compromises or interaction of approaches/objectives/solutions, i.e., balancing biological fidelity with computational efficacy.

2. **Biology:** The brain serves as working 'reference implementation' for the architecture.

3. **Ockham's razor:** Finding the simplest model that accounts for broadest set of data.

4. **Convergent multilevel modeling:** Balance between biological, cognitive and computational constraints varies between levels and questions being addressed.

5. **Learning is critical:** Much of cognitive function is acquired in ontogenetic developmental processes, where experience shapes raw neuronal material into highly functional system. The adherence to ontogenetic development comes with choosing neuronal networks, as they provide the multi-potent substrate that can learn various things from experience without excess knowledge input through the designer.

6. Networks of neurons are the fundamental information processor of the brain; each neuron works as detector for input patterns and alerts other neurons when pattern has been found.

7. Synaptic weights encode knowledge and adapt to support learning.

8. Pyramidal neurons in the neocortex are the primary information processors of relevance for higher cognition.

9. Inhibitory interneurons regulate activity levels on neocortex and drive competition; competition produces beneficial effects for learning and performance, like competing for attention or enhancing executive functions.

10. **Micro-macro interactions:** Different specialized brain areas and their related functions should be produced by different parameterizations of the micro-structural mechanisms.

11. **Interference and overlap:** As learning new information can interfere with existing memories, it is crucial to minimize the neural overlap between memories.

12. **Pattern separation and sparseness:** Reduced overlap can be produced through increasing the level of inhibitory competition.
13. **Trade-offs in separation versus overlap:** Too much pattern separation reduces the ability to generalize knowledge across experiences.
14. Activation-based memory is more flexible than weight-based memory changes and is crucial for exerting top-down control.
15. There is as trade-off between updating and maintenance.
16. A dynamic gating system resolves the trade-off between rapid updating and robust maintenance.
17. Meaning lies in the activity pattern across neurons, not in the messages of individual neurons.
18. Complex processing require hierarchical stages.
19. Evolutionarily newer brain areas have strong bidirectional connections with older areas.
20. Motivation and reward must be grounded.

Modularization

On the large scale, the architecture features three brain areas divided by the learning rate and representation style: The posterior cortex uses slow learning rates for semantic and perceptual processing, while the hippocampus is specialized on fast learning of arbitrary information, both of which are modulated by the frontal cortex/basal ganglia. Those serve the active maintenance of goals and context information, but also represent the neuromodulatory dimension that is biologically driven by other areas (ventral tegmental area, locus ceruleus, amygdala, substantia nigra pars compacta). It is important to note that the differentiated structures in Leabra do not provide built-in cognitive functionality (O'Reilly et al., 2017, p. 97). The division into three domains arises from algorithmic properties on the micro-level, which guide and constrain the learning mechanism and outcomes, but is itself constrained by adherence to biological plausibility. "Each brain area uses the same set of core neural mechanisms, yet they differ in their parameterization to maximize different mutually incompatible cognitive functions," (Lebiere et al., 2008, p. 4).

Learning

This division means the algorithm integrates otherwise alone-standing neural learning and processing mechanisms in a biologically plausible way. It exploits increments of Hebbian learning, as well as instances of error-driven and reinforcement learning. Together they produce better overall learning than any one mechanism alone (Jilk et al., 2008, p. 203).

The learning rule is constrained top-down by computational desiderata and bottom-up by biology. Bottom-up it is inspired by the model of synaptic plasticity, which adheres to the dependency of synaptic plasticity on postsynaptic Ca++ levels. From top-down the idea of the BCM (Bienenstock, Cooper and Munro) learning rule, which is a modified version of Hebbian learning, introduces a floating threshold for determining the level of activity. With this, long-term potentiation and long-term depression are mediated by a homeostatic effect, which is useful for self-organized learning when implemented with a longer timeframe. Further, if this constant is manipulated to a short-term floating threshold, it resembles error-driven learning (O'Reilly et al., 2020, p. 66).

Cycle

To realize the error-driven learning, the timescale of the floating threshold is adapted according to physiological values. Those are inspired by the layered organizations of the cortex and their associated functions and frequencies. The timescale for the error-driven learning is chosen as follows: deep cortical layers (5 and 6) associated with (motor) output oscillate at 10 hz, while superficial layers (2 and 3) function as hidden layers of the network and are associated with classification; they oscillate at a gamma frequency (40hz) (O'Reilly et al., 2020, p. 45, 72).

XCAL Unit	Explanation	Anatomic Structure	Frequency
Quarter	first three quarters form expectation (minus) phase, last one the outcome (plus) phase	cortical layer 2/3 (hidden)	25 msec (40hz)
Trial	one sequence of expectation-outcome learning organized in 4 quarters	cortical layer 5/6 (output)	100 msec (10hz)
Cycle	each neuron updates its membrane potential	individual neurons (detecting)	1 msec

Table 4.1: The timescale of different levels updating in Leabra

The component of bidirectional excitation is introduced by exploiting attractor dynamics. This is helpful to explain phenomena like top-down imagery, top-down ambiguity resolution and pattern completion. The so-called XCAL function, *eXtended Contrastive Attractor Learning*, operates on the timescale presented in Table 4.1 (O'Reilly et al., 2020, p. 73):

Contrastive attractor learning means that the learning is driven by the contrast between the final attractor state in which the system settles and the networks activation trajectory as it approaches this final state. It is called *eXtended* when the two states do not have to be compared at discrete points in time but the baseline is averaged over the whole settling period (O'Reilly et al., 2020, p. 73).

To regulate excitement and avoid overboarding activity, local inhibitory interneurons keep the activation of neurons per region at 15-25%. This enforces sparse distributed representation, as it increases competition between neurons, which drives them to specialize (O'Reilly et al., 2020, p. 57). Inhibition can be realized in two modes: *feedforward* anticipates the level of excitement from the input, while *feedback* reflects the actual excitement of the target neuron (O'Reilly et al., 2020, p. 58).

4.2.3 SAL

The idea of working together sprung from the realization that both teams approach modeling from a pluralistic stance on both a vertical (coarse vs. granular) and horizontal (multiple learning strategies in Leabra and exchangeable modules in ACT-R) level. Additionally, the authors found that their respective architectures exhibited quite some overlap and complementary strengths, which had potential to synthesize well together, despite coming from very different theoretical backgrounds (Jilk et al., 2008, p. 205). ACT-R needed everything translated to pure symbols, while Leabra was missing the large-scale implementation it was advocating for. Through the combination it became possible for the cognitive system to extract meta-cognitive measures from a biologically plausible deep learning system to produce more robust and open-ended performance (Lebiere et al., 2022).

Modular Integration

An initial form of integration was formed through replacing ACT-R's vision module with that of Leabra, which lead to better performance on a navigation and search task (Lebiere et al., 2008). This straightforward integration showed the value that can arise from the synthesis, meanwhile shedding light on a few theoretical problems. Regarding the symbol grounding problem, SAL demonstrates how the neural network can be trained to associate images of certain objects to the corresponding symbolic representation. Also, the way that the control system enforces visual attention toward certain objects provides a basis for reflecting on the influence of top-down control on bottom-up processing in vision (Jilk et al., 2008, p. 208–209). Building on this, Vinokurov, Lebiere, Herd, and O'Reilly experimented with realizing a tighter connection of the systems within the visual module to implement a classifier that enables object recognition in dynamic environments (Vinokurov et al., 2011). Jilk et al. further laid out two future research trajectories: the mapping track, and the integration track.

Integration Through Mapping

Through mapping structural and emergent features of both architectures onto each other, the researchers want to better understand the components and their interaction. Specifically, how do the buffers and production rules of ACT-R correspond to working memory and cognitive control in Leabra's prefrontal cortex and basal ganglia? How can the basic production cycle of ACT-R be neurally realized through Leabra? (Jilk et al., 2008, p. 210). By this approach ACT-R is supposed to function as a developing framework for Leabra, while Leabra provides necessary constraints for ACT-R, which poses further questions for the individual architectures (Jilk et al., 2008, p. 211):

1. In ACT-R, how can *partial matching* of production rules soften traditional production conditions?
2. How can partial matching be implemented without producing degenerate behavior?
3. How can the utility parameters that control production selection be learned in a way that is grounded in plausible assumptions about the nature of feedback available?
4. How can production matching and selection be modulated by context without the explicit specification of precise conditions?
5. In Leabra, how can new procedural learning build on prior learning, instead of using trial and error for each new behavior?
6. How can verbal instructions be integrated and processed?
7. How does the system choose between exploration and exploitation, and manage strategic search through the space of actions and policies?

Tight Principled Integration

This proposed track should research how performance and robustness can be increased by the integration of symbolic and sub-symbolic systems. The authors advocate for a functional differentiation, because sub-symbolic, distributed processes are faster, more robust and have a larger processing capacity, which makes them ideal for representation. Symbolic systems, on the other hand, are inspectable, tractable, and efficient in learning sequential control, making them ideal for control structures. The goal of the integration track is a deeper synthesis, where integration does not happen by merely connecting distinct modules along a vertical axis, but rather happens within the modules themselves. The SAL framework thus would construe the symbolic/sub-symbolic interface in the bidirectional interactions between the ACT-R elements associated with prefrontal cortex, and the Leabra elements associated with posterior cortex (Jilk et al., 2008, p. 213).

Summary for SAL

What slightly disenchants the idea of SAL is that, according to Kotseruba and Tsotsos (who classify SAL as a fully integrated hybrid), it has not been actively developed since 2015 (Kotseruba & Tsotsos, 2020, p. 20, 25). A search for literature confirmed that there were no publications corresponding explicitly to research on SAL since then. The research communities of ACT-R and Leabra, in contrast, remained active. Nevertheless, SAL's explicitly pluralistic stance, together with the unintended theoretical conversion of its components, made it a highly interesting system, especially since Jilk et al. pursued multiple strategies for integration. In all presented integration tracks, SAL exhibits *vertical*, *modular* and *mechanistic pluralism*, and achieves symbolic-emergent hybridization that maps its structures to dissociable systems in the human brain (Jilk et al., 2008, p. 214). Further, the contributors state that the collaboration was fruitful because 'teaching' their architecture to the colleague of the other team improved their own understanding of the architecture they were researching beforehand. They also saw that the combination led to a broadening of the scope that SAL can account for. Nevertheless, Jilk et al. point out that these successes do not warrant an ontological privilege for SAL.

4.3 Sigma

The Sigma cognitive architecture is likewise classified as a fully integrated hybrid by (Kotseruba & Tsotsos, 2020, p. 25). Its design is influenced by Rosenbloom's work on Soar, successes of ACT-R, and the *graphical hypothesis*, and strives to attain *grand unification, functional elegance, sufficient efficiency*, and *generic cognition* (P. S. Rosenbloom et al., 2016, p. 1). The latter two were inherited from Soar. The functional elegance is an evolution of what Newell called *listening to the archtitecture*. Grand unification, however, clearly departs from the high-level focus in Soar.

- **Grand unification** specifies the integration of higher-level thought processes like planning or reasoning and lower-level functions critical for behavior in human-like environments, such as perception, motor control, or emotions (P. S. Rosenbloom et al., 2016, p. 2). In Sigma, this is realized by using a combination of discrete and continuous representations, as well as symbolic and probabilistic methods. It uses graphical models, in particular factor graphs with the summary product algorithm and piecewise continuous functions (P. Rosenbloom, 2013, p. 1).

- **Generic cognition** means that both natural and artificial cognition can be represented by Sigma at a proper level of abstraction.

- **Functional elegance** aims at achieving grand unification through a simple and theoretically elegant base, and has led to Sigma's non-modular design (P. S. Rosenbloom et al., 2016, p. 4).

- **Sufficient efficiency** refers to the demand of being fast enough for real-time interactions with virtual human and intelligent agents, as well as for general modeling purposes (P. S. Rosenbloom et al., 2016, p. 2).

Other important characteristics in Sigma are the exploitation of the graphical hypothesis, the nested tri-level control structure, and the strongly coupled hybridity.

4.3.1 The Cognitive Architecture

Sigma's cognitive architecture is grounded in an equally simple and elegant graphical architecture, which in turn builds on a low-level common LISP. The combination between the two architectures is facilitated through a straightforward compiler, translating between the two languages (P. S. Rosenbloom et al., 2016, p. 6). "The dynamic tension, interplay and constraint between these two architectures, in both their development and execution, is the source of much of Sigma's unique contributions to cognitive architectures, artificial (general) intelligence, and cognitive science," (P. S. Rosenbloom et al., 2016, p. 6).

Cognitive Language

Sigma's cognitive language is a form of *statistical relational languages*. It allows for symbolic and numeric (discrete and continuous) variables, that can be combined into sets to create *functions*. The relations among those sets of typed arguments are expressed through *predicates* (P. S. Rosenbloom et al., 2016, p. 32–34). As part of the three different forms of memory, a predicate may have multiple functions associated with it. A function can serve as the the current state or goal in working memory, or represent a learned model for a predicate in LTM. *Conditionals* define the interactions among predicate patterns through relating conditions, actions, condacts (fusion of conditions and actions, which both match WM and suggest changes to it) and functions. Also, multiple predicates and conditionals can use the same function (P. S. Rosenbloom et al., 2016, p. 35–37). The pervasive availability of functions and the bidirectional processing of the condacts that typically accompany them, is the extension that stands out most. This full support of hybrid and mixed processing is derived directly from making use of the graphical architecture, because functions and condacts are inherent to graphical models (P. S. Rosenbloom et al., 2016, p. 42).

Cognitive Cycle

Sigma's *cognitive cycle* is inspired by the $\tilde{5}0$ msec cycle found in humans, which is exploited by many other cognitive architectures, as well (P. Rosenbloom, 2013, p. 5). It consists of two minor (input, output) and two major (elaboration, adaptation) phases. Input and output, which respectively start and end the cycle, are the phases that enable the system to interface with its environment. Sensory and motor processing are included in the cycle to achieve both grand unification and functional elegance. During *elaboration phase*, knowledge about the current situation is processed to draw conclusions from it. The graphical architecture updates the distributions over variables until no further updates are available, i.e., *quiescence* is reached. Both representation and processing comprise of signals (continuous), probabilities (discrete and continuous) and symbols (discrete) (P. S. Rosenbloom et al., 2016, p. 16–17). The *adaptation phase* is largely non-monotonic and consists of meta-architectural processing, decision-making and memory changes on the basis of the distributions provided by the *elabortation phase*. Changes include short-term modifications to working memory, alterations to the system's affective and attentional states and changes in long-term memory (P. S. Rosenbloom et al., 2016, p. 17). In the graphical architecture the two big phases map onto *graph solution* (accessing memory in the elaboration phase) and *graph modification* (decision and learning in the adaptation phase) (P. Rosenbloom, 2013, p. 4).

Nested Tri-Level Control Structure

Sigma's control structure consists of three levels that are nested analog to the nesting in Soar. Through the design of the cognitive cycle, Sigma has a direct *reactive capability*. It means that a task like recognizing a fragment of speech can be achieved in a single cycle (P. S. Rosenbloom et al., 2016, p. 28). The *deliberative capability* builds on the reactive level and uses it for internal simulation, thereby achieving sequential, knowledge-driven or algorithmic behavior. The *reflective capability* again builds on the deliberative level for internal simulation to reason about its reasoning and help resolve impasses. Whenever an impasse occurs that makes progress at that level impossible, new (sub-)goals and corresponding meta-levels are initiated (P. S. Rosenbloom et al., 2016, p. 29).

Levels of Attention

There are two main mechanisms through which attention is modulated. Reflective attention makes use of the the control structure just described. Thus, each impasse that occurs focuses the meta-level processing that happens in response to it (P. S. Rosenbloom et al., 2016, p. 31). Otherwise, attention is regulated by a mechanism that selectively abstracts functions to make them simpler and easier to process. It works through combining two signals through an approximation to the probabilistic *OR*; one of them arises from bottom-up (based on surprise), the other one arises from top-down (based on goal or task relevance) (P. S. Rosenbloom et al., 2016, p. 31).

Learning

The *adaptation phase* – roughly corresponding to Soar's decision phase – includes long-term modifications to long-term memory (LTM), thereby facilitating learning (P. S. Rosenbloom et al., 2016, p. 17). The changes to LTM are attained in Sigma through *gradient descent*. That means that distributions over parameters as well as values of relations are learned, without necessarily affecting more complex structures in LTM. Key parameters of the algorithm are the learning rate, the smoothing threshold, the maximum increment and the form of normalization (P. S. Rosenbloom et al., 2016, p. 26). The guiding principle of functional elegance leads to different types of learning being implemented via *cognitive idioms*, rather than architectural modules (P. S. Rosenbloom et al., 2016, p. 12). A cognitive idiom consists of specific knowledge and gradient descent, and is effectively Sigma's analogues to software idioms, design patterns, libraries and services (P. S. Rosenbloom et al., 2016, p. 6). Thereby Sigma can produce (P. S. Rosenbloom et al., 2016, p. 27):

- concept learning (supervised and unsupervised)
- reinforcement learning and inverse reinforcement learning
- learning of action models
- learning of transition and perception models
- learning of maps
- multi-agent learning

For more complex behaviors involving, e.g., semantic memory and learning, Sigma makes use of a *supraarchitectural* cognitive idiom that straightforwardly integrates with other such capabilities in a functionally elegant manner (P. S. Rosenbloom et al., 2016, p. 15).

Theory of Mind

Effective social interaction can be facilitated through having a *Theory of Mind*. That makes it possible for an agent to develop beliefs about other individuals, produce expectations about how others will act, and utilize those predictions to guide its own decision-making. Sigma exhibits two mechanisms for producing a theory of mind. One of them arises from the tri-level nested control structure, which enables impasse detection (Pynadath, Rosenbloom, Marsella, & Li, 2013, p. 105). The other one is integrated in the reactive structure via a *trellis graph*. It expresses the transitions that represents the agent's (re)actions and backwards passes the utilities achieved through them (Pynadath et al., 2013, p. 104).

4.3.2 The Graphical Architecture

Generally, graphical models work by decomposing complex multivariate formulas into products of simpler functions and then translating them into graphs (P. S. Rosenbloom et al., 2016, p. 43). The big advantage of factor graphs is their ability to express both functions usually tied to Markov or Bayesian networks. This means with the same type of graph multivariate as well as probabilistic functions, cyclic and acyclic structures can be represented (P. S. Rosenbloom et al., 2016, p. 44).

The Rete algorithm (commonly used for rule matching in expert systems) is extended in Sigma to express conditionals, as well as actions and condacts as factor graphs. As a large-scale graphical idiom, it is leveraged by the compiler to achieve this conversion (P. S. Rosenbloom et al., 2016, p. 59). This implementation detail deeply enriches the expressibility of the system, because conditional functions can express more than conditional probabilities (P. Rosenbloom, 2013, p. 6–7).

In the graphical architecture the three memories are interconnected, as they refer to the different structures of the graph. One memory layer is holding the factor nodes, one the variable nodes and one is holding the undirected links.

"Each memory in the cognitive architecture is thus a graphical idiom defined in terms of particular combinations of structures in the graphical architecture's memories," (P. S. Rosenbloom et al., 2016, p. 44). The compiler builds fractions of the overall graph based on the types, predicates, etc. specified in the cognitive language. The meaning of the graph is determined in a bottom-up fashion through nodes and their connectivity (P. S. Rosenbloom et al., 2016, p. 44).

Representation Style in the Graphic Architecture The most recent version of Sigma uses n-dimensional arrays of orthotopic piecewise linear regions to represent functions, because they are more compact than lists and easier to traverse. It can be viewed as an array of pixels or voxels with (1) any number of dimensions, (2) varying size along each dimension, and (3) linear expression within each region (P. S. Rosenbloom et al., 2016, p. 46). At its core, the representation is continuous but also supports mixed/hybrid processing. That means that continuous functions can be approximated through adaptation of region size, as well as symbolic and discrete functions can be represented (P. S. Rosenbloom et al., 2016, p. 47). Further, it is possible to use a non-boolean function over a symbolic domain to express, e.g., the uncertainty over a choice between symbols (although pure symbolic functions are necessarily boolean) (P. S. Rosenbloom et al., 2016, p. 47). To support desirable outcomes, the standard notion of factor graphs has been extended for Sigma in many ways. Some of them are very in-line with traditional factor graph semantics, while others are not fully understood, which leads to loss of functional elegance. A straightforward extension are the *junction trees*, which make it possible for a variable node to correspond to more than one function variable (e.g. shape and color). What is ambiguous in the semantics of these combined variables is the endeavor to regulate direction of influence among a link, which is necessary to implement actions and conditions. (P. S. Rosenbloom et al., 2016, p. 48). Furthermore, there are special factor nodes for filters, action combination, and negation, which appear in the periphery of the graph. Based on that, the authors discuss them as a form of preprocessing, yet note that incorporating them in the solution of the graph may provide a more uniform way to implement these necessary components of and in the cognitive cycle (P. S. Rosenbloom et al., 2016, p. 49).

Graph Solution

Factor graphs are solved by inference algorithms. Via *evidence* (fixed values at the periphery), either the marginal distributions over individual nodes or the modal value over all of them is computed. Typically, not all regions of the graph can be computed with evidence, so for those unspecified regions, some default values will be set. In Sigma, these defaults will be determined by whether the system operates in an *open world* (value=1) or a *closed world* (value=0) (P. S. Rosenbloom et al., 2016, p. 49).

Types of Nodes **Assumption nodes** are a sub-goup of factor nodes that cannot change their output message during graph solution. They are analogous to assumption-based truth maintenance systems (ATMS) and include the following factors: *Evidence factors* used for assessing working memory in the cognitive architecture, factors for *predicate* and *conditional functions* connected to long-term memory, *perceptual factors* as well as *discount factors* through which current working memory contents become the default of the upcoming cycle (P. S. Rosenbloom et al., 2016, p. 50). At **variable nodes** the outgoing message of one link is the pointwise product over all messages incoming to that node via all other links. **Factor nodes** normally compute the same pointwise product but include their own function, meaning that all variables of the node are included in the calculation. However, in order to send an appropriate message along a link, only the variables corresponding to the link should be included in the calculation. This is achieved by a form of disjunctive combination, facilitated by the **summary product algorithm** that summarizes select variables out. Any message with a single (1) and the rest (0) works effectively as an index. Special purpose implementation of factor nodes do not merely involve multiplying the product of sending links with a factor function (P. S. Rosenbloom et al., 2016, p. 51–52). **Affine factor nodes** exploit variable swapping, which is used for object rotation. Currently, only 90-degree rotations and its multiples are available (P. S. Rosenbloom et al., 2016, p. 53).

Optimization Sigma's solution algorithm can be described as a *lifted* inference algorithm, meaning it only reasons over first-order structures (knowledge) and the messages passed along the graph are sliced "to distinguish regions with distinct functions" (P. S. Rosenbloom et al., 2016, p. 53). The removal of unnecessary slices does not change the meaning of the message, but its size in terms of number of regions. Further, Sigma uses message memory to not send redundant messages, that is, only messages which deviate more than the factor ϵ from the previous one are sent. For functional values, ϵ will be relative to the size of the numbers compared; for any other value there is an absolute ϵ that is implemented as a collection of distributed structures. Message memory also enables reuse of messages over cognitive cycles (P. S. Rosenbloom et al., 2016, p. 54). Since Sigma employs serial message passing, the order in which they are processed strongly influences processing time. Parallel processing is experimented with but stays future work. Processing is over when *quiescence* (meaning no significantly different messages are passed anymore) is reached (P. S. Rosenbloom et al., 2016, p. 55).

Graph Modification

The step after graph solution is graph modification, during which the functions maintained in factor nodes can be changed. Currently, only a small subset of functions associated with assumption nodes can be modified, yet this corresponds to changes in working memory, long-term memory and perceptual buffer functions. Thus, it achieves changes of state on the cognitive level; however, it does not yet yield structural learning (P. Rosenbloom, 2013, p. 4). The input for modification comes from messages or assumptions (P. S. Rosenbloom et al., 2016, p. 55). To achieve state changes in working memory, the functions are stored in specific assumption nodes. In some cases a selection needs to be made based on a distribution arriving. As a tiebreaker, the current memory will be multiplied with .01 before combining it disjunctively. If the old selection is as good as the best ones from the distribution, it is kept, otherwise, one of the best ones is chosen randomly. This *preference memory*, inspired by Soar, is not a mere implementational detail, but part of a more general memory structure. Another issue within this selection process is that the architecture needs to 'be aware' of whether a variable is discrete or continuous in order to apply the right unit (ϵ when continuous and 1 when discrete or symbolic) for selecting the region around the value.

The distinction between messages that drive graph solution or modification is made analogous for WM and LT. Messages coming into the memory drive modification and messages leaving the memory drive solution. The algorithm that facilitates learning is different, however (P. S. Rosenbloom et al., 2016, p. 57).

4.3.3 Senses

The two senses currently implemented in Sigma are auditory and visual perception (Kotseruba & Tsotsos, 2020, p. 29). Visual perception in Sigma has been achieved through two mechanisms: The first one is a conditional random field (CRF) that computes distributions over perceived objects from noisy feature data. The second is a localization graph (SLAM) that produces distributions over (current and past) locations from distributions over (present and past) objects and a map. Those two can be combined to obtain distributions over locations based on noisy feature information (P. Rosenbloom, 2013, p. 7). Auditory signals used for speech recognition are processed via Hidden Markov Models based on a reusable generic time slice. The transition predicate represents a function stored in a factor node, and represents the joint conditional probability of the upcoming state given the currently selected word and phone (P. S. Rosenbloom et al., 2016, p. 69–70).

4.3.4 Limitations and Outlook

The authors state they have only begun to explore Sigma's relationship to neural networks by experimenting with special one-way factor nodes, which can compute logistic regressions. This might in the future outweigh some of the current limitations, as on of them is the approximation of continuous functions via piecewise-linear functions. Further, they recognize a deficit in learning from functions without child nodes, which they hope to amend by applying a Soar-like chunking mechanism (P. S. Rosenbloom et al., 2016, p. 91). Further, Sigma is missing many cognitive functions that come to bear at the graphical level, like appraisal, motivation, attention, personality and a notion of bodily state. Within the cognitive architecture, P. S. Rosenbloom et al. mention less ample gaps; however, they say planning, abductive and analogical reasoning were not part of or are not central to Sigma's cognitive functions. On the rational band (supervening the cognitive band) Sigma does not feature an explicit logic, which the authors do not perceive as a limitation. The current infringement of Sigma on that level they attribute to the absence of large bodies of diverse knowledge that can be used in goal pursuit (P. S. Rosenbloom et al., 2016, p. 92–93). Also, Sigma clearly exhibits constraints from both robotics and cognitive science to achieve generic cognition, but it is not yet clear whether that will yield both effective artificial systems and sufficiently accurate models of natural intelligent systems at the end(P. S. Rosenbloom et al., 2016, p. 94).

4.3.5 Summary of Sigma

Summing up, it can be very nicely observed in Sigma how the selected desiderata influence decision making processes in the design on lower levels, e.g., the use of factor graphs as a main facilitator of functional elegance instead of modularization. Factor graphs provide state-of-the-art processing across signals, probabilities, and symbols from a single representation and processing algorithm. Also interesting is how a dimension like biological fidelity can be handled very differently at the multiple levels. On a low-level, factor graphs are lacking biological plausibility, especially compared to the neural network that Leabra (4.2.2) uses. On a more abstract level, however, the elegance and expressiveness of the graphical models makes them an interesting and valid choice for achieving natural cognitive plausibility. This fit to higher level biological plausibility is expressed in the fact that a single cognitive cycle facilitates basic reactivity.

4.4 Summary for Hybridization

The taxonomies of hybridization in section 4.1 displayed multiple degrees of freedom in constructing hybrid architectures: granularity, modularity, heterogeneity, coupling between symbolic and sub-symbolic components, as well as types of learning and memory, and direction of integration. The two architectures presented thereafter take highly different approaches towards realizing hybridization. Both achieve integrating multiple types of representations and learning strategies across multiple levels. While SAL exhibited modularity on all levels and depicted vertival integration, Sigma strives for a non-modular way, with horizontal integration. What becomes very clear when delving into their design and construction is, that those high-level properties are crucially influenced by decisions that might appear to be implementational details. Sigma can only realize the functional elegance because of the high expressiveness of factor graphs. SAL, on the other hand, exhibits modularity which, in Leabra, is caused strictly though changing parameters in its learning algorithm.

A similarity between the two might be drawn regarding the division of labor that Jilk et al. described for SAL, where parts of Leabra were commissioned with low-level, robust, fast processing, and representation, while ACT-R was responsible for control and planning. This can analogously be said about Sigma, where the graphical architecture is responsible for representations and manipulations of them, while the cognitive architecture orchestrates the low-level processing. Irrespective of their differences, both routes of hybridization examined here yielded interesting results and showed just how multifaceted hybridization can be.

Chapter 5

Implications and Conclusion

5.1 Implications regarding research questions

5.1.1 RQ1: How is this integration of seemingly competing theoretical standpoints realized?

As was shown in section 4.1, as well as laid out by the examples in section 4.3 and section 4.2 there is not one way of achieving integration between cognitivist and emergent ideas. There are so many, in fact, that I *now* understand why I could not find a well defined and nicely constrained hybrid paradigm. The combinatorial potential was too big to make substantial predictions on the future convergence of a main stream yet. By providing a comprehensive, albeit non-exhaustive, list of degrees of freedom that researchers have when designing cognitive architectures, this thesis depicted the multidimensionality of such theories. The approach of decomposing the paradigms into many singular design choices enables reflecting on their implications. From this point it can be openly assessed which implications arise from the specific combination of assumptions built into a system in question. By that I showed how the construction of hybrid cognitive architectures may help to break free from the infringements of a particular theory and the scope that it was once formulated for.

5.1.2 RQ2: Which theoretical problems may arise from such realizations?

From here, I conclude that the building of hybrid cognitive architectures is not what brings about large theoretical problems. What can become an issue regarding classification and analysis of such systems is wanting to stick to the labels and preexisting mutually exclusive categories that have been created around the symbolic and emergent paradigm. If one views hybrid systems through a lens of hybridity, however, where every design choice has certain influence on how the system as a whole will function and how it can be classified, the combination does not cause unresolvable contradictions. What it does cause is extra work in reflecting the influence of each choice made. In turn, this can lead to loss of tractability, because the multidimensionality of such systems makes them hard to compare. Admittedly, this also holds for non-hybrids.

What helps alleviate the problem of tractability within a given system is careful documentation and reflection on the choices that researchers make in the construction process. Additionally, the structure itself ensures accountability by exhibiting the choices that have been made in previous steps in design. The option to simply connote certain terms slightly differently from case to case in order to make a misfit disappear (which can be done with verbal concepts quite easily) is not given here. Hybrid architectures are a safe and methodologically sound way of exploring the boundaries between paradigms and the effects of single and compounding assumptions. Hybrids explore the space of *cross pollination* that lies between cognitivism and emergentism, which is crucial for testing and developing new unified theories of cognition. Through that, they can facilitate the integrative pluralism that is needed to catalyze the development of cognitive science (not cognitive sciences) as a field, and potentially lead the way to a new paradigm.

5.2 Conclusion

5.2.1 Contrasting points

While there are good arguments for exploring hybridity, the work being done in one paradigm or even with one particular class of algorithms is paramount to the field, and all the while as necessary as researching hybrids. I like the approach that Marcus takes in his opening statement of the AI debate with Bengio. Marcus argued for neural-symbolic integration and described how a group of researchers kept investing time and effort into working on neural networks throughout another *connectionist winter*, and with that facilitated the powerful deep neural networks we have today. "Luckily people like Bengio, LeCun and Hinton kept plugging away despite resistance. I hope people doing symbols will keep plugging away," (MILA – Quebec AI Institute. [Montreal.AI] , 2019, timestamp 26:14). The same thing holds regarding the sub-disciplines of cognitive science. Despite the great value in achieving unification, doing highly specialized and localized research in the sub-disciplines pushes the boundary of knowledge and delivers the facts that are needed to build unified theories of cognition.

5.2.2 Perspective on further research

Throughout the thesis I argued that exploring the space of *cross pollination* between emergent and cognitivist approaches is one way of facilitating integrative pluralism within cognitive science, and that cognitive architectures are a methodologically sound way of doing to achieve that. However, I also showed – implicitly and explicitly – that this integration is effortful. This especially holds true in the beginning the process, where the attempt to combine different sets of knowledge is likely to be chaotic and the arising theory is likely to be less parsimonious than the parts that were used to construct it. It might also turn out that many ways of integration in the long run do not yield the expected results. At this point in time, however, there is not any cognitive architecture or other theory that achieves what P. S. Rosenbloom et al. call grand unification. Since parsimony is only choosing between models that adequately cover the data, discarding certain approaches due to lack thereof seems premature (Marcus, 2001, p. 31). On the other hand, it is also premature to praise the path of hybridity simply because it bridges a divide that current processing methods are opening. The success of processing techniques throughout the development of computer science has always been bound to the availability and development of hardware. With the advent of quantum computers and their eventual successors, perhaps such a divide between high-level and low-level will cease to be. For now, "forcing ourselves away from general polar comparisons and toward theories that could deal with complex psychological situations" (Newell, 1990, p. 2) is a promising path. Decomposing existing paradigms into different assumptions might help view the research field of cognitive modeling again with the eyes of pioneers, embrace the complexity, and relieve the field of heavy notional baggage.

5.2.3 Reflection

Whether hybrid or not, cognitive architectures are highly complex structures, which is what caused my fascination for them. Nevertheless, I have underestimated the time, effort, and highly specialized technical knowledge needed to understand them. Having anyways invested the time, I hope this thesis accurately reflects my learning process and displays a helpful piece of literature for any reader wo, like me, wants to explore the world of cognitive architectures. While I think that I have gained quite some understanding of the systems in question, there is still so much more to learn, and the quality of this text would have surely increased in quality, had I had the time to more deeply engage with the two architectures that I reviewed. Further research regarding the relevant design choices might arrive at an interactive visualization of the path dependencies related to certain choices. This could enable students and researchers to play around and explore different types of combinations, with the tool logging the choices made and pointing out changes brought by implicitly.

References

Abbott, A. (2014). *Digital paper. a manual for research and writing with library and internet materials.* Chicago and London: University of Chicago Press. doi: https://doi.org/10.7208/chicago/9780226167817.001.0001

Albus, J. S., & Barbera, A. J. (2005). RCS: A cognitive architecture for intelligent multi-agent systems. *Annual Reviews in Control*, *29*(1), 87–99. doi: https://doi.org/10.1016/j.arcontrol.2004.12.003

Bélanger, M., Potvin, P., Horst, S., Shtulman, A., & Mortimer, E. F. (2022). *Multidisciplinary perspectives on representational pluralism in human cognition.* New York: Routledge. doi: https://doi.org/10.4324/9781003189930

Besold, T. R., d'Avila Garcez, A., Bader, S., Bowman, H., Domingos, P., Hitzler, P., ... Zaverucha, G. (2017, November). *Neural-symbolic learning and reasoning: A survey and interpretation.* arXiv. arXiv. doi: https://doi.org/10.48550/arXiv.1711.03902

Blaha, L. M., & Gluck, K. A. (2023, April). Model validation, comparison, and selection. In R. Sun (Ed.), *The Cambridge Handbook Of Computational Cognitive Science* (Second ed., pp. 1165–1200). Cambridge and New York and Melbourne and New Delhi and Singapore: Cambridge University Press. doi: https://doi.org/10.1017/9781108755610.042

Boden, M. A. (2023, April). An evaluation of computational modeling in cognitive sciences. In R. Sun (Ed.), *The cambridge handbook of computational cognitive science* (Second ed., pp. 1228–1249). Cambridge and New York and Melbourne and New Delhi and Singapore: Cambridge University Press. doi: https://doi.org/10.1017/9781108755610.044

Bogen, J. (2020). Theory and Observation in Science. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Summer 2020 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/sum2020/entries/science-theory-observation/.

Bzdok, D., Altman, N., & Krzywinski, M. (2018). Statistics versus machine learning. *Nat. Methods*, *15*(4), 233–234. doi: https://doi.org/10.1038/nmeth.4642

Chella, A., Cossentino, M., Gaglio, S., & Seidita, V. (2012). A general theoretical framework for designing cognitive architectures: Hybrid and meta-level architectures for BICA. *Biologically Inspired Cognitive Architectures*, *2*, 100–108. doi: http://dx.doi.org/10.1016/j.bica.2012.07.002

Curtis-Trudel, A. (2022). Why do we need a theory of implementation? *The British Journal for the Philosophy of Science*, *73*(4), 1067–1091.

d'Avila Garcez, A. S., Lamb, L. C., & Gabbay, D. M. (2009). *Neural-symbolic cognitive reasoning.* Berlin, Heidelber: Springer. doi: https://doi.org/10.1007/978-3-540-73246-4

Farrell, S., & Lewandowsky, S. (2018). *Computational modeling of cognition and behavior.* Cambridge :: Cambridge University Press,.

Frigg, R., & Hartmann, S. (2020). Models in Science. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Spring 2020 ed.). Stanford, CA: Metaphysics Research Lab, Stanford University. Retrieved 2021-06-23, from https://plato.stanford.edu/archives/spr2020/entries/models-science/

Gallant, S. I. (1993). *Neural network learning and expert systems.* Cambridge, Massachusetts: MIT Press.

Gelman, A., & Loken, E. (2013). *The garden of forking paths: Why multiple comparisons can be a problem , even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time.* Department of Statistics, Columbia University. Retrieved 2023-05-13, from http://www.stat.columbia.edu/~gelman/research/unpublished/p_hacking.pdf

Hendler, J. A. (2014). *integrating marker passing and problem solving.* Psychology Press. doi: https://doi.org/10.4324/9781315801797

Jilk, D. J., Lebiere, C., O'Reilly, R. C., & Anderson, J. R. (2008). Sal: an explicitly pluralistic cognitive architecture. *Journal of Experimental & Theoretical Artificial Intelligence*, *20*(3), 197–218. doi: https://doi.org/10.1080/09528130802319128

Kotseruba, I., & Tsotsos, J. K. (2020). A Review of 40 Years of Cognitive Architecture Research: Core Cognitive Abilities and Practical Applications. *AI Review*, *53*(1), 17–94. doi: https://doi.org/10.1007/s10462-018-9646-y

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive systems research*, *10*(2), 141–160.

Lebiere, C., Cranford, E. A., Martin, M., Morrison, D., & Stocco, A. (2022). Cognitive architectures and their applications. In *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)* (p. 55-60). Atlanta, GA, USA. doi: https://doi.org/10.1109/CIC56439.2022.00018

Lebiere, C., O'Reilly, R., Jilk, D., Taatgen, N., & Anderson, J. (2008, January). The SAL Integrated Cognitive Architecture. In *Papers from the 2008 AAAI Fall Symposium: Biologically Inspired Cognitive Architectures* (pp. 98–104).

Lewandowsky, S., & Farrell, S. (2011). *Computational Modeling in Cognition: Principles and Practice.* SAGE Publications, Inc. doi: https://doi.org/10.4135/9781483349428

Lieto, A. (2021). *Cognitive design for artificial minds* (first ed.). London: Routledge. doi: https://doi.org/10.4324/9781315460536

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., & Wu, J. (2019). The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International conference on learning representations.*

Marcus, G. F. (2001). *The algebraic mind: integrating connectionism and cognitive science /.* Cambridge, Massachusetts: MIT Press.

McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*(1), 11–38. doi: https://doi.org/10.1111/j.1756-8765.2008.01003.x

MEi:Cogsci. (2017). *Cognitive science.* Website. Retrieved 2023-09-03, from `https://www.meicogsci.eu/cognitive-science.html`

MILA – Quebec AI Institute. [Montreal.AI] . (2019). *Ai debate: Yoshua bengio*

— *gary marcus [video].* Youtube. Retrieved 25.08.2023, from https://www.youtube.com/watch?v=EeqwFjqFvJA

Murphy, K. P. (2022). *Probabilistic machine learning : an introduction.* Cambridge, Massachusetts: The MIT Press.

Newell, A. (1990). *Unified theories of cognition.* Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving* (First ed.). Englewood Cliffs, NJ: Prentice-Hall.

O'Reilly, R. C., Hazy, T. E., & Herd, S. A. (2017, October). 91The Leabra Cognitive Architecture: How to Play 20 Principles with Nature and Win! In *The Oxford Handbook of Cognitive Science.* Oxford University Press. doi: https://doi.org/10.1093/oxfordhb/9780199842193.013.8

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., & Contributors. (2020). *Computational cognitive neuroscience.* Online Book, 4th Edition. Retrieved 24.01.2023, from https://github.com/CompCogNeuro/ed4

Polk, T. A., & Seifert, C. (2002). *Cognitive modeling.* Cambridge, MA: The MIT Press.

Pynadath, D., Rosenbloom, P., Marsella, S., & Li, L. (2013, July). Modeling two-player games in the Sigma graphical cognitive architecture. In *Artificial general intelligence* (Vol. 7999, pp. 98–108). doi: https://doi.org/10.1007/978-3-642-39521-5_11

Ritchie, J. B., & Piccinini, G. (2019). Computational implemenation. In M. Sprevak & M. Colombo (Eds.), *The Routledge handbook of the computational mind.* Oxon and New York: Routledge, Taylor & Francis Group.

Rosenbloom, P. (2013, January). The Sigma cognitive architecture and system. *AISB Quarterly*, *136*, 4-13.

Rosenbloom, P. S., Demski, A., & Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*(1), 1–103. doi: https://doi.org/doi:10.1515/jagi-2016-0001

Russell, S. J., & Norvig, P. (2021). *Artificial intelligence : a modern approach* (Fourth ed.). Hoboken, NJ: Pearson.

Samsonovich, A. (2010, January). Toward a unified catalog of implemented cognitive architectures. In (pp. 195–244). doi: https://doi.org/10.3233/978-1-60750-660-7-195

Samuels, R. (2019). The Routledge handbook of the computational mind. In M. Sprevak & M. Colombo (Eds.), *The Routledge handbook of the computational mind* (pp. 103–119). Oxon and New York: Routledge, Taylor & Francis Group.

Schinagl, D. J. (2016). *Generative adversarial networks - properties and applications.* TUGraz DIGITAL Library.

Schöner, G., & Kelso, S. (1988, March). Dynamic Pattern Generation in Behavioral and Neural Systems. *Science*, *239*(4847), 1513–1520.

Sprevak, M., & Colombo, M. (Eds.). (2019). *The Routledge handbook of the computational mind.* Oxon and New York: Routledge, Taylor & Francis Group.

Stephan, A., & Walter, S. (Eds.). (2013). *Handbuch Kognitionswissenschaften* (First ed.). Stuttgart: J.B. Metzler, Springer. (In German language)

Sun, R. (1996, March). Hybrid connectionist-symbolic modules: A report from the ijcai-95 workshop on connectionist-symbolic integration. *AI Magazine*, *17*(2), 99. doi: https://doi.org/10.1609/aimag.v17i2.1225

Sun, R. (2004). Desiderata for cognitive architectures. *Philosophical Psychology*, *17*(3), 341–373. doi: https://doi.org/10.1080/0951508042000286721

Sun, R. (Ed.). (2023a). *The Cambridge Handbook of Computational Cognitive Sciences* (Second ed.). Cambridge and New York and Melbourne and New Delhi and Singapore: Cambridge University Press. doi: https://doi.org/10.1017/9781108755610

Sun, R. (2023b, April). Introduction. In R. Sun (Ed.), *The cambridge handbook of computational cognitive science* (Second ed., pp. 1–26). Cambridge and New York and Melbourne and New Delhi and Singapore: Cambridge University Press. doi: https://doi.org/10.1017/9781108755610.002

Sun, R., & Bringsjord, S. (2007, December). *Cognitive Systems and Cognitive Architectures.* John Wiley & Sons, Inc. doi: https://doi.org/10.1002/9780470050118.ecse274

Taatgen, N., Lebiere, C., & Anderson, J. (2005). Cognition and multi-agent interaction: From cognitive modeling to social simulation. In R. Sun (Ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation* (p. 29–52). Cambridge: Cambridge University Press. doi: https://doi.org/10.1017/CBO9780511610721.003

Thomas, M. S. C., & McClelland, J. L. (2023, April). Connectionist models of cognition. In R. Sun (Ed.), *The Cambridge Handbook Of Computational Cognitive Science* (Second ed., pp. 29–79). Cambridge and New York and Melbourne and New Delhi and Singapore: Cambridge University Press. doi: https://doi.org/10.1017/9781108755610.005

Vernon, D. (2014). *Artificial Cognitive Systems: A Primer.* Cambridge, MA and London, England: The MIT Press.

Vernon, D. (2022, May). Cognitive robotics. In A. Cangelosi & M. Asada (Eds.), (pp. 191–212). Cambridge, Massachusetts and London: The MIT Press. doi: https://doi.org/10.7551/mitpress/13780.003.0015

Vernon, D., Metta, G., & Sandini, G. (2007). A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. *IEEE Transactions on Evolutionary Computation*, *11*(2), 151–180. doi: https://doi.org/10.1109/TEVC.2006.890274

Vernon, D., von Hofsten, C., & Fadiga, L. (2016). Desiderata for developmental cognitive architectures. *Biologically Inspired Cognitive Architectures*, *18*, 116-127. doi: https://doi.org/10.1016/j.bica.2016.10.004

Vinokurov, Y., Lebiere, C., Herd, S., & O'Reilly, R. (2011, January). A metacognitive classifier using a hybrid ACT-R/Leabra architecture..

Wermter, S., & Sun, R. (2000, 04). Hybrid neural systems. In (pp. 1–13). Heidelberg, New York: Springer.

List of Figures

Appendix

Abstract (english)

In the evolution and coexistence of paradigms in cognitive science, emergent and cognitivist approaches to cognition are growing and promising paradigms. To develop and validate theoretical concepts, researchers of either paradigms make use of computational modeling. While at times presented as mutually exclusive and competing for interpretational sovereignty, some researchers advocate the integration of these paradigms: *Hybrid cognitive architectures*, which make up roughly half of currently existing cognitive architectures, are the materialization of attempts of such integration practice. However, David Vernon pointed out that hybrid architectures are often neglecting the theoretical problems, that arise from combining the antagonizing paradigms. The aim of this thesis is to show how the construction of hybrid cognitive architectures and the multifold choices that have to be made on many levels, helps to break free from the restrictions of a particular theory and the scope that it was once formulated for, while also ensuring accountability for all deviations. This brings about the research questions addressed in this work:

RQ1 How is this integration of seemingly competing theoretical standpoints realized?

RQ2 Which theoretical problems may arise from such realizations?

In a literature review I collected and summarized materials that revealed the assumptions on which computational models of cognition have been built as well as the design choices that must be made when constructing a cognitive architecture. Based on that, I laid out the diversity of approaches summarized under the label of hybridity and portrayed two architectures, *SAL* and *Sigma*, as exemplars. By

providing a comprehensive list of degrees of freedom that researchers have when designing cognitive architectures I represented the multidimensionality of such theories. I argue that this makes the number of possible systems too diverse to summarize under a nicely constrained 'hybrid paradigm' yet. Through decomposing the paradigms into well tracked design choices, cognitive architectures can be a methodologically sound way of exploring this space of *cross pollination*. This is important for testing and developing new unified theories of cognition, as well as models spanning across disciplines in cognitive science. In that, hybrid cognitive architectures can facilitate the integrative pluralism that is needed to do justice to the interdisciplinarity of cognitive science.

Zusammenfassung (deutsch)

In der Entwicklung und Koexistenz von Paradigmen in der Kognitionswissenschaft, sind *emergente* und *kognitivistische* Ansätze zur Kognition wachsende und vielversprechende Paradigmen. Um theoretische Konzepte zu entwickeln und zu validieren, greifen die Forscher beider Paradigmen auf computergestützte Modellierung zurück. Obwohl diese zuweilen als unvereinbare Gegensätz welche um Deutungshoheit konkurrieren präsentiert werden, plädieren einige Forscher für die Integration der Paradigmen: *Hybride kognitive Architekturen*, die etwa die Hälfte der derzeit existierenden kognitiven Architekturen ausmachen, sind die Materialisierung einer solchen Integrationspraxis. David Vernon wies jedoch darauf hin, dass hybride Architekturen oft theoretische Probleme vernachlässigen, die sich aus der Kombination der antagonistischen Paradigmen ergeben. Ziel dieser Arbeit ist es zu zeigen, wie die Konstruktion hybrider kognitiver Architekturen und die vielfältigen Entscheidungen, die auf vielen Ebenen getroffen werden müssen, dazu beitragen, sich von den Beschränkungen einer bestimmten Theorie und dem Geltungsbereich, für den sie einst formuliert wurde, zu befreien und gleichzeitig die Verantwortlichkeit für alle Abweichungen zu gewährleisten. Daraus ergaben sich die folgenden Forschungsfragen:

RQ1 Wie wird die Integration von scheinbar konkurrierenden theoretischen Standpunkten realisiert?

RQ2 Welche theoretischen Probleme können sich aus solchen Umsetzungen ergeben?

In einer Literaturrecherche habe ich Materialien gesammelt und zusammengefasst, auf welchen Annahmen computergestützte Modelle der Kognition beruhen, sowie welche Entscheidungen beim Bau einer kognitiven Architektur getroffen werden müssen. Auf dieser Grundlage habe ich die Vielfalt der Ansätze dargelegt die unter dem Begriff der Hybridität zusammengefasst werden, und stellte als Beispiele die zwei Architekturen *SAL* und *Sigma* vor.

Durch die Bereitstellung einer umfassenden Liste von Freiheitsgraden, welche die Forscher bei der Entwicklung kognitiver Architekturen haben, bildete ich die Multidimensionalität solcher Theorien ab. Ich argumentierte, dass die Anzahl der möglichen Hybridisierungen zu vielfältig ist, um sie unter einem 'Hybriden Paradigma' zusammenzufassen. Durch die Zerlegung der Paradigmen in gut nachvolziehbare Design-Entscheidungen können kognitive Architekturen jedoch eine methodisch fundierte Möglichkeit sein, den Raum der Kreuzbestäubung zwischen den Paradigmen zu erforschen. Dies ist wichtig um neue, einheitliche Theorien der Kognition, sowie Modelle, die die Disziplinen der Kognitionswissenschaft überspannen zu entwickeln und zu testen. Insofern können hybride kognitive Architekturen den integrativen Pluralismus fördern, der notwendig ist, um der Interdisziplinarität der Kognitionswissenschaft gerecht zu werden.