Relevance Realization as a Solution to the Frame Problem in Artificial General Intelligence

A Comparison of Four Cognitive Architectures

Master Thesis Artificial Intelligence

Utrecht University

Maxim van der Kaaij

5762715

Supervisor: dr. Colin Caret

Second reader: dr. Dominik Klein

September 2022

Table of Contents

Introduction

One of the most exciting ambitions in the academic study of artificial intelligence is the creation of a so-called artificial general intelligence (AGI). That is to say, an artificial agent that is competent in not just one, but several (and ideally almost all thinkable) domains of action. Humans are often viewed as being generally intelligent, precisely because of our ability to behave intelligently in so many domains; among other things, we can do mathematics, have social interactions, plan a career for the future, become highly skilled at a sport and build rockets that go to Mars. Being able to build an artificial intelligence that can do all of these things would be revolutionary, to say the least. Most progress in the field of artificial intelligence has, however, been geared toward specific applications; so-called artificial narrow intelligence (ANI). Think for example of the widely adopted use of chatbots for internet companies, or the development of an ANI by Google Deepmind that was able to beat the best human player in the complex board game Go (Gibney, 2015). However successful these may have been, building an artificial general intelligence has been proven to be a lot harder. In this thesis, I will argue that the main reason we have not yet been able to build an AGI is because of a related unsolved problem in the philosophy of artificial intelligence: the so-called frame problem.

To build up the argument, I will first discuss both the frame problem and artificial general intelligence separately, to get a good view on the current state of these matters, after which I will make the link between the two. Next, I will introduce a different perspective on and a solution to the frame problem, the idea of relevance realization, as proposed by Vervaeke (2012). I will argue that the ability of a system to achieve relevance realization correlates with its success at solving the frame problem and that the degree to which a system can do relevance realization is the degree to which a system is generally intelligent. To put in another way, I would argue that the ability to do relevance realization is exactly what makes a system "generally" intelligent (Granted, intelligence itself involves more factors than merely the ability to do relevance realization, but any system that aims to be intelligent in a general sense will be so to the degree that it incorporates a system for relevance realization). I will propose a few new features of relevance realization, next to those already proposed by Vervaeke. After that, I will discuss several existing cognitive architectures that aim to be the framework of an artificial general intelligence, and lay out the basic design structure of each one. I will make a relative comparison between these architectures based on the features of relevance realization, and give each architecture a

score for each feature. Finally, I will make a suggestion for a design proposal for a new cognitive architecture where the best scores of the features from the different architectures are combined: this new architecture would be most capable of relevance realization and thus most promising as a design for an artificial general intelligence.

The Frame Problem

The computational frame problem

The Frame Problem is a problem in the philosophy of artificial intelligence that was first explicitly formulated by McCarthy and Hayes (1969). In its original form, it poses the question of how a system can logically represent both the effects as well as the non-effects of a given action of the system. What makes this a problem is that it is in principle impossible to account for all the non-effects of a particular action, because those would form a potential infinite list. However, if an agent wants to be able to represent an accurate description of the future state of the internal and external environment after a particular action, it must also logically represent these non-effects. To illustrate the problem, consider how the following example would be handled using a standard deductive system such as classical logic (a similar example occurs in (Brown, 1987)).

Let's suppose that we can perform two different actions on a particular object. Logically, the consequences of these actions can be described as follows.

1. Temperature(x,t) holds after Heating(x,t)
2. Shape(x,s) holds after Mold(x,s)

The first formula describes that the object x is assigned Temperature t after it has performed the action Heating to t. The second formula describes that the Shape of the object x is assigned s after the action Mold s is applied to the object. So far this is straightforward.

Now imagine that we start out with an object g(t,s) = g(50 degrees, cube). If we apply Heating(g,70 degrees), we can deduce that Temperature(g,70 degrees). However, what can we say about the shape of the object, after we apply Heating(g,70 degrees)? Intuitively, it would make sense to assume that Shape(g,cube) would hold, because only Temperature is defined to change after Heating. However, in formal logic, we cannot deduce that this is the case from just rules 1 and 2 above. This is because by only applying Temperature(g, 70 degrees), we have not explicitly ruled out that this action may have influenced the shape of G. (It could have been that since the object was heated, the shape of the object would change, which in fact is quite possible). Therefore, if in reality the shape actually cannot be changed by heating it, we would have to formulate this explicitly, by stating logically that for the first formula the Shape(x,s) holds after Heating(x,t) if Shape (x,s) was held beforehand. By stating this, we can make sure that the shape is not influenced by heating the object. These formulas, where we rule out that a particular action has no influence on the properties of an object as described by another action, are called frame

axioms. This is because these axioms "frame" the context in which we can logically deduce the consequences of a particular action; we are sure that only those properties that are described by the frame axioms actually change after an action is performed on the object.

However, one can already guess where we run into trouble with the formulation of frame axioms. Returning back to the example, if we apply Heating to the object g, we saw that we had to explicitly formulate that the shape was not affected by this. But there are of course way more properties of the object that are not going to be affected by the action Heating, and these all have to be explicitly formulated, or "framed" as well. For example, we also have to explicitly state that the color of the object does not change after heating, or the smell, or any other properties that can be described to the object that is not affected by Heating. However, the number of obvious properties that are not affected by applying an action to the object, is enormous and possible infinite. Here, we can formulate the frame problem in its original, computational form: how can we logically describe both the effects of a particular action, without explicitly having to write the infinite amount of obvious non-effects of a that action?

Several solutions to this original problem have been proposed, all from different perspectives. For the purpose of this thesis, it is not relevant to go into every solution from the literature, but at the moment most scientists agree that the original, computational frame problem is more-or-less solved (Shanahan, 1997). One of the proposed solutions is described by (McCarthy, 1986), and is based on the technique of so-called predicate circumscription.[1] The idea here is that an axiom scheme is added to the logic, which basically says that any instance of a predicate that is not formally described as being true, must be false. By stating this only once, one "circumscribes" the need to constantly explicitly formulate new formulae, which would be necessary in classical logic. By stating that any instance of a predicate that is not formally described as being true must be false, one does not have to add new axioms each time a new action is performed. In this way, this approach solves the computational frame problem as all the non-effects are now accounted for. However, in practice, it is often the case that an action sometimes has a lot of unintended side-effects. Since these effects have not been described by the initial logic, the solution of predicate circumscription automatically assumes that there are no unintended side-effects. Here we see that while the solution formally works, in practice it still seems to be problematic. Other solutions run into

[1] For interested readers, another solution using modal logic can be found in Schwind (1978).

similar problems when considered from a practical standpoint, which is why philosophers such as Dennett (1984) suspected a deeper problem. In the next chapter, this deeper problem will be introduced and discussed.

The epistemological frame problem

While philosophers generally agreed that the frame problem in its original, computational form is more or less solved, there were other philosophers that suspected a way deeper, epistemological frame problem, such as Dennett (1984) and Fodor (1987). The deeper epistemological problem is best portrayed by an example that Dennett (1984) gives. In the example, one imagines a robot whose task it is to retrieve an object from a wagon in a room nearby. However, there is also a bomb on the wagon, that would go off when the wagon would be moved. The robot can perceive the bomb, but cannot see that the bomb would go off the moment the wagon would be moved, or that it would go off eventually if the robot takes too long to do anything. By some simple reasoning and observations, the robot can deduce a way to retrieve the object from the room. First, it notices that the object is on the wagon. Second, it observes that the wagon has wheels, and can therefore be moved. The robot concludes that it should thus pull the wagon out of the room, such that his goal is accomplished. However, the problem now of course is that the bomb will explode, and the object with it, such that the robot has failed to accomplish his goal. Therefore, the robot needs some revision, such that it can account for all (side) effects of his actions. The robot thus decides to work out all the consequences of its actions before doing anything. However, to accomplish this, one would have to keep an enormous list of all the consequences of each possible action. So far, the problem resembles the original, computational frame problem. Dennett describes what would happen:

> *"It had just finished deducing that pulling the wagon out of the room would not change to color of the room's walls, and was embarking on a proof of the further implication that pulling the wagon out would cause its wheels to turn more revolutions than there were wheels on the wagon—when the bomb exploded." (Dennett, 1984, p. 129).*

Of course, we can now observe that the problem the robot faces is that it calculates the consequences of actions that are completely irrelevant to the goal at hand. So, the designers of the robot conclude that the robot should not calculate all the consequences of its actions, but only those consequences relevant to the problem at hand. However, here we get at the heart of the epistemological frame problem: how can a robot, or any cognitive

agent for that matter, *a priori* determine what is and is not relevant to the problem at hand, without explicitly considering every possible consequence of its action? After all, how can a cognitive agent ever determine that a consequence of an action is irrelevant, if it has not explicitly considered that consequence at all? The problem of determining which consequences of action are relevant and which are not without explicitly considering the relevance of every possible consequence (because there are an indefinitely large number of possible irrelevant consequences), is defined as the epistemological frame problem.

Is the epistemological frame problem really a problem?

The interesting thing about the epistemological frame problem, is that even though it is still considered a real problem, one could see that the problem is at least approximately solved in humans. Humans are reasonably competent in determining the relevance of sensory input to a particular goal, even in environments that haven't been encountered before. We do this constantly without considering everything in our environment and labeling most things as irrelevant. We simply have to ignore them in the first place. In a way, we thus intelligently ignore a lot of our perception. But how is that we can do this reasonably successfully? After all, perhaps a lot of what we ignore in our perception is actually relevant? In any way, it is clear that humans at least implement an approximate solution to the epistemological frame problem in the sense that we do, in fact, reasonably determine what consequences of an action are relevant for us in any moment in any given domain without considering every possible consequence of the actions we are considering.

The fact that humans implement a solution prompts us to consider whether the epistemological frame problem is really a problem at all. Arguably, in the deepest sense, the epistemological frame problem is a problem that can never, in principle, be completely solved, in the sense that a perfect solution to the problem does not exist. To illustrate why, consider by contrast that a such a perfect solution exists. This would mean that a system capable of doing this would perfectly know the relevance as well as the irrelevance of every consequence of a particular action it intends to execute. However, we have previously seen that the list of possible consequences of an action that is irrelevant is potentially infinite! After all, one could list an infinite number of possible consequences of an action that are completely irrelevant to the system. Assuming that representing an infinite number of irrelevant consequences is computationally intractable, we can thus conclude that such a perfect solution does not exist. The quest for a solution to the epistemological frame

problem should thus not be seen as looking for the perfect answer, rather, a solution to the problem will always be approximate, but never perfect.

The global workspace theory as a solution to the epistemological frame problem

One proposed solution to the epistemological frame problem from the literature comes from Shanahan and Baars (2005), in the form of the Global Workspace Theory (GWT) of consciousness. The GWT is a model of consciousness or cognition in general, that proposes that human cognition is run by various specialist parallel processes, all of which are connected to a global workspace. The global workspace determines which of these parallel processes gets access to the global workspace, and after access has been granted, the global workspace broadcasts the contents from the specialized process to the entire set of specialized processes. In figure 1, this idea is illustrated.

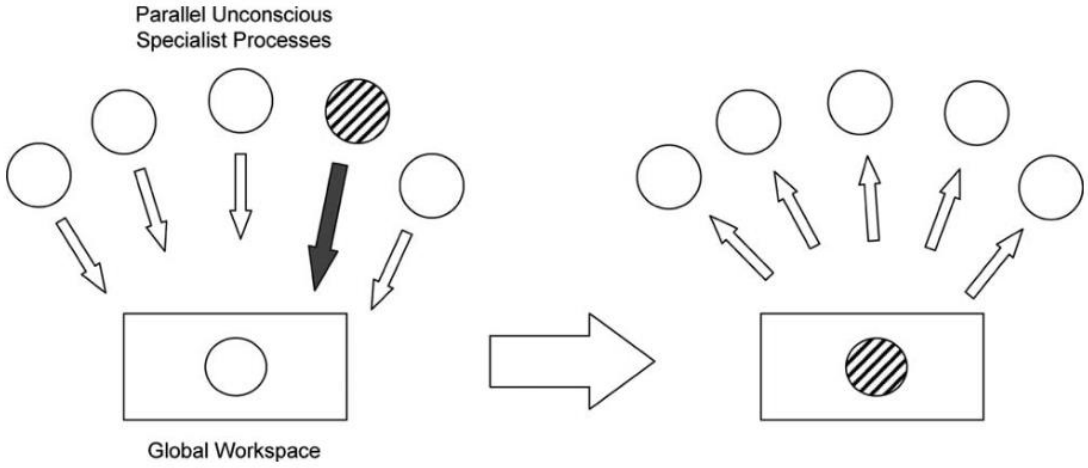

Figure 1. Illustration of the concept of a global workspace. One local workspace is selected, sent to the global workspace, and then signaled to all the local workspaces (Shanahan & Baars, 2005).

Remember that the epistemological frame problem deals with the question of determining relevant input from perception without explicitly considering every possible perceptual input, as that would require too much computation (how is perception 'framed' when there are an almost infinite ways to frame a perception?). Shanahan and Baars argue that the intuition that is used in the description of this problem is that determining relevancy of perceptual input supposedly happens in a serial manner; in other words, they argue that the epistemological problem is only a problem when determining relevancy is done in serially as that would require too much computation. They propose that the GWT, by making use of a massive number of parallel processes, can deal with the epistemological frame problem, by outsourcing the determination of relevance to these parallel processes.

By making use of parallel processes that each are responsible for determining relevancy that can all run at the same time; they argue that the frame problem resolves as the amount of computing power is greatly reduced.

While I think that the GWT design structure does contribute to a solution to the epistemological frame problem, I would argue that Shanahan and Baars overstate the capacity of the GWT to fully solve the epistemological frame problem. This is because even when using a massive number of parallel units to determine relevancy, thereby reducing the need for computing power, the required computing power still diverges to infinity. An example would be the playing of a game of chess; after playing the first move; the player can calculate what the best response to any move is by the opponent (for example, by using GWT where a parallel process for each possible move tries to calculate the best response), however, the possible responses in the game after each possible move that can by analyzed by the parallel process is still close to infinite; using parallel processes here does not solve the problem as the amount of possible consequences of actions are still way too much too calculate.

Another argument against the GWT as a complete solution to the frame problem is that Shanahan and Baars assume that the relevancy of input of each parallel process is independent of input by any other parallel process, but I would argue that this may not be the case. The relevancy of certain perceptual input may only be relevant insofar the system has perceived something else as relevant from a different parallel process. To check this, the system would still need to make use of using a serial way of determining if input b is present in any of the parallel processes (because only if so, input a is relevant to the system). I would thus argue that in some instances one cannot circumvent the necessity of serial processing to determine relevancy.

However, while not being a full solution, the GWT is useful in assisting in an approximate solution the epistemological frame problem. The GWT will come back later on when discussing cognitive architectures, as one of them implements this theory.

Artificial General Intelligence

In the academic study of Artificial Intelligence, one can broadly distinguish between two types of artificial intelligence. The first one can be defined as "artificial narrow intelligence", first explicitly defined by Kurzweil (2005). This type of artificial intelligence refers to applications that are competent at carrying out one specific task, but cannot extend their skills to novel domains. Most applications of artificial intelligence today concern applications that fall under the category of artificial narrow intelligence. Voice-assistants, recommendation algorithms and gaming AI's are all examples of narrow artificial intelligence; the AI is very competent for the task at hand, but cannot its domain of competence beyond its original task (an artificial narrow intelligence that is designed to be very skilled at chess for example, has no idea how to interpret language and recommend restaurants nearby). The other category of artificial intelligence would be the integration of various artificial narrow intelligences: artificial general intelligence. In contrast to an artificial narrow intelligence; an artificial general intelligence displays competence across several domains, and has an ability to transfer knowledge from one or several domains in such a way that it can intelligently use it in a novel, never encountered domain.

While the creation of such a general intelligent machine has always been the most ambitious goal since the start of the field, the development of such a machine appeared to be quite difficult. One of the first attempt at the creation of an artificial general intelligence was the General Problem Solver (GPS) by Newell and Simon (1959). In the General Problem Solver, a problem was defined as having four elements: a representation of the initial state, a representation of the final state, a representation of all the operators that can be used to transform one state into another state, and finally, a set of path constraints to limit the set of possible solutions. A solution to any problem formed by the General Problem Solver would then entail a description of the exact path that brings the problem from its initial to its final state, by using the set of operators that are allowed, and conforming to the path constraints. However, the General Problem Solver admitted of two big problems, as Vervaeke (2012), points out. First, the GPS assumes that any real-world problem can be represented as a particular state in a computational machine, in other words, the GPS assumes that all problems are well-defined. However, most real-world problems are not well-defined but ill-defined. The problem of having a productive conversation for example is not well-defined: there does not exist a one-size-fits-all solution of the perfect productive conversation. A second problem is that, in the case a problem is

well-defined, the set of possible paths that one can take from the initial state to the solution (final state), get exponentially larger by each step. For example, if each state can be transformed into three new possible states, with every step on the path, the possible paths are multiplied by 3: after the first iteration, 3 states are possible, after the second, each of these three states can be transformed into three other states giving a total of 9 possible paths, for the third iterations we get to 27, etcetera. It is clear that the number of possible paths that lead to the final state are therefore 'combinatorally explosive', as Vervaeke (2009) puts it. The number of possible paths from the initial to the final state will quickly run to infinity due to the exponential nature of the path expansion. It was thus clear that a real general problem solver was quite far from reaching that point. Therefore, most of the field has since focused on the creation of artificial narrow intelligences, which seemed to be more successful due the fact that they concern well-defined domains and problems. These efforts have resulted in successful technologies such as voice recognition software, chatbots, image generators, etcetera.

However, in recent years, renewed interest in the development of an artificial general intelligence has been sparked. The most notable example of this was the creation of the annual Artificial General Intelligence conference, which was first organized in 2008 (AAAI, 2008). The conference was initiated as means for researchers interested in the creation of artificial general intelligence to come together, share ideas and goals, and make progress in the field. In his seminal review paper on artificial general intelligence, Goertzel (2014) provides several characteristics and concepts of general intelligence that are mostly agreed upon by scientists within the artificial general intelligence community, and these are as follows. First, there will be no such thing as unlimited or arbitrarily general intelligence, due to resource constrains in the real world, and the fact that the learning time and difficulty may differ per task or domain, resulting in bias for more intelligence for some tasks over others. Second, humans currently display a higher level of general intelligence than existing AI applications, and finally, that it seems very unlikely that the general intelligence as displayed by humans is the maximum achievable intelligence.

Goertzel furthermore elucidates the distinction between the study of artificial narrow intelligence and artificial general intelligence, and formulates this distinction in his "Core AGI hypothesis", which goes as follows:

> *The creation and study of synthetic intelligences with sufficiently broad (e.g. human-level) scope and strong generalization capability, is at bottom*

qualitatively different from the creation and study of synthetic intelligences with significantly narrower scope and weaker generalization capability.

This highlights that the two categories of artificial intelligence are indeed conceived of as very distinct projects, and may require fundamentally different methods. I think this hypothesis is likely true, for several reasons. A practical reason is that we currently have success with ANI and not with AGI, which seems to indicate that the latter involves dealing with a qualitatively different problem. More importantly, the goals of ANI and AGI are completely different. The first is concerned with designing an optimal machine capable of dealing with a well-defined specific task. For the latter, the real crux of the problem is not the developing of competence at one domain (even though that is still a part of it), but rather how the system can transfer knowledge from one domain to another and deal with novel, not before encountered domains of actions and show learning and competence there. The real problem in AGI is therefore to effectively deal with new information and transform this new information into new competence. For the purpose of this thesis, I will presume the hypothesis to be true.

While some characteristics of artificial general intelligence can be defined, it should be emphasized that a full overarching definition of artificial general intelligence is still lacking. This is mostly a result of the absence of final definition for (general) intelligence itself. The final goal of what would constitute an artificial intelligence may therefore shift as progress is being made and our understanding of it changes with it. Goertzel (2014) proposes two main characteristics of AGI, which I have hinted at before.

- An AGI should be able to carry out a variety of tasks, achieve a variety of goals, and do this in a variety of different domains and contexts. This characteristic thus referrers to the "general" capacity of AGI.
- Furthermore, the AGI should be able to effectively transfer knowledge from one domain of competence to another, this thus also refers to the capacity to generalize to other domains.

However, I would argue that there is another characteristic that is absolutely necessary for artificial general intelligence, and that is the ability to effectively deal with new information, learn from this information, and transform it into new knowledge and competence. This is because any successful AGI must be able to handle new situations well, and this can only be done when it knows how to zero in on the right information from this domain and knows how to transform this into new competence.

Different approaches to artificial general intelligence

Goertzel (2014) describes several approaches or perspectives in defining artificial general intelligence, which I will discuss here. One of these approaches, the cognitive architecture approach, will be discussed in more detail, as it will form the main focus of this thesis.

A first approach to conceptualizing artificial general intelligence is formulated by Nilsson (2005) and is defined as the pragmatic approach. This approach derives its pragmatic aspect from the fact that accomplishment of artificial general intelligence is defined as the ability to perform most of the tasks or jobs that a human being would do, making the tasks for humans obsolete. Artificial general intelligence in this sense is not explicitly defined, but is expressed in terms of how well a system is able to imitate humans by performing their tasks, thereby assuming that humans are the prime example of general intelligence. It also does not matter if the system is actually intelligent, whatever that would mean, the only metric that is looked is the practical results of its actions; namely, can this artificial intelligence perform as well as humans on most tasks? Note that in this definition, an artificial general intelligence beyond human level general intelligence is not considered. However, as stated earlier, most AI researchers agree that general intelligence most probably extends well beyond human level general intelligence. In a sense, this conceptualization is therefore limited in its scope.

A second approach, the psychological approach, is a way to artificial general intelligence that tries to elucidate the display of general intelligence by describing underlying mechanics and design features. This differs from the practical approach, where only the outcome mattered. The approach is similar however in the sense that artificial general intelligence is compared to achieving human level intelligence. Some of the well-known characteristics of general intelligence in psychology concern the intelligence quotient and the so-called g-factor (Parnassum & Klee, 1998). The g-factor was introduced because researchers noted that performance on one specific task, for example linguistic capability, was often strongly correlated with performance on other specific task (for example, mathematical capability). Adams et al. (2012) have compiled a list of underlying psychological capabilities or categories that are suggested to be required for artificial general intelligence as seen in humans. These are as follows: perception, actuation, memory, learning, reasoning, planning, attention, motivation, emotion, modeling self and other, social interaction, communication, quantitative/mathematical, building/creation. These 12 capabilities may differ in importance but are generally agreed upon to be necessary for an artificial intelligence to be general.

The third, mathematical approach is fundamentally different from the two approaches described earlier; while the pragmatic and psychological approach specifically concerned the accomplishment of human-like general intelligence, the mathematical approach extends beyond that, and asks how general intelligence could be defined irrespective of human-level general intelligence. This approach therefore assumes that general intelligence is not limited to human level general intelligence.

The adaptationist approach conceptualizes artificial general intelligence as the capability for a system to adapt to new environments using limited resources (Wang, 2006). In this sense, one could make a parallel with the theory of evolution: those offspring that display the highest fitness by being most adapted to the environment, are in this definition considered to be most intelligent. In an interesting way, the degree of general intelligence could therefore be compared to the chances of survival in new environments. Humans possess the highest degree of general intelligence of all known organisms according to this definition, as they display the highest capacity to adapt to different environments.

Another approach to describing artificial general intelligence is the embodiment approach, which argues that general intelligence can only be displayed by physical systems in a physical environment. From this perspective, general intelligence should be viewed as the proper modulation of the physical systems in relation to the environment it is in (in order to achieve its goal). Stronger, Pfeifer and Bongard go as far as to argue that a software AI cannot considered to be intelligent (Munari, 2009), since it is not embodied. This approach to artificial general intelligence will come back later in the thesis when discussing the theory of relevance realization.

A final conceptualization to developing artificial general intelligence, and which will form the main focus of this thesis, is the idea of cognitive architectures. Cognitive architectures are basically frameworks describing how several parts of cognition within an artificial system relate to each other, and how processing within subparts of the system is executed. Thórisson & Helgasson (2012) emphasize that cognitive architectures are not simply a description of how independent parts of cognition do their processing; they should also provide a framework of how the different parts of cognition work together and are integrated as a whole, thereby forming one structure that is more than the sum of its parts. They furthermore describe the cognitive architecture approach to developing artificial general intelligence as the most promising approach within the field, which is also the reason why this approach is used as the main focus of this thesis.

Relevance Realization

So far, I have discussed the epistemological frame problem and AGI as two separate subjects. In this chapter, I will argue for a link between the two and introduce a framework that serves as an attempt of a solution to the epistemological frame problem and as a design framework for artificial general intelligence. To see the connection between the epistemological frame problem and general intelligence, recall that the epistemological frame problem revolves around the problem of determining what consequences are and are not relevant to a particular action. More broadly speaking, this not only concerns the consequences of action, but actually involves all domains of cognition, most notably perception. Determining what data for cognition is relevant and what is not is the epistemological frame problem in its broadest sense. Now, let's turn back to some characteristics of general intelligence. One of the most important properties of general intelligence is the ability to be competent at many domains of action, and also have the ability to effectively learn to be competent in a completely new environment. More specifically, one thing that is integral to general intelligence is the ability to quickly adapt and develop competence in basically any new, never before, encountered domain of action. This is also again what distinguishes AGI from ANI; an ANI could in principle never learn to be competent in any new domain that it wasn't specifically designed for.

The question then arises: what makes system most able to learn in a completely new environment? Arguably, at the heart of the capability to effectively adapt and develop competence in a new domain of action is the ability to determine relevancy, in its broadest sense, but most importantly of new perceptual input. Vervaeke and Ferraro (2013) have also argued for this close connection between general intelligence and the ability to determine relevancy. They discuss various core parts of cognition, such as problem solving, communication, categorization, rationality and environment interaction. Vervaeke (2012) also shows how the ability to determine what is or is not relevant is at the heart of the ability to perform well in each of these important domains of cognition. For the purpose of this thesis, it is too much to discuss every domain separately (for details one can refer to the paper), but as an example, take the cognitive ability of categorization. To be able to internally categorize various perceptual input, one must be able to determine shared characteristics between two objects such that they form a category. For example, take a tennis ball and a football. Do these belong in the same category? Well, most people would say so, because they share similar features: both are round, are used for a sport, and can

move around. However, we can also name as many differences. One is a lot bigger than the other, the outside textures are different, and their weights are different. Based on the latter differences, one could just as well argue that they do not belong in a category. But which one is right? Well, it depends on what characteristics are relevant to the categorization that is made. There are almost infinite similarities and differences between any two objects, but categorizing them together involves determining which of those properties are relevant to the category.

For every other mentioned domain of cognition, one eventually runs into the same problem of determining relevancy. Hence, we can see that it is central to successful cognition and indispensable for general intelligence. While a technical and complete argument for this connection may require more philosophical argumentation and consideration, for the purpose of this thesis, I will assume that the ability of a system to determine relevance, in its broadest sense (being able to do it quick, accurately and effectively) as being equal to the degree of which a system is generally intelligent.

Now, one of the first attempts to a solution of the problem of determining relevance and thus of the epistemological frame problem comes from Vervaeke (2012), which I mentioned a bit before. In this outline of his theory of 'relevance realization', as he has refers to it, he formulates a design framework by which human beings, or artificial systems for that matter, are able to determine the relevance of both sensory input and internal input. Before going into the specifics of this theory, the first point to get clear and that he mentions is that a theory of relevance itself is scientifically unviable. This is because every scientific theory or explanation has to deal with a stable, homogenous class of data that it purports to give a theory or explanation about. For example, the theory of the force of gravity is a scientific theory since it describes that every object with a mass will be under the influence of this force. All objects are homogenous in the sense that they all have a mass and stable in the sense that their property of mass does not change in a different environment. If we look at the property 'relevance', of an object however, we see that this property does not relate to a stable, homogenous class; the relevance of an object is namely very much dependent on the environment and task at hand. There are no class of things that are relevant on a Thursday, for example. It does not make sense to talk of set of homogenous relevant objects, and it is not stable because the relevance of an object may thus differ depending on the task at hand or the environment it is in. It is therefore clear that a theory of relevance in itself does not make sense scientifically. However, a theory of relevance realization, a scientific theory of the process by which relevance is realized may actually

exist. The analogy that Vervaeke makes is to that of the concept of biological fitness: there is no scientific theory about the fitness of a particular organism, but there exists a scientific explanation for how the fitness of an organism is realized, namely the theory of evolution.

I have argued for and assumed that relevance realization is a valuable skill for any general intelligence as it relates to virtually all domains of cognition. So how would an attempt at a formulation of a framework of relevance realization look like? Vervaeke explains that his proposal is by no means a full account of a final theory, but proposes three characteristics that such a theory should incorporate at minimum. To restate, the goal of a theory of relevance realization is a description of how a cognitive agent that exhibits general intelligence is able to determine relevant perceptual and internal input from a potentially infinite number of possible information, across a wide variety of domains, including new domains that haven't been encountered before. A theory of relevance realization should therefore describe the mechanics of how the relevant input for the task at hand is determined in any domain of competence, including new ones. Again, the theory does not describe what will be labeled as relevant input in any domain per se, since we have seen that the set of relevant inputs in any domain is not a stable homogenous class. First, I will discuss three characteristics that Vervaeke proposes a theory of relevance realization should incorporate. After that, I will introduce two more characteristics which I would argue also are necessary components of a theory of relevance realization. The goal here again is not to deliver a complete theory of relevance realization, but rather to illustrate what the necessary characteristics are that such a theory should have.

Self-organization

The first of the characteristics that Vervaeke proposes is that the machinery of relevance realization is a self-organizing process. Self-organization of a system refers to the ability of the system with many components to generate a particular functionality without any central control. The functionality is generated from local interaction in the system, and via feedback mechanisms these local parts interact to generate a particular behavior for the system as whole. The organization thus becomes an automatic process that emerges from local interaction, and can therefore be called self-organizing (e.g. no external input or control is needed to direct its organizational and functional structure. The main argument that Vervaeke makes for the importance of dynamically self-organizing and constantly changing framework for relevance realization instead of preprogrammed rigid system

refers back the idea there exists no scientific theory of relevance, due to its inherently goal and context-depending nature). To determine relevancy in any new domain of action may require a completely new design structure to realize relevance that wasn't needed before. Therefore, a self-organizing framework, that can constantly adapt itself to new needs and environments is something that a system capable of relevance realization should have. Another way of seeing this is again making the parallel with the theory of evolution; the optimal fitness of an organism is constantly changing dependent on the environment that constantly changing; therefore, an organism that has rigid design cannot deal effectively with new environments.

Bio-economic model

A second characteristic that Vervaeke proposes as a principle of a theory of relevance realization is that it should make use of a bio-economic model of allocating computational resources. A bio-economic model of cognition stands in contrast to the functional computational model of cognition. The latter model, popularized by Fodor (Green, 1996), makes use of symbolic or syntactic representations and manipulates this to work out the action for the agent to achieve its goal at hand (this is therefore similar to the computational theory of mind). However, the problem with functional computationalism is that it cannot determine from the syntactic representation what is relevant input to the goal at hand. Fodor recognizes this, and explains that the reason for this is that syntax is locally defined, but that relevance is globally defined (e.g. the syntactic representation are relevant to a global goal, but this global goal is often very abstract and cannot therefore be syntactically defined). Therefore, a theory of relevance realization needs a model whereby local representations and manipulations are defined by certain local rules that also have a direct global effect on the goal at hand for the agent in that particular environment. Here, the global goal does not need any syntactical representation, but we can infer from local rules what the effect on the global goal will be.

An analogy to a real economy would be for example the goal of the growth of domestic production of all goods. To syntactically represent this goal in terms of the symbols of all local syntactical representation would results in an infinite calculation that would be impossible to represent. Instead, the relevance of any local input is determined directly to what kind of influence it has on the global goal. If one company produces more flowers than it did last year, we can deduce this has a positive effect on the global goal

(increase in domestic production), and this local interaction is therefore relevant to the goal at hand. We can thus make use of certain local rules that influence the global goal, without explicitly representing the global goal in terms of all the local rules. This property is interestingly relatable to a complex system, which will be discussed later on. The global behavior or goal of a complex system is not defined in terms of the whole of the system, but emerges from the sum of all local interactions that contribute to that goal. The representation of the global goal is therefore thus defined by local interactions.

Balancing of constraints by opponent processing

The final characteristic that Vervaeke argues for is that the mechanics of relevance realizations should involve the balancing of multiple constraints that serves opposite functions, rather than a specialized machinery. He argues that a specialized machinery for relevance realization would not circumvent the epistemological frame problem that it intends to solve; it would face combinatorial explosion in the face of all possible input that could be relevant to possible action, where inputs could even depend on one another for being relevant. Furthermore, a specialized machinery for relevance realization would basically mean that it is a general-purpose learning algorithm, which has been shown to be impossible to create (Wolpert & Macready, 1997). The so-called No Free Lunch Theorem that Wolpert describes shows that all learning algorithms can never be completely objective and have at least some sort of bias towards certain functions over others. Any ambition therefore to come up with some sort of general-purpose learning algorithm, necessarily has to use a bias or heuristic to certain functions it uses. Therefore, the best solution to this seems to be the goal of dynamically balancing certain constraints/functions, starting out with a bias towards one constraint and constantly changing parameters to see whether one gets closer to the goal. More specifically, the balancing should be between two functions that to do the exact opposite. Think for example of the regulation of blood sugar in humans; the goal there is to maintain an optimal (relevant) value of the blood sugar in the blood such that the organism as a whole can best serve its goals. In an organism, this is not solved by a specialized machinery for maintaining blood sugar within range, but rather two simple processes that to do the exact opposite: the release of insulin to lower blood sugar when it is too high, and the opposite, the release of glucagon to increase blood sugar when it is too low. There is no overarching machinery here that incorporates these two processes, rather, there is a dynamic self-organizing interplay between these two processes depending on the

current blood sugar value; if it is too high, there will be a bias towards insulin release, if it is too low, it will be based towards glucagon release. Note that the opposite function to take in the same value as input (current blood sugar value). The same kind of opponent processing happens for several domains in relevance realization, Vervaeke argues. He argues that due to the continuous dynamic balancing between opposing functions, the explicit binary relevance of an object is never calculated, but rather the degree to which it may be relevant (just as the explicit optimal value for blood sugar does not exist, but rather a range of possible good values). Relevance realization by opponent processing of function also ties in with the economic model of representation: depending on the value of a certain input in relation to the goal of two opposing functions, there will be more cognitive resources to one function rather than its opponent function, which thus is a bio-economic way of dividing computational resources.

Complex systems theory

So far, we have discussed three characteristics for a theory of relevance realization that Vervaeke has proposed. Before introducing two other characteristics that are also arguably important to the framework of relevance realization, I would first like to discuss the concept of a complex system and complex systems theory, as the two added characteristics follow from the premise that the machinery of relevance realization in the brain, or in artificial system for that matter, is analogous to a complex system. Boschetti (2011) discusses the concept of a complex system and its characteristics. There currently exists no explicit, set-in-stone definition of a complex system, but there are nevertheless several aspects of a complex system that most academics agree on.

Examples of well-known complex systems are the climate, the economy or our brain. Complex systems are systems that operate on the border between order and disorder; they are neither mechanistic and fully predictable, nor completely random. They consist of many individual parts that are connected to one another, where each individual part is governed by some set of rules that determine the interaction with other connected parts, resulting in certain global behavior. Furthermore, complex systems show emergent properties, meaning that the system as a whole shows behavior that is more than the sum of the individual parts and thus cannot be reduced to its parts. They are self-organizing, meaning that the system as a whole is configured to a particular state without central

control. Finally, a complex system is networked in a particular way as to optimize efficiency and robustness, the network is often scale-free, small-world and clustered.

Interestingly, if we compare these properties of a complex system with the three proposed characteristics of a theory of relevance realization by Vervaeke, we see some striking similarities: self-organization, economic allocation (whereby global behavior results from local interactions) and balancing of constraints (which can be achieved by allocating a specific rule set to individual agents in the complex system) seem to all be properties of a complex system as well. Here, I will argue that the mechanics of relevance realization within in the brain can therefore best be described by a complex system. If this is right, then a theory of relevance realization should be a theory of complex systems. More specifically, the mechanics behind relevance realization require a scale-free, small-world clustered network of processing units and the complex system behind relevance realization will show emergent behavior, e.g. higher order phenomena in determining relevance. Finally, I would argue that a system capable of relevance realization must necessarily be embodied and cannot simply be an abstract program. This last feature of embodiment is not a necessary feature of a complex system per se (the economy or climate are not 'embodied' in any meaningful way) but I would argue is still necessary for doing relevance realization, as the relevance of a certain object is often dependent on it being useful for the execution of a physical task in relation to that object, which can only be performed by a physical body. I will discuss these two aspects separately.

Complex network characteristics

Since the mechanics behind relevance realization closely resemble those described by a complex system, it is highly likely that the mechanics behind relevance realization show properties typical of a complex system. One of these properties is that that the system can be described as a complex network, meaning it shows properties of a scale-free, small world, clustered network of processing units. These properties of a network are important, as they result in a network that has the optimal balance between efficiency (processing speed within the network) and robustness (The ability of the network to be resilient against changes within the network, e.g. the sudden removal of a particular node). These two properties of a network make the network most capable of dealing with new information and optimizing computing power, essential for doing relevance realization. The property of a network of being small-world refers to the fact that the maximum distance between

any two nodes in the network is small. In practice, this means that given any node in the network, any other node can be reached via a relatively short number of other nodes. This property is important for quick coordination within a system, because information that is transferred from any node in the network can be reached by other places in the network relatively quickly. In figure 2, a small-world network is depicted, where the maximum distance between any two nodes in the network is minimized. Interestingly, this structure of the network is neither fully regular (ordered) nor fully random (chaotic). As described earlier, any complex system operates on the border between order and disorder, and this aspect is thus also typical of a small-world network.

Figure 2. Examples of regular, small-world and random networks.(Watts & Strogatz, 1998)

The property of a network being scale-free refers to the fact that within the network there exists an inverse relation between the number of nodes and the number of links that node has with other nodes. For example, if there exists a network that has 4 nodes with each 3 links to another node, we can expect the network to have 2 nodes with each 6 links, and 1 node with 12 links. This type of network is governed by a so-called power law; since the degree (= the number of links a node has) is proportional to the negative power of some constant of that degree. To see how a self-organizing network often exhibits power-law behavior in the distribution of its links, consider the fact that in the beginning of the formation of a network, nodes will form links with neighboring

nodes at random, resulting in some nodes with more links to others than other nodes. Given this situation, if a new node is introduced to the network at random, it has a higher chance of being close to a node with many links than being close to a node with only few links (simply due to the fact that a node with many links is close to any node in the network). This implies that the new node is more likely to be close to a node with many links, resulting in a higher chance of that node later on pairing with that node with many links. This is a positive feedback loop where nodes with many links eventually attract more nodes than nodes with fewer links. These dynamics results in the earlier described power-law behavior of scale-free networks. The name "scale-free" results from the fact that on whatever scale the network is considered, the functional structure of the network will be the same as the number of links per node are governed by the same rule, no matter how big the network may get; the number of nodes with a particular number of links will always be inversely proportional to the number of links (see figure 3).

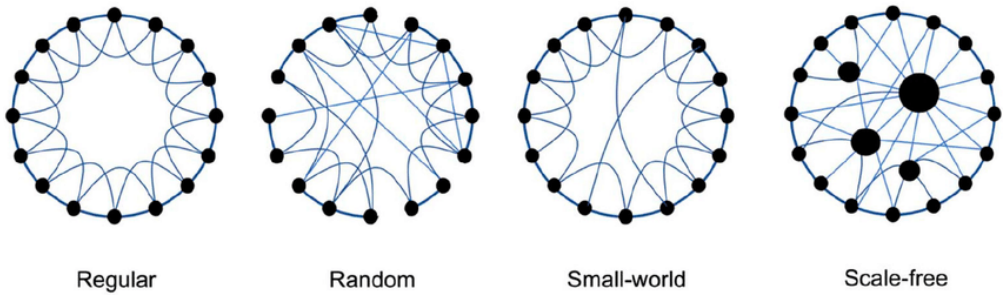

Figure 3. Examples of regular, random, small-world and scale-free networks.

Finally, a self-organizing complex network often shows clustering behavior. The reason for this is similar due the reason of power-law behavior: The probability of two nodes A and B to pair is larger if A is paired to C and B is paired to C compared to when neither are A and B are paired to C. This results in a higher chance of clustering behavior compared to the network having random links for every node.

Embodiment of system

As I have also argued before, the mechanics behind the process of relevance realization can be described as an embodied complex system. To see why embodiment may be necessary for a general intelligent system, Miracchi (2022) builds up an argument from the premise of so-called "semantic efficacy." This concept relates to the fact that semantic content of mental processes or attitudes are causally relevant, thus that particular semantic

content is directly causally linked to earlier processed semantic content. An example of this would be that when your body senses it gets warm, it will release sweat to compensate for this. This is a direct causal semantic link that doesn't require a complex set of internal computations. This combined with the premise of "semantic externalism", which states that content is generally externally metaphysically determined, meaning that mental content is generally derived from a source external to the agent itself, results in an argument for embodied cognition; that mental processes are generally metaphysically externally determined. Computational reductionism, or the Computational Theory of Mind (CTM), Miracchi argues, denies semantic efficacy. To see why, consider again that the CTM states that mental processes can be fully described by a series of computations over a certain input or representations. The output of a particular input is thus fully determined by internal processing. The input of a CTM can be semantic, but semantic content is not necessary by design. The system can be described purely by internal processes alone. Semantic efficacy however states that there is a direct link between certain semantic input and the formation of new semantic processes, without using a complex set of internal computations to accomplish that, hence why it is called semantic efficacy; a direct causal link between semantic processes results in much quicker processing that using input over a complex set of internal computations. By using a semantic efficacious process that accomplishes the exact opposite of another semantic efficacious process, a higher order balance can emerge from these two processes (as described earlier under the section of balancing of constraints). Interestingly, one could also see how from these opposing set of semantic efficacious processes a higher-order feature of the system emerges. Emergence is typical of a complex system as well, further indicating that opponent processing of semantic efficacious processes resulting in higher-order emergence, such as is suggested to be a part of a theory of relevance realization, is indicative of the mechanics of relevance realization being a complex system.

What follows from these arguments, is that one could argue that the degree of embodiment of a system is basically degree to which the system makes use of (opposing sets of) semantic efficacious processes. Embodiment thus reduces the need for computing power in a system. The more a system can directly interact with the environment and process new semantic content directly without first using a complex set of internal computations, the quicker and more efficient the system can behave as a whole. In conclusion, a system capable of relevance realization would prefer a system that is capable of as much semantic efficacious computations as possible as to reduce the need for internal

computations, and it thus would prefer a system that is embodied as much as possible, which would be the argument for the necessity of embodiment for a theory of relevance realization.

Interconnectedness of the five features

In total, there are now five features that a theory of relevance realization should have that have been discussed. Three of these were inherited from Vervaeke (self-organization, economic model, balancing of constraints), and I have suggested adding two more to this list (complex network design and embodiment). Taken together, the current proposal is a system capable of relevance realization consists of a large complex network of processing units that is self-organizing, has an internal economy that makes use of reward mechanisms to steer cognitive resources, is embodied in such a way that it can physically interact with its environment and uses several opponent processes/constraints to reduce the need for complex internal computations. While these features have been discussed separately, in a sense they all work together in one system: the complex network is self-organizing, and the units in the network make use of internal rewards and punishing mechanisms after which cognitive resources are allocated in self-organizing fashion, and opponent processing is directly involved in embodiment and the internal economy of reward mechanisms.

Cognitive Architectures: Definition and Evaluation

Cognitive architectures are abstract descriptions of an artificial intelligence that describe the processes by which the system as a whole operates and functions. They describe how input is processed, how the system may be compartmentalized, how these parts are related to each other, and how the system generates an output or performs an action. It is not a complete and large description of all the technical details involved in the system, but rather a more abstract description that gives a broad idea of the functionality and processing within the system.

These cognitive architectures may or may not be instantiated in an actual physical system. To date, a lot of different cognitive architectures for general intelligence have been developed, all with different design features and characteristics. There are many ways to classify cognitive architectures, but Duch et al. (2008), in their review paper, have created a broad categorization of these architectures, based on their fundamental design structure. These three categories are symbolic, emergent and hybrid. In many cognitive architectures there is overlap between these three categories; the distinctions are, therefore, mostly based on which category is most, not exclusively, represented. Symbolic cognitive architectures are built on the premise that the mind simply works by representing aspects of the world and itself internally as symbols, and manipulating these symbols in order to execute actions in relation to its goal. The symbolic approach of AI is therefore analogous to the computational theory of mind, which states that the mind is simply a machine that makes computations over symbolic representations. Another category of cognitive architectures is based on the so-called the emergentist approach, which is based on the premise that all of the symbolic processing of the mind emerges from deeper subsymbolic processing (hence the term emergentism). Neural networks, one of the most successful applications of artificial intelligence, are an example of subsymbolic processing. This type of processing is often successful when one needs to extract patterns from a large amount of data.

Both the symbolic and subsymbolic approaches have their advantages and disadvantages: symbolic architectures are good at for example abstract reasoning, and subsymbolic architectures are good at extracting patterns from large quantities of data. However, symbolic approaches lack the ability to do quick pattern recognition based on sensory input, which is often required for intelligent behavior. Subsymbolic approaches on the other hand miss the higher-level organization of data to be able to do abstract reasoning or comprehend and produce language, which is also essential for intelligent behavior. A

third approach to cognitive architectures, the hybrid approach, therefore, uses both symbolic and subsymbolic parts in its architectures and combines these to take the best of both worlds and create one whole.

Assessing performance of cognitive architectures for AGI

There already exists some literature on how to assess performance of a cognitive architecture for AGI. In particular, I will discuss the assessment characteristics as proposed by Laird et al. (2009) and Thórisson and Helgasson (2012). Laird et al. proposed a set of necessary characteristics for the creation of a cognitive architecture for artificial general intelligence. These characteristics were compiled after the first scientific workshop on artificial general intelligence in 2008, as described earlier, where scientists in the field shared their vision of what would constitute the creation of an artificial general intelligence system. Laird et al. construe these characteristics in such a way that they can be scientifically tested and validated. By conceptualizing the characteristics in terms of testable claims, the approach of creating an artificial general intelligence has become a valid scientific enterprise. The claim is often about a hypothesis of relations between a particular characteristic and other measurable factors. In each of the claims, the characteristic (independent factor) is varied and several other factors relating to the characteristic (dependent variables) are measured, as a means to validate the claim. Laird et al. (2009) categorize four types of claims that can be made about an artificial general intelligence:

- A claim about a cognitive capability or behavior similar to a human cognitive capability, such as the ability to improve performance after experience or to understand natural language.
- A claim about the improvement of an artificial general intelligence by modifying its cognitive architecture. Improvement here means an expansion of the set of problems the system is able to solve.
- A claim about the difference in performance between two separate artificial general systems
- A claim about the similarity between behavior of the artificial system and human behavior. Note that this is not the same as the first category, where cognitive capabilities in artificial systems are evaluated on its own metrics, claims of this

category are made where human behavior is the target metric instead. (this type claim is therefore a special type of the third category).

Laird et al. describe the types of independent variables that can be studied within these types of claims as follows:

- System components; these can be seen as new modules or parts of the cognitive architecture that result in a change in performance of the system, for example vision, talking etc.; these components are categorical and as a consequence not quantifiable.
- Amount of knowledge; this concerns how much knowledge is being represented within a cognitive architecture, often per system component.
- System parameters; these are global factors that influence the performance of the system as a whole. An example of a global parameter in a human being would be gender; this is a global parameter that affects many parts of the whole system, including varying performance outcomes.

Finally, they describe a set of dependent variables or measurements that a cognitive architecture can be studied on:

- Performance metrics; these are the most straightforward and measurable metrics of a system. Think for example here of the speed of performance for a specific task, or the quality of the solution that was found. This metric is almost always used in assessing the performance of a narrow artificial intelligence.
- Scalability metrics; in contrast to artificial narrow intelligence, scalability metrics are quite important in assessing the performance of an artificial general intelligence. Since a general intelligence operates in a multitude of domains, it requires a vast amount of knowledge that can be built on each other; scalability metrics are therefore essential to the assessment of an artificial general intelligence. This metric is also quantifiable.

Next to these quantifiable metrics, artificial general intelligent systems exhibit behavior that is not easily captured by quantifiable metrics. These more abstract measures are as follows:

- Task and Domain Generality; this metric concerns how well a system performs across a set of different domains; the hallmark of what constitutes an artificial general intelligence
- Expressivity: this refers to what types of knowledge a system can use in its cognitive processing (e.g. symbolic, relational)
- Robustness: this metric refers to the extent of which the performance of the system changes when new knowledge is added or removed; this metric is of interest due to the inherent flexibility that is often required of a system aiming for the capacities of general intelligence; it often has to use impartial or partially wrong knowledge when trying to reach a goal. Measuring the change of system performance in relation to change of knowledge used is therefore of interest.
- Instructability: this refers to the ability of system to capacity to effectively learn new knowledge from other systems; and thus related to all things related to learning new tasks or gaining new knowledge.
- Taskability: this concerns the creative part of a general intelligent system; how well is that system able to generate new task that are not explicitly programmed, but are relevant to the goal at hand? Human beings are an example of general intelligence that perform well on this metric.
- Explainability; this final metric refers to the ability of a system why it has performed in a certain way.

As mentioned earlier, the naming of explicit metrics on which a cognitive architecture for artificial general intelligence should be evaluated results in a clearer view of what constitutes progress towards the goal of creating artificial general intelligence by means of a cognitive architecture. Other researchers have also tried to independently come up with some metrics that a cognitive architecture should be assessed on for its progress towards an artificial general intelligence. Thórisson and Helgasson (2012) suggest an intimate relationship between autonomy and general intelligence: they propose that autonomous systems are systems automatically perform tasks in an environment in order to achieve a particular goal, where unforeseen changes in both the tasks and environments can occur that can be mediated by some type of automatic learning and adaptation. The system is in this sense autonomous because it does not require an external agent to modify its architecture when faced with new challenges. General intelligence, as described before,

concerns an ability to adapt to unforeseen changes in environments and tasks at hand; and Thórisson and Helgasson therefore argue that these two concepts are closely linked.

They then propose several metrics that the performance of an autonomous system can be evaluated on, and that therefore also pertain to a general intelligent system. These evaluation metrics are of particular interest since they seem to be closely related to earlier discussed features of relevance realization.

- Realtime system operation: the ability of the system to perform in relation to the environment as quickly as possible. The quicker a system can handle new input either from internal processes or from the environment and act on it accordingly, the more efficient it will operate and the more it will be able to do in a set amount of time. This is also important for relevance realization as determining relevance also requires a continuous environment interaction.
- Resource management: the ability of the system to allocate its computational and sensory resources in an appropriate manner as to optimize its performance. This is a necessary feature given that resources are limited but that an intelligent system in a complex environment has to deal with way more information than it can process in a timely manner. This relates to the bio-economic model of allocating resources in relevance realization as described before.
- Learning: the ability of the system to improve its performance over time. The better it learns, the shorter the amount of time is to improve its performance. This in a way relates to the self-organizing property in relevance realization, as learning involves a continual change in the structure of representation.
- Meta-learning: the ability of the system to change the way it learns new capabilities and to change its internal processes as a consequence. Also refers to the self-organizing aspect of relevance realization.

A new set of evaluation metrics

So far, two papers have been discussed that propose a set of metrics to evaluate the performance of a cognitive architecture. Earlier on, I have argued that in order for an artificial intelligent system to exhibit general intelligence, it must deal with the epistemological frame problem, and consequently have a mechanism for relevance realization. Therefore, I would propose an entire new set of evaluating cognitive

architectures for artificial general intelligence based on the theory of relevance realization: the more a cognitive architecture incorporates each of the five discussed features of relevance realization, the better it is equipped to serve as an architecture for general intelligence. Again, the reasoning here is that if a system incorporates the five features well, it consequently has a good basis for doing relevance realization, and is thus fit as a design for artificial general intelligence.

Discussion and Comparison of Cognitive Architectures

In this section, I will discuss and analyze four different cognitive architectures based on their the guidelines discussed above. Since there is a relatively tight connection between the evaluation metrics for autonomy studied by Thórisson and Helgasson and the features of a mechanism for relevance realization, I will pick a few cognitive architectures that get a good score on all the dimensions for autonomy as my starting point. These will be the CLARION, LIDA, AKIRA and IKON FLUX cognitive architectures. Most of these systems score relatively well on autonomy metrics and should, therefore, be promising candidates for relevance realization. AKIRA does not score that well on autonomy metrics, but is of particular interest here since it was especially designed to incorporate features of a complex system. Lastly, the cognitive architecture that will also get special attention will be LIDA, as it is an instantiation of the Global Workspace Theory (Baars, 2005), which has been proposed as a solution to the epistemological frame problem (Shanahan & Baars, 2005), as discussed earlier in the chapter on the frame problem.

First, the cognitive architectures will be discussed on their own shortly to give a broad overview of their design features. Afterwards, a comparison between them will be made in a similar fashion to Thórisson and Helgasson (2012). For each evaluation metric, a cognitive architecture can get 1 to 5 points, where 1 point means the cognitive architecture does not incorporate the characteristic in its design at all, and 5 stars means the cognitive architecture has the characteristic fully and close to optimally integrated. Granted, it can be quite hard to objectively evaluate how well each characteristic is integrated in the architecture. Therefore, the scoring will be mostly on a relative basis, e.g. an architecture will get a high score if the feature is more integrated compared to the other architectures, and a low score if the feature is not well-integrated in the architecture compared to others. The comparison will be made for each feature of relevance realization, thus there will be subchapters for each feature where the four architectures are compared to each other and are given relative strengths. In the end, I will suggest a short design proposal for a new architecture that takes the strengths from each architecture and combines these, such that a novel architecture is generated that incorporates design features for relevance realization well and consequently should be a good candidate as an architecture for artificial general intelligence.

CLARION

The CLARION architecture is discussed in (Sun, 2007), and consists of four main interacting modules. These modules are the Action-Centered Subsystem (ACS), the Non-Action-Centered Subsystem (NACS), the Motivational Subsystem (MS) and the Meta-Cognitive Subsystem. In figure 4, you can see the interactions between these modules and their subcomponents.

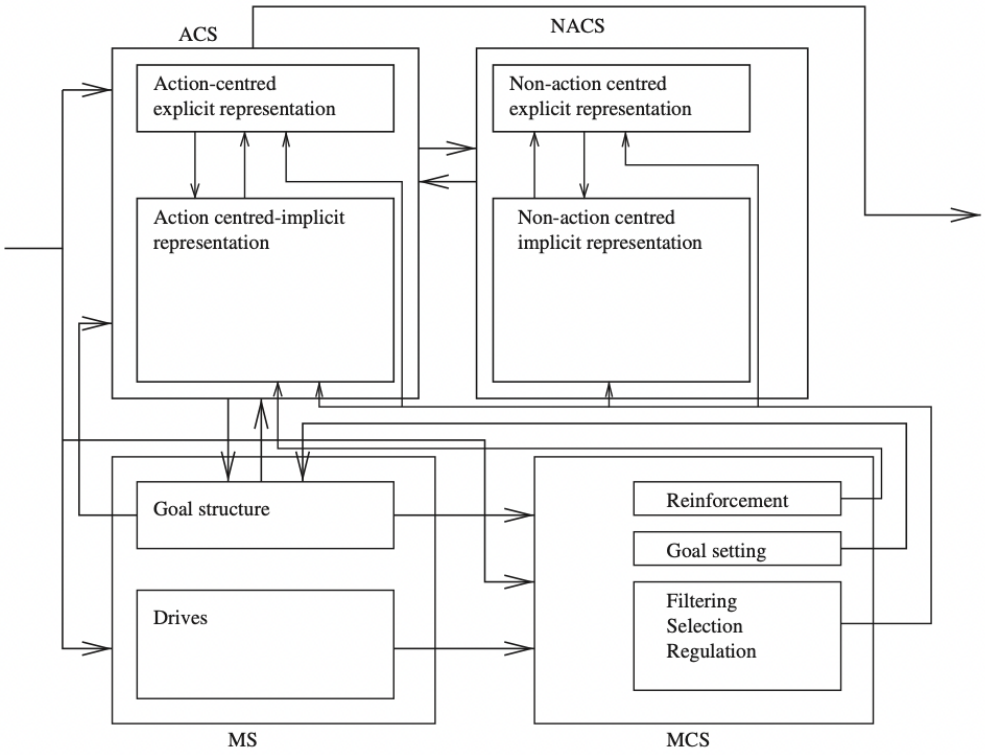

Figure 4. Overview of the CLARION architecture (Sun, 2007),

The ACS concerns itself with the execution of actions, both in the environment as well as internally. The NACS deals with the maintenance and representation of general knowledge. As can be seen from the figure, both have a subpart specific for explicit (symbolic) and implicit (subsymbolic) representation. Both these subparts also interact and influence each other. The MS is concerned with the motivation and underlying perception, action and cognition. It provides impetus and feedback for the system. Finally, the MCS provides monitoring, modulation and control for all the other submodules, but mostly on the ACS. Even though it may not be explicitly stated in the figure, the MS and MCS modules also incorporate both a symbolic representation that is top-down, as well as subsymbolic representation that is bottom-up, so all modules consist of this basic structure. These two subparts communicate with and influence each other, and output from these

subparts is often a combination of suggestions from both the symbolic and subsymbolic parts. The author described that the cognitive architecture was developed with the following intentions and thus shows signs of these features. First, the system should be able to learn with or without *a priori* knowledge. This means that it can generate new knowledge based on existing knowledge representations, but also that it can generate new knowledge without those representations, thus purely from perceptual input. Second, the system should be able to continuously generate new knowledge based on on-going experience in its environment. It should also be able to learn different types of knowledge (for example procedural vs declarative knowledge). Finally, the system should incorporate motivational and meta-cognitive processes.

LIDA

The LIDA cognitive architecture (Franklin et al., 2007) is a cognitive architecture based on the Global Workspace Theory of consciousness (GWT) as developed by Baars (2005), as described earlier in the chapter on the frame problem. The operation of the LIDA cognitive architecture is based on a series of so-called cognitive cycles. The cognitive cycle of LIDA with all of its modules and relationships is displayed in figure 5.

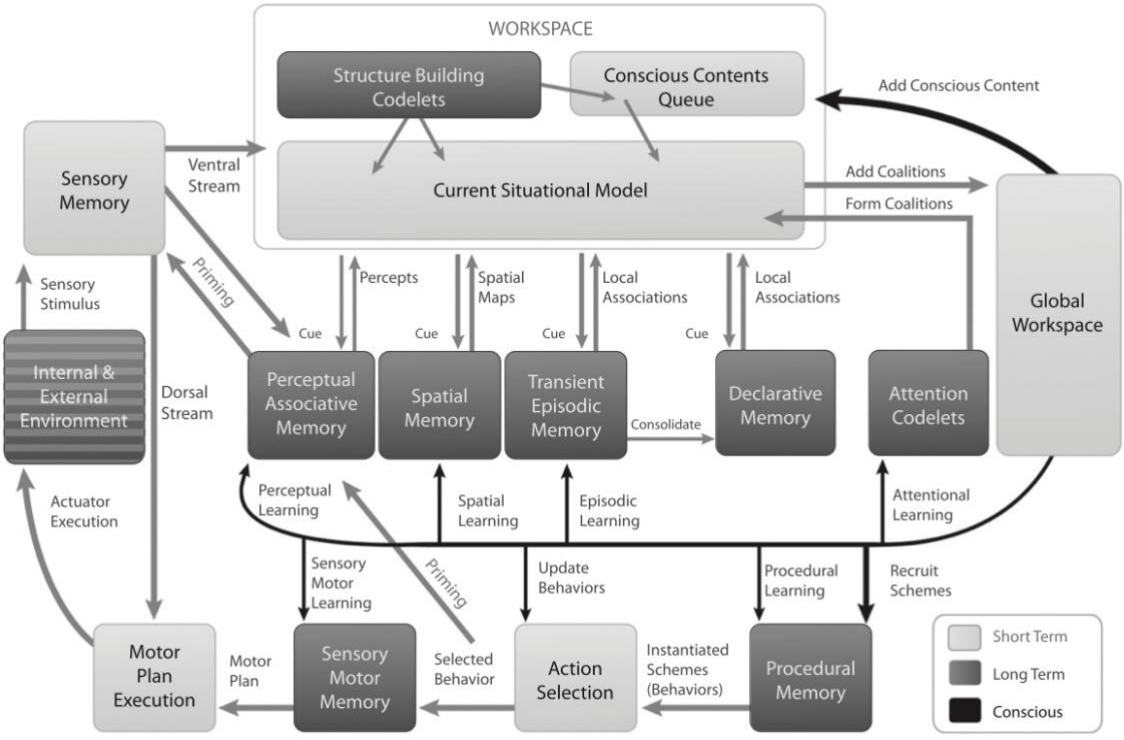

Figure 5. Overview of the LIDA cognitive architecture (Franklin et al., 2012).

Each cognitive cycle consists of three phases: sense, attend and action selection. In the sense phase, the current knowledge of both the internal and external environment is updated by means of new perceptual input. Low-level features of this input are sent to the perceptual memory, where this can be further classified as higher-order content. It is then sent to the local workspace (top of the figure) and compared with existing episodic and declarative memory to see the perceptual input matches input from memory. This series of events constitutes the sense phase, as it now has constructed a new representation from recent perceptual input. The attend phase is concerned with dividing the input from the local workspace into coalitions of data that are functionally related, which is done by the so-called attention codelets. A competitive process then results in one of these coalitions having the highest importance, and only that coalition is then sent to the global workspace. The global workspace then sends the information from this coalition to several parts of the system that involve the cognition of the system, such as the several types of memory, action selection and the attention codelets. The information sent to the procedural memory and attention codelets results in learning of these parts from the information of the coalition in this cycle. Finally, in the action selection phase, the procedural memory computes possible actions, in part based on the current input from the global workspace. These actions are then sent to action selection component which selects one action to execute.

AKIRA

The AKIRA architecture, whilst scoring relatively low on the autonomy metrics used by Thórisson and Helgasson, is of particular interest here because it was built with the intention to create a system that shares features of a complex biological system. In their 2007 paper, the authors Pezzulo and Calvi (2007) mention that the inspiration for the design of AKIRA came from particular characteristics of complex systems, namely self-organization, adaptivity and robustness. Figures 6 and 7 explain the design of AKIRA.

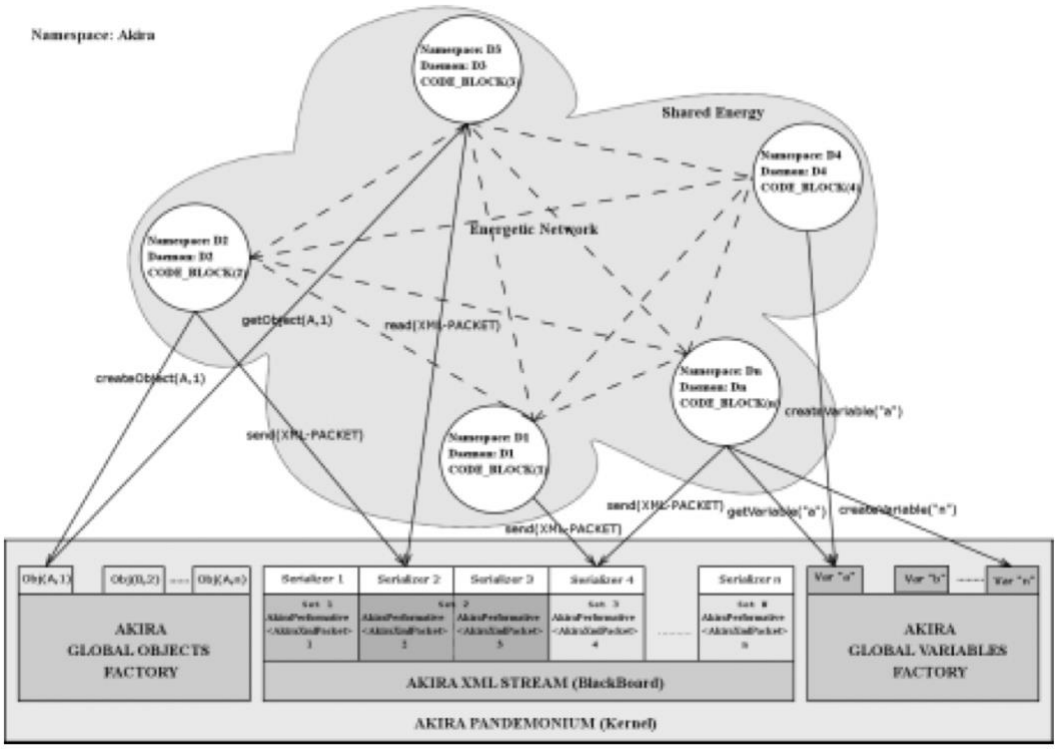

Figure 6. Overview of the AKIRA cognitive architecture (Pezzulo & Calvi, 2007).

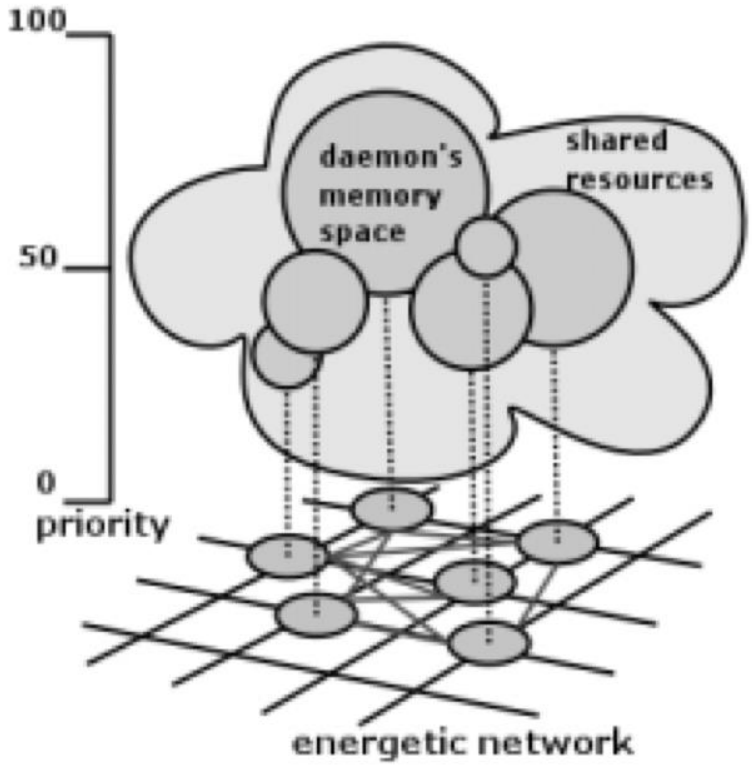

Figure 7. Illustration of the energetic network in AKIRA (Pezzulo & Calvi, 2007).

The basic idea behind AKIRA is that it consists of a large set of so-called modules. Each module is responsible for one part of cognition and can be either simple such as moving an arm, or complex, such as visual perception. Each module is represented as a node in an energetic network and has connections to some other nodes in the network. It also has an activation value that represents the number of resources the node is able to use. There is one pool of resources that all nodes can take from, and the activation value of every node determines what proportion of this shared resources pool is allocated to that module (see figure 7, a higher activation value results in a higher priority value and thus more energy allocation). The modules can run asynchronously and in parallel. A module is also called a 'daemon', and the network of daemons is connected to a monitoring system called the 'pandemonium'. The pandemonium is responsible for the execution and monitoring of the whole system. It consists of a blackboard that is responsible for communication and coordination, plus the global variables and objects that apply to every daemon.

IKON FLUX

IKON FLUX is a proto-architecture that has design features similar to that of AKIRA, but additionally offers a self-learning feature that results in ability of the whole system to continually and in real-time grow and adapt to its environment and redesign its own architectures (meta-learning), starting from only minimally specified initial conditions. Being a proto-architecture, the complete design is not worked out yet, but its proposed design features are nevertheless interesting, hence the inclusion of this architecture. IKON FLUX is similar to AKIRA in the sense that its design also consists of many modules. However, the modules in IKON FLUX are a lot smaller and there are much more of them. K. R. Thórisson & Nivel (2009) call this feature *peewee granularity*. IKON FLUX generates new modules via bottom-up information coming from for example sensory data, combined with top-down models existing programs. Another design feature of IKON FLUX is that it incorporates *computational homogeneity*. This refers to the fact that modules can be built up only from a very small set of one particular computational substrate, with the size of one peewee (the smallest possible computation is thus a set combination of operations and has the size of one module). The structure of this small computational substrate allows for growth and increase complexity when these small modules are combined in a particular way. The best analogy to the design of IKON FLUX may in fact be the structure of the

human brain, which also consists of a very large amount of small modules (neurons) and thus exhibits peewee granularity. Furthermore, every small module has a given set of operations and is homogenous for every module in the network, just like each neuron in the brain has the same basic design structure and functionality, thereby also reflecting computational homogeneity. In IKON FLUX, as well as the brain, coalitions of modules/neurons can form to generate higher-order cognitive functions.

Comparison on Features of Relevance Realization

So far, the four different architectures have been described in sparse detail to give a general impression of their design features and differences. In this section, these architectures will be compared to each other and evaluated on the basis of each feature of relevance realization. This will be done per feature, so each of the five features will get a different paragraph in which each of the four architectures are discussed.

Self-organization

As discussed before, the feature of a cognitive architecture to be self-organizing means that it has an ability, at least to some degree, to build up its own architecture without a central control that directs the organization. Rather, the architectures organization follows from the local interaction and dynamics between its parts organically. Essential for a self-organizing cognitive architecture is also that the final organization must show some emergent new behavior; an architecture can be built organically from local interactions between parts, but if there does not emerge a higher order functionality from that, the interactions between the parts do not form any meaningful self-organization. In conclusion, it refers to the ability of the architecture to organize itself meaningfully as if it were directed by a central control mechanism, without there actually being a central control mechanism.

A look from the overall view of CLARION reflects a predetermined overall organization of the architectures, namely its 4 subparts that are each responsible for a part of its cognition. In that sense, CLARION cannot be said to be self-organizing. However, each of the four subparts all incorporate a so-called dual representational structure, where explicit, top-down representations interact with implicit, bottom-up representations in a dynamic fashion to produce a meaningful output, as discussed before. For example, in the ACS, an action is chosen by observing its current state x from sensory data and computing he quality for each possible action from the implicit representation. In CLARION, this is done with the help of neural network. The same observation of state x is sent to the explicit representation. The explicit representation determines all possible moves based on the input and the already existing represented rules for action. Then, the possible moves from the explicit representation are chosen from the implicit representation, and the best action gets chosen.

After this action is performed, the new state of the system is reflected back to both the implicit and explicit part; to update the algorithm and its rule set respectively if the action is successful. In this sense, the choosing of an action, the expansion of the rule set, and the parameters of the implicit representation are all updated in a self-organizing manner as to produce a higher-order functionality, namely an improved ability to choose the appropriate action. Next to this, the system also does not have a strictly reductive approach in its design for various aspects of cognition, such as separate modules for memory, reasoning, problem-solving etc. Rather, these capabilities emerge from dynamic interactions between the four parts of action representation, non-action representation, motivation and meta-cognition. In this sense, the architecture can also be said to be self-organizing, as these aspects of cognition are not predetermined but emerge organically.

On to LIDA, we again do see a predetermined structure of its higher-order organization in various modules. In that sense, the architecture cannot be said to be completely self-organizing. However, the subpart responsible for prioritizing which coalitions of functionally related data get sent to the global workspace does so in a competitive, self-organizing manner, according to Franklin et al (2007).AKIRA, which distinguishes itself from both CLARION and LIDA in the sense that part of the intention of the design was to create an architecture capable of self-organization without a central control. The main part of AKIRA's design that implements the self-organizing idea is its large reservoir of modules which dynamically interact for resources and selection. Due the finite number of resources that all modules have access to, their emerges a competition for resources in a self-organizing manner, the authors argue. The limited number of resources results in systemic features such as cooperation, hierarchical organization, exploitation and context awareness. They explain that for behavior such as selection and cooperation of modules, a central control is generally required, but that AKIRA is able to achieve these things without a central control by means of the competitive interaction between modules and a limited total amount of resources.

Finally, IKON Flux is similar to AKIRA as the intention of its design was to create a self-organizing architecture. However, compared to AKIRA, IKON FLUX trumps the ability of self-organization, since the whole structure builds itself from the ground up (mainly through environment interaction) from only a small set of initial conditions. Basically, the whole architecture is built up in a self-organizing fashion. Of all the architectures, IKON FLUX thus best incorporates this feature.

Bio-economic model

A second feature of relevance realization that Vervaeke proposes is the presence of a bio-economic model of rewards or punishments to appropriately allocate cognitive resources. The term economic refers to the fact that the system makes use of internal reward or punishment mechanisms per unit of the 'economy' (e.g. the whole system/network), which influences the share that that particular unit gets from the total of cognitive resources. In other words, there is a finite amount of cognitive resources that the system can use in any moment for internal processing or perception, and the system should thus allocate resources based on those units that require it the most in that moment. An example would be a module in system where the action from a moment ago resulted in successful behavior; that module then gets a reward and signals that to the entire system, after which it is allocated more cognitive resources. In a way, this feature is tied in with the self-organizing ability of the system, as the sum of successful and unsuccessful actions of different units/modules results in an automatic and dynamically self-organizing allocation of cognitive resources. No central hub is thus necessarily required for this mechanism either.

Let us first take a look at CLARION. For its ability to allocate cognitive resources, the two main modules to look at are the MS (Motivational Subsystem) and MCS (Meta-Cognitive Subunit). The MS is concerned with the different drives of the system (e.g. hunger, sex as a biological example) and how these drives interact as to influence action selection. Some of these drives are predetermined (such as the basic ones for survival), but second-order drives can be derived from satisfying or not satisfying the basic drives and can change over time. The Meta-Cognitive Subsystem works closely with the Motivational Subsystem and provides the control and monitoring for the system as a whole. On the basis of certain active drives in the MS, the MCS can decide to for example interrupt an action from the ACS, set parameters for the NACS etc. as to optimize the performance of the system as a whole. These two systems thus work together two provide a mechanism for the allocation of cognitive resources; for example, based on certain perceptual input (low energy), the MS can prioritize the drive to find food, thereby pointing cognitive resources to look for food. Each cycle the system gives feedback to the system as a whole, after which drives may change or the MCS may interfere with the performance of the system.

CLARION thus does have a mechanism for allocating cognitive resources, and it is somewhat self-organizing. Several drives are already pre-programmed and cannot be changed, thus here we see a slight lack of self-organization. However, for the most part,

the system makes use of internal rewards in the ACS, sends this back to the MS, and consequently the drives and thus cognitive resources are re-evaluated, and this happens dynamically in a self-organizing manner (because it is constantly based on perceptual input and output from the system a moment ago). So in conclusion, the system has a fairly good economic mechanism for allocating cognitive resources.

When we look at LIDA, we see that is particularly designed to incorporate a self-organizing, economic way of allocating cognitive resources. The idea behind the global workspace theory, which is incorporated in the design of LIDA, is that several different modules all compete for selection to be sent to the global workspace. The module with the highest prioritization gets that access and that information is sent back to all modules from the global workspace. In LIDA, so-called attention codelets form coalitions of similar data in the local workspace, after which these coalitions compete for selection to be sent to the global workspace. While it would be interesting to consider the precise mechanism of competition between coalitions, Franklin et. al (2012) do not specify this in their paper. It can be speculated though that prioritization is given to coalitions that deal with mostly new information. Incoming perceptual input is compared with working and long-term memory, and coalitions that have information that is not present in memory would probably be more likely to win competition for cognitive resources, as the system has more interest in dealing with new information compared to old information which it has already processed. The important thing to note though is that LIDA, by implementing the global workspace theory, makes use of a completely self-organizing dynamic way of allocating cognitive resources, without a central control. On this feature, the LIDA architecture thus performs well.

AKIRA has a design that shows a lot of similarity to LIDA when it comes to dealing with resource allocation. As discussed before, the system consists of a lot of different modules that all compete for a finite number of resources, e.g. implicitly competing for resources from the system in a dynamically self-organizing manner. The difference with LIDA is that in LIDA there is only one coalition that wins in each cycle (a winner-takes-all, discrete mechanism), whereas in AKIRA, multiple modules at once can get resources, albeit that some modules get more resources than others. The resources or activation that each module in AKIRA receivers is determined by three elements: Base Priority, Energy Tapped and Energy Linked. The Base Priority is a predetermined value that is given to a module that is private and not shared with the other modules. The Base Priority of a module can be higher or lower based on the type of module (for example a feature vs.

concept). Energy Tapped refers to the amount of energy the module gets from the energy pool, the total amount of resources. Energy Linked refers to the energy that it gets from other modules in the network that it is linked to. The energy a module gets from other modules is determined by the strength and number of the links to other modules. This in turn is determined by the success of the module; if in previous cycles the module shows successful behavior (e.g. predicting matching outcome), the number of links and strength of links to other modules can be increased. To determine the score on this feature, AKIRA can again be best compared to LIDA, which scored already performs quite well on this feature. The mechanism for resource allocation is similar to LIDA, but since AKIRA has a more dynamic, continuous mechanism for resource allocation (multiple modules can get resources at once relative to their activation level), the mechanism in AKIRA is in my estimation more sophisticated. Both allocate resources in a dynamic, self-organizing fashion, but AKIRA's design is even better in my estimation due to its relative allocation of resources instead of a winner-takes-all strategy in LIDA. Therefore, AKIRA performs a little better than LIDA.

Finally, we move on to IKON FLUX. Since the design of IKON FLUX is based on a very large number of very small modules, that can all work in parallel, an efficient way of allocating resources is necessary, since the system of cannot attend to all small modules at the same time. In IKON FLUX, every module has a certain activation value, that determines the relative importance of the module. Perceptual input of the environment as well as internal input is also given an activation value. The authors do not specify what exactly determines this activation value, but for perceptual input this will most likely correlate again with novelty, and for existing modules this will most likely correlate with relevance of that module relevant to the current goal structure. For input (perceptual or internal) to be attended to by IKON FLUX, both the input as well as a corresponding model that is able to deal with the input must have an activation value that exceeds a given threshold. This mechanism of resources allocation thus also happens economically (local modules activation determine global resource allocation) and in a self-organizing manner (no central control is required for resource allocation; it is automatically assigned to modules and input with large enough activation values). However, due to the lack of description of the precise mechanism, the performance of IKON FLUX will not be rated as high as AKIRA.

Balancing of constraints by opponent processing

The next feature for relevance realization is the appropriate balancing of constraints by opponent processing, as argued by Vervaeke. As explained before, this proposed feature for relevance realization is about the presence of certain processes where each process has two mechanisms that each try to accomplish the exact opposite of each other, resulting in a homeostatic balance. Vervaeke argues that the presence of such opponent processes is a requirement for relevance realization, and that relevance realization emerges from the self-organizing, dynamic balancing of these constraints.

Looking at CLARION first, each of its 4 submodules has a dual representation structure, as discussed before. This means that each module has a top-down, explicit representation part and a bottom-up, implicit representation part. These two parts work together to form for example an action selection or they work together in learning new skills. Technically speaking, this dual representational structure does not entail two opponent processes where each process tries to attain the exact opposite of the other, e.g. the implicit and explicit representation actually work together to try to reach the same goal instead of working in direct opposition of each other. In conclusion, while a dual representational structure is one of the core design features in CLARION as it is present in each of its 4 submodules, it does not work strictly by balancing opponent processes. Therefore, CLARION scores average for this feature.

When we look at the design structure of LIDA, we unfortunately do not see any specific mechanism that make use of a balancing of opponent processes. Where it does have a sophisticated mechanism for resource allocation as discussed earlier, there is no place in its design where two opponent processes compete such as to establish a homeostatic balance. Competition does take place within the coalitions in the local workspace, but again this competition does not involve any processes that do the exact opposite; at least insofar that has been described by the authors. On this feature, LIDA therefore scores relatively low.

In AKIRA then, we at first again do not see specific processes that each try to do the exact opposite of the other in its basic design. However, one could see that this could in fact in principle be instantiated if two modules in AKIRA develop tasks that are opposite to each other. For example, let's say that one module develops a responsibility to make sure the system gets enough food, e.g. its goal is to acquire new energy. Another module on the other hand is concerned with for example training its stamina, thereby using energy in trying to reach that goal. Both take in the variable energy (as an example, not to confuse

with the total Energy the AKIRA system can divide amongst its modules). To determine its actions, however, the result of both processes is opposite; if energy is low, the module for foods searching will get more priority from the system and the training for stamina module will be mostly deactivated, if energy is high on the other hand, it will be the exact way around. During the lifetime of the system there will be a continuous balancing between these two opponent processes. In conclusion, AKIRA's system is, due to its flexible nature, able to develop modules that do opponent processing and therefore scores relatively high on this feature.

Finally, when we look at IKON FLUX, we see a design similar to IKON FLUX but with many more small modules and even higher flexibility than AKIRA in the development of those. Naturally, one would expect that even more modules with opponent tasks develop during its lifetime, and therefore, in a way similar to AKIRA, should be able to make use of opponent processes in balancing constraints even better. In conclusion, IKON FLUX therefore scores the highest on this feature, because of the highest potential it has to develop such opponent processes.

Complex network characteristics

This feature, not explicitly mentioned by Vervaeke but argued before by me earlier on, deals with the overall structure of the cognitive architecture, and whether it has as design structure that resembles features of a complex network, e.g. does the system have a lot of interacting parts that are connected to each other in some way (such that it would form a network), and second, does the type of connectivity resemble that of a complex network in particular, meaning it has the properties of being small-world, scale-free and clustered? Again, the idea behind a cognitive architecture having such a design is that it is optimally designed to balance processing speed (efficiency) and its ability to deal with and incorporate new information (robustness).

CLARION's design structure is not typical of a (complex) network. Again, it is divided into only four main modules. Granted, these modules do interact quite intensely with one another, plus there is some communication with each module. However, a real network structure is not present in its design, not to mention a complex network structure. Therefore, on this feature, CLARION scores relatively low.

LIDA suffers from a similar problem, its design is mostly predetermined and does not consist of a large network of interacting units. The only place where a network exists

in the design is concerning its competition of the coalitions as prepared by the attention codelets. The coalitions all interact with each other in a competitive way as to attain a winner coalition that is sent to the global workspace. The authors do not specify again how this competition takes place and thus there are also no details about how the coalitions would interact, but since there can be quite a lot and competition between them takes place, there must be some form of interconnectivity in the coalition which would resemble a network-like structure. In conclusion, while its design has, at least compared to CLARIO, a more networked structure in the competitive process of determining a winning coalition for the global workspace, the overall design is not resemblant to that of a large complex network of interacting modules. LIDA therefore scores average on this feature.

Moving onto AKIRA however, we see a totally different design, and in this architecture there is definitely a network structure present. In AKIRA, part of the design philosophy was to create a large set of dynamically interacting modules, that all have a connection to each other. In this sense, AKIRA aims to haves a classic network structure. Moreover, the number of links that a module has is partially dependent on its success as a module. Naturally, the number of successful modules will be lower than the number of unsuccessful modules. However, this results in a network structure that resembles a scale-free network, where a few successful nodes have many links to other nodes, and more unsuccessful nodes have few links to other nodes. Next to this, close modules can also form coalitions to work together, resulting in clustering. In conclusion, AKIRA scores relatively high on this feature.

IKON FLUX also has this large interacting network of modules, with even more and smaller modules than AKIRA. Since its structure is almost exclusively self-organizing, IKON FLUX will most likely develop a network structure resemblant to that of a complex network, in a similar fashion as what happens in AKIRA. However, since IKON FLUX has an even larger number of modules, the network can be a lot bigger and thus allows for more flexibility and robustness than the smaller network in AKIRA. Therefore, IKON FLUX scores the highest of the four on this feature.

Embodiment of system

The final feature for a cognitive architecture for relevance realization is that it requires some form of embodiment. The embodiment of a system refers to the ability of the system to directly semantically interact with and respond to its environment, without having to

constantly refer to internal computation to determine its next move. An example of this in humans is the regulation of body temperature; there are sensors all over our body that directly interact with the environment by measuring the temperature. If the temperature gets too high, there is a direct response from the body to release sweat as to cool down the body's temperature. No internal computations are required, sweat is released as a direct consequence of a measurement of high body temperature. As I have argued before, the more embodied a system is, e.g. the more it can directly semantically interact with its environment without internal computation, the more efficient the system can behave as a whole. Embodiment can thus be seen as a form of 'outsourcing' of computation as to increase processing speed and efficiency. The feature of embodiment ties in closely to the feature of opponent processing as described earlier; the balancing of opponent processing often involves sensory processes that play a role in embodiment. Since all architectures are mere design descriptions and do not necessarily involve a physical system that incorporates them, I will - for the purpose of this discussion – judge the degree of embodiment of a system in relation to the number of various sensory modules the system possesses. It will however unfortunately be relatively short as the information on specific sensory modules in some architectures is lacking.

Starting again with CLARION; we see that the system senses its current state x after having performed an action, and this information is sent to the ACS, MS and MCS. The paper unfortunately does not specify what exactly the sensory information about its state entails; but this will most likely concern all kinds of parameters relating to certain goals it is trying to achieve. It is thus unfortunately hard to conclude how much sensory information CLARION deals with, but it is clear that the system uses new sensory information every time it performs an action, and that is an indication that it consistently interacts with its environment.

LIDA faces a similar problem, as the authors only describe that sensory information is used to update representations of the external and internal environment, but what this sensory information exactly entails is not specified. It is similar to CLARION in that with every cognitive cycle, sensory information is updated, which also indicates a constant interaction with its environment and thus a good potential for embodiment.

AKIRA and IKON FLUX both also do not explicitly specify the actual content of sensory information. AKIRA however has one more advanced feature and that is that access to sensors is also competed for (just like access to energy and other resources). In this sense, AKIRA thus allows for a bit more nuanced and complex sensory processing;

where particular modules that have higher priority have more access to sensors and thus allow for more sensory processing. Relative to CLARION and LIDA, AKIRA thus allows for more nuanced sensory processing and scores a bit higher on this feature.

IKON FLUX does not mention much about sensory processing, only that new modules are partially being built from new sensory information; remember that IKON FLUX's structure builds itself up from a very small set of initial conditions and does this continuously on the basis of environment interaction and internal information. In a sense, sensory information is thus very important for the final structure of IKON FLUX, and thus plays a big role in its development. Due to this fact, IKON FLUX pays a high importance to sensory information as it is vital to its development, and thus also scores relatively high on this feature.

Conclusions and a Short Proposal for Optimal Cognitive Architecture

Based on the analysis of the previous chapter, I will attempt to evaluate and score each cognitive architecture on each of the five discussed features relative to each other. For every feature, an architecture gets a score from 1 to 5, where 1 means that the particular feature is not incorporated at all, and 5 means that it is perfectly incorporated. The results of this scoring can be seen in table 1.

	Self-organization	Bio-economic model	Balancing of opponent processes	Complex network characteristics	(potential for) Degree of embodiment	Total
CLARION	●●	●●●	●●●	●●	●●●	13
LIDA	●●●	●●●●●	●●	●●●	●●●	16
AKIRA	●●●●	●●●●	●●●●	●●●●●	●●●●	21
IKON FLUX	●●●●●	●●●●	●●●●	●●●●●	●●●●	22

Table 1. Overview of the scores of every feature per cognitive architecture.

As can be seen from table 1, there are two cognitive architectures that perform significantly better than the two other architectures; both AKIRA and IKON FLUX attained a relatively high cumulative score when all features for relevance realization are combined. This means that both AKIRA and IKON FLUX are promising architecture designs when it comes to having potential for a mechanism for relevance realization and are thus most promising in their potential for artificial general intelligence.

This naturally prompts the question: What sets AKIRA and IKON FLUX apart from CLARION and LIDA that results in a significantly higher score for these two architectures? It seems that the overarching design feature that is most important to possess is having a large number of various small modules that all interact and communicate with each other. Both AKIRA and IKON FLUX have such a basic design, where there is a lot of flexibility possible in its basic design structure due to development and prioritization of new modules, whereas both CLARION and LIDA have a more classic design where the overall structure is mostly determined from the beginning. Recall that the five discussed features for relevance realization all tie in to each other; the goal of relevance realization is

to continuously determine what external and internal input is relevant to the system at that point. To be able to do this as well as possible, it was suggested that the system must be self-organizing, dynamic system that is able to allocate its resources by economic constraints that to opponent processing, where there are multiple units that are connected to each other in a complex network, and where the system as a whole is embodied, in that it has a lot of direct sensory interaction with its environment and can respond quickly to that sensory input. A design that has incorporated all these features, I have argued, will be a design that has most potential for being able to do relevance realization and consequently will be most promising for an artificial general intelligence.

Optimal cognitive architecture for AGI

We have seen that both IKON FLUX and AKIRA are significantly more promising for their potential of incorporating a mechanism for relevance realization compared to CLARION and LIDA. However, we can still make a distinction between the performance on specific features between the different architectures. If we take design features of the different architectures that score best on each respective feature and combine these into a novel architecture, naturally this novel architecture would score even better than IKON FLUX and AKIRA do currently and would thus be even more promising as a design for artificial general intelligence.

In general, the greatest strength of IKON FLUX and AKIRA regarding relevance realization was the presence of a large network of small interacting modules. In a new design, this is something that should be preserved. However, we saw that LIDA scored best on the feature of a bio-economic model of cognition, due to its incorporation of the global workspace theory. Thus, a novel architecture that combines the strength of having a large complex network of small interacting modules with the incorporation of the global workspace theory where in each cognitive cycle a particular module with the highest prioritization is chosen and then sent to a global workspace, after which it is sent back to the entire network, would result in a design that could have even more potential for relevance realization and thus for artificial general intelligence.

Below I will give a rough sketch of how such a design would look like. Note that it only serves as an abstract design sketch; specifics about interactions and content of modules are omitted. It is just an illustration of a basic design structure that a novel architecture may incorporate.

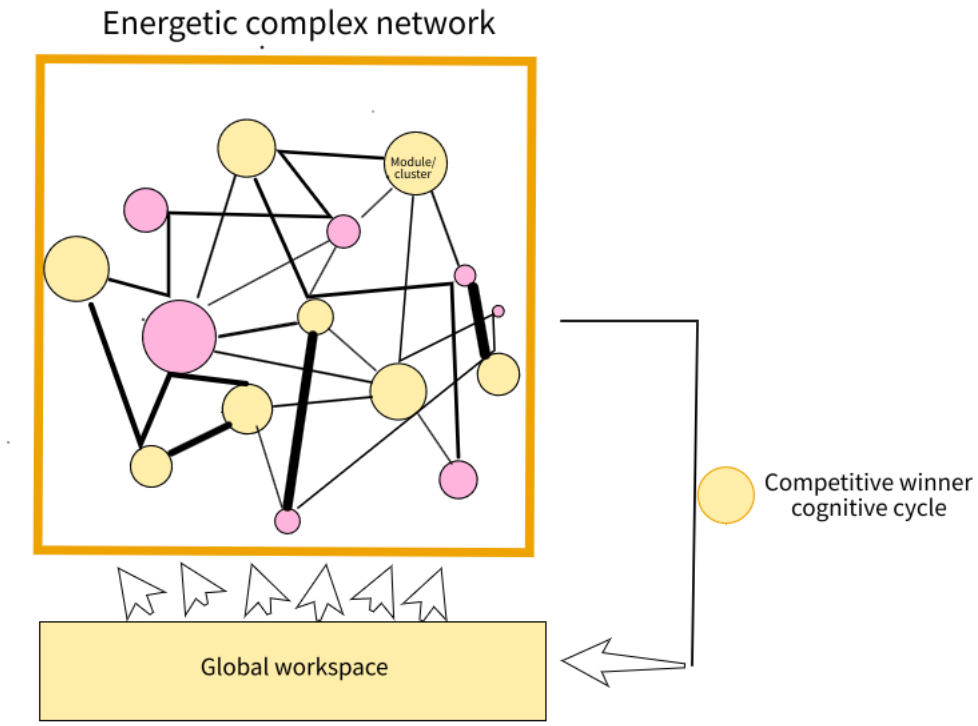

Figure 8: A rough sketch of a novel architecture design that incorporates the network structure characteristics of IKON FLUX and AKIRA with the global workspace design of LIDA.

The idea behind this architecture is similar to all the discussed architectures in that it consists of a series of cognitive cycles. Just like in LIDA for example, each cognitive cycle can be really short (on the order of milliseconds). The basic design consists of a complex (meaning scale-free, mostly small-world, and clustered) network. The network principles are largely based on AKIRA; the network share one pool of energy resources, and the individual modules or clusters compete for these resources. Large circles represent either large modules or a cluster of modules that work together for the same purpose. A few modules have a large number of links, and most modules only a few, making it close to scale-free. Every module can furthermore reach every other module in a maximum of three links, making it small-world. In the network, every module can take on any function, from very simple tasks such as a representation of an explicit memory, or it can be the machinery behind visual perception. The different colors indicate opponent processes; Most modules come in pairs, in that the yellow modules biases itself to one particular task in order to achieve a goal, while a corresponding pink module biases itself to the exact opposite such that the interaction between the modules results in a balancing of these constraints (this feature makes sure that opponent processing is incorporated). Each module has access to certain sensors, and every cognitive cycle each module updates its

internal state with the new sensory input and shares its relevancy to the rest of the network based on how much the sensory input matches the goal that module is trying to achieve. New modules can be developed automatically in a similar fashion as in IKON FLUX. Every cognitive cycle, the modules compete for energy, and the module or cluster with the highest energy wins the cognitive cycle and is sent to the global workspace. After that, the global workspace sends the information of the winning module back to the entire network, such that the network is notified that that particular module currently has the highest relevance. In response, the network can send more energy to that module such that it has more access to computational power and sensors. After a while, the module loses relevancy since the sensory or internal input is not relevant to the goal of the module anymore, and another module has a chance to win the competition and be sent to the global workspace, after which the cognitive cycle is repeated. Finally, while this design sketch seems abstract, it must be instantiated in an embodied physical system, e.g. certain modules with sensors are actually parts of the body of the system; for example, the system can have a module for sensing touch, and this module would not be present in the 'brain' of the system, but rather in peripheral parts of its body.

This design arguably allows for optimal relevance realization as it incorporates all the earlier discussed features. Again, it is meant as a rough sketch to indicate a design of an architecture where a complex network is combined with an implementation of the global workspace theory, and the actual design would require a lot more specification and nuance in its design. However, for the sake of argument, since such a design is most promising for relevance realization, it follows that it is a design that is also most promising for artificial general intelligence, based on previous argumentation.

Final discussion

In this thesis, I have discussed the challenge of creating artificial general intelligence and argued that the reason there is so much difficulty in building this is due to the unsolved epistemological frame problem. I have argued that one of the main aspects of general intelligence involves the ability to quickly adapt and learn in any new particular environment. In order to achieve this effectively, an AGI must in the basis be able to appropriately frame its perception, due to the infinite potential information such a new environment brings. In other words, the AGI must have a mechanism for determining the relevancy of perceptual input in any new environment, and the better the system is able to

do this, the more competent the system can perform in this new environment. I introduced the framework of relevance realization as discussed by Vervaeke (2012). While not providing a full answer of what such a framework would look like, he suggested several features that it should possess at minimum: self-organization, a bio-economic model of cognition, and balancing of constraints by opponent processing. I introduced two more, complex network characteristics and embodiment, and then compared several existing cognitive architectures for artificial general intelligence (CLARION, LIDA, AKIRA and IKON FLUX) on the basis of these five features; evaluating to what degree these architectures incorporate these features and then concluding which of the architectures does so best. From this analysis, AKIRA and IKON FLUX scored highest in total.

Finally, I attempted to construe a rough sketch of a novel cognitive architecture for artificial general intelligence, with the idea of taking the highest scores of each architecture on each feature and combining these into a new design. Naturally, this design looks mostly like the architectures AKIRA and IKON FLUX, but it also incorporated the model of allocating cognitive resources from LIDA. The design is only meant as an abstract description of such an architecture, so technical and elaborate details are lacking. Nevertheless, the design of the cognitive architecture such as I proposed seems like a solid basis for future design of architectures for artificial general intelligence, since it incorporates all features of relevance realization (and I have argued for a close connection between the ability to do relevance realization and general intelligence). Future research could focus more on the technical details of such an architecture (how exactly do all parts relate and influence each other, what modules can the system incorporate, etc.), and see if such a design may perhaps be instantiated in a physical system. On the theoretical side, future research could also focus more on developing the framework of relevance realization, as the theory is not yet complete and requires a fuller description of the precise mechanism of how relevance is determined and calculated.

References

Adams, S. S., Arel, I., Bach, J., Coop, R., Furlan, R., Goertzel, B., Hall, J. S., Samsonovich, A., Scheutz, M., Schlesinger, M., Shapiro, S. C., & Sowa, J. F. (2012). *Mapping the Landscape of Human-level Artificial General Intelligence*. *2009*, 25–42.

Baars, B. J. (2005). Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. *Progress in Brain Research*, *150*, 45–53.

Boschetti, F. (2011). Rationality, complexity and self-organization. *Emergence: Complexity and Organization*, *13*(1–2), 133–145.

Brown, F. M. (1987). *The Frame problem in artificial intelligence : proceedings of the 1987 workshop, April 12-15, 1987, Lawrence, Kansas*. Morgan Kaufmann Publishers.

Dennett, D. C. (1984). Cognitive wheels: The frame problem of AI. *Minds, machines and evolution*, 129-151.

Duch, W., Oentaryo, R., & Pasquier, M. (2008). Cognitive Architectures: Where do we go from here? *Contemporary Longterm Care*, *171*, 122–136.

Fodor, J. A. (1987). Modules, frames, fridgeons, sleeping dogs, and the music of the spheres. In *Modularity In Knowledge Representation And Natural-Language Understanding*. MIT Press.

Franklin, S., Ramamurthy, U., D'Mello, S. K., McCauley, L., Negatu, A., Rodrigo Silva, L., & Datla, V. (2007). LIDA: A computational model of global workspace theory and developmental learning. *AAAI Fall Symposium - Technical Report*, *FS-07-01*, 61–66.

Gibney, E. (2015). DeepMind algorithm beats people at classic video games. *Nature*, *518*(7540), 465–466.

Goertzel, B. (2014). Artificial General Intelligence: Concept, State of the Art, and Future Prospects. *Journal of Artificial General Intelligence*, *5*(1), 1–48.

Green, C. D. (1996). Fodor, functions, physics, and fantasyland: Is ai a mickey mouse discipline? *Journal of Experimental and Theoretical Artificial Intelligence*, *8*(1), 95–106.

Kurzweil, R. (2005). *The singularity is near: When humans transcend biology.* Penguin.

Laird, J. E., Wray, R. E., Marinier, R. P., & Langley, P. (2009). Claims and challenges in evaluating human-level intelligent systems. *Proceedings of the 2nd Conference on Artificial General Intelligence, AGI 2009*, *1995*, 91–96.

McCarthy, J., & Hayes, P. J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Readings in Artificial Intelligence* (pp. 431–450). Elsevier.

McCarthy, John. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, *28*(1), 89–116.

Munari, L. (2009). How the body shapes the way we think — a new view of intelligence. *Journal of Medicine and the Person*, *7*(2), 110–111.

Newell, A., Shaw, J. C., & Simon, H. A. (1959, June). Report on a general problem solving program. In *IFIP congress* (Vol. 256, p. 64).

Nilsson, N. J. (2005). Human-level artificial intelligence? Be serious! *AI Magazine*, *26*(4), 68–75.

Parnassum, A., & Klee, P. (1998). *The General Intelligence Factor*.

Pezzulo, G., & Calvi, G. (2007). Designing modular architectures in the framework AKIRA. *Multiagent and Grid Systems*, *3*(1), 65–86.

Schwind, C. B. (1978). Representing actions by state logic. *3rd Conf. of the Society for AI and the Simulation of Behaviour (AISB 78)*, 304–308.

Shanahan, M. (1997). *Solving the frame problem: a mathematical investigation of the common sense law of inertia*. MIT press.

Shanahan, M., & Baars, B. (2005). Applying global workspace theory to the frame problem. *Cognition*, *98*(2), 157–176.

Sun, R. (2007). The importance of cognitive architectures: An analysis based on CLARION. *Journal of Experimental and Theoretical Artificial Intelligence*, *19*(2), 159–

AAAI, The First Conference on Artificial General Intelligence (2008). Retrieved December 11, 2021, from http://agi-conference.org/

Thórisson, K., & Helgasson, H. (2012). Cognitive Architectures and Autonomy: A Comparative Review. *Journal of Artificial General Intelligence*, *3*(2), 1–30.

Thórisson, K. R., & Nivel, E. (2009). Achieving artificial general intelligence through peewee granularity. *Proceedings of the 2nd Conference on Artificial General Intelligence, AGI 2009*, 222–223.

Vervaeke, J., & Ferraro, L. (2013). Relevance Realization and the Neurodynamics and Neuroconnectivity of General Intelligence. In *SmartData* (pp. 57–68). Springer New York.

Vervaeke, J., Lillicrap, T. P., & Richards, B. A. (2012). Relevance realization and the emerging framework in cognitive science. *Journal of Logic and Computation*, *22*(1), 79–99.

Wang, P. (2006). Rigid flexibility: The logic of intelligence. In *Rigid Flexibility: The Logic*

of Intelligence (Vol. 34).

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *Nature*, *393*(6684), 440-442.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.