



AGI engineering

Subscribe



# ARCHITECTURE

## BUILDING BLOCKS FOR AGI



Mykola Rabchevskiy

Mar 19 2 4

According to the previous chapter, the AGI system should implement a closed control loop, which includes generating requests to sensors for information about the situation and making decisions about the necessary actions based on internal stimuli, accumulating the essential information, and increasing the rationality of decisions.

To develop the AGI system's architecture, we will analyze these processes in more detail, bearing in mind that only such a scheme can be considered actual architecture if each element has at least one implementation option.

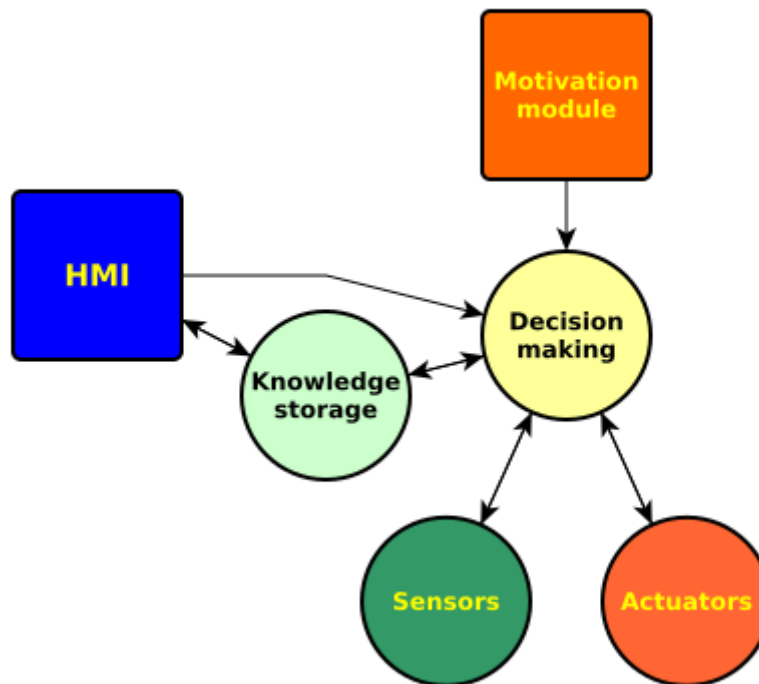
From the AGI system's requirements, it is clear that it should include a store of accumulated knowledge, communication channels with *effectors* (sensors and actuators), a component that analyzes the data, evaluates the situation, and decides what actions should do. Finally, analysis and decision-making are fundamentally dependent on the internal stimuli generated by the motivation module. Thus, the three essential components are the *knowledge store*, the *decision-making module*, and the *motivation module*. Of these three components, the motivation module is the most specialized component because it determines how much the system's behavior corresponds to the mission assigned to the system.

© 2021 Mykola Rabchevskiy. See [privacy](#), [terms](#) and [information collection notice](#)



Publish on Substack

AGI engineering is on [Substack](#) – the place for independent writing



However, this structuring level does not meet our requirement for at least one implementation option: the schema is too abstract and requires more detail.

The knowledge store should be convenient and efficient for several "varieties" of knowledge. There is knowledge of a declarative type that is relatively stable; they expand (replenish) but rarely change (and some never). AI system developers usually refer to them as *ontologies*. Examples: "an ant is an insect"; "banana is edible"; "shades of blue are called cool colors, and shades of yellow are called warm". The most complex elements of this type include *sequences* intended to operate as a whole (actions, signs, situations) and *rules* - as in the good old expert systems. In what follows, this type of knowledge will be called *semantic*. Two aspects determine the specificity of semantic knowledge:

- semantic knowledge operates with logical objects (as opposed to numerical data)
- their volume grows as the system acquires new knowledge.

In a stable environment, the growth in the volume of information slows down as the environment becomes more and more studied, and under certain conditions, the size can stabilize. The natural way to represent such knowledge is a *semantic graph*; a separate chapter will be devoted to details.

Another kind of knowledge is what happened and what order happened in the past (actions, events); that is, it is a kind of biography, a system diary, or experience. We will use the term "*chronicle*" to refer to such a module. These data are the basis for finding cause-and-effect relationships, including predicting the possible consequences of

specific actions in a given situation. There are three main aspects of this knowledge:

- it is a sequence of events in which the same element can occur many times
- sequence intended to represent the process, so sequence elements reflect changes in the situation over time
- chronicle elements are references to logical entities in the storage of semantic information - the sequence defines only the order of events in time

In AI, the concept of *episodic memory* is known, which looks similar to the diary described above. In reality, these things serve different purposes and are used differently, so we do not use the term "*episodic memory*."

Since the same event can present multiple times, a sequence can contain multiple references to the same logical entity. The stored sequence of events is continuously increasing even if the size of semantic knowledge has stabilized; therefore, due to the system's limited resources, the outdated or least essential data should be excluded (forgotten).

The search for cause-and-effect relationships is based on finding repeating fragments of a sequence of events, so the data structure must ensure such a search's effectiveness. One of the following chapters will be devoted to a data structure that effectively implements such a search and compactly represents a sequence, replacing repeated fragments with references to them.

Finally, the third and most traditional knowledge is **data** (quantitative and textual): phone numbers, sizes, coordinates, speeds, dates, and so on, which is similar to the content of commonly adopted relational databases. This data can be presented in the well-known "**object-attribute-value**" format. The **object** and **attribute** are logical entities present in the storage of semantic information (references to objects also contained in the chronicle). The specific of such data is that it can change while logical connections between objects are not changed; that is, changes in data do not entail changes in the semantic knowledge or events' sequence.

As a result, the knowledge store turns out to be composed of three components with a shared set of logical entities used for them: a **semantic graph**, a **chronicle sequence**, and **quantitative data**. All three components of the knowledge base are responsible for storing information, but not for analyzing it; the functions of these modules include a set of operations typical for databases: adding information, modifying, deleting, and

searching by specific criteria.

The next component requiring detail is the decision-making module.

A reasonable decision depends on the current state of both the environment and the controlled system. Therefore, data describing the current state is required and a permanently active function of updating it to maintain relevance. The corresponding module updates the current state's data by using data received from the sensors and actuators. This module brings together data from many sensors and removes obsolete data from the description of the situation; this functionality includes what is called *data fusion*.

Situation description is a model of reality, and this model in the case of the natural environment always simplifies reality. The effectiveness (quality) of solutions naturally depends on how those aspects and details of the environment are selected to represent the current state, which details of reality noticed when collecting information, and which details are omitted (which details attracted attention and which did not). This aspect is controlled by the *attention* subsystem, which analyzes the situation and determines which of them require more detailed information, more frequent updates, and more detailed analysis. The attention module, on the one hand, uses data about the current situation, and on the other hand, supplies the state updating module with information about what to pay attention to.

A feature of the attention subsystem, like the motivation module, is that its specific design depends significantly on the mission (purpose) of the system, including dependence on the available set of sensors and actuators. A critical piece of information for decision-making is state-changing data; state-changing events detected by the same module and added to the system diary.

Deciding actions is the choice of such an action from among the available ones, for which the forecast of the consequences is most favorable. Since we are talking about finding the most favorable option, this becomes an *optimization problem*, and it requires a *criterion* for comparing options. Furthermore, since the *future* consequences are compared, it is naturally necessary to make *forecasts* based on the accumulated knowledge.

The evaluation of specific actions' consequences dependent on the current situation and current intentions, so optimization *criteria must be formed dynamically* by the motivation module. The *motivation module* defines the purposeful activity of the system. Naturally,

the design of this module is entirely determined by the mission of the system.

The system's level of intelligence depends on the forecast's adequacy and how long-term consequences it represents. The most primitive variant with an evaluation of the situation immediately after the first action is insufficient for the practical system. An analysis requires several steps ahead, that is, for the predicted possible consequences of the first step, the consequences of each choice of the second step are analyzed, and so on. Such a process makes building the "forecast tree" the most resource-intensive part of the system.

An unobvious aspect of forecasting is the need to analyze *all possible*, and not only the *most probable*, consequences of specific actions (for example, if a predator is guided only by the fact that the most likely attempt at hunting leads to failure with a guaranteed waste of energy, the result will not be satisfactory).

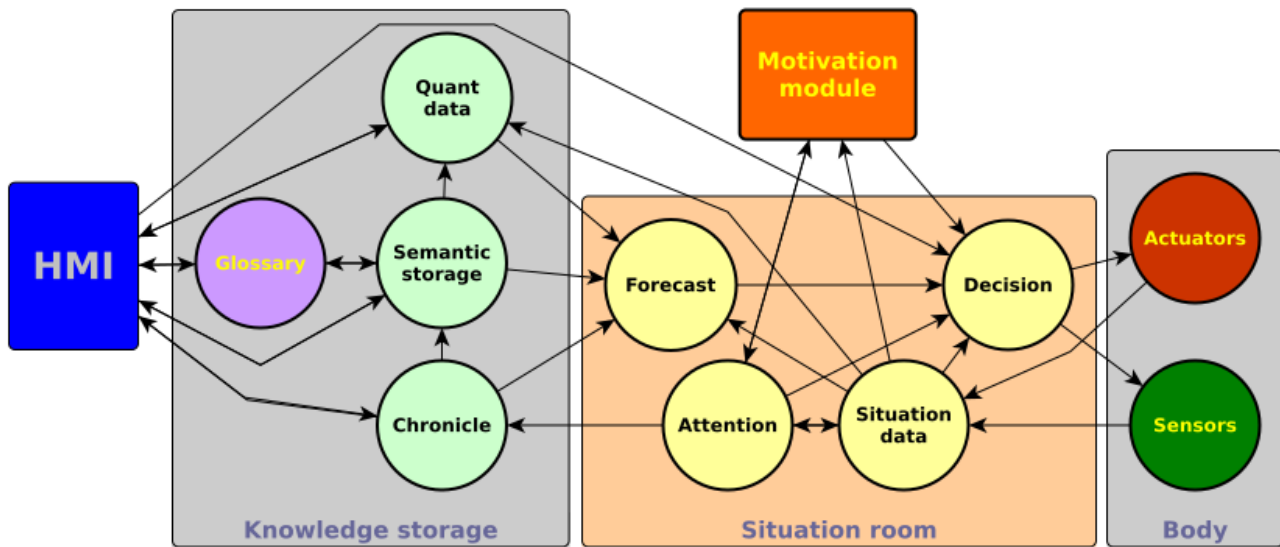
Finally, in *real-time* systems, the forecast needs to be available before it becomes irrelevant; in rapidly changing (stressful) situations, one should confine oneself to a less extended forecast in time but faster in making a forecast.

At first glance, it may seem, choosing the best possible course of action is not a difficult task if there is plenty to choose from (there are forecasts). In reality, however, optimization problems have their specifics, which determine the boundaries of what is possible for AGI and makes it fundamentally impossible to build such an AGI version that will be better than others in all aspects and all cases ("perfect AGI").

As a result, the initial decision-making module turns into a complex set of autonomous but interacting sub-modules for updating the situation description, attention, forecasting, forming the current criterion for choosing actions, and choosing an action. The software implementation of the module can use computational parallelism.

Overall, this turns out to be a quite recognizable structure, known in politics and military affairs as the "*situation room*," and we will use this term. The principles of intelligent decision-making are the same for society, organism, and artificial systems. Note also that the situation room's functionality is mainly in line with what is called a blackboard architecture.

The detailed structure of the AGI system takes on the following form (arrows directed from the data source to the consumer):



## SUMMATION

- The motivation module provides the system with criteria for evaluating forecasts
- The attention module prioritizes the collection of information
- The information collection module forms the current description of the situation and extends the experience
- The forecasting module builds the forecasting tree using collected experience
- The decision-making module uses forecast and criteria provided by the motivation module
- Accumulated knowledge is stored in a semantic graph, a chronicle, and a quantitative data
- Modules interact, working in parallel and mostly asynchronously



Subscribe

← Previous

Next →



Write a comment...



**supralibrix@gmail.com** Apr 17 Liked by Mykola Rabchevskiy

I'd like to connect, I'm working on a parallel project.

<https://physix.world/>

[supralibrix@gmail.com](mailto:supralibrix@gmail.com)

♥ 1 Reply



**James Kitching** Mar 25

Hi

I have been writing up an explanation of a new computational theory of mind. I have also been trying to sort out the resources that I could use to implement this theoretical system.

In the article, I am writing I would like to quote your articles and reproduce two of your diagrams as part of my explanation. I want to explain how I believe memories are part of our encoded consciousness system and are in fact a type of editable, reuseable and extendable simulation component. Interesting I believe we could potentially describe memories using XML (Yes!) and some open source software called XDEVS.

I am on the Cognitive architecture FB group you are on. So could perhaps PM you there or FB friend you if you would like to discuss more.

Here is my article intro so far:

I believe our minds contain something that is very much like computer simulation software. Theoretically at least simulation software could be used to mimic life and, therefore, also potentially a type of human intelligence.

I believe we should recognise simulation technology as the next big thing in developing more human-like AI. I will explain how I believe we could create computers that appear to convincingly and literally understand and use any and all human languages. We could in fact go further and potentially implement an artificial ability to understand and use human languages in an entirely human like. I believe it will be a long while before we start asking our computers to be completely human-like. I do not currently see the need for an AI poet or playwright. People, particularly the elderly, do have a need for AI social companions. Such a

[Expand full comment](#)

♥ 1 Reply

2 replies by Mykola Rabchevskiy and others

2 more comments...

**Ready for more?**

**Subscribe**