Chapter 2

A New Approach Toward AI

In this book, a new approach toward AI will be presented. This chapter first informally introduces the basic ideas.

2.1 To define AI

As described in the previous chapter, different schools of AI research are based on different working definitions of "intelligence." Therefore, I start by clarifying my definition, which encapsulates the goal of the research.

2.1.1 Information System

The working definition of intelligence, no matter what it is, should distinguish one type of system from another type of system. More concretely, here I want to distinguish one type of *information system* from another type.

The concept "information," like "intelligence," is also a concept used differently by different people. In this book, this concept is used to set the "platform" or "background" for the discussion on intelligence, so I only state my working definition for it, that is, what I mean by it, without a detailed discussion about why it is better than alternative definitions.

An information system, or information-processing system, is a system whose internal activities and interactions with its environment can be described *abstractly* — that is, without specifying the concrete entity and process (hardware) that carries out the activities and interactions.

Usually, such a system has certain *tasks* (also called *goals*) to carry out, given by the environment or generated by the system itself. To do this, the system takes various *actions* (also called *operations*), guided by its *knowledge* (also called *beliefs*) about how the actions and the tasks are related. Any internal activity costs the system some *resources*, especially, processing time and memory space.

The *environment* of such a system may be the physical world (if the system has sensorimotor capability), or other information-processing systems (human or computer). In either case, the interactions are specified by the *experiences* (or *stimuli*) and the *behaviors* (or *responses*) of the system, which can be described as streams of input and output information, respectively. For the system, recognizable patterns of input and producible patterns of output constitute its *interface language*.

According to this definition, all human beings and computer systems, as well as many animals and automatic control systems, can be described as information systems.

To call a system an "information system" means to describe the system at an abstract level, and many low-level details will be omitted from the description. A computer server is an information system to a remote user, who does not care about its size, color, and weight. However, to a worker who is moving the server from one room to another, it is no longer suitable to treat it as an information system. If you throw a ball to a friend as a prearranged signal for something, your action is information transformation, and where the ball goes does not matter. However, it would be silly to call the ball-throwing "information transformation" in a baseball game — it is not wrong, but contributes little to our understanding of the game.

Even if a entity or process cannot be treated as an information system, it often can be "modeled" or "simulated" in an information system. It means that the system can be described at an abstract level, and another system can be built that has the same high-level description, which contains essential features of the system to be modeled, though these two systems are completely different at a lower level. For example, a hurricane can be modeled in a computer, so that its movement can be predicted. However, "being a kind of movement of air" is a defining property of hurricane, which is not in the computer simulation. In this sense, a model is not the entity or process to be modeled, and they are only similar at a high level of description.

However, if the entity or process being modeled is such a system that all of its major properties are shown at the "information-processing" level, then we no longer call the above procedure "modeling" or "simulating," but call it "reproducing," "replicating," or "implementing." For example, arithmetic calculation are manipulations of symbolic entities and relations. Whether it is done by stones, abacus, or pen and paper usually has little importance to the result. When such a process is carried out by a computer, we do not say that the arithmetic calculation is "simulated" in the computer — unlike a hurricane, the calculation in a computer is genuine.

Sensitive readers would have realized why I start the discussion about AI with information system. Actually, to ask "To what extent can intelligence be produced by a computer?" is the same as to ask "To what extent can intelligence be described as information processing?" and the latter question will be answered by this book, using the information-processing terminology introduced above.

2.1.2 A working definition of intelligence

Following the preparation of the previous subsection, I propose here a working definition of intelligence:

> *Intelligence is the capability of an information system to adapt to its environment while operating with insufficient knowledge and resources.*

Here the meanings of "information system," "environment," "knowledge," and "resources" have been clarified previously. However, the two major components of the definition: "adaptation" and "insufficient knowledge and resources" remain to be explained.

In terms of behavior change, we can distinguish three types of systems:

Instinctive system: The behaviors of the system remains the same, and do not change over time.

Erratic system: The behaviors of the system change, but not as a function of its experience.

Adaptive system: The behaviors of the system change according to its experience. It attempts to improve its performance in carrying out the tasks, under the assumption that its future experience will be similar to its past experience.

On the other hand, we can roughly distinguish three kinds of environment, in terms of its interaction with a system in it:

Constant environment: The environment is deterministic, and the results of system behaviors never change. In this kind of environment, the best way to carry out a task is by an instinctive system specially built for the task. An adaptive system may be able to learn the behavior, but it is less efficient.

Random environment: The environment is completely unpredictable. In this situation, all systems are equally bad in the long run.

Stable environment: The environment may change, though not randomly, and the results of system behaviors are usually predictable, but with exceptions from time to time. Adaptive systems work better in this kind of environment, so long as its adaptation-rate is not too slow compared to the speed of change in the environment.

In this sense, an adaptive system is not necessarily better than a non-adaptive system. Actually, if a problem can be handled without adaptation (i.e., can be solved "mechanically"), it is better to do it that way. Adaptation is needed only when there is no predetermined solution available.

Just being "adaptive" is not enough for being "intelligent." A complex system can be called "adaptive" only because a few parameters in it can be tuned by itself according to its experience. Intelligence requires more than that, and this is why I have another component in the working definition of intelligence.

Insufficient knowledge and resources means that the system works under the following restrictions:

Finite: The information-processing capability of the system's hardware is fixed.

Real-time: All tasks have time constraints attached to them.

Open: No constraint is put on the content of the experience that the system may have, as long as they are representable in the interface language.

Not all information-processing systems take the insufficiency of knowledge and resources into full consideration. Non-adaptive systems, for instance, simply ignore new knowledge in their interactions with their environment. As for artificial adaptive systems, most of them are not finite, real-time, and open, in the following senses:

1. Though all concrete systems are finite, many theoretical models (for example, Turing Machine) neglect the fact that the requirements for processor time and/or memory space may go beyond the supply capability of the system [Hopcroft and Ullman, 1979].

2. Most current AI systems do not consider time constraint at run time. Most real-time systems can handle time constraint only if they are essentially deadlines [Strosnider and Paul, 1994].

3. In most systems, various explicit or implicit constraints are imposed on what a system can experience. For example, only questions that can be answered by retrieval and deduction from current knowledge are acceptable, new knowledge cannot conflict with previous knowledge, and so on.

Many computer systems are designed under the assumption that their knowledge and resources, though *limited* or *bounded*, are nevertheless *sufficient* to fulfill the tasks that they will be called upon to handle. When facing a situation where this assumption fails, such a system simply panics, and asks for external intervention, usually from a human manager of the system.

For a system to work under the *Assumption of Insufficient Knowledge and Resources*, hereforth known as *AIKR*, it should have mechanisms to handle the following situations:

> A processor is required when all processors are occupied;
>
> A piece of memory is required when all memory is already full;
>
> A task comes up when the system is busy with something else;
>
> A task comes up with a time constraint, so exhaustive processing is not affordable;
>
> New knowledge conflicts with previous knowledge;
>
> A question is presented for which no sure answer can be deduced from available knowledge;
>
> etc.

For traditional computing systems, these situations usually either require human intervention, or simply cause the system to reject the task or knowledge involved. However, for a system designed under AIKR, these are *normal situations*, and should be managed smoothly by the system itself.

The two main components in the working definition, *adaptation* and *insufficient knowledge and resources*, are related to each other. An adaptive system must have some insufficiency in its knowledge and resources, for otherwise it would never need to change at all. On the other hand, without adaptation, a system may have insufficient knowledge and resources, but make no attempt to improve its capability. Such a system acts, for all intents and purposes, as if its knowledge and resources were sufficient.

According to the above definition, intelligence is indeed a "highly developed form of mental adaptation" [Piaget, 1960]. This assertion is consistent with the usages of the two words in natural language: we are willing to call many animals, computer systems, and automatic control systems "adaptive," but not "intelligent," because the latter has a higher standard than the former.

When defining intelligence, many authors ignore the complementary question: what is unintelligent? If everything is intelligent, then this concept is empty. If every computer system is intelligent, it is better to stay within the theory of computation. Even if we agree that intelligence, like almost all properties, is a matter of degree, we still need criteria to indicate what makes a system more intelligent than another. An unintelligent system is not necessarily incapable or gives only wrong results. Actually, most ordinary computer systems and many animals can do something that human beings cannot. However, these abilities do not earn the title "intelligent" for them. What is missing in these capable-but-unintelligent systems?

According to the working definition of intelligence introduced previously, an *unintelligent* system is one that does not adapt to its environment. Especially, in artificial systems, an unintelligent system is one that is designed under the assumption that it only works on problems for which the system has sufficient knowledge and resources.

An intelligent system is not always "better" than an unintelligent system for practical purposes. Actually, it is the contrary: when a problem can be solved by both of them, the unintelligent system is usually better, because it guarantees a correct solution. As Hofstadter said, for tasks like adding two numbers, a "reliable but mindless" system is better than an "intelligent but fallible" system [Hofstadter, 1979].

2.1.3 Comparison with other definitions

According to the classification of AI schools in the previous chapter, the above working definition of intelligence belongs to the attempts that treat intelligence as being defined by some underlying *principles*.

Furthermore, it is an attempt to attack the whole AI problem, rather than part of it. Therefore we can also classify it as an "AGI," using the term introduced in the previous chapter. Personally, however, I would rather stay with the term "AI," because to me, "intelligence" is a domain-independent capability, so it is redundant to say "general intelligence." However, since many other people do not use the term in this way, I do not mind to use AGI or similar terms to stress the "general purpose" nature of the related systems.

How is my working definition of intelligence different from the others discussed in the previous chapter?

- In the following chapters, I will show that a system developed on such a foundation has many cognitive *functions*, but they are better thought of as emergent phenomena than as well-defined tools used by the system.

- By learning from its experience, the system potentially can acquire the *capability* to solve hard problems — actually, "hard" problems are exactly those for which the system has insufficient knowledge and resources. However, this capability is not built into the system, and thus, without proper training, no capability is guaranteed, and acquired capability can even be lost.

- Because the human mind also follows the above principles, such a system can be expected to behave similarly to human beings, but the similarity would exist at a more abstract level than that of concrete *behavior*. Due to the fundamental difference between human experience and the experience of an AI system, the system will not accurately reproduce masses of psychological data or guarantee to pass a Turing Test.

- Although the internal *structure* of the system has some properties in common with a description of the human mind at a certain level, it is not an attempt to simulate a biological neural network or the brain as a whole.

To be sure, what has been proposed in my definition is not entirely new to the AI community. Few would deny that adaptation, or learning, is important for intelligence (though many people are still working on "AI" projects that have no learning capability — to them, learning is something that can be added into the picture at a later time). Moreover, "insufficient knowledge and resources" is the focus of many subfields of AI, such as heuristic search, reasoning under uncertainty, real-time planning, and machine learning. Besides the various types of "rationality" listed in the previous chapter, similar attitudes toward intelligence can be found in the following quotations:

> Medin and Ross: Much of intelligent behavior can be understood in terms of strategies for coping with too little information and too many possibilities. [Medin and Ross, 1992]
>
> Michalski: By "intelligence," we mean a set of capabilities that let a system with limited resources (energy, time, and memory) operate under limited input information (incomplete, uncertain, inconsistent, or incorrect). [Hearst and Hirsh, 2000]

Given all that has already been said, what is *new* in my approach? As far as the working definition is concerned, this approach is new in the following aspects:

1. An explicit and unambiguous definition of intelligence as "adaptation under insufficient knowledge and resources."
2. A further clarification of the phrase "with insufficient knowledge and resources" as meaning *finite*, *real-time*, and *open*.
3. The design of all formal and computational aspects of an AGI project keeping the two previous definitions foremost in mind.

Though many AI systems are designed according to some kind of "bounded rationality", and assume some restrictions in knowledge and resources, few of them can be said to be based on AIKR. We will see detailed analysis on this topic in Part III, where my approach is compared with other approaches.

2.2 Intelligent reasoning systems

As stated in the previous chapter, a typical AI project consists of an informal theory, a formal model, and a computer implementation. The previous working definition belongs to the theory level. Now let us see how to use it to choose a proper formal language to build a model.

2.2.1 Different traditions in formalization

Formalization means, in the current context, the process of describing the status and activities of a system in a formal (artificial, symbolized) language.

In the current AI research, there are different traditions of formalization. The major ones are the following:

Dynamic system: In this tradition, the state of a system at a given time is specified by the values of a fixed set of attributes. Intuitively, it corresponds to a point in a multi-dimensional space, where each dimension corresponds to an attribute, and the coordinate of the point on that dimension corresponds to the system's value on the attribute. In this representation, the change of state (caused either by the system itself or by an outside factor) is a trajectory line in the space, indicating how one state follows another. The regularity of the system is represented by equations that describe the possible trajectory lines. AI inherits this type of formalization from dynamics, system theory, and cybernetics, and uses it in pattern recognition systems, connectionist models, and so on.

Reasoning system: In this tradition, the state of a system at a given time is specified by a set of sentences in a formal language. Each sentence represents a belief of the system, or a piece of knowledge about the environment. The change of state can either be caused by the system's inference activity (i.e., using a fixed set of rules to derive new sentences from existing ones), or by its communication activity (i.e., the input and output of the sentences). The regularity of the system is represented by the set of inference rules. AI inherits this type of formalization from mathematical logic, and uses it in various types of "knowledge-based systems."

Computing system: In this tradition, the state of a system at a given time is specified by a data structure, in which individual data items are organized together. The change of state is caused by the execution of a program, which modifies the data structure, and produces certain side effects. The regularity of the system is specified by algorithms, which are abstract representation of the programs. AI inherits this type of formalization from computer science, and uses it in searching, learning, and many other techniques.

All the three traditions are very powerful, and in principle they can emulate one another, in the sense that a virtual machine specified in one tradition can be implemented by another virtual machine specified in a different tradition. In this sense, they have equivalent expressive and processing capacity. It is also possible to build a hybrid system, in which multiple formalization traditions are integrated.

However, for a given problem, one tradition may be more natural, convenient, and efficient than the others. Therefore they are not always equivalent under practical considerations for a given problem.

I choose to formalize my working definition of intelligence in the framework of a reasoning system, mainly based on the following considerations:

- It is a general-purpose system. Working in such a framework keeps us from being bothered by domain-specific properties, and also prevents us from cheating by using domain-specific tricks.

- It uses a rich formal language, especially compared to the "language" used in multi-dimensional space, where a huge number of dimensions are needed to represent a moderately complicated situation.

- Since the activities of a reasoning system consists of inference steps, it allows a natural combination of the *rigidness* (i.e., justifiability) of each individual step and the *flexibility* (i.e., context-dependency) of the inference processes, especially compared to the algorithm-governed processes, where the linkage from one step to the next is fixed, and the process usually cannot stop in the middle.

- Compared with cognitive activities like low-level perception and motor control, reasoning is at a more abstract level, and is one of the cognitive skills that collectively make human beings so qualitatively different from other animals.

- As will be displayed by this book, the notion of "reasoning" can be extended to cover many cognitive functions, including learning, searching, categorizing, planning, decision making, and so on.

- Most research on reasoning systems is carried out within a school based on assumptions directly opposed to AIKR. By "fighting in the backyard of the rival," we can see more clearly what kinds of effects the new ideas have.

In summary, I believe that an *intelligent reasoning system* provides a suitable framework for the study of intelligence, though being a reasoning system is neither necessary nor sufficient for being intelligent. Here I will not justify the above claims, but leave that task to the end of the book, after the whole formal model is described and discussed.

2.2.2 Reasoning systems and logics

An automatic (computerized) reasoning system is an information-processing system that consists of the following major (conceptual) components:

1. *a language*, defined by a formal grammar, for the (external) communication between the system and its environment, and for the (internal) knowledge representation within the system;[1]

2. *a semantics* of the language that provides the principles to determine the meanings of the words and the truth values of the sentences in the language;

3. *a set of inference rules* that is defined formally, and can be used to match questions with knowledge, to infer conclusions from premises, to derive sub-tasks from tasks, and so on;

4. *a memory* that systematically stores tasks, beliefs, and so on, as well as provides a work place for inferences;

5. *a control mechanism* that is responsible for resource management, including to choose premises and inference rules in each inference step, and to allocate memory space.

[1]It is possible to have the two functions be accomplished by two different languages, with a translation mechanism in between.

The first three components are usually referred to as a *logic*, or the *logical part* of the reasoning system, and the last two as the *control part* of the system.

Before showing how an intelligent reasoning system is designed, let us first see its opposite — that is, a reasoning system designed under the assumption that its knowledge and resources are *sufficient* to answer the questions asked by its environment (so no adaptation is needed). By definition, such a system has the following properties:

1. No new knowledge is necessary. All the system needs to know to answer the questions is already there at the very beginning, expressed by a set of *axioms*.

2. The axioms are *true*, and will remain true, in the sense that they correspond to the actual situation of the environment.

3. The system answers questions by applying a set of formal rules to the axioms. The rules are sound and complete (with respect to the valid questions), therefore they guarantee correct answers for all questions.

4. The memory of the system is so big that all axioms and intermediate results can always be stored within it, and so effective that any content can be retrieved faithfully whenever needed.

5. There is an algorithm that can carry out any required inference in finite time, and it runs so fast that it can satisfy all time requirements attached to the questions.

This is the type of system dreamed of by Leibniz, Boole, Hilbert, and many others. It is usually referred to as a "decidable axiomatic system." The attempt to build such systems has dominated the study of logic for a century, and has strongly influenced the research of AI. Many researchers believe that such a system can serve as a model of human thinking.

However, if intelligence is defined as "to adapt under insufficient knowledge and resources," what we want is the *contrary*, in some sense, to an axiomatic system, though it is still *formalized* or *symbolized* in a technical sense. That is why *Non-Axiomatic Reasoning System*, NARS for short, is chosen as the name for the intelligent reasoning system to be introduced in this book.

2.3 Major design issues of NARS

Before going into detailed formal descriptions of NARS in the chapters of Part II, this section provides an informal introduction to the major design issues of the system. For each topic, the problem is summarized first, then the solution provided in NARS is briefly described. Full discussions and comparisons with other approaches will be given in Part III of the book, because the NARS solutions to many problems are provided by multiple components of the system.

2.3.1 Validity and rationality

A central issue of NARS is: when a system has to work with insufficient knowledge and resources, what is the criteria of *validity* or *rationality*? This problem is obviously related to Hume's problem of induction [Hume, 1748] — if the future is different from the past, how can we predict the former by the latter?

This issue needs to be addressed, because the aim of NARS is to provide a normative model for intelligence in general, not a descriptive model of human intelligence. It means that what the system does should be "the right thing to do," that is, can be *justified* against certain simple and intuitively attractive principles of validity or rationality.

In traditional logic, a "valid" or "sound" inference rule is one that never derives a *false* conclusion (that is, it will be contradicted by the future experience of the system) from *true* premises [Copi, 1982]. However, such a standard cannot be applied to a system that has to work under AIKR, since by definition, such a system has no way to guarantee the infallibility of its conclusions. On the other hand, it does not mean that every conclusion is equally valid.

As discussed previously, since intelligence is a special kind of adaptation, and in an adaptive system the behavior is determined by the assumption that the future experience will be similar to the past experience, in NARS a "valid conclusion" is one that is most consistent with the evidence in the past experience, and a "valid inference rule" is one whose conclusions are supported by the evidence provided by the premises used to derive them.

Furthermore, restricted by insufficient resources, NARS cannot exhaustively check every possible conclusion to find the best conclusion for every given task. Instead, it has to settle down with the best it can find with available resources.

In this sense, NARS can also be called an "adaptive reasoning system," whose central principle of rationality is "to predict the (unknown) future according to the (experienced) past, and to satisfy the (potentially infinite) resource request with the (actually finite) resources supply."

The following components are designed according to this principle.

2.3.2 Semantics

As was stated earlier, semantics studies how the items in a language are related to the environment in which the language is used.

Model-theoretic semantics is the dominant theory in the semantics of formal languages. For a language **L**, a model **M** consists of the relevant part of some domain described in another language **ML**, and an interpretation **I** that maps the items in **L** onto the objects in the domain (labeled by words in **ML**). **ML** is referred to as a "meta-language," which can be either a natural language, like English, or another formal language.

Given the above components, the *meaning* of a term in **L** is defined as its image in **M** under **I**, and whether a sentence in **L** is *true* is determined by whether it is mapped by **I** onto a "state of affairs" that holds in **M**. For a reasoning system, valid inference rules are those that always derive true conclusions from true premises.

With insufficient knowledge and resources, what relates the language **L**, used by a system **R**, to the environment is not a *model*, but the system's *experience*. For a reasoning system like NARS, the experience of the system is a stream of sentences in **L**, provided by a human user or another computer.

In such a situation, the basic semantic notions of "meaning" and "truth" still make sense. The system may treat terms and sentences in **L**, not solely according to their syntax (shape), but in addition taking into account their relations to the environment, according to the system's experience. Therefore, What we need is an *experience-grounded semantics.*

Under AIKR, NARS should not (and cannot) use "true" and "false" as the only truth values of sentences. To handle conflicts in experience properly, we need to determine what counts as positive evidence in support of a sentence, and what counts as negative evidence against it, and in addition we need some way to measure the *amount* of evidence in terms of some fixed unit. In this way, a truth value will simply be a numerical summary of available evidence.

Similarly, the meaning of a term (or word) is defined by the role it plays in the experience of the system, that is, by its relations with other terms, according to the experience of the system.

As was mentioned above, "experience" in NARS is represented in **L**, too. Therefore, in **L** the truth value of a sentence, or the meaning of a word, is defined by a set of sentences, also in **L**, with their own truth values and meanings — which seems to have led us into a circular definition or an infinite regress.

The way out of this seeming circularity in NARS is "bootstrapping." In the following, I will first define a very simple subset of the language, with its semantics. Then, I will use it to define the semantics of the whole language.

As a result, the truth value of statements in NARS uniformly represents several types of uncertainty, such as randomness, fuzziness, and ignorance. The semantics specifies how to understand sentences in the language, and provides justifications for the inference rules.

2.3.3 Grammar and inference rules

When presenting NARS, I take a path that is opposite to the usually accepted one. Instead of first defining a language formally, then attaching a semantics to it, I analyze the desired semantics first (guided by my working definition of intelligence), then analyze the language that can support such a semantics. The advantage of such an approach is argued in [Ellis, 1993].

From the previous discussion, we can see that what NARS needs is a language in which the meaning of a term is represented by its relation with other terms, and the truth value of a sentence is determined by available evidence. For these purposes, the concept of (positive or negative) evidence should be naturally introduced into the language. Unfortunately, the most popular formal language used in First-Order Predicate Logic does not satisfy the requirement, as revealed by the "Confirmation Paradox" [Hempel, 1943].[2] A traditional rival to predicate logic is known as *term logic*. Such logics, exemplified by Aristotle's Syllogistic, have the following features: [Bocheński, 1970, Englebretsen, 1981]

1. A typical sentence is *categorical*, which consists of a *subject term* and a *predicate term*, related by a *copula* intuitively interpreted as "to be."

2. A typical inference rule is *syllogistic*, which takes two sentences that share a common term as premises, and from them derives a conclusion formed by the other two terms.

Traditional term logic has been criticized for its poor expressive power. In NARS, this problem is solved by introducing various types of *compound terms* into the language, to represent *sets*, *intersections*, *differences*, *products*, *images*, *statements*, and so on.

The inference rules in this extended term logic carry out inheritance-based inference. Basically, each of them indicates how to use one item as another one, according to the experience of the system. Different rules correspond to different combinations of premises, and use different truth-value functions to calculate the truth values of the conclusions from those of the premises, justified according to the semantics of the system.

The inference rules in NARS uniformly carry out *choice, revision, deduction, abduction, induction, exemplification, comparison, analogy, compound term composition and decomposition*, and so on.

2.3.4 Inference control

Under AIKR, NARS cannot guarantee to process every task perfectly — with insufficient knowledge, the best way to carry out a task is

[2]This issue will be discussed in detail in Section 9.2.2.

unknown; with insufficient resources, the system cannot exhaustively try all possibilities.

Since NARS still needs to do its best in this situation, its solution is to let the items and activities in the system compete for the limited resources. Again, the validity of the resource allocation policy is justified according to the past experience of the system (rather than its future experience), and the aim is to satisfy the goals of the system as much as possible.

In the system, different data items (task, belief, or concept) have different *priority values* attached, according to which the system's resources are distributed. These values are determined according to the past experience of the system, and are adjusted according to the change of situation.

A special data structure is developed to implement a probabilistic priority queue with a limited storage. Using it, each access to an item takes roughly a constant time, and the accessibility of an item depends on its priority value. When no space is left, items with low priority will be removed.

The memory of the system contains a collection of concepts, each of which is identified by a term. Within the concept, all the tasks and beliefs (i.e., pieces of knowledge) that have the term as subject or predicate are collected together.

The running of NARS consists of individual inference steps. In each step, a concept is selected probabilistically (according to its priority), then within the concept a task and a belief are selected (also probabilistically), and the applicable inference rules take the task and the belief as premises to derive new tasks and beliefs, which are added into the memory.

The system runs continuously, and interacts with its environment all the time, without stopping at the beginning and ending of each task. The processing of a task is interwoven with the processing of other existing tasks, so as to give the system a dynamic and context-sensitive character.