Motivation Management in AGI Systems

Pei Wang

Department of Computer and Information Sciences
Temple University, Philadelphia, USA
pei.wang@temple.edu

Abstract. AGI systems should be able to manage its motivations or goals that are persistent, spontaneous, mutually restricting, and changing over time. A mechanism for handles this kind of goals is introduced and discussed.

1 Properties of goals in AGI systems

In a broad sense, all AI systems are "goal-oriented", in that every activity in it serves certain purpose. Researchers have been using notions like "motivation", "drive", "need", "goal", "task", and "intention" to indicate this *teleological* aspect of the system. In this paper, they are all called "goals", since the differences among these notions are not significant for this discussion. No matter what we call it, a process in such a system points to a certain destination, and it is against this destination that the system's progress and success are evaluated. In this broad sense, we do not require every goal to be explicitly represented or consciously known to the system.

In the context of AGI, some related topics have been discussed from different perspectives [1–4], though there are still many issues to be resolved. In this paper, we do not focus on the *content* of goals in AGI systems, like [1, 3], but on the general *properties* of goals, as well as on how they should be managed in the system. In AI, existing works are summarized in [5–7], though the situation is more or less different in the context of AGI, with the stressing on the versatility and unity of the system.

In the following we will discuss several questions:

- Can the system achieves its goals one after another? If not, when to switch the effort from one goal to another?
- Can the co-existing goals be assumed to be compatible with each other? If not, how to handle their conflicts?
- Can the goals change over time? If yes, why and how?
- Can the system produce its own goals? If yes, will it be out of control?

Before discussing these questions in the AGI context, let us consider the classical case of a *computation process* in a Turing Machine [8]. In this situation, the unique "goal" of the process is specified by the *final states* of the machine, which are predetermined, constant, and reachable. A traditional computer system typically has multiple running programs at any given moment, and each of which

corresponds to such a goal-guided process. Usually these processes are mutually *compatible*, in the sense that one does not prevent another to be reached. From time to time, there are processes start, while some others stop, so the "current goals" change, though remains a subset of all the possible goals of the system, which correspond to the programs in the system. All these goals are given be the designers and users of the system, so no goal really comes from the system *itself*. Since this situation is very simple, it is unnecessary to be described using fancy words like "goal" or "motivation".

However, the situation is not so simple for AGI systems. In the following let us discuss the four questions raised previously one by one.

Transient vs. persistent

The goals in traditional computer systems are *transient*, in the sense that each of them only exists for a relatively short time, from its creation to its satisfaction, corresponding to the beginning and ending of a computational process. Even when a program is repeatedly executed, the corresponding goal is usually not explicitly related to its previous occurrence.

Many AI systems also specify their goals in this way, that is, as "states satisfying particular conditions" [9], where the process stop and the system is reset to its initial state with respect to this process. The most typical examples are the systems doing state-space search, such as GPS [10].

On the contrary, in AGI systems, many (though not all) goals will be *persistent*, in the sense that once created, such a goal may last in the lifetime of the system. Examples of persistent goals can be found everywhere in the human mind, and many of them are also clearly desirable or inevitable in AGI systems, such as "be self-protective" and "to acquire resources" [1, 11].

This type of goals cannot be treated as final states where process stops. For one reason, such a "state" may never be actually reached, but serves merely as the direction for the system to move. Furthermore, even if it is achieved at a given moment, the system should not consider it done and does not think about it anymore, but has to prevent the achievement from being destroyed by future events.

For the above reasons, the system cannot treat a persistent goal as the ending point of a process, but a destination to be approached or a status to be preserved, and to decide when to stop the process by some other criteria, such as the quality of the obtained result (such as a "satisfying threshold") or the cost of the processing (such as an "expense budget").

Now we see that an optimization problem fall into this category, as far as the system cannot prove whether a given candidate answer is optimal or not. In AI, many learning techniques have this nature, such as genetic algorithm [12] and reinforcement learning [13]. In such a system, the goal is to optimize a measurement ("fitness", "reward", or "utility"), and the processing typically stops before all possibilities have been tried. If there is a fixed threshold, then the persistent goal is converted into a transient goal, by treating all states above the threshold as final states. However, this is not the only option. The persistence

nature will be handled better if such a goal is pursued using an anytime algorithm [14] and let the stop decision be made in a context-sensitive way, or the pursuing of the goal may never stop, though become dormant from time to time.

Compatible vs. restricting

Generally speaking, all interesting systems have multiples goals, since a non-trivial goal is almost always achieved though the achieving of its subgoals or derived goals. Even so, in traditional systems it is usually more fruitful to consider a single goal at a time. This is valid, because in the terminology of graph theory, the goals in such a system can be considered as a "forest", consisting of trees where the predecessor-successor relation between nodes represents the supergoal-subgoal relation between goals. Normally, the top-level, or *root*, goals in disjoint trees are *mutually compatible*, in the sense that the achieving of one does not prevent another from being achieved, otherwise the goals cannot coexist in the same system.

In many systems, each goal-tree can be represented by its root, because

1. The subgoals are recursively created as means to achieve the root goal;
2. As far as the goal-derivation process is designed correctly, the effects of the subgoal should be implied by the effects of the root goal;
3. The duration in which each subgoal exists is a sub-interval of the duration in which the root goal exist.

For these reasons, to analyze the goals of such a system, it usually suffices to only consider the top-level goals. Their subgoals may cause some issues, such as one may have another as a prerequisite, or the two may compete for a piece of resource, but these issues usually can be resolved by careful scheduling.

For AGI systems, however, this is not the case anymore. Even if the compatibility of the top-level goals can still be assumed (actually even this assumption is shaky), it definitely cannot be assumed for the subgoals derived from them. This is the case because a realistic AGI system is not omniscient, and at the same time has to deal with goals for which it has uncertain and incomplete information. Consequently, the goal derivation is only based on the system's current beliefs, which are not absolutely true. For example, if the system beliefs that event E_1 implies event E_2, then when the latter becomes a goal, the former may be derived as another goal. However, this situation is different from the above classical supergoal-subgoal relation, because E_1 and E_2 may turn out to be irrelevant, or even contradictory, to each other.

When the goals involved are persistent, the situation become even more complicated, because the existing period of a "subgoal" may be beyond that of the "supergoal" from which it was derived. Though it may sound irrational, there is an explanation for an adaptive system to do so, since a goal derived for one reason may be valuable for another purpose, or for similar purposes in the future, so becomes desirable *for its own sake*. It should not sound too strange to us, because many human motivations initially appear as means to achieve other ends.

Psychologist Allport called this phenomenon "functional autonomy of motives" [15], and it can also explain many Freudian notions, such as "compensation" and "sublimation". It has been argued that in an adaptive system working with insufficient knowledge and resources, such a phenomenon is inevitable [16].

It means that in such a system though goal H is derived as a way to achieve goal G, the former nevertheless may gradually gain independence. For this reason, the traditional "supergoal-subgoal" relation cannot be assumed anymore between an original goal and a derived goal. The relation between the two may only be *historical*, rather than *logical*.

As a result, the goals in an AGI system should be considered as *mutually restricting*, in the sense that the achieving of one sometimes does prevent another from being achieved, or at least makes it more difficult. To handle that requires the goal management mechanism to prioritize the existing goals for resource allocation, as well as to resolve their conflicts in action selection.

Constant vs. variable

There are several reasons to assume that in an AGI system the goals may change from time to time: the environment changes, the system's internal needs change (such as its energy reserves), and as discussed above, the overall *goal complex* of the system evolves as new goals are derived, even when the original goal remains the same.

Due to the resource restriction, an AGI system usually cannot take all of its existing goals into consideration at every moment. Instead, it has to focus on different goals at different moments. As a result, even though the system in its whole lifetime has many goals, at a moment usually only a small number of them are in effect in determining which action to take. These "effective goals" are what matters when the system's behavior is predicted or explained, not the dormant goals, though the latter do exist in the system, and some may have higher levels of significance in the system's lifetime.

If we take the goal complex of an AGI system as a whole, we should assume that it changes as the system runs, and the change is not circular, nor does it converge to a stable state — a system may never have identical goal-states in its lifetime, and that is arguably the case for a human being. On the other hand, the change is not pure random, or can be specified according to a probability distribution, because there will be new goals generated, which cannot be logically reduced into the previous goals.

For these reasons, it is not proper to assume that an AGI system always chooses or evaluates its actions according to a constant goal, no matter how that goal is specified or interpreted.

Mandatory vs. spontaneous

Many authors have expressed the opinion that a truly intelligent system should be "autonomous" [5, 6, 17, 4] or "self-motivated" [2], though what that exactly

means differ from author to author. Intuitively speaking, the consensus is that such a system should behave according to goals of its own choice or creation.

Some people consider this expectation impossible or even self-contradictory. After all, an AI system is designed, directly or indirectly, by human designers, who, among other things, specifies the system's (initial) goals. In this situation, how can the system have any goal that is not created, directly or indirectly, by its designer?

Actually we have answered this question. Previously, it has be explained that for an adaptive systems, even though all of its *initial* goals are specified by its designer as part of the system's initial state, the same cannot be said about the *derived* goals, which are decided by both the initial goals and the beliefs of the system. When the beliefs are learned from the system's experience, the goal complex of the system does not only depend on its initial design (its *nature*), but also on its experience (its *nurture*). When the system's experience is complicated enough, especially when it is not folly controlled by a tutor, the system may have goals that cannot be fairly attributed to anyone but *the system itself*.

Such a system still have *mandatory* goals that are either built-in by its designer, or imposed-upon by a user via its user interface. But at the same time, the system derives new goals recursively from the existing goals, and some of them can be considered as *spontaneous*, in the sense that they are not destined by the system's design, but mostly come out of the system's idiosyncratic history. Due to the functional autonomy phenomenon, these goals are not logically related to the initial goals, though they are derived from the latter. As the system gets more and more experience, it becomes more and more autonomous, in the sense that its behaviors are more and more oriented to its own goals.

2 Motivation management in NARS

As a concrete example of systems with goals that are *persistent, mutually restricting, variable*, and *spontaneous*, in the following we will introduce the representation and processing of motivations in NARS.

NARS is an AGI built in the framework of a reasoning system, based on the theory that "intelligence" is the ability of adaptation with insufficient knowledge and resources [16, 18]. This paper only describes the motivation management, plus the directly related aspects, of the system.

As many other systems, NARS can be analyzed at more than one level of description, where some "motivations" or "goals" can be recognized. For example, obviously every program consisting of NARS can be seen as goal-oriented, where the "goal" can be as simple as adding two numbers together. However, to analyze the system at such a level does not tell us much about its overall behaviors. Therefore, in the following we treat NARS as a whole, to see that type of "tasks" it can carry out.

Every task in NARS has a *statement* as its content, which is a sentence of a formal language whose grammar and semantics are accurately specified [16, 18]. There are three types of *task* defined in NARS:

***Judgment*:** In a judgment, the statement represents a conceptual relation experienced by the system, with a *truth-value* indicating the evidential support the statement gets. A truth-value consists of a *frequency* in [0, 1], which is the ratio of positive evidence among available evidence, and a *confidence* in (0, 1), which is the ratio of currently available evidence among all available evidence at a moment in the near future. Since the system is always open to new evidence, a confidence value can never reach is upper bound 1.0.

***Goal*:** In a goal, the statement represents a conceptual relation to be established by changing the environment or the system itself. A goal has a *desire-value* attached, which is a variant of truth-value, indicating the evidential support for the statement to be desired by the system.

***Question*:** In a question, the statement represents a conceptual relation whose truth-value or desire-value needs to be determined. A question may contain variables to be instantiated, corresponding to the *wh-questions* in a natural language.

To manage the resource competition among the tasks, in NARS each task is given a *priority-value* to indicate its relative priority in resource allocation at the moment.

Therefore, the *task* in NARS corresponds to what we call "motivation" or "goal" in general discussions, while the *goal* in NARS corresponds to a specific type of it. The other two types are distinguished from it, since they are processed differently in NARS, a reasoning system.

The tasks in NARS have two origins: *input* or *derived*, where the former are assigned to the system by its designer or user, while the latter are generated by the inference rules from the former (directly or indirectly) according to the beliefs of the system.

Input tasks can be either implanted into the system as part of its initial state, or assigned to the system through the user interface. As a general-purpose system, NARS can accept input tasks of any content, as far as they are expressible in its representation language, which allows arbitrary conceptual relations. The designer and users of the system can also assign priority-values to input tasks to influence the system's resource allocation.

NARS runs by repeating a working cycle, each time on a selected task, which can be either input or derived. What is done to a task depends on its type:

***Judgment*:** A judgment contains new information to be absorbed. The system uses it to revise the previous belief on the content to form a updated belief, to solve the pending goals or questions, and to spontaneously derive its implications using other beliefs. Unlike an ordinary database or knowledge base, NARS does not simply insert new knowledge into a storage, and let it wait there passively for future queries; instead, it actively revises and updates the system's beliefs, as well as makes predictions about future situation. This process recursively derives new judgments as tasks.

***Goal*:** When a goal is under processing, the system first checks its content against the reality to see whether somehow the request has already been satisfied. If not, the next step is to check whether there is an executable

operation that will directly satisfy the request. If neither is the case, the system will use its beliefs to derive new candidate goals as means to achieve the current goal. A candidate goal will not be directly pursued, but is used to adjust the desire-value of the corresponding statement. After the adjustment, if the desire-value of the statement is high enough, and the system believes that there is a way to achieve it, a corresponding goal will be generated, and pursued side-by-side with its "parent" goal.

***Question*:** When a question is under processing, the system keeps looking for the answer that is the current best (in terms of truth-value and simplicity). If the question is an input task, such answers are reported to corresponding user as soon as they are found. In the meanwhile, derived questions are recursively produced by using the inference rules *backwards*, so that an answer to the derived, or "child", question will produce an answer to the "parent" question. As a result, an input question may obtain multiple answers, each of which is better than the previous ones (as evaluated by the system), similar to the performance of an anytime algorithm [14].

For a task, its processing may contain any number of working cycles, depending on how many time it is selected for processing, which is proportional to its priority-value. Though an input task comes with a given priority-value, the system can adjust it according to the result of processing. For a derived task, its priority-value is initially determined and later adjusted by the system according to several factors. Overall, the priority-value of a task represents its urgency, plausibility to be achieved, and relevance to the current situation. Managed by a forgetting mechanism, all priority-values decay gradually, and tasks with the lowest priority-values will be removed when the storage space is in short supply.

Now we can see why the tasks in NARS have the properties listed previously:

- A task is *persistent*, since its processing rarely stops at its "logical end" — except in trivial situations, the system cannot exhaust all implications for a *judgment* task, nor can it find a perfect solution for a task which is a *goal* or a *question*. Instead, each time a task is processed, it is *partially achieved*, so its priority-value is deceased. When a task stops being processed, it is because its priority-value is too low, not because it has been fully achieved. How long a task lives depends on many factors.
- Tasks are *mutually restricting* because there is no requirement for the input tasks to be logically consistent in what they want the system to do. Furthermore, the task derivation is carried out according to the system's beliefs at the moment, which may be wrong. Finally, even compatible tasks compete with each other for the system's limited resources, so the achieving of one may cause another to be ignored temporarily or permanently.
- The overall task complex is *variable* because new (input and derived) tasks are added constantly to the system, while some old tasks get forgot gradually. Also, due to resource restriction, only a small part of the task complex is effective at a given moment, and controls the system's behaviors. Which task is in this active region changes from time to time.

– Certain tasks are *spontaneous* in the sense that they are only historically and remotely related to input tasks, and owe their existence mostly to the system's experience. Therefore, they should be considered as the system's *own* tasks. As the system runs, it tends to become more and more autonomous and self-motivated.

3 Implication and discussion

The above analysis shows that unlike the situation in ordinary computer systems and "narrow AI" systems, motivation management in an AGI system is more similar to the situation in the human mind. This is a natural consequence of the requirement of being general-purpose and working in realistic environments.

On one hand, an AGI system should not be considered as a problem-solving system that processes its goals one by one, as in BDI agents [19]. On the other hand, it should not be considered as guided by a constant ultimate goal, from which all the other motivations are logically derived as subgoals.

From a pure mathematical point of view, it is possible to refer to the whole goal complex or motivational mechanism as a single "goal" (like talking about the resultant of several forces in different directions), which changes from time to time, as the guidance of the system. However, to actually design or analyze an AGI system in this way is very difficult, if not impossible, and it is much easier and more clear to explicitly identify the individual factors, which may come and go from time to time, and compete with each other on what the system should think and do at each moment. For this reason, it is not a good idea for an AGI system to be designed in the frameworks where a single goal is assumed, such as evolutionary learning, program search, or reinforcement learning, despite of their other advantages [20, 21].

The major conclusion argued in this paper is that an AGI system should always maintain a goal structure (or whatever it is called) which contains multiple goals that are separately specified, with the properties that

– Some of the goals are accurately specified, and can be fully achieved, while some others are vaguely specified and only partially achievable, but nevertheless have impact on the system's decisions.
– The goals may conflict with each other on what the system should do at a moment, and cannot be achieved all together. Very often the system has to make compromises among the goals.
– Due to the restriction in computational resources, the system cannot take all existing goals into account when making each decision, and nor can it keep a complete record of the goal derivation history.
– The designers and users are responsible for the input goals of an AGI system, from which all the other goals are derived, according to the system's experience. There is no guarantee that the derived goals will be logically consistent with the input goals, except in highly simplified situations.

One area that is closely related to goal management is AI ethics. The previous discussions focused on the goal the designers assign to an AGI system ("super

goal" or "final goal"), with the implicit assumption that such a goal will decide the consequences caused by the A(G)I systems. However, the above analysis shows that though the input goals are indeed important, they are not the dominating factor that decides the broad impact of AI to human society. Since no AGI system can be omniscient and omnipotent, to be "general-purpose" means such a system has to handle problems for which its knowledge and resources are insufficient [16, 18], and one direct consequence is that its actions may produce unanticipated results. This consequence, plus the previous conclusion that the effective goal for an action may be inconsistent with the input goals, will render many of the previous suggestions mostly irrelevant to AI ethics.

For example, Yudkowsky's "Friendly AI" agenda is based on the assumption that "a true AI might remain knowably stable in its goals, even after carrying out a large number of self-modifications" [22]. The problem about this assumption is that unless we are talking about an axiomatic system with unlimited resources, we cannot assume the system can accurately know the consequence of its actions. Furthermore, as argued previously, the goals in an intelligent system inevitable change as its experience grows, which is not necessarily a bad thing — after all, our "human nature" gradually grows out of, and deviates from, our "animal nature", at both the species level and the individual level.

Omohundro argued that no matter what input goals are given to an AGI system, it usually will derive some common "basic drives", including "be self-protective" and "to acquire resources" [1], which leads some people to worry that such a system will become unethical. According to our previous analysis, the producing of these goals are indeed very likely, but it is only half of the story. A system with a resource-acquisition goal does not necessarily attempts to achieve it at all cost, without considering its other goals. Again, consider the human beings — everyone has some goals that can become dangerous (either to oneself or to the others) *if pursued at all costs*. The proper solution, both to human ethics and to AGI ethics, is to prevent this kind of goal from *becoming dominant*, rather than from *being formed*.

A similar analysis can be applied to the "the instrumental convergence thesis" of Bostrom [11]: though it is reasonable to assume the generation of certain "intermediary goals", there is no enough reason to believe that they will converge, independent of the system's experience. The problem comes from the belief that a "superintelligence" would be "more likely to achieve her final goals" [11]. Even though it is possible for an AGI to have more computational power and more experience than human beings, that does not make it omniscient and omnipotent. As argued in detail in [16], an AGI will still be bounded by insufficient knowledge and resources, which means it cannot realize all of its goals.

In summary, "intelligence" and "autonomy" are arguably two sides of the same coin. Therefore, the motivational mechanism in AGI systems will have properties that are more similar to those of the human beings than those of the traditional computer systems. Some of these properties are desired, while some others provide challenges to AGI research. None of the challenges has been proved unsolvable, though they demand novel ideas and approaches.

References

1. Omohundro, S.M.: The basic AI drives. In: Proceedings of AGI-08. (2008) 483–492
2. Liu, D., Schubert, L.: Incorporating planning and reasoning into a self-motivated, communicative agent. In: Proceedings of AGI-09. (2009) 108–113
3. Bach, J.: A motivational system for cognitive AI. In: Proceedings of AGI-11. (2011) 232–242
4. Thórisson, K.R., Helgasson, H.P.: Cognitive architectures and autonomy: A comparative review. Journal of Artificial General Intelligence **3**(2) (2012) 1–30
5. Beaudoin, L.P.: Goal processing in autonomous agents. PhD thesis, School of Computer Science, The University of Birmingham (1994)
6. Norman, T.J.F.: Motivation-based direction of planning attention in agents with goal autonomy. PhD thesis, Department of Computer Science, University College London (1997)
7. Hawes, N.: A survey of motivation frameworks for intelligent systems. Artificial Intelligence **175**(5-6) (2011) 1020–1036
8. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Language, and Computation. Addison-Wesley, Reading, Massachusetts (1979)
9. Wellman, M.P., Doyle, J.: Preferential semantics for goals. In: Proceedings of AAAI-91. (1991) 698–703
10. Newell, A., Simon, H.A.: GPS, a program that simulates human thought. In Feigenbaum, E.A., Feldman, J., eds.: Computers and Thought. McGraw-Hill, New York (1963) 279–293
11. Bostrom, N.: The superintelligent will: Motivation and instrumental rationality in advanced artificial agents. Minds and Machines (2012)
12. Holland, J.H.: Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems. In Michalski, R.S., Carbonell, J.G., Mitchell, T.M., eds.: Machine Learning: an artificial intelligence approach. Volume II. Morgan Kaufmann, Los Altos, California (1986) 593–624
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, Massachusetts (1998)
14. Dean, T., Boddy, M.: An analysis of time-dependent planning. In: Proceedings of AAAI-88. (1988) 49–54
15. Allport, G.W.: The functional autonomy of motives. American Journal of Psychology **50** (1937) 141–156
16. Wang, P.: Rigid Flexibility: The Logic of Intelligence. Springer, Dordrecht (2006)
17. Bach, J.: Seven principles of synthetic intelligence. In: Proceedings of AGI-08. (2008) 63–74
18. Wang, P.: Non-Axiomatic Logic: A Model of Intelligent Reasoning. World Scientific, Singapore (2012) (in press).
19. Rao, A.S., Georgeff, M.P.: BDI-agents: from theory to practice. In: Proceedings of the First International Conference on Multiagent Systems. (1995)
20. Schmidhuber, J.: The new AI: General & sound & relevant for physics. In Goertzel, B., Pennachin, C., eds.: Artificial General Intelligence. Springer, Berlin (2007) 175–198
21. Hutter, M.: Feature reinforcement learning: Part I. Unstructured MDPs. Journal of Artificial General Intelligence **1** (2009) 3–24
22. Yudkowsky, E.: Artificial intelligence as a positive and negative factor in global risk. In Bostrom, N., Cirkovic, M., eds.: Global Catastrophic Risks. Oxford University Press (2008) 308–345