Goal Generation and Management in NARS

Christian Hahm(✉), Boyang Xu, and Pei Wang

Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
{christian.hahm,boyangxu,pei.wang}@temple.edu

Abstract. AGI systems should be able to pursue their many goals autonomously while operating in realistic environments which are complex, dynamic, and often novel. This paper discusses the theory and mechanisms for goal generation and management in Non-Axiomatic Reasoning System (NARS). NARS works to accomplish its goals by performing executable actions while integrating feedback from its experience to build subjective, but useful, predictive and meaningful models. The system's ever-changing knowledge allows it to adaptively derive new goals from its existing goals. Derived goals not only serve to accomplish their parent goals but also represent independent motivation. The system determines how and when to pursue its many goals based on priority, context, and knowledge acquired from its experience and reasoning capabilities.

Keywords: Goal · Motivation · Task · Aim · Objective · Drive · Intention · Desire · Target

1 Introduction

Non-Axiomatic Reasoning System (NARS) is a general-purpose AI (AGI) system, which means that it can accept and pursue arbitrary goals. The ability to pursue arbitrary goals must be a fundamental aspect of AGI systems since they are meant to be capable of "general" problem-solving similarly to humans; that is to say they should have the ability to at least attempt to solve any problem they are presented with[1]. Indeed, a hallmark of any cognitive system's intelligence is its capability to achieve its goals. In the general sense, a system's "goal" is simply a motivation for the orientation of its activities. The goal is an objective that the system is working towards, whether successfully or not. It answers the question, "What is the system's aim?". Goals can be very specific (e.g. "Pick up that apple") or more general (e.g. "Be good").

All AI systems may be considered as having at minimum one goal since each is built to accomplish one or more tasks. AGI systems in particular will necessarily have multiple different goals at the same time due to the complexity of their environments. Although an AGI system's initial "seed" goals are predefined by a user, the system will invariably need to generate new goals (such as in subgoaling) in order to solve problems autonomously. The system also needs to effectively manage its existing goal complex: not all goals are equal, and certain goals may be more or less important depending on

[1] A human (or other intelligent system) will not necessarily be able to achieve any arbitrary goal, especially with limited knowledge and resources.

B. Goertzel et al. (Eds.): AGI 2021, LNAI 13154, pp. 96–105, 2022.
https://doi.org/10.1007/978-3-030-93758-4_11

context. Some goals will be actively pursued by the system, while others can lie dormant for any period of time, depending on each goal's desirability and relative priority. Goals may be transient, existing in the system for only a short time (such as when the system is completing a quick task), or persistent, existing in the system for a long time (possibly eternally, over the system's lifetime) [4,8].

2 Relevant Works

A useful AI system should exhibit autonomous behavior with respect to handling and accomplishing its goals. In the simplest case, an automation program has a single goal and one or more goal-specific actions available to it. Perhaps it responds to some stimulus or is otherwise programmed to execute some action when a predefined condition is met (e.g. "stamp the item when you detect it rolling past on the conveyor belt"), or performs some domain-specific algorithm.

More sophisticated systems use subgoaling to deal with a goal that requires multiple steps to achieve. When a goal cannot be achieved immediately, subgoals are derived from it (using the system's knowledge and backward chaining [4]) that may lead to its achieving. This subgoaling process is recursive, creating a tree-structure of goals when viewed graphically, as each subgoal may itself require further subgoaling [8]. The tree bottoms out at its leaves, either with "atomic actions the system can execute, goals that need no further subgoaling since they are already complete, or otherwise goals that cannot be subgoaled further due to insufficient knowledge.

An AI should be able to handle multiple goals simultaneously, and that is where things become complicated, since any cognitive system has only limited knowledge and capabilities. While operating, the system needs to intelligently reason about both its goals and what it knows about its environment to decide what actions to take. In many system designs, goals are assumed to be orderable and/or compatible using some priority ranking or planning, but this is not always a valid assumption to make as it is not always possible to rank goals by "objective" importance. This autonomous goal-managing capability of AI systems is a challenging problem and has been the target of many efforts, some of which are addressed here.

Belief-Desire-Intention (BDI) systems are a classical approach towards modeling autonomy with respect to goals. BDI is a term used to describe the separation of a system's internal knowledge: its beliefs (knowledge about itself and the world), desires (latent goals), and intentions (active goals) [7]. This distinction is agreeable, but such systems also tend to have the problem of assuming their goals to be static (in terms of meaning) and orderable (in terms of priority).

Goal-Driven Autonomy (GDA) is a recent approach in AI where the system dynamically selects, generates, and pursues novel (i.e. indirectly related) goals based on perceived changes and discrepancies in its environment, ultimately helping the system to achieve its goals more effectively. In [5], ARTUE escorts an ally in a strategy simulation and derives a novel goal to fortify the area when an enemy appears. Although fortifying the area is not a direct subgoal of escorting the ally, the goal is nonetheless useful to the mission because it protects the AI system and the escortee; the goal is adaptively derived in response to unexpected environment changes (i.e. an enemy appearing nearby). MIDCA [6] is tasked to build houses in a block world, but a hidden arsonist lights blocks on fire, hindering progress. The system derives goals to investigate and eventually apprehend the arsonist, preventing further fires. The GDA systems' ability to reason about environment changes and generate goals in response to anomalous situations allows the systems to outperform traditional models not based on GDA.

MicroPsi [2] is an AGI system that executes actions based on its active motives, which represent some events that the system desires to achieve or prevent. The system creates its motives in response to its urge signals, which are produced internally by a presupposed set of basic systemic needs (e.g. "competence", "sustenance", "rest", etc.). The system uses reward signals to indicate how the system's current actions and the changing state of the environment are impacting the satisfaction of the system's urges, providing feedback to the system about its success in the environment and allowing it to plan its next actions accordingly.

3 NARS Overview

Non-Axiomatic Reasoning System (NARS) is a unified model of general intelligence, where intelligence is defined by adaptation under an Assumption of Insufficient Knowledge and Resources (AIKR). The system operates using a formal logic called Non-Axiomatic Logic (NAL) which defines relationships between concepts and allows reasoning on them under uncertainty. The system is "non-axiomatic" in that its knowledge does not represent the true and complete state of the world, but rather represents knowledge about the world as based on the system's accumulated experience. The representation language of NAL is called Narsese.

AIKR means NARS works in real-time with finite processing capacity while being open to knowledge and tasks of any content. Under this assumption of resource and knowledge restriction, NARS incrementally processes all of its many beliefs, goals, and questions in parallel using its control mechanism. NARS absorbs information through its input channels and revises its knowledge during the course of its operation to build more reliable and useful internal models of the world. This can be considered a form of adaptation and is usefully paired with executable *operations* that NARS can perform to autonomously gain a richer understanding of its environment and pursue goals [9].

Similarly to the relevant works, NARS' goals drive the operations it performs. The system engages in common goal generation behaviors like subgoaling but is also unique in that the system explores many potential goal solution paths in parallel and optimizes these solution paths by reasoning on its existing beliefs and new incoming experience. Additionally, there are not many restrictions on the content, meaning, and usefulness of the goals themselves due to the flexibility of NARS' logic and formal language. The system can accept external goals via input, and also generates its own new goals based on the content of its memory. As long as a goal remains desirable to the system and there is no evidence to the contrary, it may continue to be pursued, while a completed or very unimportant goal may be abandoned.

NARS' goal complex changes throughout its lifetime as it adapts to its environment and the system gains new experience. Indeed, throughout a very long lifetime, the system may gain entirely new sets of goals and may no longer desire the goals with which it was initially programmed[2]. Successful self-control and autonomy in any AI system, including NARS, depends on the system's aptitude for generating and identifying useful goals. The system must also holistically manage its overall goal complex, because goals can interplay and in many situations it does not suffice to blindly pursue every goal at the exact moment it arises. Some goals require long-term planning and execution, and a NARS operating in an open environment could develop or acquire goals that conflict with each other.

NARS' internal memory can be visualized as a graph, where each node is some concept, and edges between nodes are relationships between concepts. A concept is named by a term (e.g. "cat").

In NARS, a piece of information is represented by a Narsese *sentence* called *judgment*, which consists of a *statement* and its corresponding *truth-value*. It has the following format:

$$\langle judgment \rangle ::= [\langle tense \rangle]\langle statement \rangle.\ \langle truth\text{-}value \rangle$$

The statement represents a relationship between a subject concept and a predicate concept (e.g. an inheritance statement "$<S \rightarrow P>$" means S is a specialization of P). The truth-value is a pair of numbers $(f,\ c) \in [0,1] \times (0,1)$, where f is the *frequency* (representing how *positive* or *negative* [true or false] the statement is according to the system's evidence), and c is the *confidence* (representing how much evidence the system has to justify its assessment of *frequency*).

For example, to represent that you are very confident that "a dog is a type of animal", a judgment can be written as "$<dog \rightarrow animal>.\ \langle 1.0,\ 0.9 \rangle$". This statement is directly related to 3 concepts: dog, $animal$, and $<dog \rightarrow animal>$. When a judgment is first absorbed into NARS, new subject, predicate, and statement concepts are created in the system's memory (unless they already exist). Additional judgments of the same content are accumulated as additional evidence within the corresponding concept's *belief table* and thus acquire stronger confidence values—the set of all these accumulated judgments within the system's memory make up the system's *beliefs*.

A judgment is optionally associated with a timestamp representing when its truth-value is valid; in that case, we can also call it an *event*. It represents temporal knowledge.

NARS operates by continually executing a working cycle, during which *tasks* and beliefs are processed. *Tasks* are *sentence* containers, outside of the system's memory in a buffer. Sentence types that can be processed by NARS including *judgment*, *question*, and *goal*. The exact mechanism depends on the specific NARS implementation, but in general, during a given working cycle two sentences are selected for processing. Tasks and other objects in the system's memory have a decaying *priority* $p \in [0,1]$ value that represents the object's relative urgency compared to other objects of the same type; priority can also be raised in various ways, allowing objects to compete for the system's time and resources. In each working cycle, the objects with the highest priority tend to be selected for processing, whereas objects with the lowest priority are pruned from the object container if it overflows [3].

[2] This potential temporary nature of goals may be overcome by periodically re-inputting the goals into NARS.

4 Goal Generation and Management

A goal in NARS is represented by a type of Narsese sentence called *goal*. It is essentially a special piece of knowledge representing that NARS desires an event to occur (or not occur) in the current moment. Similarly to how each concept in memory has a belief table which holds accumulated judgments, every concept also has a *desire table* which holds accumulated goals. A goal can either be contained in a *task* (in a buffer, ready to be processed) or referred to as a *desire* (once it is integrated into the system's memory).

New goals accumulate into desires of varying confidence which by extension drive the executable operations that the system performs. A goal in NARS has the following format:

$$\langle goal \rangle ::= \langle statement \rangle ! \ \langle desire\text{-}value \rangle$$

Here, the statement is an event that is desired by the system to occur (or be prevented) in the present moment, and NARS will actively work to achieve it.[3]

A *desire-value* is a pair of numbers $(f,\ c) \in [0,1] \times [0,1)$ that summarizes evidence for and against the event being desirable to NARS (i.e. how desirable the event is). Formally, desire-value is equivalent to the truth-value of a virtual judgment representing the extent to which the event S implies NARS' overall desired state D:

$$S! \ \langle f, c \rangle \iff <S \mid\Rightarrow D>. \ \langle f, c \rangle$$

Therefore, the frequency f represents the event's *desirability*, and confidence c represents how confident the system is in its assessment of desirability.

A *positive goal* is an event (S) with a high desirability (near 1.0), where the system wants to make S as *positive* (true) as possible in the present moment, because the event S's occurrence is sufficient for NARS' overall desired state (D).

A *negative goal* is a negated event ($\neg S$) with a high desirability (near 1.0), where the system wants to make S as *negative* (false) as possible in the present moment. The event S's non-occurrence[4] is sufficient for NARS' overall desired state (D).

For example, if you wanted to give your NARS a goal to "open the door", you could express it as either a positive goal ("NARS wants the door to be open") or a negative goal ("NARS wants the door to be not closed"), respectively:

$$<\{door\} \rightarrow [open]>! \ \langle 1.0, 0.9 \rangle \qquad \neg<\{door\} \rightarrow [closed]>! \ \langle 1.0, 0.9 \rangle$$

Given either of these goals, NARS will begin planning ways to open the door, though positive goals are preferred since they are syntactically simpler.

Goal management in NARS is handled innately by the system's design. The system is constantly processing new information coming from its sensors and thinking about what it already knows (i.e. reasoning on the information in its memory). The system's memory consists of concepts which contain collections of beliefs and desires. When NARS selects a strong desire for processing, the system works to achieve it.

[3] To represent events that are desired to occur in the future, we can simply add a temporal condition to the goal.

[4] Equivalently, the event $\neg S$'s occurrence.

Like all tasks in NARS, goals have one of two origins: *input* or *derived*. An input goal is assigned to the system from an external source (e.g. a user) through an input channel, whereas a derived goal is derived internally by backward inference rules using existing goals and beliefs as premises [8]. Either way, when a goal is selected for processing for the first time by NARS, it undergoes initial processing in the form of *revision*. The content of the new goal is merged into its corresponding concept's desire table, keeping NARS' overall assessment of the goal event's desirability up-to-date (see *Algorithm 1*).

Whenever it is subsequently selected by NARS again, the desire will undergo continued processing during which NARS may work to actively pursue it. A desire is actively pursued by NARS according to the Decision-Making Rule:

$$d > T_d$$

Where $d \in (0, 1)$ is the event's *expected desirability*, the *expectation*[5] of its desire-value (f, c) (where $d = c \times (f - 0.5) + 0.5$). The system's *desire threshold*, $T_d \in [0, 1.0]$, is a personality parameter representing NARS' cautiousness[6].

When a desire exceeds the desire threshold, it may be considered an active goal that NARS works to pursue - such a desire is like "intention" in BDI models, whereas those which have low but non-zero desirability are like the latent "desire" in BDI. Usually, a goal must be derived and accumulated many times before it reaches this threshold; that is, the system has derived and accumulated lots of evidence that the goal in question should be pursued, or alternatively the goal was input with high expected desirability directly by a user.

During a given working cycle, when a desire should be pursued based on the Decision-Making Rule, first the system checks if the goal event is already occurring (as based on the value in its belief table) – in this case, the goal does not require further action, and the system can move on to its next task. If the goal is not satisfied, the system next checks if the goal itself is an executable operation (e.g. $\Uparrow walk$, $\Uparrow jump$, $\Uparrow speak$), and if so immediately executes that operation. If the goal is not an operation, then the system will use related beliefs to derive new (potential) goals that, through their achieving, may lead to the achieving of the original goal (see *Algorithm 2*).

The system derives new goals using the knowledge it has acquired from its lifetime of experience, using a desire in backward inference with temporally related beliefs. If NARS has an empirical belief that a desired event will occur after the occurrence of

Algorithm 1. *Goal* initial processing (Narsese pseudocode)

```
Require: NARS selects task G for processing for the first time
Require: G is a goal of form: $\langle goal \rangle ::= \langle statement \rangle ! \langle desire\text{-}value \rangle$
$S_G \leftarrow \langle statement \rangle_G$
$C_G \leftarrow GetConcept(S_G)$                    ▷ Get concept named by $S_G$
Merge $G$ into $C_G$'s desire table               ▷ Keep concept's desirability up-to-date
```

[5] *Expectation* is an estimate of future frequency.

[6] Although currently this threshold is a constant, in future implementations it can be treated as a context-dependent variable.

some other events, then new goals for those precondition events will be derived. Formally, backward inference rule b derives new goal G' from existing goal G and related belief B, if and only if forward inference rule f derives J_G (the judgment satisfying G) from $\mathrm{J}_{G'}$ (the judgment satisfying G') and B:

$$\{G, B\} \vdash_b G' \iff \{\mathrm{J}_{G'}, B\} \vdash_f \mathrm{J}_G$$

In this way, a *derived goal* G' is derived *initially* as a potential subgoal of the original goal G—the system derives G' that, when achieved, may lead to achieving G. However, the role of G' in NARS is not restricted to subgoal of G. On the one hand, it is possible that the achieving of G' could also result in the achieving of other goals or the discovery of new knowledge that could be helpful in the achieving of G. On the other hand, G' is derived using the system's beliefs which (under AIKR) may be incomplete or even incorrect, since the system can only know what it has learned so far during its lifetime of experience; there is no guarantee that G' is useful for G.

If the achieving of a derived goal G' tends to garner outcomes that are favorable to the system, G' can be considered valuable in its own right, taking on a different role independent from subgoal of G. Conversely, the achieving of G' may ultimately end up being irrelevant or even inhibitory to the achieving of G, but at the time of derivation it is a reasonable guess. For these reasons, in NARS derived goal G' is not managed as a subgoal but is instead treated independently[7] as a goal that must compete with the other goals in the system for relevance and attention. Thus, derived goals are simply added to the task buffer where they may be absorbed to compete with the system's other goals. This independent nature of motivations is referred to as the "functional autonomy of motives" and is a phenomenon observed in humans [1].

Algorithm 2. *Goal* continued processing (Narsese pseudocode)

```
Require: $G$ is a goal of form: $\langle goal\rangle ::= \langle statement\rangle!\ \langle desire\text{-}value\rangle$
Require: NARS selects desire $G$ for continued processing
Require: $T_d$ is the system's desire-threshold
$d \leftarrow Expectation(\langle desire\text{-}value\rangle_G)$    ▷ Get desirability of $G$
if $d \leq T_d$ then return    ▷ Decision-Making Rule; stop if $G$ is not desirable enough
$S_G \leftarrow \langle statement\rangle_G$
$G_N \leftarrow$ Answer ($\neg S_G$¿ )    ▷ Get negative goal $G_N = \neg S_G!\ \langle f_1, c_1\rangle$
$d_N \leftarrow Expectation(\langle desire\text{-}value\rangle_{G_N})$    ▷ Get desirability of $G_N$
if $d_N > d$ then return    ▷ Inhibition; stop if $G_N$ is more desirable than $G$
$J_G \leftarrow$ Answer ($S_G$?);    ▷ Get judgment $J_G = S_G.\ \langle f_2, c_2\rangle$
$e \leftarrow Expectation(\langle truth\text{-}value\rangle_{J_G})$    ▷ Get truth expectation of $J_G$
if $e$ is positive then return    ▷ Stop processing if $G$ is already achieved
if $S_G$ is an operation then $Execute(S_G)$ return    ▷ Execute if $G$ is an operation
if No known sufficient precondition for $S_G$ then return    ▷ Stop if no beliefs related to $G$
$B \leftarrow$ Predictive belief related to $S_G$    ▷ e.g. $<(E\ \&/\ \Uparrow op)\ /\!\!\!\Rightarrow S_G>.\langle f_3, c_3\rangle$
$G' \leftarrow$ Derive new goal $\{G, B\} \vdash_b$    ▷ e.g. $(E\ \&/\ \Uparrow op)!\langle f_4, c_4\rangle$
```

[7] Though a limited record is kept regarding its origins.

The meaning and usefulness of each goal changes as a function of the system's experience, which continuously shapes the system's *many* interacting conceptual relationships, goals, and activities. A system with tens of thousands changing conceptual relations and constant streams of incoming sensorimotor experiences will derive and accumulate many pieces of evidence for and against a goal. Evidence for goals should be accumulated from across various beliefs, and kept up-to-date in the current moment according to context and what is currently known.

This highlights the important distinction between treating a derived goal (G') independently versus treating it as a subgoal. When a derived goal is treated purely as a subgoal, its usefulness is limited in that the system manages it solely for the greater purpose of achieving its corresponding supergoal. Many AI systems may try to pre-specify these goals, or assume that achieving the subgoals is absolutely necessary or sufficient for the achieving of the supergoal. However, under AIKR, the system cannot assume this is always the case, since there may be a different solution path to achieving the supergoal that doesn't require the derived subgoal, and perhaps uses different subgoals entirely, derived using very different concepts and knowledge. Conversely, treating derived goals *independently* allows them to take on a usefulness (or uselessness) of their own, based on their own merits. NARS treats goals independently by processing its many concepts and goals in parallel using a probabilistic priority queue—every goal has a chance to be processed in a given working cycle, and the system constantly takes in knowledge from its experience buffers to keep its memory up-to-date. A derived goal's usefulness includes, of course, how well it helps to achieve the goal from which it was derived. If the same goal is derived often without evidence to the contrary, it may gain extremely high confidence and priority.

The types of goals that may be derived in NARS are also unique due to the system's extensive inference capabilities. Usually, the process of subgoaling is purely deductive, but NARS' inference capabilities allow it to derive knowledge using induction, abduction, and analogy between two statements. When using this type of knowledge in backwards inference, the derived goal is essentially a guess – but an educated guess, which can be justified by the system's current knowledge.

NARS' goal derivation capabilities allow the system to adaptively react to its environment as a function of new experience gained. In the example in Fig. 1a, the NARS begins with only one initial goal: **touch the finish line**. The system is given some background knowledge, including that it can jump to navigate around obstacles. During its journey, the system encounters an obstacle and is obstructed. As a result, the system generates and achieves a goal to jump over the obstacle, allowing the system to continue making progress towards its original goal of moving to the finish line.

As long as NARS was assigned some initial goal and has related knowledge to infer new goals, the system will have multiple goals at any given time. Each goal's priority value denotes its urgency and is a factor in how much time the system will spend trying to achieve it. Depending on the situation, certain goals should be prioritized over others that may not be as relevant (e.g. when an AI robot's battery is low, the goal to find an electrical outlet to recharge is more relevant and should have high priority). Furthermore, the system is finite and under AIKR does not have the time, resources, or knowledge to achieve all goals fully or at once, especially as new goals can arrive

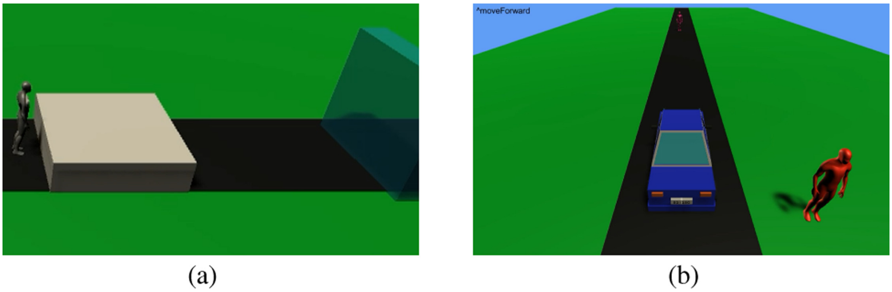

Fig. 1. (a) NARS reaches the finish line by generating new goals to jump over obstacles. [https://www.youtube.com/watch?v=iYHpk-JtIRQ], (b) A NARS vehicle dynamically handles goals that suddenly become conflicting. [https://www.youtube.com/watch?v=sNLWVtfk5tA]

at any moment; instead, the system must efficiently pick from its many goals so as to, over time, achieve them all to the system's best extent.

The overall goal complex of NARS may be disjointed, in that some goals are quite different from each other, and potentially even *conflicting* such achieving one makes it more difficult or impossible to achieve another. In that case, NARS may be forced to resolve conflicting goals at a moment's notice. The example in Fig. 1b shows a NARS car handling conflicting goals. The system is taught some simple background knowledge and the fact that it can execute operations to drive forward ("$\Uparrow drive$") and brake ("$\Uparrow brake$"). The NARS is initially input with 2 reasonable goals: 1. **drive forward**, and 2. **don't hit pedestrians**. However, these goals become conflicting when NARS encounters a pedestrian standing in the middle of the road. When the pedestrian appears in front and NARS processes highly-desired goal #2, the system may derive evidence that braking is desired (e.g."$\Uparrow brake!\ \langle 1.0, 0.9\rangle$") since it will help the system to not hit the pedestrian. The system could also derive evidence that preventing the execution of $\Uparrow drive$ is desired (e.g."$\neg \Uparrow drive!\ \langle 1.0, 0.9\rangle$"), since executing $\Uparrow drive$ will result in the failure of goal #2. These derivations working in tandem will encourage the NARS to execute $\Uparrow brake$ while inhibiting the NARS from executing $\Uparrow drive$.

5 Comparison and Conclusion

NARS is a real-time system that can accept and process new goals at any moment. However, AIKR implies there is no guarantee that a given goal can be achieved perfectly, or at all. There may not be a way for the system to achieve the goal immediately, and the path to achieving it may be uncertain (if the system lacks the knowledge) or impossible (if the system lacks the resources or capabilities).

NARS works to achieve all of its goals to the best extent by combining and extending multiple aspects of AI goal management. Similarly to GDA, the system is goal-driven and decides which goals to actively pursue based on desirability, priority, and the system's current beliefs. The system is adaptive, constantly incorporating new knowledge into its memory, and the system's goals change over time. NARS can also be described as roughly following a BDI model, in that its beliefs are separate from its goals, and its goals may be either actively pursued or latently desired. Such a distinction between active and latent desires becomes clearly necessary in the case of conflicting goals, where it may be necessary to suppress some goals for the achievement of others. Similarly to MicroPsi's appetitive and aversive goals [2], NARS also distinguishes between positive and negative goals, where the system desires to either achieve or prevent an event respectively.

However, NARS is also quite different from other goal-driven AI. The system's flexible formal language and various logical inference rules give it the freedom to be creative and adaptive in problem-solving situations. The system does not have any single leading motive or supergoal but instead works to explore and achieve all of its many goals in parallel, though to different extents depending on priority. Whether a given goal is latently desired or actively pursued is not permanently decided, but instead changes over time according to the system's learned experience. Currently, every new instance of NARS starts out empty with no goals, and instead acquires its initial goals at run-time.

Perhaps one of the more unique features of NARS is its independent treatment of derived goals. Since a derived goal is prioritized independently, its time-resource budget and length of existence may go beyond any of its ancestor goals and eventually play its own larger role in the system. In humans, the functional autonomy of motives is apparent and seems to be an integral aspect of adaptability. The meaning of goals is not static, but changes as the system gains new experience and the holistic content and organization of the memory changes. By virtue of its adaptive goal complex, the system's psychology can be compared to a human's in that the system can be said to develop its own dynamic traits, attitudes, and interests that evolve over its lifetime [1,8].

References

1. Allport, G.: The functional autonomy of motives. Am. J. Psychol. **50**, 141–156 (1937)
2. Bach, J.: Modeling motivation in MicroPsi 2. In: Bieger, J., Goertzel, B., Potapov, A. (eds.) AGI 2015. LNCS (LNAI), vol. 9205, pp. 3–13. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21365-1_1
3. Hammer, P., Lofthouse, T., Wang, P.: The OpenNARS implementation of the non-axiomatic reasoning system. In: Steunebrink, B., Wang, P., Goertzel, B. (eds.) AGI -2016. LNCS (LNAI), vol. 9782, pp. 160–170. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41649-6_16
4. Hawes, N.: A survey of motivation frameworks for intelligent systems. Artif. Intell. **175**(5), 1020–1036 (2011). https://doi.org/10.1016/j.artint.2011.02.002, http://www.sciencedirect.com/science/article/pii/S0004370211000336, special Review Issue
5. Klenk, M., Molineaux, M., Aha, D.W.: Goal-driven autonomy for responding to unexpected events in strategy simulations. Comput. Intell. **29**(2), 187–206 (2013)
6. Paisner, M., Cox, M., Maynord, M., Perlis, D.: Goal-driven autonomy for cognitive systems. In: Proceedings of the Annual Meeting of the Cognitive Science Society, No. 36 (2014)
7. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: from theory to practice. In: Icmas, vol. 95, pp. 312–319 (1995)
8. Wang, P.: Motivation management in AGI systems. In: Bach, J., Goertzel, B., Iklé, M. (eds.) Proceedings of the Fifth Conference on Artificial General Intelligence, pp. 352–361 (2012)
9. Wang, P.: Non-Axiomatic Logic: A Model of Intelligent Reasoning. World Scientific, Singapore (2013)