Advances in Cognitive Systems 10 (2023) 15–34 Submitted 5/2020; published 5/2023

The Problem with Problems

Michael T. Cox MICHAEL.COX@WRIGHT.EDU
Department of Computer Science & Engineering, Wright State University, Dayton, OH 45435 USA

Abstract

For over sixty years, the artificial intelligence and cognitive systems communities have represented problems to be solved as a combination of an initial and goal state along with some background domain knowledge. In this paper, I challenge this representation because it does not adequately capture the nature of a problem. Instead, a problem is a state of the world that limits choice in terms of potential goals or available actions. To capture this view of a problem, a representation should include a characterization of the context that exists when a problem arises and an explanation that causally links the part of the context that contributes to the problem with a goal whose achievement constitutes a solution. The challenge to the research community is not only to represent such features but to design and implement agents that can infer them autonomously.

1. Introduction

The task of problem solving was a central cognitive process examined during the genesis of the field of artificial intelligence. Like humans, a machine should be capable of solving difficult problems if it were to be considered intelligent. To illustrate such behavior, programs like the *General Problem Solver (GPS)* were given an initial starting state and a goal description, and they would output a sequence of steps that would achieve the goal if executed (Newell & Simon, 1963; Newell, Shaw, & Simon, 1959). This sequence of steps was considered a solution to the problem. Problem-solving itself was cast as heuristic search through the state space implicit in a given body of knowledge (in the case of GPS, inherent in its difference table) to find a combination of steps that met the goal criteria (Amarel, 1968; McCarthy & Hayes, 1969).[1]

Over the years, many types of problems have been studied. Initially, scientists developed procedures for various puzzles and games, such as the Towers of Hanoi[2] (e.g., Ernst, 1969; Knoblock, 1990), chess (e.g., Bilalić, McLeod, & Gobet, 2008; Chase & Simon, 1973; Hsu, 2002), and the Eight Puzzle and its derivations (e.g., Ratner & Warmuth, 1986; Russell, & Norvig, 2003). As research matured, attention turned toward complex design and planning tasks. For design problems, solutions are configurations for an artifact that meet given functional requirements and structural constraints (Chandrasekaran, 1990; Dinar et al., 2015; Goel, 1997; Maher, Balachandran,

[1] The Logic Theorist (Newell & Simon, 1956) proved theorems, where the given axioms formed an initial state, and the proposition to be proved represented the goal. The logical deductions from the initial state to the goal became the solution, but the representations used in GPS are more appropriate for this paper.

[2] At least 340 articles were published on the game in the 100 years from its invention in 1883–1983 (Stockmeyer, 2013), and apparently even ants can learn to solve an isomorphic version of the problem (Reid, Sumpter, & Beekman, 2010).

& Zhang, 1995; Vattam, Helms, & Goel, 2010). For automated planning, solutions are sequences of actions (i.e., steps) that achieve a goal (Fikes & Nilsson, 1971; Ghallab, Nau, & Traverso, 2016). This paper will focus on planning problems to illustrate our arguments in some depth.

Further, we will distinguish puzzles from problems. Puzzles do not contain a threat, entail risk, or in any significant way limit the choices available to an agent as do problems. We claim that the defining attribute of a problem is the restriction of an agent's choice. The contributions of this paper are to question the commonly accepted assumptions of the classical problem representation and to offer a formal alternative along with a computational implementation serving as an example.

This paper continues with three major sections. The first outlines the classical representation of a problem and enumerates some drawbacks of this construction. The second proposes an alternative problem representation and then challenges our research community to take seriously the three computational tasks of recognizing a problem, explaining what causes it, and generating a goal to remove the problem's cause. The third section illustrates how such concepts can be implemented. The subsequent section discusses related research and the closing section briefly reiterates the central challenge.

2. The Classical Problem Definition

What is a problem? An initial state, a goal state, and the means to get from one to the other.

2.1 Classical Problem Representation

Over time, the representation of a problem has been formalized with a standardized notation. Here we adapt the notation used by the automated planning community (e.g., Bonet & Geffner, 2001; Ghallab et al., 2016). Variations across AI, however, all have similarities to the format below.

Formal Problem Definition: A problem, $\mathcal{P}$, is a triple consisting of an initial state, s_0, a goal expression, g, and a transition model for the domain.

$$\mathcal{P} = (\Sigma, s_0, g) \text{ where } s_0 \in S, g \in G \subset S \tag{1}$$

State Transition System: This model is represented as a triple composed of the set of all possible states, S, a set of available actions, A, and a successor function, $\gamma\colon S \times A \rightarrow S$, that returns the next state, s_{i+1}, given a current state, s_i, and one or more actions, $\alpha \in A$.

$$\Sigma = (S, A, \gamma) \tag{2}$$

Problem Solution: The solution to a problem is an ordered sequence of n actions, π (i.e., a plan). In this paper, $\pi[i]$ denotes the i_{th} action, α_i, in the sequence, and $\pi[i..j]$ is the subplan starting with action α_i and ending at α_j.

$$\pi\colon 2^A = \alpha_1 \mid \pi[2 \dots n] = \langle \alpha_1, \alpha_2 \dots \alpha_n \rangle \tag{3}$$

Plan Execution: Starting from the initial state, s_0, recursive action executions result in the goal state, s_g that entails the goal expression, g. We say entails because a goal may be abstract (e.g., have *some* block on top of block A); thus, many states may satisfy the goal expression.

$$\gamma(s_0, \pi) = \gamma(\gamma(s_0, \alpha_1), \pi[2 \dots n]) \rightarrow s_g \vDash g \tag{4}$$

2.2 Problems with the Classical Representation

Significant issues exist with the classical representation of a problem. Representations of the form shown in equation (1) amount to arbitrary states to achieve and hence constitute a class of puzzles rather than problems. The problematic characteristics for the agent posed by the initial state and the relative attractiveness of the goal state are lacking in the representation. At best, we might say that s_0 may be of lower utility than s_g. The reason that achieving the goal state solves the problem is not actually clear. Indeed, this choice of representation leaves the problem itself implicit and opaque rather than declarative and open to inspection by the cognitive system. Instead, the causal justification for classifying $\mathcal{P}$ as a problem remains in the head of the researcher; the machine has no access to it and thus must blindly follow its set of problem-solving procedures. Reasoning about problems that arise in dynamic environments, formulating new goals as a result, and changing them as needed are essentially outside of the scope of the agent and remain the responsibility of a human.

Summarizing these arguments, the classical problem representation tends to possess three important limitations:

1. What is wrong with the initial state is left implicit;
2. The desirability of the goal state is opaque and cannot be explained; and
3. Problems must be provided by humans rather than inferred by a cognitive system or agent.

Hence, the representation for a problem is often overly simplified in the literature. Consider the Blocks World planning domain (Gupta & Nau, 1992; Winograd, 1972). Initial states in this domain are random configurations of blocks, and so too are the goals. For example, in the first panel of Figure 1, the initial state (a) is the arrangement of three blocks on the table, and the goal state (c) is to have block A on top of block B. The planner executes a plan to pick up A and stack it on B, but the planner has no reason why this goal state is valued. If the world changes dramatically, the agent simply adapts the plan to maintain the intended state without a causal justification for the adaptation other than the goal was given to it by a human. It does not have a solid basis to reason about the nature of the problem or its solution except perhaps for minimizing the solution's cost.

In the second panel, we assume a larger context such as the construction of buildings and towers. In this context, the planner wishes to have the triangle D on the block A to keep water out when it rains. Here the pyramid D represents the roof of the house composed of A, B, and C. Water being able to get into a person's living space is a problem for the person; stacking random blocks in various arrangements is not.

The next section examines a novel alternative to the classical definition of a problem. We explore what it means for a problem to be cast as a situation that restricts an agent and its problem-solving ability. A house with no roof allows a thief to steal the owner's property and the weather to ruin it. Property (e.g., an instrument such as a tool or a resource such as fuel) enables effective actions that achieve one's goals. Therefore, without the property, the owner has limited choice in terms of the goals that can be achieved and the actions that can be executed. Indeed, a house with no roof (or even a leaky one) represents a serious problem for any homeowner.

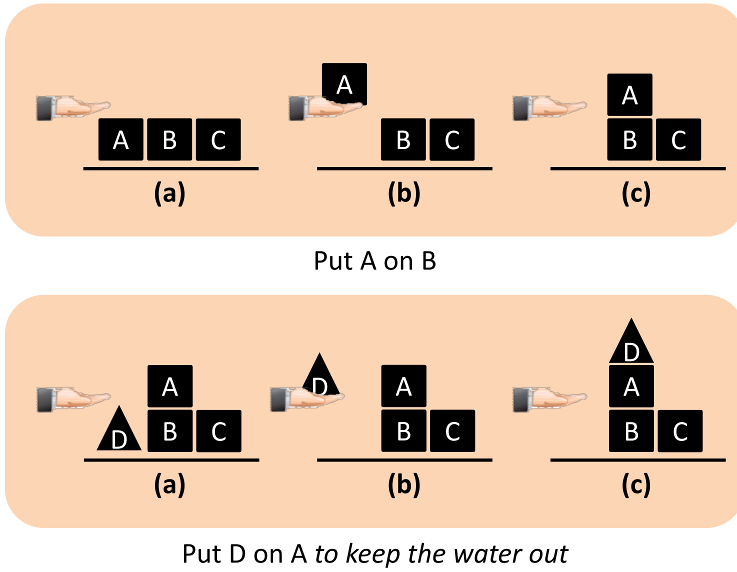

Figure 1. Blocks World state sequences that distinguish a justified problem in the lower panel from an arbitrary problem in the upper panel (adapted from Cox, 2013).

3. An Alternative Problem Definition

What is a problem? A situation that limits choice in terms of potential goals or available actions.

Problems are not simply puzzles or arbitrary states to be achieved. A problem is a situation relative to an agent (or agents) with some existing history of intent, executed actions, and decisions. Furthermore, problems arise even as one is working on other, independent problems. We claim that a situation is a problem for an agent whenever a significant risk exists (either immediate or latent) of a loss in ability to achieve its current or future goals or to select and execute various actions.

Potential goals are those that might be possible to formulate in the future; *kinetic goals* are those currently in an agent's agenda. Risks to either can pose a particular class of problems. For example, the loss of home value due to negative neighborhood trends (e.g., uncut lawns and abandoned vehicles) is a problem for a house's owner. It limits the potential goal of having the house sold, even if the owner does not currently have the desire to do so.

Alternatively, a problem can stem from a restricted action set, A. If an agent lacks the required action models (i.e., planning operators) to achieve its goals, then a limitation of choice also exists. For example, such a situation can occur when new technology is introduced into the workplace and older workers lack the necessary skills to perform a manufacturing job. In a sense, environmental change can cause similar outdating if an agent cannot learn new actions or adapt old ones.

Formal Problem Definition: In contrast to $\mathcal{P}$ in equation (1), the current problem, $\mathcal{P}_c$, is a tuple consisting of the currently observed and expected states, s_c and s_e, the background knowledge, Bk, an episodic problem-solving history, H_c, a causal explanation of the problem, χ, and a new goal,

g', whose achievement solves $\mathcal{P}_c$. The next three subsections examine each of these in turn. In particular, H_c (defined by equation 15 on page 21) includes components described in Section 3.1 and Section 3.2 .

$$\mathcal{P}_c = (s_c,\ s_e,\ Bk, H_c,\ \chi, g') \text{ where } s_c, s_e \in S \tag{5}$$

3.1 Representing the Intent Context

The key to understanding problems is to recognize the importance of goals or the intended future directions of an agent. The area of research called *goal reasoning* has attempted to develop cognitive systems with a capability to reason about their goals, to change them when warranted, and to formulate new goals when confronted with new problems (Aha, 2018;[3] Cox, 2007, 2013; Hawes, 2011; Klenk, Molineaux, & Aha, 2013; Muñoz-Avila, 2018; Vattam, Klenk, Molineaux, & Aha, 2013). To do so, problems must include a representation of the agent's dynamic context with respect to its intent. This includes: the agent's background knowledge, Bk; an interpretation function, β, that can change or formulate goals; the changing trajectory, $\vec{g}$, of the current goal; the system's current goal agenda, $\hat{G}_c$; and the agenda's history of change, $\hat{G}_h$.

Background Knowledge: The system's background knowledge, Bk, consists of the state transition system (see Section 2.1 equation 2) along with a set of goal operations, $\Delta = \{\delta | \delta: G \to G\}$, an interpretation function, β, and a planning function, φ (Section 3.2 expression 12).

$$Bk = (\Sigma, \Delta, \beta, \varphi) \tag{6}$$

Here, the action models within Σ enable an agent to predict subsequent states, s_e, and to use these expectations in comparison with observed states, s_c, to suspect the presence of problems (Dannenhauer & Muñoz-Avila, 2015; Dannenhauer, Muñoz-Avila, & Cox, 2020)

Interpretation Function: Given a state and a (possibly empty) goal, the interpretation function, β, performs *goal operations* from Δ outputting a desired goal expression (Cox, 2017; Cox, Dannenhauer, & Kondrakunta, 2017). This cognitive process is the dual to the planning function, φ, defined in the next section.

$$\beta: S \times G \to G \tag{7}$$

A specific operation from Δ is represented as the 4-tuple $\delta = (head(\delta), parameter(\delta), pre(\delta), res(\delta))$, where $pre(\delta)$ and $res(\delta)$ are its preconditions and result. The transformation's identifier is $head(\delta)$, and its input goal argument is $parameter(\delta)$. There are two essential goal operations. *Goal formulation* ($\beta(s, \emptyset) \to g$) infers a new goal given some state (Cox, 2007, 2013; Paisner, Cox, Maynord, & Perlis, 2014), whereas, *goal change* ($\beta(s, g) \to g'$) transforms an existing goal into another (Choi, 2011; Cox & Veloso, 1998; Cox & Dannenhauer, 2016).[4]

Goal Trajectory: The trajectory represents the original goal, g_1, and its evolution into the agent's current goal, g_c. It consists of an ordered sequence of state-goal pairs.

$$\vec{g} = \langle (s_0,\ g_1), (s_i, \beta(s_i, g_1)), \ldots (s_j, g_c) \rangle \tag{8}$$

[3] This paper summarizes work presented at the Robert S. Engelmore Memorial Lecture by David Aha at the Twenty-Ninth Conference on Innovative Applications of Artificial Intelligence in San Francisco.

[4] Goal formulation is implemented as the *insertion transformation* $\delta^*(\emptyset) \to g$. A trivial example of goal change would be the *identity transformation* $\delta^I(g_i) \to g_i$ for all $g_i \in G$, i.e., the tuple $(identity, g, \{true\}, g)$. Cox (2017) provides further detail, whereas Cox and Dannenhauer (2017) offer a more expressive goal representation.

Goals do not always remain as given or first formulated. They are malleable objects that change over time as an agent changes its intent. Goals go through arcs or trajectories in a goal hyperspace over time (Bengfort & Cox, 2015; Eyorokon, 2018; Eyorokon, Panjala, & Cox, 2017; Eyorokon, Yalamanchili, & Cox, 2018).

Current Goal Agenda: This set includes all goals that the agent intends to achieve. The current goal being solved, g_c, may be one, some, or all the goals in the agenda.

$$\hat{G}_c = \{g_1, g_2, \ldots g_n\} \tag{9}$$

Agenda History: This knowledge structure records the evolution of the goal agenda up to and including its current instance, $\hat{G}_c$. It is a simple sequence of the variations the agenda has undergone.

$$\hat{G}_h = \langle \hat{G}_1, \hat{G}_2, \ldots \hat{G}_c \rangle \tag{10}$$

3.2 Representing the Problem-Solving Context

Finally, the problem representation requires a formalism for the problem-solving process itself and its unfolding solution to a goal. The reason for this requirement is that new problems can arise during the act of solving a previous problem or during plan execution. To capture the problem-solving process so that a system can reason about potential limitations restraining it, this section describes the plan, π, the planning function, φ, the planning trajectory, $\vec{\pi}$, and the current execution episode, ε_c. These formalisms complete the constituents of the episodic, problem-solving history, H_c, first mentioned at the beginning of Section 3.

Plan: The dynamically executing plan consists of the previously executed steps (including current step, $\alpha_{\boldsymbol{c}}$) concatenated with all remaining steps ($\pi_{\boldsymbol{r}}$) of the plan.

$$\pi: 2^A = \langle \alpha_{\mathbf{1}}, \alpha_{\mathbf{2}}, \ldots \alpha_{\boldsymbol{c}} \rangle \circ \pi_{\boldsymbol{r}} = \pi_{\boldsymbol{c}} \circ \pi_{\boldsymbol{r}} \tag{11}$$

Planning Function: Given a state, a goal, and a (possibly empty) plan, the planning function, φ, performs a (re)planning operation using Σ (Cox, 2017).

$$\varphi: S \times G \times 2^A \rightarrow 2^A \tag{12}$$

Traditional plan generation is of the grounded form $\pi_1 \leftarrow \varphi(s_0, g_1, \emptyset)$. If the goal was inferred instead of given, then we have $\pi_1 \leftarrow \varphi(s_0, \beta(s_0, \emptyset), \emptyset)$. Replanning (e.g., Kunze et al., 2018; Langley et al., 2017; Pettersson, 2005) takes the form $\pi_{k+1} \leftarrow \varphi\big(s_i, g_j, \pi_k\big)$. Replanning with goal change would instead be $\pi_{k+1} \leftarrow \varphi\big(s_i, \beta(s_i, g_j), \pi_k\big)$.

Planning Trajectory: This trajectory is the sequence over time of changing plans paired with the goals they purport to solve starting from the first goal and plan (g_1, π_1) and ending with the current goal and the remainder of the plan that awaits execution (g_c, π_r).

$$\vec{\pi} = \langle (g_1, \pi_1), \Big(g_i, \varphi\big(s_j, g_i, \pi_1[k \ldots n]\big)\Big), \ldots (g_c, \pi_r) \rangle \tag{13}$$

Sometimes the plan changes because of exogenous events in the world or because previous uncertainty is removed; sometimes it changes because the goal changed. In other circumstances, both conditions may precipitate an alteration to the plan.

Current Execution Episode: The episode consists of the sequence of all states and executed actions that occurred up to but *not* including the current state, s_c.

$$\varepsilon_c = \langle s_0, \alpha_1, \gamma(s_0, \alpha_1), \alpha_2, \ldots s_{c-1}, \alpha_c \rangle \tag{14}$$

Episodic Problem-Solving History: This final knowledge structure encapsulates the goal, agenda, plan, and execution trajectories. It represents the dynamical, problem-solving context within which a problem is understood and solved by the cognitive system.

$$H_c = (\vec{g}, \hat{G}_h, \vec{\pi}, \varepsilon_c) \tag{15}$$

The new work developed in this paper centers about this representational structure and enables a cognitive system to reason about the full scope and content of problems, including the intent context (Section 3.1) and the overall problem-solving context in which intent is situated.

3.3 The Cognitive Systems Challenge: Inferring the Problem

Now we have the prerequisites for specifying a problem and the restriction of choice it represents for an agent. For example, if an agent is building a physical structure to contain its possessions and to safely house itself, it will have a typical set of goals to achieve and reasons for each. The goal to add the roof is causally connected to the need for guarding one's possessions and for personal safety and comfort. However, these ancillary needs are not threatened at construction time given that the possessions are safe elsewhere, it is not raining, and the agent does not currently live in the house. But if possessions are moved into the house and a proper roof is not in place, the possessions will lose substantial value when it rains. Lost value signifies reduced benefit and therefore fewer choices. This explanation (or others like it that relate the current state to what can occur in the future) supports the goal of having a roof placed on the structure. Such relationships become institutionalized in best practices (e.g., building codes), but they are crucial in relatively novel situations that pose new problems for any agent.

Problem Explanation: The explanatory graph consists of sets of vertices (V) and edges (E) that link the current state, s_c, causally to the limitation of choice:

$$\chi = (V, E) \tag{16}$$

To be fully effective in complex, uncertain, and changing environments, an intelligent agent must do more than just solve problems and achieve the goals given to it. Rather, it should be able to (1) recognize problems on its own; (2) explain what caused them; and (3) formulate an independent goal to solve the problem or remove the cause (Cox, 2013). Preliminary findings show benefits to this approach (Kondrakunta et al., 2019; Gogineni, Kondrakunta, Molineaux, & Cox, 2018, 2020), although it is quite difficult to cleanly separate out "true" problems from minor discrepancies encountered by an agent in such environments.

The claim is that the combination of these three tasks constitutes the next grand challenge for the AI community, especially for the cognitive systems community. Although the community has investigated non-traditional settings such as design tasks that require problem reformulation (e.g., Grace & Maher, 2016) and insight tasks that require viewing problems from different perspectives (e.g., Ohlsson, 2012), the representation of a problem often resembles the classical formulation, and much is externally provided by the user or researcher. If cognitive systems are to be genuinely autonomous with a human-like measure of independence, they should infer both the explanation χ and the new goal g' by themselves (placing the latter in their agenda, $\hat{G}_c$). They should not simply generate some plan π and then wait for a human to given them further direction.

Reduced Problem Definition: In accordance with this challenge, a current problem would be represented as a 4-tuple adapted from equation (5) on page 18. Neither the goal nor the explanation would be given a priori.

$$\mathcal{P}_c = (s_c, s_e, Bk, H_c) \tag{17}$$ [5]

Therefore, instead of a sole plan, π, the solution to $\mathcal{P}_c$ would be a 3-tuple $(\chi, g', \pi_{g'})$, where χ is an explanation that justifies a new goal g' and $\pi_{g'}$ is a plan to achieve it. Prevailing over time in the above tasks will enable cognitive systems to manage problems flexibly on their own and, if necessary, to explain to others the reasons for their choices, appropriately outputting χ when asked about a new goal or g' when asked about unexpected actions.

In summary, this section has presented a novel formalization of a problem that serves as an alternative to the classical formulation. The representation encodes the problem context in terms of the agent's intent and a history of the agent's current problem solving when a new problem arises. This knowledge is encapsulated in the episodic problem-solving history, which is composed of four cognitive trajectories:

- *The goal trajectory*: How the current active goal set has changed over an interval of time;
- *The agenda history*: How the entire set of active and inactive goals has changed over time;
- *The plan trajectory*: How the current plan has changed; and
- *The current execution episode*: A record of actions and resulting states during plan execution.

Existing systems cannot make full use of such knowledge structures or completely generate the solutions and explanations discussed here. However, the next section and its examples demonstrate some initial steps and how an implemented system could begin to exploit these representations and start to address the challenge of Section 3.3.

4. Computational Implementation and Example

This section examines how the alternative representation for problems can be implemented in an existing computational framework. Subsection 4.1 briefly characterizes this framework, after which Subsection 4.2 describes the mine clearance domain used for its implementation. Finally, Subsection 4.3 examines the structure of an example problem and details how the implementation uses the pieces of the problem representation described in Section 3.

4.1 The MIDCA Framework

The *Metacognitive, Integrated, Dual-Cycle Architecture (MIDCA)*[6] (Cox et al., 2016; Cox, Oates, & Perlis, 2011; Paisner, Cox, Maynord, & Perlis, 2014) is an architectural framework for intelligent cognitive systems. Figure 2 depicts the cognitive layer as an iterative repetition of six phases along with an abstract depiction for the metacognitive layer. MIDCA consists of an "action-perception"

[5] In many ways, Bk is like long-term memory and H_c is like a working memory. Then, considering an agent's memory to be $M = (Bk, H_c)$, the problem becomes $\mathcal{P}_c = (s_c, s_e, M)$.

[6] See http://www.midca-arch.org and, for the code repository, https://github.com/COLAB2/midca. Dannenhauer et al. (2022) provide documentation for the implementation.

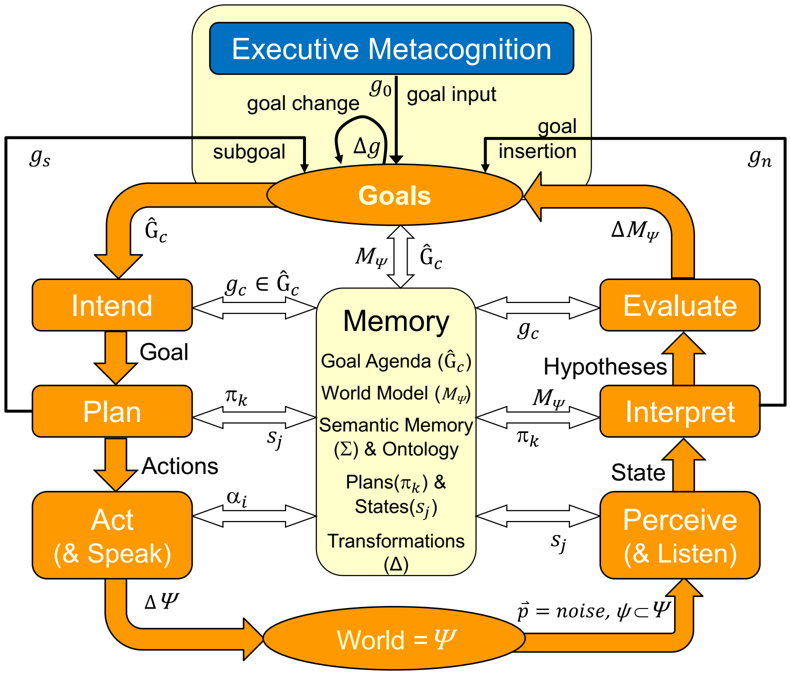

Figure 2. A functional decomposition of the major cognitive processes in MIDCA: The Perceive, Interpret, Intend, Plan, and Act phases. Although abstracted here, these are duplicated at the meta-level. A similar six-phase cycle operates at the metacognitive layer. Note that Interpret also formulates new goals (g_n).

cycle at both levels. The output side of each cycle consists of intention, planning, and action execution; whereas the input side consists of perception, interpretation, and goal evaluation.

The Intend phase selects a current goal set (g_c) from its goal agenda ($\hat{G}_c$) and commits to achieving it. The Plan phase then generates a sequence of actions (i.e., the plan π_k) to achieve the goal, and subsequently the Act phase executes the next action (α_i) from the plan to move the current state (s_c) toward the goal state (g_c). The Perceive phase observes percepts ($\vec{p}$) that represent changes to the environment resulting from each action; the Interpret phase interprets the resulting state (s_j) with respect to the plan; and the Evaluate phase assesses the interpretation (M_Ψ) with respect to the goal, determining whether or not it has been achieved. Note that the Interpret phase implements the three tasks mentioned in Section 3.3, including components that (1) recognize a problem (anomaly detection), (2) explain what causes it (explanation), and, if necessary, (3) generate a new goal, g_n (goal formulation). Details are available in Cox (2013, 2017) and Paisner, Cox, Maynord, and Perlis (2014).

The meta-level cycle is analogous to the cognitive cycle, consisting of Intend, Plan, and Control for the problem-solving (output) side and comprising Monitor, Interpret, and Evaluate on the comprehension (input) side. The Monitor phase "perceives" a trace of the cognitive-level process phases and the Control phase executes meta-level planning "actions" to regulate the cognitive-level cycle. The focus of the current paper is on cognitive-level problem representations, but Dannenhauer, Cox, and Muñoz-Avila (2018) and Cox et al., (2021) report further details of the metacognitive cycle in MIDCA.

4.2 The Mine Clearance Domain

To prepare a harbor for use during maritime operations, it is essential to conduct mine clearance activities to ensure that ships can operate safely as they transit between the open sea and a port. A network of safe shipping lanes is typically established to reduce the size of the area within the harbor. Such a system is known as a Q-route (Li, 2009). For experimentation, we modeled the mine clearance domain (Gogineni, Kondrakunta, Molineaux, & Cox, 2018; Kondrakunta et al., 2018) with a fixed Q-route that consists of a single shipping lane (see Figure 3). In this simulation, MIDCA controls a Remus autonomous underwater vehicle through an interface to the MOOS IvP software (Benjamin, Schmidt, Newman, & Leonard, 2010) and performs both mine detection and clearance. For planning in this domain, MIDCA uses the Pyhop *hierarchical task network (HTN)* planner that is based on the SHOP2 HTN planner (Nau et al., 2003).[7] Actions include deploying, transiting between locations, clearing mines from areas, avoiding obstacles, and being picked up.

For this domain, we developed several test scenarios. In each scenario, the agent knows of two previously identified areas within the Q-route (i.e., green area one, $GA1$, and green area two, $GA2$) where mines are expected (see again Figure 3). MIDCA is given goals to clear each area, although the location and number of mines are not known in advance. An area is clear if all mines within it satisfy the is-cleared relation (the effect of a do-clear action).

$$\text{cleared}(area) \Leftrightarrow \forall m, l \mid \text{location}(l) \wedge \text{mine}(m) \wedge \text{within}(area, l) \wedge \text{at-location}(m, l) \\ \rightarrow \text{is-cleared}(m)$$

As such, any mines encountered that do not lie within GA1 or GA2 constitute discrepancies. However, only mines within the Q-route are classified as problems, because ones outside the route will not pose a hazard to shipping. The agent's role is to determine how to respond to all mines in each scenario.

4.3 Representing Mine Clearance Problems

Now we will describe how MIDCA represents and solves new problems in this domain as they arise. At initialization time, each element of the problem-solving history, H_c, from equation (15) is initialized to empty sequences such that $H_c = (\vec{g} \leftarrow \langle\,\rangle, \hat{G}_h \leftarrow \langle\,\rangle, \vec{\pi} \leftarrow \langle\,\rangle, \varepsilon_c \leftarrow \langle\,\rangle)$. MIDCA always starts with the Perceive phase to establish the initial state, s_0, and to set the execution episode from equation (14) to $\varepsilon_c = \langle s_0 \rangle$. The Interpret phase detects the initial three goals (i.e., $g_1 = \text{cleared}(GA1)$, $g_2 = \text{cleared}(GA2)$, and $g_3 = \text{stored}(p)$) and adds them to the starting goal agenda from equation (9), $\hat{G}_c \leftarrow \{g_1, g_2, g_3\}$. The Evaluate phase checks to see if the goal state is achieved (it is not), after which the Intend phase chooses all three goals by setting the current goal expression, g_c, as their conjunct.

$$g_c \leftarrow g_1 \wedge g_2 \wedge g_3$$

[7] The link https://bitbucket.org/dananau/pyhop/src/master/ points to the open source code for Pyhop.

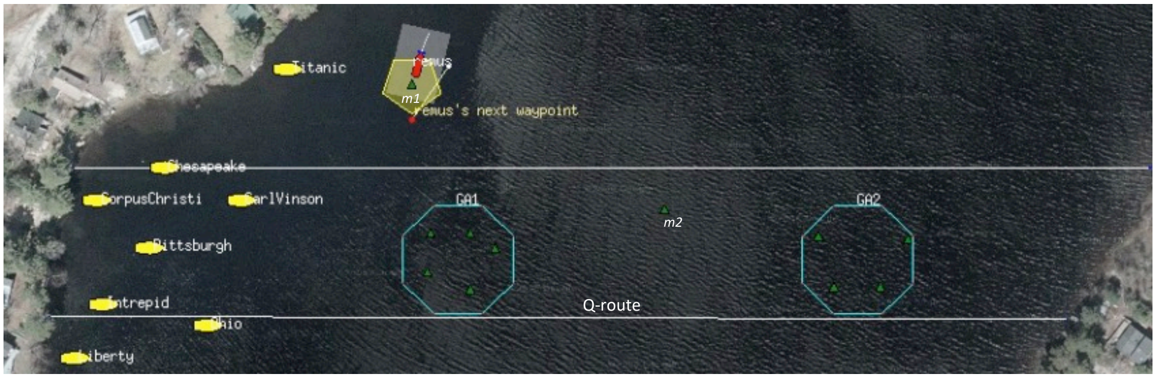

Figure 3. Simulation of the mine clearance domain in Moos IvP. The Q-route extends from the left to the right side of the map. Shipping (shown in yellow) awaits on the left side of the map, and the Remus platform (in red) encounters a mine ($m1$ in the pentagon) as it transits to the $GA1$ location.

Subsequently, the Plan phase produces a seven-step plan, π, to achieve the goals and sets the beginning plan trajectory to $\vec{\pi} = \langle (g_c, \pi) \rangle$, as shown in expression (12).

$$\pi[1..7] \leftarrow \varphi(S_0, g_c, \emptyset) = \left\langle \begin{array}{c} \text{deploy}(p, start), \text{transit}(p, start, GA1), \\ \text{do-clear}(p, GA1), \text{transit}(p, GA1, GA2), \text{do-clear}(p, GA2), \\ \text{transit}(p, GA2, dest), \text{pick-up}(p, dest) \end{array} \right\rangle$$

Finally, the Act phase executes the first step, $\text{deploy}(p, start)$, and sets the execution history to $\varepsilon_c \leftarrow \varepsilon_c \circ \langle \alpha_1 \rangle = \langle s_0, \text{deploy}(P, \text{start}) \rangle$. These six phases are then repeated in succession. At each instance i throughout the MIDCA cycle, Act changes $\varepsilon_c \leftarrow \varepsilon_c \circ \langle \alpha_i \rangle$.

4.3.1 First Encountered Problem: Discrepancy, Explanation, and Goal

MIDCA discovers a surprise after it starts to execute the second action of its plan above. Figure 3 shows the state of the environment (s_2) during the transit from the starting position to $GA1$. Here, the Remus' side-scanning sonar detects the mine $m1$. Perceive then adds s_2 to the current execution episode, ε_c, and changes the second action from transit(p, *start*, *GA1*) to $\text{transit}\big(p, \text{start}, \text{loc}(m_1)\big)$.

$$\varepsilon_c = \langle s_0, \text{deploy}(p, start), s_1, \text{transit}\big(p, start, \text{loc}(m_1)\big), s_2 \rangle$$

MIDCA's Interpret phase recognizes a discrepancy because it expects the transit area to be clear, but it observes a mine in the area. That is, the expectation, s_e, is equivalent to the expression $\forall l \mid \text{location}(l) \land \text{within}(clear\text{-}area, l) \land \nexists m | \text{mine}(m) \land \text{at-location}(m, \text{l})$, and the observed predicate $\text{at-location}(l, m) \subset s_2$ violates it. At this point, MIDCA has established a new episodic problem-solving history that lets it reason about changes in the future. Now instantiated from equation (17), the current problem is:

$$\mathcal{P}_c = (s_2, \nexists m1, (\Sigma, \Delta, \beta, \varphi), H_c) \text{ where } H_c = (\vec{g}, \hat{G}_h, \vec{\pi}, \varepsilon_c)$$

The Interpret phase explains that this might have been placed in the area by an enemy mine-laying vessel (see Figure 4) and that, because it is outside of the Q-route, it does not represent a

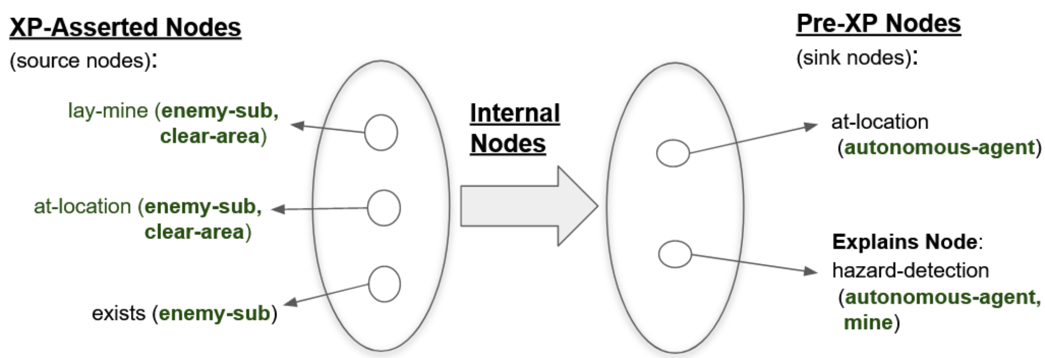

Figure 4. The abstract Mine-XP (taken from Kondrakunta et al., 2019). *Explanation patterns (XPs)* (Schank, 1986; Schank, Kass, & Riesbeck, 1994) map observed Pre-XP nodes to inferred XP-Asserted nodes that cause the Explains node. Bold symbols represent variables that are matched against and unified with objects and relations in the state.

problem to friendly shipping.[8] Instead, it generates a goal to avoid the mine itself, adds this to the goal agenda, and updates the agenda history to reflect the new status.

$$g_4 \leftarrow \beta(s_2, g_c) = \text{avoided}(m1)$$
$$\hat{G}_c \leftarrow \hat{G}_c \cup g_4$$
$$\hat{G}_h \leftarrow \left(\hat{G}_h \circ \langle \hat{G}_c \rangle\right) = \langle \{g_1 \wedge g_2 \wedge g_3\}, \{g_1 \wedge g_2 \wedge g_3 \wedge g_4\} \rangle$$

The Evaluate phase does nothing since the goal has not yet been achieved, but the Intend phase adds g_4 to the current goal conjunct, i.e., $g_c \leftarrow g_c \wedge g_4$, after which Intend updates the goal trajectory.

$$\vec{g} = \langle (s_0, g_1 \wedge g_2 \wedge g_3), (s_2, g_1 \wedge g_2 \wedge g_3 \wedge g_4) \rangle$$

MIDCA's Plan phase then modifies the remaining current plan fragment, $\pi_r = \pi[3..7]$, to achieve the new current goal by adding two steps to the front of the plan. The phase also changes the plan trajectory to incorporate the expanded current goal and the newly updated plan.

$$\pi' \leftarrow \varphi(s_2, g_c, \pi_r) = \langle \text{avoid}(p, m1), \text{transit}(p, \text{loc}(m1), GA1) \rangle \circ \pi_r$$
$$\vec{\pi} = \langle (g_1 \wedge g_2 \wedge g_3, \pi), (g_1 \wedge g_2 \wedge g_3 \wedge g_4, \pi') \rangle$$

4.3.2 Second Encountered Problem: Discrepancy, Explanation, and Goal

After continuing execution from the location of $m1$, the Remus platform continues to $GA1$ and clears all mines in that location. During the transit from $GA1$ to $GA2$, however, MIDCA encounters the mine $m2$, as shown in Figure 5. The presence of this mine also constitutes a discrepancy because no mines were expected in the area between $GA1$ and $GA2$.

[8] Previous papers provide further details on explanation patterns, their representation, and how they are retrieved, selected, and applied (Cox, 2011; Cox & Ram, 1999; Gogineni et al., 2018, 2020; Kondrakunta et al., 2019; Ram, 1990; Schank, 1986; Schank et al., 1994).

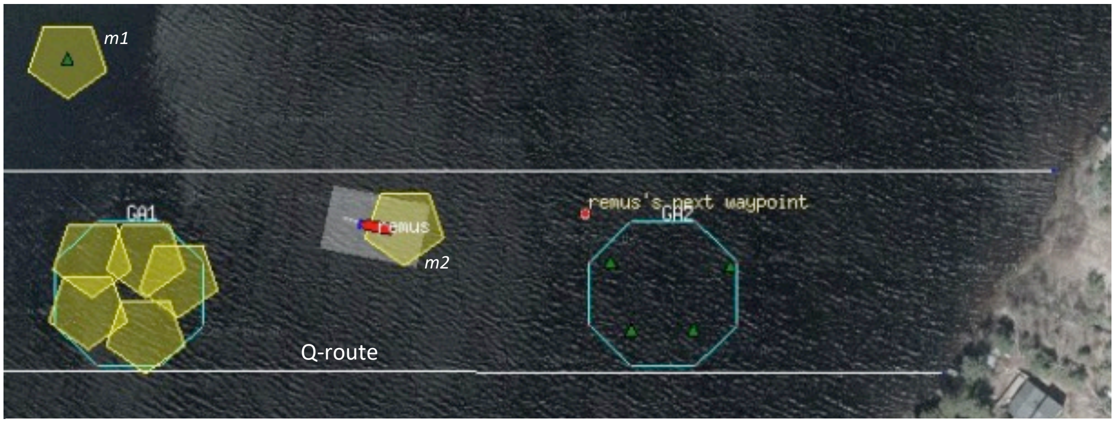

Figure 5. The Remus encounters another surprise in the mine clearance domain. The mine $m2$ is within the Q-route and hence represents a problem to the ships as they traverse the channel.

At this point, the Perceive phase updates the current execution episode. As in the previous example from Section 4.3.1, it replaces α_4 in ε_c with transit$(p, GA1, \text{loc}(m2))$ and adds s_6.

$$\varepsilon_c = \left\langle \begin{array}{c} s_0, \text{deploy}(P, \text{start}), s_1, \text{transit}(p, \text{start}, \text{loc}(m_1)), s_2, \text{avoid}(p, m_1), s_3, \\ \text{transit}(p, \text{loc}(m1), GA1), s_4, \text{do-clear}(p, GA1), s_5, \text{transit}(p, GA1, \text{loc}(m2)), s_6 \end{array} \right\rangle$$

Here, the Interpret phase recognizes another discrepancy because it expects the transit area between the two target areas to be clear, but it observes the mine $m2$. Once again, the discrepancy is caused because it expects no mine (i.e., $\nexists m2$) and it observes one in state s_6. The problem is:

$$\mathcal{P}_c = (s_6, \nexists m2, (\Sigma, \Delta, \beta, \varphi), H_c)$$

As before, the system explains that this might have been placed in the area by an enemy mine-laying vessel, but in this case the mine is inside the Q-route and so *does* represent a problem to friendly shipping. A new goal is formulated to clear m2 and it is added to the agenda. Subsequently, MIDCA updates the agenda history.

$$g_5 \leftarrow \beta(s_6, g_1 \wedge g_2 \wedge g_3 \wedge g_4) = \text{is-cleared}(m2)$$
$$\hat{G} \leftarrow \hat{G} \cup g_5$$
$$\hat{G}_h \leftarrow \hat{G}_h \circ \langle \hat{G} \rangle$$

Again the Evaluate phase does nothing, but the Intend phase adds g_5 to the current goal and updates the goal trajectory.

$$g_c \leftarrow g_c \wedge g_5 = (g_1 \wedge g_2 \wedge g_3 \wedge g_4) \wedge g_5$$
$$\vec{g} \leftarrow \vec{g} \circ \langle (s_6, g_c) \rangle$$

In the Plan phase, MIDCA takes the remaining plan, $\pi'_r = \pi'[5..7]$, and generates a new plan. As a result, it also adjusts the plan trajectory. This new plan can now be carried out by the Act phase with the result that ships can safely traverse the channel to deliver supplies in the harbor.

$$\pi" \leftarrow \varphi(s_6, g_c, \pi'_r) = \langle \text{do-clear}(p, m2), \text{transit}(p, \text{loc}(m2), GA2) \rangle \circ \pi'_r$$
$$\vec{\pi} = \langle (g_1 \wedge g_2 \wedge g_3, \pi), (g_1 \wedge g_2 \wedge g_3 \wedge g_4, \pi'), (g_1 \wedge g_2 \wedge g_3 \wedge g_4 \wedge g_5, \pi") \rangle$$

Finally, the Evaluate phase checks that the current state entails the goal state and clears the agenda. As such, this example demonstrates the potential of the alternative representations specified in this paper. Because the knowledge structures capture not just the initial and goal state but the problem-solving context within which problems arise, the agent has access to much more problem-relevant information when reasoning about the direction the agent is headed (both cognitively and physically) within a dynamically changing world.

5. Related Research

An alternative formal model (Johnson, Roberts, Apker, & Aha, 2016; Roberts et al., 2015, 2014) treats goal reasoning as *goal refinement*. Using an extension of the plan-refinement model of planning, Roberts and colleagues model goal reasoning as refinement search over a *goal memory* M, a set of *goal transition operators* R, and a transition function *delta* that restricts the applicable operators from R to those provided by a fundamental *goal lifecycle*. Unlike the formalism here, which represents much of the goal reasoning process with the function β, they propose a detailed lifecycle that consists of goal formulation, selection, expansion, commitment, dispatching, monitoring, evaluation, repair, and deferment. Thus, many of the differential functionalities in β are distinct and explicit in the goal reasoning cycle, but problems are represented classically.

Both goal reasoning and explainable AI (Aha et al., 2017; Cox, 2011, 1994; Gunning, 2016; Lane et al., 2005) are research areas that question the status quo and push the frontiers of what we think machines should be able to accomplish on their own. These lend support to the proposition that both goals and explanations of problem solving or performance are important for representing and understanding problems. The planning community is beginning to entertain the view that planners are more than generators of action sequences; they must consider dynamic and uncertain environments where decisions, action execution, collaboration, and replanning interact (Ghallab, Nau, & Traverso, 2014, 2016). Yet the representation of a problem remains much the same as it has for some sixty years (cf., Patra, Traverso, Ghallab, & Nau, 2018).

Many researchers in the cognitive systems community have proposed problem representations and specified numerous problem-solving mechanisms. However, most of these assume some variation on the basic representation of an initial state and goal description given by a human or otherwise input to the system. Although progress has been made, existing research emphasizes developing methods to produce solutions. Thus, most problems that appear in the literature are closer to puzzles. For example, Klenk and Forbus (2009) developed an analogical method that solves AP Physics problems. These problems consist of a set of given facts and a goal query that seeks a particular value for some quantity. Langley et al. (2016) use heuristic search through a space of candidate decompositions of a problem, but problems themselves consist of state-goal pairs. Still, many cognitive systems such as PUG (Langley et al., 2017) recognize that goals are not simple predicate states. Instead, they differ widely according to utility and other attributes, and problem solutions must be monitored in dynamic environments.

Additionally, the concept of a *MacGyver problem* (Sarathy & Scheutz, 2018) is quite interesting because it represents a problem that resides partially outside the transitive closure of the existing background knowledge of the agent, hence requiring insight for a solution. However, like most other representations in the community, it assumes the formalism from equation (1), although with a novel twist that can be stated:

$$\mathcal{P}_M = (\mathbb{W}^t, s_0, g) \text{ where } \mathbb{w}^t = (\mathrm{S}^t, \mathrm{A}^t, \gamma^t) \tag{18}$$

Like the state transition system of equation (2), the *world* $\mathbb{w}^t$ is composed of a set of possible states, actions, and a successor function, each specific to agents of type t. This world contains a portion of a larger *universe* $\mathbb{u}$ that includes further possible states, actions, and transitions not initially available to the agent. To solve $\mathcal{P}_M$, an agent must learn or infer missing constituents. Although the representation of MacGyver problems suffer from many of the limitations enumerated in Section 2.2 , Sarathy and Scheutz also represent the evolving context of the agent:

$$\mathbb{C}_i = (\Sigma_i^t, s_i) \text{ where } \Sigma_i^t = (S_i^t, A_i^t, \gamma_i^t) \tag{19}$$

The *context* $\mathbb{C}_i$ consists of the current state s_i and the subdomain existing at time step i. A *subdomain* Σ_i^t represents the perceptions and actions currently available to an agent within its world. Therefore, a solution to $\mathcal{P}_M$ is obtained by iteratively extending (or contracting) its domain using a set of domain modifications Δ until the goal is reachable from its current state. At this point, the solution π to the problem can be output. Although these conceptualizations are certainly steps in the right direction, such work accepts most assumptions that underly the classical representation.

Finally, Simon (1973) distinguishes between well-structured and ill-structured problems. The former, like the classical representation, has initial and goal states, along with a means to recognize goal achievement and a model of state transitions. Despite this similarity, Simon claims that all problem domains defy precise formalization. Instead, each domain lies along a spectrum of well-understood characteristics and weakly understood ones. He further asserts that all problems are more or less ill-structured when examined closely. The actual problem solving occurs with the researcher who establishes the state representations, the solution evaluator, and the range of action models or operators used by a cognitive system. Simon's core critique thus remains an impediment to those of us who study problem solving and believe we are making significant progress.

6. Conclusion

This paper redefined a problem as a state of the world that limits choice in terms of potential goals or available actions. It proposed a formal notation to support this definition and presented an implemented example to illustrate its application. Unlike the traditional notion of a problem, this definition has the benefit of declaratively representing the larger problem-solving context within which problems arise and thus lets cognitive systems reason about the causal factors that make the current situation a problem and the opportunities that exist for solving it. Furthermore, we claim this can be done even while managing pre-existing goals that may be independent of any new one.

However, the task of independently recognizing a problem remains an open question. This paper is not about complete solutions; rather, its focus is to recast the problem we are trying to solve as a community. The challenge I have posed constitutes a significant research issue that borders on many of the scientific questions we already address. Thus, under any theoretical framework or within any implemented cognitive system, the fundamental research question becomes "How can a system recognize, represent, and reason about a new problem given the backdrop of a current set of physical and cognitive activities?" The vision is to develop an alternative to the current overdependence upon human monitoring of the larger situation and subsequent manual intervention. Although this paper does not address the equally important issue of properly circumscribing an agent's capacity to act independently, it looks at an old research question in a new light. Most importantly, the work reexamines underexplored issues central to the understanding of human cognition and problem solving.

Acknowledgements

This work is supported by AFOSR Grant FA2386-17-1-4063, by ONR Grant N00014-18-1-2009, and by NSF Grant 1849131. We thank Cindy Marling, Matt Molineaux, and Mak Roberts for their suggestions, along with Sravya Kondrakunta and Sampath Gogineni for help with the examples.

References

Aha, D. W. (2018). Goal reasoning: Foundations, emerging applications, and prospects. *AI Magazine, 39*, 3–24.

Aha, D. W., Darrell, T., Pazzani, M., Reid, D., Sammut, C., & Stone, P. (Eds.) (2017). *Proceedings of the IJCAI-17 Workshop on Explainable AI*. Melbourne, Australia.

Amarel, S. (1968). On representations of problems of reasoning about actions. In D. Michie (Ed.), *Machine Intelligence 3* (pp. 131–171). Edinburgh: Edinburgh University Press.

Bengfort, B., & Cox, M. T. (2015). Interactive reasoning to solve knowledge goals. *Goal reasoning: Papers from the ACS workshop* (pp. 10–25). Tech. Rep. No. GT-IRIM-CR-2015-001. Atlanta: Georgia Institute of Technology.

Benjamin, M. R., Schmidt, H., Newman P. M., & Leonard, J. J. (2010). Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics, 27,* 834–875.

Bilalić, M., McLeod, P., & Gobet, F. (2008) Expert and “novice” problem solving strategies in chess: Sixty years of citing de Groot (1946). *Thinking & Reasoning, 14*, 395–408.

Bonet, B., & Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence, 129*, 5–33.

Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI Magazine, 11,* 59–71.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4*, 55–81.

Choi, D. (2011). Reactive goal management in a cognitive architecture. *Cognitive Systems Research*, *12*, 293–308.

Cox, M. T. (1994). Machines that forget: Learning from retrieval failure of mis-indexed explanations. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 225–230). Hillsdale, NJ: Lawrence Erlbaum Associates.

Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI Magazine, 28*, 32–45.

Cox, M. T. (2011). Metareasoning, monitoring, and self-explanation. In M. T. Cox & A. Raja, *Metareasoning: Thinking about thinking* (pp. 131–149). Cambridge, MA: MIT Press.

Cox, M. T. (2013). Goal-driven autonomy and question-based problem recognition. *Poster Collection* of the *Second Annual Conference on Advances in Cognitive Systems* (pp. 29–45). Palo Alto, CA: Cognitive Systems Foundation.

Cox, M. T. (2017). A model of planning, action, and interpretation with goal reasoning. *Advances in Cognitive Systems, 5*, 57–76.

Cox, M. T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Muñoz-Avila, H., & Perlis, D. (2016). MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 3712–3718). Palo Alto, CA: AAAI Press.

Cox, M. T., & Dannenhauer, D. (2016). Goal transformation and goal reasoning. *Working Notes of the Fourth Workshop on Goal Reasoning*. New York.

Cox, M. T., Dannenhauer, D., & Kondrakunta, S. (2017). Goal operations for cognitive systems. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 4385–4391). Palo Alto, CA: AAAI Press.

Cox, M. T., & Dannenhauer, Z. A. (2017). Perceptual goal monitors for cognitive agents in changing environments. *Poster Collection of the Fifth Annual Conference on Advances in Cognitive Systems* (pp. 1–16). Palo Alto, CA: Cognitive Systems Foundation.

Cox, M. T., Mohammad, Z., Kondrakunta, S., Gogineni, V. R., Dannenhauer, D. & Larue, O. (2021). Computational metacognition. *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*. Palo Alto, CA: Cognitive Systems Foundation.

Cox, M. T., Oates, T., & Perlis, D. (2011). Toward an integrated metacognitive architecture. *Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium* (pp. 74–81). Tech. Rep. No. FS-11-01. Menlo Park, CA: AAAI Press.

Cox, M. T., & Ram, A. (1999). Introspective multistrategy learning: On the construction of learning strategies. *Artificial Intelligence, 112*, 1–55.

Cox, M. T., & Veloso, M. M. (1998). Goal transformations in continuous planning. *Proceedings of the 1998 AAAI Fall Symposium on Distributed Continual Planning* (pp. 23–30). Menlo Park, CA: AAAI Press.

Dannenhauer, D., Cox, M. T., & Muñoz-Avila, H. (2018). Declarative metacognitive expectations for high-level cognition. *Advances in Cognitive Systems*, *6*, 231–250.

Dannenhauer, D., & Muñoz-Avila, H. (2015). Raising expectations in GDA agents acting in dynamic environments. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence* (pp. 2241–2247). Buenos Aires, Argentina.

Dannenhauer, D., Muñoz-Avila, H., & Cox, M. T. (2020). Expectations for agents with goal-driven autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*, *33*, 867–889.

Dannenhauer, D., Gogineni, V. R., Kondrakunta, S., Mitchell, A., & Cox, M. T. (2022). MIDCA, Version 1.5: User manual and tutorial for the Metacognitive Integrated Dual-Cycle Architecture. Tech. Rep. No. COLAB2-TR-6. Collaboration and Cognition Laboratory, Wright State University, Dayton, OH.

Dinar, M., Danielescu, A., Maclellan, C., Shah, J. J., & Langley, P. (2015). Problem Map: An ontological framework for a computational study of problem formulation in engineering design. *Journal of Computing and Information Science in Engineering, 15*, 031007/1–10.

Ernst, G. W. (1969). Sufficient conditions for the success of GPS. *Journal of the Association for Computing Machinery, 16,* 517–533.

Eyorokon, V. (2018). *Measuring goal similarity using concept, context and task features*. Master's thesis, Wright State University, College of Engineering and Computer Science, Dayton, OH.

Eyorokon, V., Panjala, U., & Cox, M. T. (2017). Case-based goal trajectories for knowledge investigations. *Proceedings of the Thirtieth International FLAIRS Conference* (pp. 477–482). Palo Alto, CA: AAAI Press.

Eyorokon, V., Yalamanchili, P., & Cox, M. T. (2018). Tangent recognition and anomaly pruning to trap off-topic questions in conversational case-based dialogues. *Case-Based Reasoning Research and Development: Proceedings of the Twenty-Sixth International Conference* (pp. 213–228). Berlin: Springer.

Fikes, R. E., & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*, 189–208.

Ghallab, M., Nau, D., & Traverso, P. (2014). The actor's view of automated planning and acting: A position paper. *Artificial Intelligence, 208*, 1–17.

Ghallab, M., Nau, D., & Traverso, P. (2016). *Automated planning and acting*. Cambridge, UK: Cambridge University Press.

Goel, A. K. (1997). Design, analogy, and creativity. *IEEE Expert, 12*, 62–70.

Gogineni, V. R., Kondrakunta, S., Molineaux, M., & Cox, M. (2020). Case-based explanations and goal specific resource estimations. *Proceedings of the Thirty-Third International Conference of the Florida Artificial Intelligence Research Society* (pp. 407–412). Menlo Park, CA: AAAI Press.

Gogineni, V., Kondrakunta, S., Molineaux, M., & Cox, M. T. (2018). Application of case-based explanations to formulate goals in an unpredictable mine clearance domain. *Twenty-sixth International Conference on Case-Based Reasoning: Workshop Proceedings – Case-based Reasoning for the Explanation of Intelligent Systems* (pp. 42–51). ICCBR-18.

Grace, K., & Maher, M. L. (2016). Surprise-triggered reformulation of design goals. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 3726–3732). Palo Alto, CA: AAAI Press.

Gunning, D. (2016). *Explainable Artificial Intelligence (XAI)*. DARPA-BAA-16-53. Washington, DC: The U.S. Department of Defense. Retrieved on April 20, 2020, from URL https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf

Gupta, N., & Nau, D. (1992). On the complexity of blocks-world planning. *Artificial Intelligence, 56,* 223–254.

Hawes, N. (2011). A survey of motivation frameworks for intelligent systems. *Artificial Intelligence, 175*, 1020–1036.

Hsu, F.-H. (2002). *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton: Princeton University Press.

Johnson, B., Roberts, M., Apker, T., & Aha, D. W. (2016). Goal reasoning with information measures. *Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems.* Palo Alto, CA: Cognitive Systems Foundation.

Klenk, M., & Forbus, K. (2009). Analogical model formulation for transfer learning in AP Physics. *Artificial Intelligence, 173,* 1615–1638.

Klenk, M., Molineaux, M., & Aha, D. W. (2013). Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence, 29*, 187–206.

Knoblock, C. A. (1990). Abstracting the Tower of Hanoi. *Working Notes of the AAAI-90 Workshop on Automatic Generation of Approximations and Abstractions* (pp. 13–23). Boston.

Kondrakunta, S., Gogineni, V. R., Brown, D., Molineaux, M., & Cox, M. T. (2019). Problem recognition, explanation and goal formulation. *Proceedings of the Seventh Annual Conference on Advances in Cognitive Systems* (pp. 437–452). Tech. Rep. No. COLAB2-TR-4. Collaboration and Cognition Laboratory, Wright State University, Dayton, OH.

Kondrakunta, S., Gogineni, V. R., Molineaux, M., Muñoz-Avila, H., Oxenham, M., & Cox, M. T. (2018). Toward problem recognition, explanation and goal formulation. *Working Notes of the Sixth Goal Reasoning Workshop.* Stockholm, Sweden.

Kunze, L., Hawes, N., Duckett, T., Hanheide, M., & Krajník, T. (2018). Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robotics and Automation Letters*, *3*, 4023–4030.

Lane, H. C., Core, M. G., van Lent, M., Solomon, S., & Gomboc, D. (2005). Explainable artificial intelligence for training and tutoring. *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology* (pp. 762–764). Amsterdam: IOS.

Langley, P., Choi, D., Barley, M., Meadows, B., & Katz, E. P. (2017). Generating, executing, and monitoring plans with goal-based utilities in continuous domains. *Proceedings of the Fifth Annual Conference on Advances in Cognitive Systems* (pp. 1–12). Palo Alto, CA: Cognitive Systems Foundation.

Langley, P., Pearce, C., Bai, Y., Barley, M., & Worsfold, C. (2016). Variations on a theory of problem solving. *Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems*. Palo Alto, CA: Cognitive Systems Foundation.

Li, P. C. (2009). *Planning the optimal transit for a ship through a mapped minefield*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA.

Maher, M. L., Balachandran, M. B., & Zhang, D. M. (1995). *Case-based reasoning in design*. New York: Psychology Press.

McCarthy, J., & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence, 4*, 463–502.

Muñoz-Avila, H. (2018). Adaptive goal-driven autonomy. *Case-Based Reasoning Research and Development: Proceedings of the Twenty-Sixth International Conference* (pp. 1–10). Berlin: Springer.

Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research, 20*, 379–404.

Newell, A., & Simon, H. A. (1956). *The logic theory machine: A complex information processing system*. Tech. Rep. No. P-868. RAND, Santa Monica, CA.

Newell, A., & Simon, H. A. (1963). GPS, a program that simulates human thought. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought* (pp. 279–293). New York: McGraw.

Newell, A., Shaw, J. C., & Simon, H. A. (1959). Report on a general problem-solving program. *Proceedings of the International Conference on Information Processing* (pp. 256–264). RAND, Santa Monica, CA.

Ohlsson, S. (2012). The problems with problem solving: Reflections on the rise, current status, and possible future of a cognitive research paradigm. *The Journal of Problem Solving, 5*, 101–128.

Paisner, M., Cox, M. T., Maynord, M., & Perlis, D. (2014). Goal-driven autonomy for cognitive systems. *Proceedings of the Thirty-Sixth Annual Conference of the Cognitive Science Society* (pp. 2085–2090). Austin, TX: Cognitive Science Society.

Patra, S., Traverso, P., Ghallab, M., & Nau, D. (2018). Controller synthesis for hierarchical agent interactions. *Proceedings of the Sixth Annual Conference on Advances in Cognitive Systems, Poster Collection*. Palo Alto, CA: Cognitive Systems Foundation.

Pettersson, O. (2005). Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems, 53,* 73–88.

Ram, A. (1990). Decision models: A theory of volitional explanation. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 198–205). Mahwah, NJ: LEA.

Ratner, D., & Warmuth, M. K. (1986). Finding a shortest solution for the n x n extension of the 15–puzzle is intractable. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 168–172). Menlo Park, CA: AAAI Press.

Reid, C. R., Sumpter, D. J. T., & Beekman, M. (2011). Optimization in a natural system: Argentine ants solve the Towers of Hanoi. *Journal of Experimental Biology, 214,* 50–58.

Roberts, M., Vattam, S., Alford, R., Auslander, B., Apker, T., Johnson, B., & Aha, D. W. (2015). Goal reasoning to coordinate robotic teams for disaster relief. *Proceedings of the Third Workshop on Planning and Robotics* (pp. 127–138). Jerusalem, Israel.

Roberts, M., Vattam, S., Alford, R., Auslander, B., Karneeb, J., Molineaux, M., Apker, T., Wilson, M., McMahon, J., & Aha, D. W. (2014). Iterative goal refinement for robotics. *Proceedings of the Second Workshop on Planning and Robotics* (pp. 106–116). Portsmouth, NH.

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Sarathy, V. & Scheutz, M. (2018). MacGyver problems: AI challenges for testing resourcefulness and creativity. *Advances in Cognitive Systems, 6*, 31–44.

Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C., Kass, A., & Riesbeck, C. K. (Eds.) (1994). *Inside case-based explanation*. Hove, England: Psychology Press.

Simon, H. A. (1973). The structure of ill structured problems. *Artificial Intelligence, 4*, 181–201.

Stockmeyer, P. (2013). *The Tower of Hanoi: A bibliography*. Department of Computer Science, The College of William and Mary, Williamsburg, VA. Retrieved on April 20, 2020, from URL https://www.researchgate.net/publication/250056612_The_Tower_of_Hanoi_A_Bibliography

Vattam, S., Helms, M., & Goel, A. K. (2010). A content account of creative analogies in biologically inspired design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 24*, 467–481.

Vattam, S., Klenk, M., Molineaux, M., & Aha, D. W. (2013). Breadth of approaches to goal reasoning: A research survey. *Goal Reasoning: Papers from the ACS workshop* (pp. 111–126). Tech. Rep. No. CS-TR-5029. Department of Computer Science, University of Maryland, College Park.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology, 3*, 1–19.