The Problem with Problems

Michael T. Cox MICHAEL.COX@WRIGHT.EDU
Wright State Research Institute, 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 state and a 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 begin 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 (AI)*. Like humans, a machine should be capable of solving difficult problems if it were to be considered intelligent. To illustrate such behavior, programs such as the *General Problem Solver (GPS)* were given some initial starting state and a goal state, and then they would output a sequence of steps that would achieve the goal if executed (Newell & Simon, 1963). This sequence of steps was considered a solution to the problem. Problem-solving itself was cast as a 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 meet the goal criteria (Amarel, 1968; McCarthy & Hayes, 1969).[1]

Over the years, many types of problems have been studied. Initially, scientists developed algorithms 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 8-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 design configurations for an artifact that meet specific functional requirements and structural constraints (Chandrasekaran, 1990; Dinar, et al. 2016; 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. However, 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 (Ghallab, Nau, & Traverso, 2004). 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 follows with three major sections. The first outlines the classical representation of a problem and enumerates some problems with this construction. The second proposes an alternative problem representation and then challenges our community to take serious the three computational tasks of recognizing a problem, explaining what causes it, and generating a goal to remove the cause and thereafter solve the problem. The third section illustrates how such concepts can be implemented. Related research follows, and I briefly reiterate our challenge in a closing section.

2. The Classical Problem Definition

What is a problem? An initial state, the 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 standard notation. Here we adapt the particular notation used by the automated planning community (e.g., Bonet & Geffner, 2001; Ghallab, Nau, & Traverso, 2004), but variations across AI also are similar to the following.

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 a set of possible states, S, a set of available actions, A, and a successor function, $\gamma: S \times A \rightarrow S$, that returns the next state, s_{i+1}, given a current state, s_i, and an action, $\alpha \in A$.

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

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 π_i and ending at π_j.

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

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.

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

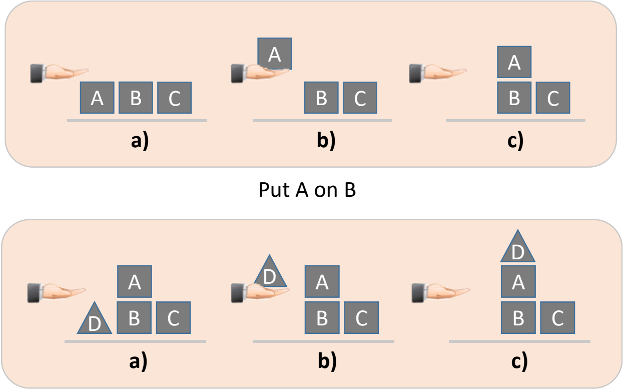
2.2 Problems with the Classical Representation

Significant issues exist with the classical representation of a problem, however. 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 is lacking in the representation. At best, we might say that s_0 may be of lower utility than s_g. But, this choice of representation leaves the problem 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 necessary 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 significant limitations.

1. What is wrong with the initial state is left implicit;
2. The need for the goal or why its achievement is a solution to a problem is opaque and cannot be explained;
3. Problems must be provided by humans rather than inferred by a cognitive system or agent.

Indeed, the representation for problems is often overly simplified in the literature. Consider the blocksworld 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 is the arrangement of three blocks on the table, and the goal state 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 the basis to reason about the nature of the problem or its solution except for minimizing the solution's cost perhaps.



Put A on B

Put D on A *to keep the water out*

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

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; stacking random blocks is not.

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 achieve. A problem is a situation relative to an agent (or agents) with some existing history of intent, 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 immediately or eventually) of a loss in ability to achieve its current or future goals or to select and execute particular 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. Such a situation can occur for example 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 for an agent if new actions are not learned or old actions adapted.

Formal Problem Definition: As opposed 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 it. We examine each of these in turn over the next three subsections. In particular, H_c (defined by equation 15) includes components described in sections 3.1 and 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. A new area of research called *goal reasoning* has attempted to develop cognitive systems with a capability to reason about their own 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; Munoz-Avila 2018; Vattam, Klenk, Molineaux & Aha, 2013). To do so, problems must include a representation of the dynamic context of the agent with respect to its intent. This includes the agent's background knowledge, Bk; an

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

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, $\Delta = \{\delta | \delta: G \rightarrow G\}$, 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. See Dannenhauer & Munoz-Avila (2015; Dannenhauer, Munoz-Avila & Cox, 2020) for detail.

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.

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

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* ($\beta(s, \emptyset) \rightarrow g$) infers a new goal given some state (Cox, 2007; 2013; Paisner, Cox, Maynord & Perlis, 2014); whereas, *goal change* ($\beta(s, g) \rightarrow g'$) 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.

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

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

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

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

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.

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

3.2 Representing the Problem-Solving Context

Finally, the problem representation requires a formalism for the problem-solving process and its unfolding and possibly changing solution to a goal. The reason for this requirement is that new

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

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.

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

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 (see Kunze, Hawes, Duckett, Hanheide & Krajník 2018; Langley, Choi, Barley, Meadows & Katz, 2017; Pettersson 2005) is of the form $\pi_{k+1} \leftarrow \varphi\left(s_i, g_j, \pi_k\right)$. Replanning with goal change would be $\pi_{k+1} \leftarrow \varphi\left(s_i, \beta(s_i, g_j), \pi_k\right)$

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

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

Sometimes the plan changes because of exogenous events in the world or because previous uncertainty became 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.

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

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 a given 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 cognitive systems to reason about the full scope and content of problems including both the intent context (section 3.1) and the overall problem-solving context within which intent is situated.

3.3 The Cognitive Systems Challenge: Inferring the Problem

Finally, we have the prerequisites to specify a problem along with the restriction of choice it represents for an agent. If, for example, 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 less choice. This explanation (or others like it relating 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 a relatively novel situation that poses a new problem for any agent.

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

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

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

In my opinion, the combination of these three tasks constitute the next grand challenge for the AI community and especially for the cognitive systems community. Cognitive systems or intelligent agents, if they are to be genuinely autonomous with a significant measure of independence, should themselves infer the explanation, χ, and the new goal, g' (placing the latter in their agenda, i.e., $\hat{G}_c$). 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 the following 4-tuple adapted from equation (5) on page 4. Neither the goal nor the explanation would be given a priori.

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

Therefore, instead of a sole plan, π, the solution to $\mathcal{P}_c$ would be a 3-tuple of the form $(\chi, g', \pi_{g'})$ where χ is an explanation that justifies a new goal g' and $\pi_{g'}$ is a plan to achieve it. If we prevail over time in the above tasks, it 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). Existing systems cannot fully make such inferences or completely generate such solutions or explanations. However, the following examples demonstrate some basic first steps and show how an implemented system could start to use the representations presented in this paper.

4. Computational Implementation and Example

The *Metacognitive, Integrated, Dual-Cycle Architecture (MIDCA)*[5] (Cox, Alavi, Dannenhauer, Eyorokon, Munoz-Avila, & Perlis, 2016; Cox, Oates, & Perlis, 2011) is an agent model of an intelligent cognitive system. Figure 2 shows details in the cognitive layer of MIDCA as an iterative repetition of processes together with an abstract representation for the metacognitive layer. MIDCA consists of "action-perception" cycles at both the cognitive and metacognitive layers. The output

[5] See http://www.midca-arch.org and, for the code repository, https://github.com/COLAB2/midca.

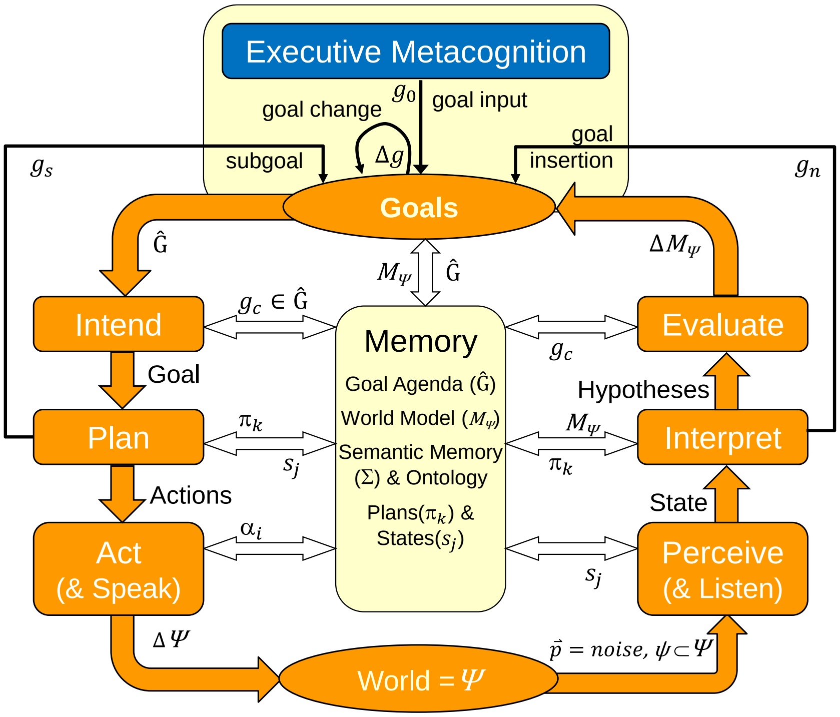


Figure 2. A functional decomposition of the major cognitive processes in MIDCA: The Perceive, Interpret, Intend, Plan, and Act phases. Although abstracted here, these are duplicated at the meta-level. A similar six phase cycle is at the metacognitive layer. Note that Interpret also formulates new goals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

side of each cycle consists of intention, planning, and action execution, whereas the input side consists of perception, interpretation, and goal evaluation. A cycle selects a goal and commits to achieving it (the Intend phase). The agent then creates a plan (Plan phase) to achieve the goal and subsequently executes a planned action (Act) to move the current state toward the goal state. The agent observes changes to the environment (Perceive) resulting from each action, interprets the percepts (Interpret) with respect to the plan, and evaluates the interpretation (Evaluate) with respect to the goal.

4.1 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 the 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 and developed several test scenarios (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. 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. 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 the state of all mines within it is equal to is-cleared.

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

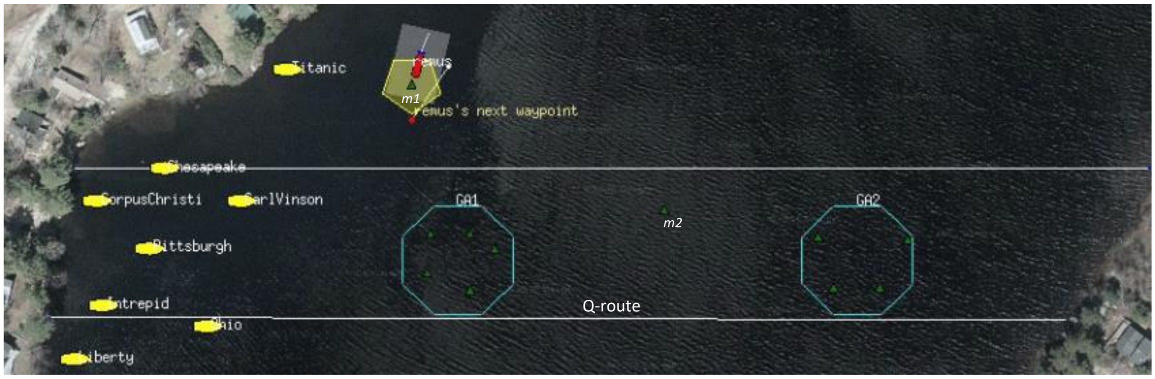


Figure 3. Simulation of the mine clearance domain in Moos IvP. The Q-route extends from the left to the right side of the map. Shipping (shown in yellow) awaits on the left side of the map, and the Remus platform (in red) encounters a mine ($m1$ in the pentagon) as it transits to the $GA1$ location.

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

4.2 Mine Clearance Problems

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 $g_1 =$ cleared$(GA1)$; $g_2 =$ cleared$(GA2)$; and $g_3 =$ 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), and then the Intend phase chooses all three goals by setting the current goal expression, g_c, as a conjunct of the three.

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

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

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

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

4.2.1 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 sees the mine $m1$. Perceive then adds s_2 to the current execution episode, ε_c, and changes the second action from $\text{transit}(p, start, GA1)$ 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 given 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 $\forall l \mid \text{location}(l) \wedge \text{within}(clear\text{-}area, l) \wedge \nexists m | \text{mine}(m) \wedge \text{at-location}(m, \text{l})$; whereas, the observed predicate $\text{at-location}(l, m) \subset s_2$ violates it. Hence, the discrepancy. At this point, MIDCA has established a new episodic problem-solving history that enables it to reason about changes in the future. Now instantiated from equation (17), the current problem is as follows.

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

Interpret 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 problem to friendly shipping.[6] Instead, it generates a goal to avoid the mine itself, adds the goal to the goal agenda, and updates the agenda history to reflect the new status.

$$g_4 \leftarrow \beta(s_2, g_c) = \text{avoided}(m1)$$

$$\hat{G}_c \leftarrow \hat{G}_c \cup g_4$$

$$\hat{G}_h \leftarrow \big(\hat{G}_h \circ \langle \hat{G}_c \rangle\big) = \langle \{g_1 \wedge g_2 \wedge g_3\}, \{g_1 \wedge g_2 \wedge g_3 \wedge g_4\} \rangle$$

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

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

[6] See past work for further details on explanation patterns, their representation, and how they are retrieved, selected and applied (Cox, 2011; Cox & Ram, 1999; Gogineni, et al., 2020; 2018; Kondrakunta, et al., 2019; Ram, 1990; Schank, 1986).

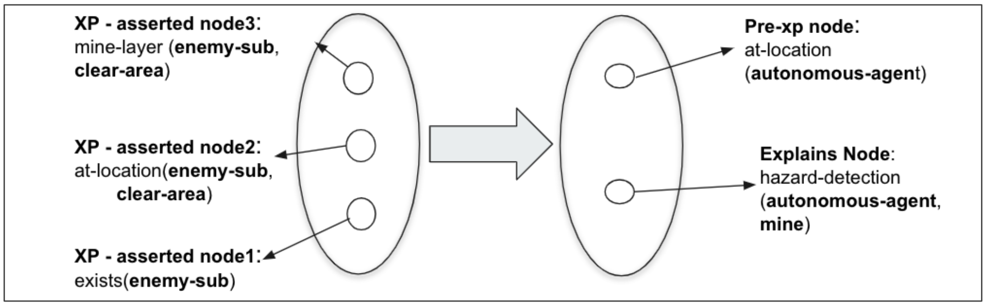

XP - asserted node3:
mine-layer (enemy-sub, clear-area)
XP - asserted node2:
at-location(enemy-sub, clear-area)
XP - asserted node1:
exists(enemy-sub)
Pre-xp node:
at-location (autonomous-agent)
Explains Node:
hazard-detection (autonomous-agent, mine)

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

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 given the expanded current goal and the newly updated plan.

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

$$\vec{\pi} = \langle (g_1 \wedge g_2 \wedge g_3, \pi), (g_1 \wedge g_2 \wedge g_3 \wedge g_4, \pi') \rangle$$

4.2.2 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 $m2$ (see Figure 5). The presence of this mine also represents a discrepancy because no mines were expected in the area between $GA1$ and $GA2$.

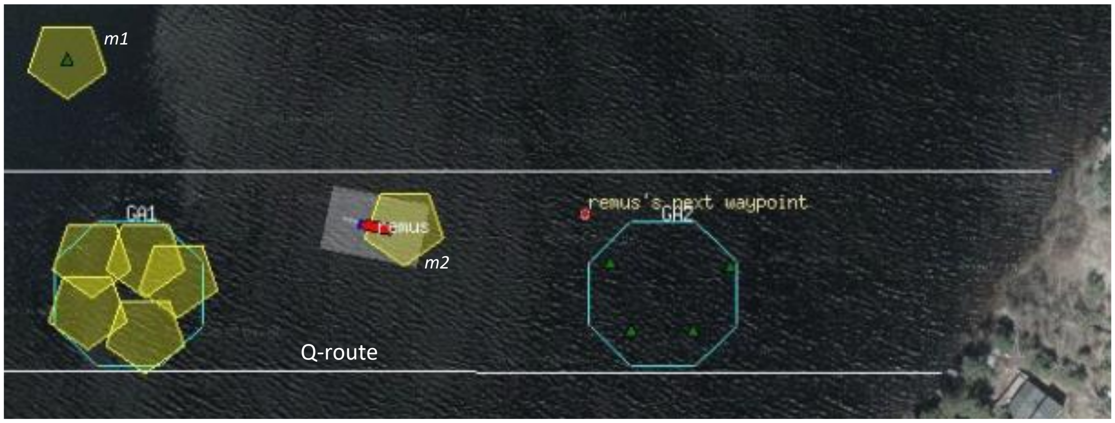


Figure 5. The Remus encounters another surprise in the mine clearance domain. The mine $m2$ is within the Q-route and hence represents a problem to the shipping as they traverse the channel.

At this point, the Perceive phase updates the current execution episode. Like the previous example from section 4.2.1 it replaces α_4 in ε_c with transit$(p, GA1, \text{loc}(m2))$, and it adds s_6.

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

Here, the Interpret phase recognizes another discrepancy given 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 as follows.

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

Like before, it 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 added to clear m2, and it is added to the agenda. Subsequently, MIDCA updates the agenda history.

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

As before, 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 shipping can safely traverse the channel to deliver supplied 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, Evaluate checks that the current state entails the goal state and clears the agenda.

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 that represents much of the goal reasoning process with the function β, Roberts proposes a detailed lifecycle consisting 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. However, problems are represented classically.

Both goal reasoning and *explainable AI* (Aha, et al., 2017; Cox, 2011, 1994; Gunning, 2016; Lane, Core, van Lent, Solomon, & Gomboc, 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 open up to 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 all interact (Ghallab, Nau, & Traverso, 2014; 2016). Yet, the representation of a problem remains much the same as it has for some sixty years (c.f., Patra, Traverso, Ghallab, & Nau, 2018).

Many researchers in the cognitive systems community have proposed various problem representations and specified numerous problem-solving mechanisms. But, most of these assume some variation on the basic representation of an initial state and goal state given by a human or otherwise input to the system. Although progress has been made, the research focus tends to be upon developing methods to produce solutions. Problems are closer to puzzles in many of the cases found in the literature. For example, Klenk and Forbus (2009) developed an analogical method that solves AP Physics type problems. These problems consist of a set of given facts and a goal query that seeks a particular value for some quantity. Langley, Pearce, Bai, Barley, and Worsfold (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, Choi, Barley, Meadows & Katz, 2017) do recognize that goals are not simple predicate states. Instead, they differ widely according to utility and other attributes, and problem solutions need to be monitored given dynamic environments.

Finally, 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) but with a novel twist as shown below in equation (18).

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

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 same limitations as those enumerated in section 2.2 , Sarathy and Scheutz also represent the evolving context of the agent (shown in equation 19).

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

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 of the assumptions underlying the classical representation.

6. Conclusion

This paper redefines a problem as a state of the world that limits choice in terms of potential goals or available actions and presents a formal notation to support this definition and an implemented example to illustrate its application. I argue that, unlike the traditional definition of a problem, this new definition has the benefit of declaratively representing the larger problem-solving context within which problems arise and thus allows cognitive systems to reason about the causal factors that make the current situation a problem along with the opportunities that exist for solving it, even while managing pre-existing goals that may be independent of any new one. Given this position, I have challenged the community to consider how computational systems can autonomously recognize problems on their own and form their own response.

I do not claim to have solved the task of independently recognizing a problem. This paper is not about solutions; rather, its focus is about recasting the problem itself we are trying to solve as a community. The challenge I pose constitutes a significant research issue that borders on many of the scientific questions we already address. So, under any theoretical framework or within any implemented cognitive system, the fundamental research question becomes "How can a system recognize, represent and then 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 an over-dependence 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 does look at an old research question in a new light. Most importantly, this work re-examines underexplored issues central to fully understanding 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. I thank Cindy Marling, Matt Molineaux, Mak Roberts and the anonymous reviewers for their clear comments and suggestions. I thank 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*(2), 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 (XAI)*. International Joint Conference on Artificial Intelligence.

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. D. W. Aha (Ed.), *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*(6): 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*:4, 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*(4). 59-71.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4*(1), 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. In A. Ram & K. Eiselt (Eds.), *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*(1), 32-45.

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

Cox, M. T. (2013). Goal-driven autonomy and question-based problem recognition. In *Second Annual Conference on Advances in Cognitive Systems 2013, Poster Collection* (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., Munoz-Avila, H., & Perlis, D. (2016). MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Vol. 5 (pp. 3712-3718). Palo Alto, CA: AAAI Press.

Cox, M. T., & Dannenhauer, D. (2016). Goal transformation and goal reasoning. In *Proceedings of the 4th Workshop on Goal Reasoning*. New York, IJCAI-16.

Cox, M. T., Dannenhauer, D., & Kondrakunta, S. (2017). Goal operations for cognitive systems. In *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. In *Proceedings of the Fifth Annual Conference on Advances in Cognitive Systems, Poster Collection* (pp. 1-16). Palo Alto, CA: Cognitive Systems Foundation.

Cox, M. T., Oates, T., & Perlis, D. (2011). Toward an integrated metacognitive architecture. In P. Langley (Ed.), *Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium* (pp. 74-81). Technical Report 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. In M. desJardins (Ed.), *Proceedings of the 1998 AAAI Fall Symposium on Distributed Continual Planning* (pp. 23-30). Menlo Park, CA: AAAI Press / MIT Press.

Dannenhauer, D., & Munoz-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., Munoz-Avila, H., & Cox, M. T. (2020). Expectations for agents with goal-driven autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*. DOI: 10.1080/0952813X.2020.1789755

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*(4):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.

Eyorokon, V., Panjala, U., & Cox, M. T. (2017). Case-based goal trajectories for knowledge investigations. V. Rus & Z. Markov (Eds.), *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. In M. T. Cox, P. Funk, & S. Begum (Eds.), *Case-Based Reasoning Research and Development: Proceedings of the 26th International Conference* (pp.213-228). Berlin: Springer.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Francisco: Morgan Kaufmann.

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: Cambridge University Press.

Goel, A. K. (1997). Design, analogy, and creativity. *IEEE Expert 12*(3), 62-70.

Gogineni, V. R., Kondrakunta, S., Molineaux, M., & Cox, M. T. (2020). Case-based explanations and goal specific resource estimations. In *Proceedings of the 33rd 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. In M. Minor (Ed.), *26th International Conference on Case-Based Reasoning: Workshop Proceedings - Case-based Reasoning for the Explanation of Intelligent Systems* (pp. 42-51). ICCBR-18.

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. *AIJ, 56,* 223–254.

Hawes, N. (2011). A survey of motivation frameworks for intelligent systems. *Artificial Intelligence, 175*(5-6), 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. (2016). Goal reasoning with information measures. In *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*(2), 187-206.

Knoblock, C. A. (1990). Abstracting the Tower of Hanoi. In *Working Notes of 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. In M. T. Cox (Ed.), *Proceedings of the Seventh Annual Conference on Advances in Cognitive Systems* (pp. 437-452). Tech. Rep. No. COLAB2-TR-4. Dayton, OH: Wright State University, Collaboration and Cognition Laboratory.

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

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(4), 4023-4030.

Lane, H. C., Core, M. G., van Lent, M., Solomon, S., & Gomboc, D. (2005). Explainable artificial intelligence for training and tutoring. In C.-K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.), *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. In *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.

Munoz-Avail, H. (2018). Adaptive goal-driven autonomy. In M. T. Cox, P. Funk, & S. Begum (Eds.), *Case-Based Reasoning Research and Development: Proceedings of the 26th International Conference* (pp. 1-10). Berlin: Springer.

Newell, A., & Simon, H. A. (1956). *The logic theory machine: A complex information processing system*. Technical Report P-868. Santa Monica, CA: RAND.

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.

Paisner, M., Cox, M. T., Maynord, M., & Perlis, D. (2014). Goal-driven autonomy for cognitive systems. In *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. In *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. In K. Forbus, D. Gentner & T. Regier (Eds.), *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 × n extension of the 15-puzzle is intractable. *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. (2015). Goal reasoning to coordinate robotic teams for disaster relief. In A. Finzi, F. Ingrand, & A. Orlandini (Eds.) *Planning and Robotics: Papers from the ICAPS Workshop*. Jerusalem, Israel: AAAI Press.

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. In *ICAPS 2014*.

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*. LEA.

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. (2013). Breadth of approaches to goal reasoning: A research survey. In D. W. Aha, M. T. Cox, & H. Munoz-Avila (Eds.), *Goal Reasoning: Papers from the ACS workshop* (pp. 111-126). Tech. Rep. No. CS-TR-5029. College Park, MD: University of Maryland, Department of Computer Science.

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