Extending an Embodied Cognitive Architecture with Spatial Representation and Reasoning

Pat Langley[1], Edward P. Katz[2]

[1]Center for Design Research, Stanford University, Stanford, CA 94305 USA
[2]Stanford Intelligent Systems Laboratory, Stanford University, Stanford, CA 94305 USA

Abstract
In this paper, we review the PUG cognitive architecture for embodied agents, which posits grounded and graded concepts, integration of symbolic planning with continuous control, and mental simulation to evaluate candidate trajectories. We also describe extensions that will let the framework represent spatial knowledge about places and use this content to move an agent through its environment.

Keywords
Cognitive architectures, Spatial cognition, Planning and control

Introduction

Research on cognitive architectures aims to develop unified theories of intelligent systems (Langley, Laird, & Rogers, 2009; Langley, 2017). This paradigm incorporates many ideas from cognitive psychology and focuses on agents that operate over time. Common assumptions include: short-term memories are distinct from long-term stores; both memories contain modular symbolic structures; relational pattern matching accesses long-term content; processing occurs in recognize-act cycles; and cognition dynamically composes mental structures as needed.

The architecture community has reported many successes, ranging from demonstrations of human-level performance on complex tasks, such as piloting simulated jet fighters (Jones et al., 1999), to explanations of key psychological phenomena, such as driver behavior (Salvucci, 2006). However, an emphasis on generality has led researchers to avoid making commitments about the representation and processing of spatial content. The field needs additional work that addresses this critical omission and here we report some initial progress in this direction.

The PUG Architecture for Embodied Agents

In previous research, we have developed PUG, a cognitive architecture for embodied agents (Langley et al., 2016). The framework incorporates ideas from classical architectures, such as separating long-term from short-term memory and encoding their content as modular mental structures. A distinguishing feature is that it combines symbolic and numeric representations

Third International Workshop on Human-Like Computing, September 28–30, 2022, Windsor Great Park, UK
EMAIL: langley@stanford.edu (P. Langley); edpkatz@stanford.edu (E. P. Katz)
URL: http://www.isle.org/~langley/ (P. Langley)
ORCID: 0000-0001-5260-7048 (P. Langley)

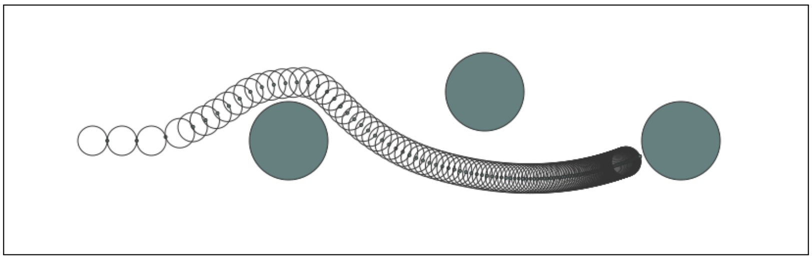

Figure 1: A simulated scenario in which a PUG-controlled robot (small circle) uses skills to approach and turn toward a target object while avoiding two obstacles along the way. The system relies on additional skills for turning to the left and right around obstructions. The trace shows the robot's position and orientation at equal time intervals. Skill execution continues until the the robot is close enough.

and processing to support a unified account of discrete planning and continuous control. Key theoretical commitments of the PUG architecture include:

- Symbolic *concepts* are *grounded* in constituent physical objects and their associated numeric attributes. *Beliefs* are always instances of these defined concepts.
- Symbolic concepts are *graded*, in that they match to different degrees based on constituent objects' attributes. Thus, derived beliefs have greater or lesser veracity.
- Symbolic *skills* incorporate equations for control attributes that are functions of *mismatch* between associated target concepts and the environment.
- *Execution* involves the iterative, reactive use of skill instances to compute values for skills' control attributes based on the degree of their target concepts' mismatches.
- To determine values for control attributes, execution takes the *vector sum* of results from active skill instances, much as in potential field approaches to continuous control.
- *Processes* specify equations that predict changes to state attributes based on state and control attributes. The agent can affect the environment only indirectly through such processes.
- *Mental simulation* uses skills and processes to predict trajectories in the state space over time as a function of anticipated situations and the agent's responses.
- *Motion planning* involves search through a space of sets of skill instances that, taken together, will achieve high-utility goals (desired beliefs) and avoid low-utility ones.
- *Task planning* involves search through a space of sequences of possible motion plans that, again, will achieve high-utility goals and avoid low-utility ones.
- *Trajectories* produced by mental simulation are used to evaluate both candidate motion plans and task plans in terms of their summed expected utilities over time.

PUG provides a high-level programming language, which we have used to create simple robot agents that operate in simulated two-dimensional environments. Demonstrations have included tasks in which robots must move to static objects or pursue mobile ones, in both cases avoiding obstacles along their paths, as depicted in Figure 1. We have also developed an extended framework, PUG/X, that integrates generation, execution, and monitoring of task and motion plans to let agents detect and respond to unexpected events (Langley et al., 2017).

Adding Spatial Cognition to the PUG Architecture

The architecture supports agents that operate in dynamic physical environments and it unifies symbolic approaches to planning with continuous approaches to control, but, like other cognitive architectures, it does not incorporate strong theoretical commitments about how to encode or reason over spatial content. In response, we have devised extensions to the framework that will address these representational and processing limitations. These include a number of generic claims, which we illustrate with examples from the diagram in Figure 2:

- Agents encode their spatial relations to nearby objects in the environment in terms of *continuous attributes* that are represented in *egocentric polar coordinates*. In the diagram, the robot R perceives its distance and angle to the object X and to the object Y.
- A *place* P is a *virtual object* defined by a set of other objects, either observable or defined, along with distances to them. In the figure, place P is defined by its distances to the reference objects X and Y. Two landmarks suffice because the agent's pose provides additional constraints.
- An *instance* I of the place P is an inferred belief that includes the place name along with the derived distance and angle to this virtual object. In the diagram, the robot R's distances to X and Y are 6.0 and 10.0 units, whereas its angles are 85.0 and 60.0 degrees, respectively.
- To describe an instance I of place P on a given time step, the inference process calculates the distance (10.57 units) and angle (7.17 degrees) to P from the distances and angles to X and Y using the equations given below. The two values change as the agent moves relative to P.
- Inference uses the description of instance I to calculate the degree of match for relations like *(at Robot P)* and *(facing Robot P)*. E.g., the match function for *at* might be $1 - d/5$ if $d < 5$ and zero otherwise, where d is the distance d to P. The match score can also vary over time.
- Skill execution uses degree of mismatch for relations like *(at Robot P)* and *(facing Robot P)* as error signals for skill instances that let it move and turn toward P by setting relevant control values. This process continues until the beliefs match to a sufficiently high degree.
- A *map* is a collection of places that include not only distances to their constituent objects but also distances to *other places*. E.g., the definition of P might include the distance to another virtual object, Q. Such connections specify a *topological network* of distinct places.
- Skill execution and motion planning can use a place definition Q that references place P to move the agent from a position at P to one that is near Q. As before, it infers the agent's distances and angles to Q in order to calculate control values and thus guide movement.

These theoretical postulates differ markedly from those adopted by many research efforts on mobile robotics, where it is common to encode space as discretized, world-centric grids rather than in continuous, ecocentric, object-oriented terms.

We should clarify a few details about the extended architecture. First, each place is defined as a virtual object in a Prolog-like rule, but the predicate is not the place name itself. Rather, it is the same unary predicate (e.g., *object*) that identifies perceived entities. The place name is a constant argument of this predicate, while the body includes the reference objects and their fixed distances. On perceiving these landmarks, the agent infers the presence of the virtual object, its distance, and its angle. This is just another belief, so PUG can infer other beliefs from it, like *(at Robot P)*, and their match scores, then use them to drive skill execution and planning.

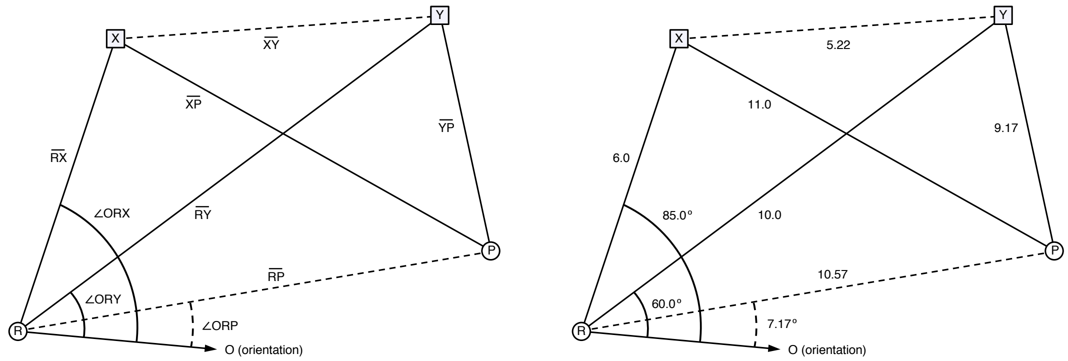

Figure 2: A scenario in which the extended PUG architecture computes the robot R's distance and angle to a place (the virtual object P) based on its perceived distances and angles to two reference objects (X and Y). The left diagram shows symbolic notations; the right gives numeric values (not to scale). Solid lines and angles are perceived or given in P's place definition; dashed lines are derived from them.

Second, the rule that defines each place P includes equations which specify the agent's distance and angle to P as a function of its distances and angles to two reference objects and to constants for the latter's distances to P. Using labels from Figure 2, the distance equation is

$$\overline{RP} = SAS(\overline{RY}, |SSS(\overline{XY}, \overline{XP}, \overline{YP}) - SSS(\overline{XY}, \overline{RX}, \overline{RY})|, \overline{YP}) \ ,$$

where $\overline{XY} = SAS(\overline{RX}, |\angle ORX - \angle ORY|, \overline{RY})$ and the equation for the angle is

$$\angle ORP = |\angle ORY - SSS(\overline{RY}, \overline{YP}, \overline{RP})| \ .$$

Here the function $SSS(a, b, c) = acos([(a^2 + c^2) - b^2]/[2 \cdot a \cdot c])$ and the other function $SAS(b, A, c) = sqrt([b^2 + c^2] - [2 \cdot b \cdot c \cdot cos(A)])$, based on the standard 'Side Side Side' and 'Side Angle Side' formulae from trigonometry. These let the rule compute the agent's distance and angle to the virtual object from its egocentric relations to the reference objects. This analysis assumes exactly two such landmarks, but larger numbers would be useful with noisy sensors.

Discussion

In this paper, we reviewed PUG, an architecture for embodied agents that combines symbolic concepts with numeric descriptors and that joins discrete planning with continuous control. We also outlined extensions that should let the framework represent and use spatial knowledge. The design adapts ideas on feedback control, potential fields (Khatib, 1986), egocentric encodings (Kuipers & Byun, 1991; Yamauchi & Langley, 1997; Yeap, 2011), and topological networks (Remolina & Kuipers, 2004; Thrun, 1998), but it embeds them in a unified agent architecture.

We are incorporating these ideas about spatial cognition into the PUG architecture and we plan to demonstrate them in the same simulated robotic environment as our earlier experiments (Langley & Katz, 2021). We also intend to elaborate the framework to handle more complex shapes and use formalisms like RCC8 (Cohn et al., 1997) to reason about their relations. Experiments will test PUG on agent localization, inter-place navigation, and other tasks that require reasoning about places rather than only about individual, perceivable objects.

Acknowledgments

The research reported here was supported by Grant No. FA9550-20-1-0130 from the US Air Force Office of Scientific Research, which is not responsible for its contents. We thank Mohan Sridharan and the reviewers for constructive comments. A briefer version of this material appeared in the Report of the Dagstuhl Seminar on Representing and Solving Spatial Problems.

References

Cohn, A. G., Bennett, B., Gooday, J., & Gotts, M. M. (1997). Qualitative spatial representation and reasoning with the Region Connection Calculus. *GeoInformatica*, *1*, 275–316.

Jones, R. M., Laird, J. E., Nielsen P. E., Coulter, K., Kenny, P., & Koss, F. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine*, *20*, 27–42.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In I. J. Cox & G. T. Wilfong (Eds.), *Autonomous robot vehicles*. New York: Springer.

Kuipers, B. J., & Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, *8*, 47–63.

Langley, P. (2017). Progress and challenges in research on cognitive architectures. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 4870–4876). San Francisco: AAAI Press.

Langley, P., Barley, M., Meadows, B., Choi, D., & Katz, E. P. (2016). Goals, utilities, and mental simulation in continuous planning. *Proceedings of the Fourth Annual Conference on Cognitive Systems*. Evanston, IL.

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 Cognitive Systems*. Troy, NY.

Langley, P., & Katz, E. P. (2021). *A cognitive systems approach to continuous control*. Unpublished manuscript, Center for Design Research, Stanford University, Stanford, CA.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141–160.

Remolina, E., & Kuipers, B. (2004). Towards a general theory of topological maps. *Artificial Intelligence*, *152*, 47–104.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, *48*, 362–380.

Schlieder, C. (1993). Representing visible locations for qualitative navigation. In N. Piera Carreté & M. G. Singh (Eds.), *Qualitative reasoning and decision technologies*, 523–532. Barcelona: CIMNE.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, *99*, 21–71.

Yamauchi, B., & Langley, P. (1997). Place recognition in dynamic environments. *Journal of Robotic Systems*, *14*, 107–120.

Yeap, W. K. (2011). How Albot0 finds its way home: A novel approach to cognitive mapping using robots. *Topics in Cognitive Science*, *3*, 707–721.