The Object Schema Model and Situational Context

Kai-yuh Hsiao and **Soroush Vosoughi**
eepness at mit dot edu, soroush at mit dot edu
MIT Media Laboratory
Cambridge, MA

Abstract

On our Trisk robotic manipulation platform, we have implemented a conceptual representation based around *object schemas*, which organize all perceptions, motor actions, planning structures, and natural language use according to the objects that they are "about." This representation provides a clear point of interaction between the discrete domains of task-level planning and language use, and the continuous dynamic domain of sensorimotor interaction. In this paper we briefly summarize our model and our results, implemented on a robotic manipulation platform in a simple tabletop domain. We conclude by proposing to use situational context to enable scalability to more complex domains.

Introduction

In (Hsiao et al. 2008), we explore *object schemas*, a model of representation based on earlier theories developed in (Roy 2005). The model provides a single level of representation that unifies responsive motor action, visual and tactile perception, task-level planning, and language use in a robotic platform. We accomplish this by assuming that perceptual, action, and planning processes are "about" objects in the real world, and organizing the processes into object schema structures accordingly.

In this paper, we summarize our object schema model, and advocate our approach as a plausible means of achieving end-to-end integration in a robotic system, from continuous sensorimotor activity to task-level planning. Finally, we propose an extension to improve the scalability of our approach.

Basics of Object Schemas

Here we describe the primary elements of our system, resulting benefits for task- and motion-level integration, and examples of implemented behaviors. Implementation has been performed on our robotic manipulator, Trisk, in a simple tabletop domain with easily identifiable and manipulable objects. Videos of the provided examples are available online at *http://www.media.mit.edu/~eepness/trisk.html*.

Interaction processes Each *interaction process* is a computational routine that, when executed, either coordinates motor outputs, accumulates sensory input (e.g. tracking a visual blob), or manipulates internal data. Interaction processes run concurrently, and respond in real-time to changes in the perceived world.

Benefit. The use of concurrent responsive processes is a key part of *behavior-based robotics*, among whose earliest proponents was Brooks (1986). Responsiveness is critical for successful completion of motor actions in an incompletely-sensed dynamic environment.

Object schemas The interaction processes are grouped into object schemas, each representing an object that each process is ostensibly "about." A *coordination process* monitors other processes in its object schema, ensuring consistency among the sensory inputs and forcing reorganization as required.

Example. Trisk is told to move the yellow ball, but accidentally grasps the green block. Visually, the green block moves, but Trisk believes it is touching the yellow ball. Sensing the discrepancy, a coordination process decouples the touch-related process from the yellow ball's object schema, to restore coherence.

Benefit. Persistence of objects is vital for successfully connecting continuous sensorimotor interactions with task-level planning. Task-level planning assumes the existence of discrete objects, but this can only hold if the continuous processes that maintain the object representations are coherent.

Interaction histories Each executing interaction process is stored with its *interaction history*, describing its history of successes and failures and associated sensory inputs, while each pre-execution interaction process is stored with a *predicted interaction history*, which predicts successes, failures, and sensory inputs based on past experiences with similar processes.

Example. Trisk attempts to grasp a red block, but it misses and its arm collides with the block instead. The collision is sensed and triggers a reassessment of the block's physical location, i.e., the grasp action is now predicted to succeed only if adjusted to the collision location. The grasp is retried and completed.

Benefit. By organizing sensory inputs in terms of objects wherever possible, perceived changes in the world are immediately propagated to the task planning level,

to alter expectations and trigger replanning.

Affordances Because each interaction process is 1) associated with an object schema and 2) stored alongside its expectations, each process represents an *affordance* (as defined by Gibson (1979)) of the object represented by the schema. For instance, if a grasping action is expected to complete successfully, then an affordance of the associated object is that it can be grasped.

Example. The vision system perceives that a ball is visually within the confines of a cup, but in order to verify that the functional relation "in" holds between the ball and the cup, the robot lifts the cup, moves it, and tilts it to see how the ball behaves. If the ball stays visually within the cup's boundaries, then the ball's expectations can be revised to denote that it *affords* being moved via the cup. This sequence is also a satisfying example of Garrod et al.'s assertion (1999) that locative prepositions such as "in" and "on" have both geometric (visual) and functional (manipulable) extent; in this example the robot is making use of both.

Benefit. Storing affordances with their objects maintains a connection between task-level planning, which manipulates discrete affordances, and the continuous nature of the motor actions needed to implement those affordances. It also provides a direct connection for human-language terms like "liftable" and "graspable," and this example shows task-level constraints ("in") being derived from sensorimotor primitives (seeing the ball remain in the cup during manipulation).

Task-level planning The affordances of an object are used in turn by the planning system to decide how best to achieve a user-provided goal. The planning system consists of two main components, top-level motivations and a plan hierarchy. The top-level motivations – safety (avoiding collisions), social (serving the user), and curiosity (attempting to move objects to explore affordances) – are each assigned a *priority score* based on recent sensory inputs, and compete with each other (as in the *action selection* problem (Tyrrell 1993)) to control the robot accordingly. The plan hierarchy is a tree of conditions and actions constructed by a STRIPS-like planner (Nilsson 1984) augmented with affordance predictions. Planning starts from the winning top-level motivation, and selected action processes begin execution, coordinating the robot's motor actions according to built-in program loops. Replanning occurs when new information invalidates the predicted affordances.

Example. Trisk is told "Group the green block and the red apple." For the one-armed Trisk, this could be accomplished by moving either object towards the other one. Trisk reaches for the red apple, but is suddenly told "The red apple is heavy." Heavy objects afford difficult grasping, forcing the planner to replan. Trisk stops and reaches for the green block instead. This is illustrated in Figure 1.

Benefit. This example illustrates task-level replanning triggered by a change in expectation of a motor action. This multi-level integration is a unique benefit of our system.

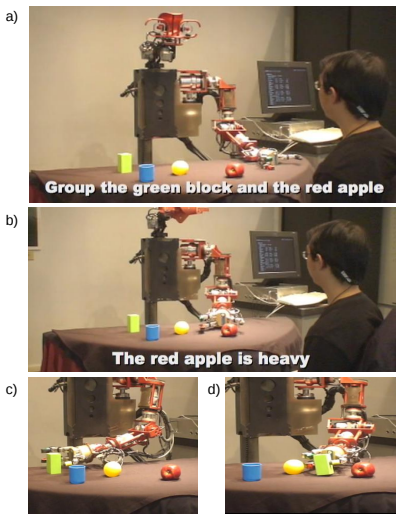

Figure 1: Trisk responds to spoken inputs that require replanning due to prior sensorimotor experience.

Affordance learning When the curiosity motivation is engaged, the robot grasps and moves objects in order to explore their affordances. Object attributes (color, shape, size, weight) are recorded alongside a history of action successes and failures. These records allow subsequent affordance predictions, which in turn influence the planning system.

Example. Trisk lifts a heavy lantern battery. As its fingers begin to sense that the object is heavy (based on force sensors), the battery slips out of its grasp. The predicted affordances for grasping of heavy objects, square objects, and gray objects will subsequently be considered less feasible.

Benefit. The learning in this example shows how events at the sensorimotor level can propagate to alter task-level planning.

Language interaction When the user speaks, the speech recognizer output triggers construction of a tree structure that influences the planning system and the predicted expectations. The language system can handle queries ("Describe the block"), directives ("Move the block to the right"), assertives ("The red apple is heavy"), and correctives ("... No, the green ball").

Example. Trisk is told "Move the green block to the left." As Trisk lifts the block and begins moving, it is then told "No... to the right." The correction replaces the plan structure at the appropriate level, and Trisk places the block to the right of its starting position.

Benefit. Natural language interaction is predicated on consistent, sustained discrete representations of a continuous world.

Summary. By organizing all motor actions, perceptions, task planning, and language use according to referent objects, our object schema model provides a single unified level of interaction for both discrete and continuous types of processing essential for real-world

robot operation.

Adding Situational Context to Object Schemas

While the value of our approach has been demonstrated in a simple tabletop domain with a limited set of physical objects, the key challenge at this point is to design for scalability. Specifically, the representations must be able to handle objects and actions of higher complexity, and the system must remain computationally tractable while dealing with higher task complexity.

Increasing the sensorimotor complexity of the system is a matter of improved perceptual processing and motor ability, which falls outside the scope of our core model. We choose instead to focus on model scalability, to prepare for increases in sensorimotor complexity. The primary limitations of the current implementation with respect to scalability are:

- Each newly instantiated object schema must include processes for all actions that could be taken towards the object, in order to represent the affordances. However, with no way to limit the number of action processes thus generated, the planning system rapidly ends up with too many actions to consider at each step.
- Each affordance's feasibility is constantly being re-evaluated in light of new perceptual inputs. This also requires extraneous computational power, especially when too many affordances are being generated.

Our proposed solution is to limit the instantiated affordances to a few actions that have previously been relevant in a similar context, and to gradually explore more actions if the goals of the system cannot be achieved. This approach is psychologically inspired by the human tendency towards *functional fixedness* – for instance, human subjects told to connect two ropes that are out-of-reach of each other (Maier 1931) do not think to use pliers as a weight to swing one rope towards the other, because pliers are typically used for grasping, not as a weight.

For our purposes, functional fixedness provides a clear way to rein in computational complexity. Humans tend not to consider actions that are uncommon within the current context, where the context might include 1) the current goal, 2) objects (instruments and patients) under consideration, 3) the agents involved, or 4) the overall situation or location.

We suggest exploring two tiers of complexity for instantiating and evaluating affordances:

1. Just as our affordance predictions are shaped by counting prior successes and failures with respect to object attributes, we can similarly measure the relevance of actions to a particular set of contextual attributes. Actions which are never taken towards the current goal or towards the current objects can be ignored.

2. The counting method is computationally cheap but cannot account for dependencies between multiple concepts (e.g., actions to be considered for a particular goal, but only with a specific object). Instead, we can use connectionist principles, such as a spreading activation network, to train these relevance relations and account for dependencies between contextual elements.

A spreading activation network is more complex to process than simple counting. Given this tradeoff it is possible to use both, by instantiating actions based only on counting, and falling back to the spreading activation network if the planning system fails, lowering the threshold for activation until a plan succeeds.

Regardless of implementation, we believe our next step is to add context sensitivity to our model in order to limit the affordances under consideration. This will allow us to add more actions (e.g., 'push', 'slide') to our current system, and handle more complex object concepts (e.g., tool use, or object merging and splitting), without becoming too computationally intensive.

Conclusion

We have described the basic elements and benefits of our object schema model, which provides a unifying representation in which visual and tactile perception, continuous motor action, discrete task planning, and natural language use are all viewed in terms of the objects that they are "about." The model has been implemented on a robotic manipulation platform, in a tabletop domain with simple objects. Furthermore, we have proposed an extension to our model to limit its computational complexity, by taking elements of situational context into account, in the hopes that this will assist in scaling it to more complex domains.

Acknowledgments

Deb Roy, our advisor, made pretty much every facet of this work possible. Thanks also to our multitude of contributors who helped with the Trisk system.

References

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2:14–23.

Garrod, S.; Ferrier, G.; and Campbell, S. 1999. In and on: investigating the functional geometry of spatial prepositions. *Cognition* 72:167–189.

Gibson, J. J. 1979. *The Ecological Approach to Visual Perception*. Erlbaum.

Hsiao, K.; Tellex, S.; Vosoughi, S.; Kubat, R.; and Roy, D. 2008. Object schemas for grounding language in a responsive robot. *Connection Science* 20(4):253–276.

Maier, N. 1931. Reasoning in humans II: The solution of a problem and its appearance in consciousness. *Journal of Comparative Psychology* 12:181–194.

Nilsson, N. J. 1984. Shakey the robot. Technical Report 323, AI Center, SRI International.

Roy, D. 2005. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence* 167(1-2):170–205.

Tyrrell, T. 1993. The use of hierarchies for action selection. *Adaptive Behavior* 1(4):387–420.