Research Challenges for Autonomous Cognitive Systems

Pat Langley
Institute for the Study of
Learning and Expertise

ONR Science of Autonomy Meeting
August 10-13, 2015

The Nature of Autonomy

Truly autonomous agents could aid the US military, and other facets of our society, in many ways.

We say that a computational system is *autonomous* when it:

- Operates in some environment *over time*
- Selects *which actions to carry out*
- Decides *how to allocate its resources*
- Determines *which goals to pursue*

In general, an agent is autonomous if it *adapts* to its situation, not only at the physical but also at the *cognitive* level.

Humans exhibit substantial cognitive autonomy, and we want to reproduce their ability in machines.

Previous Work on Cognitive Autonomy

Established paradigms already exhibit some important aspects of autonomous cognitive systems:

- Teleoreactive control systems (Nilsson, 1994; Parker, 1995)
- AI planning systems (Ghallab, Nau, & Traverso, 2004)
- Cognitive architectures (Langley, Laird, & Rogers, 2009)

Each framework has led to many successful systems that exhibit compelling forms of autonomy.

But each paradigm also makes strong assumptions that limit the adaptability of cognitive systems they support.

Four Challenges for Cognitive Autonomy

In this talk, I discuss four facets of agent autonomy that have received little attention:

- Origin of top-level goals
- Cognitive attention / motivation
- Creating and using tools
- Adaptive planning / execution

I also report some progress on each challenge, but we need far more work on all of them.

A Brief Review of ICARUS

ICARUS (Langley, Choi, & Rogers, 2009) is an architecture for intelligent agents that:

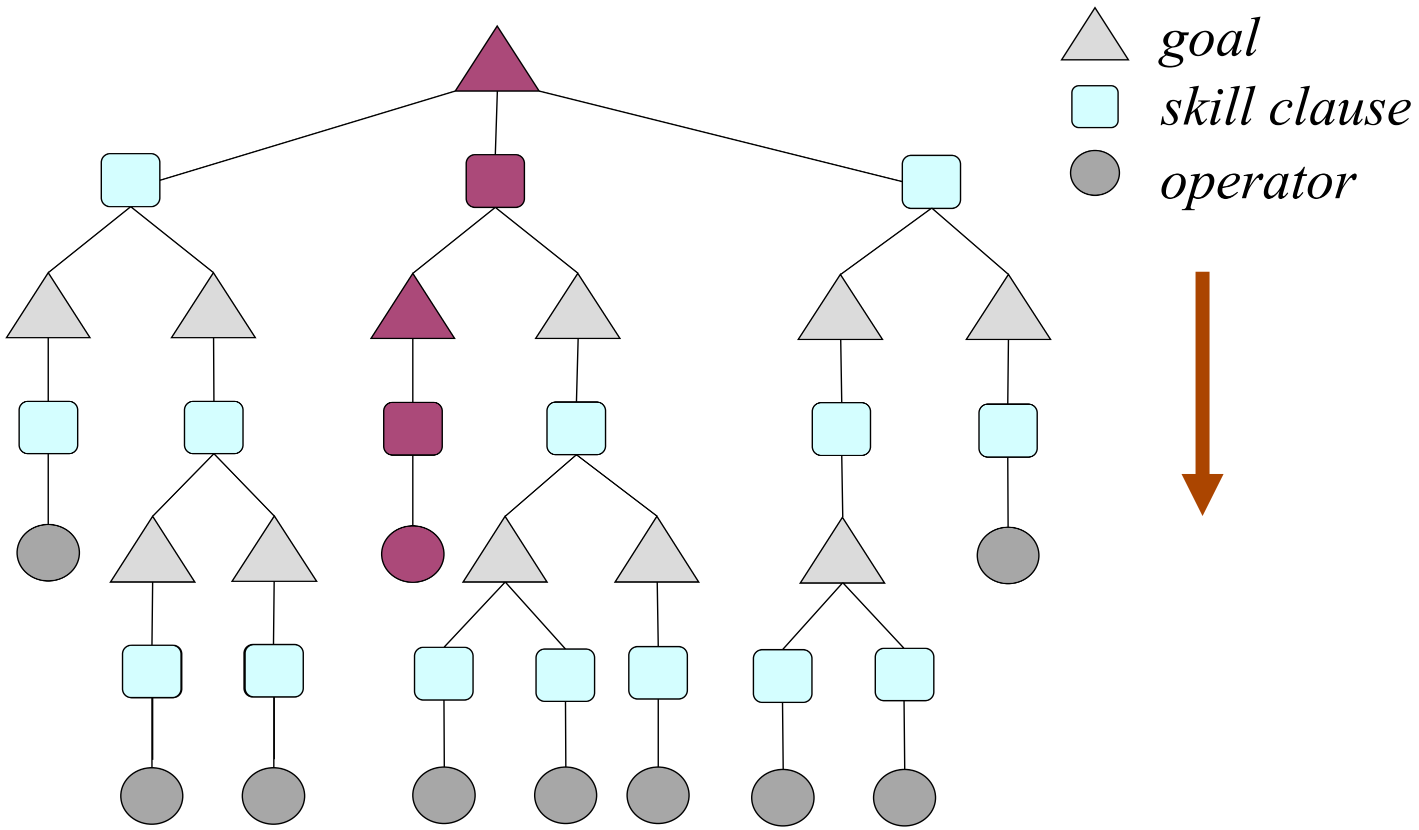
- Separates *conceptual* knowledge from *procedural* content
- Organizes both forms of knowledge into *hierarchies*
- Includes distinct modules for:
  - Conceptual inference
  - Teleoreactive execution
  - Problem solving / structural learning

The architecture controls embodied agents, so its concepts are *grounded* in percepts (descriptions of observed objects).

ICARUS has controlled both simulated characters and physical robots, but it has limitations.

Teleoreactive Execution of Hierarchical Skills

ICARUS executes skills from the top down, starting from goals, to find applicable paths through the skill hierarchy.



When it cannot find an applicable path, it falls back on problem solving to generate a novel hierarchical plan.

Challenge 1: Origin of Top-Level Goals

Most research on planning and problem solving assumes *the agent is provided with top-level goals*.

However, like humans, robotic agents on extended missions must:

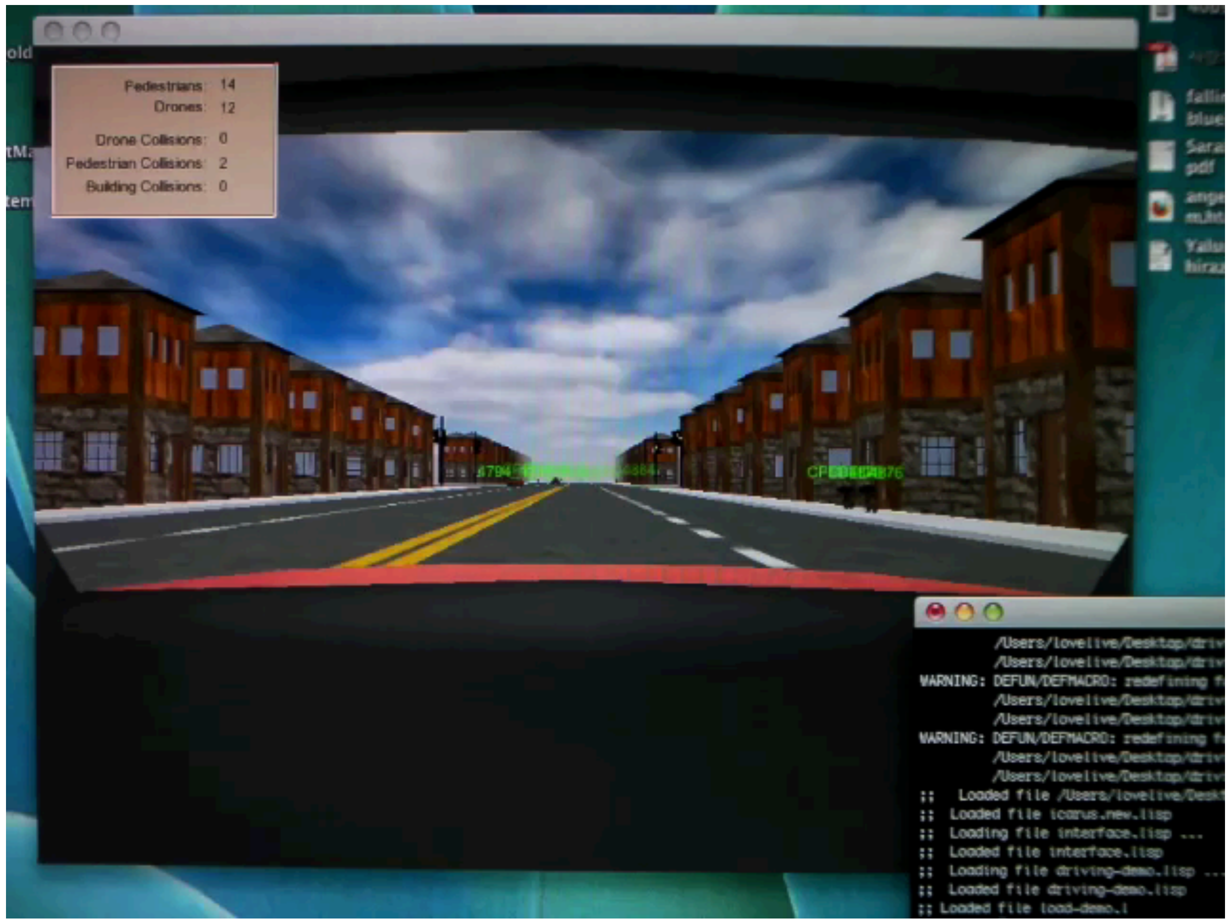
- Handle many different top-level goals
- Generate new goals when appropriate
- Abandon these goals when no longer needed

There has been little focus on this topic, although recent work in *goal reasoning* (Aha et al., 2013) has made a start.

Goal Generation in Urban Driving

We have developed ICARUS agents that drive in urban traffic with jaywalking pedestrians (Choi, 2010).

These driving agents generate new top-level goals as needed, say to avoid hitting a specific pedestrian who is running across the street.



Extending ICARUS to Generate Goals

To support this behavior, Choi (2010) has augmented the ICARUS architecture to:

- Encode knowledge about when to create goals
- Generate new goals when conditions satisfied
- Eliminate goals when conditions unsatisfied
- Use current goals to bias execution / planning

This extension relies on a distinction between goal *instances* in working memory and goal *generators* in long-term memory.

Hawes (2010), Molineaux et al. (2010), Talamadupula et al. (2010) have developed similar structures for guiding autonomous agents.

Challenge 2: Focusing Cognitive Attention

Classic autonomy research assumes that the agent pursues only a few goals.

But some missions involve many (possibly conflicting) goals, which requires the ability to:

- Encode the *priority* of each goal
- *Update priorities* as the situation changes
- *Select* which subset of these goals to pursue
- *Satisfice* when goal conflicts arise

This relates to work on *partial satisfaction planning* (Benton et al., 2012), but assumes a more dynamic setting.

Cognitive Attention and Motivation

We are building a new ICARUS-inspired architecture that focuses cognitive attention on each cycle by:

- Associating a numeric function with each goal generator
- Recalculating each goal's priority dynamically
- Using these computed priorities to:
  - Select goals that drive execution
  - Select goals and operators in planning
  - Decide when to treat a problem as solved

This unifies the traditional notions of *symbolic goals* and numeric *evaluation functions*.

This also maps directly onto the idea of *motivation* in psychology.

Challenge 3: Tool Creation and Use

Current agents operate in physical environments, but they cannot manipulate their surroundings.

In contrast, humans can design, create, and utilize *tools* that help them achieve their goals.

- They can use levers and pulleys to move heavy objects.
- They can build bridges and ramps to aid their locomotion.

A fully autonomous agent should not only adapt to its setting, but also adapt the *environment* to its own needs.

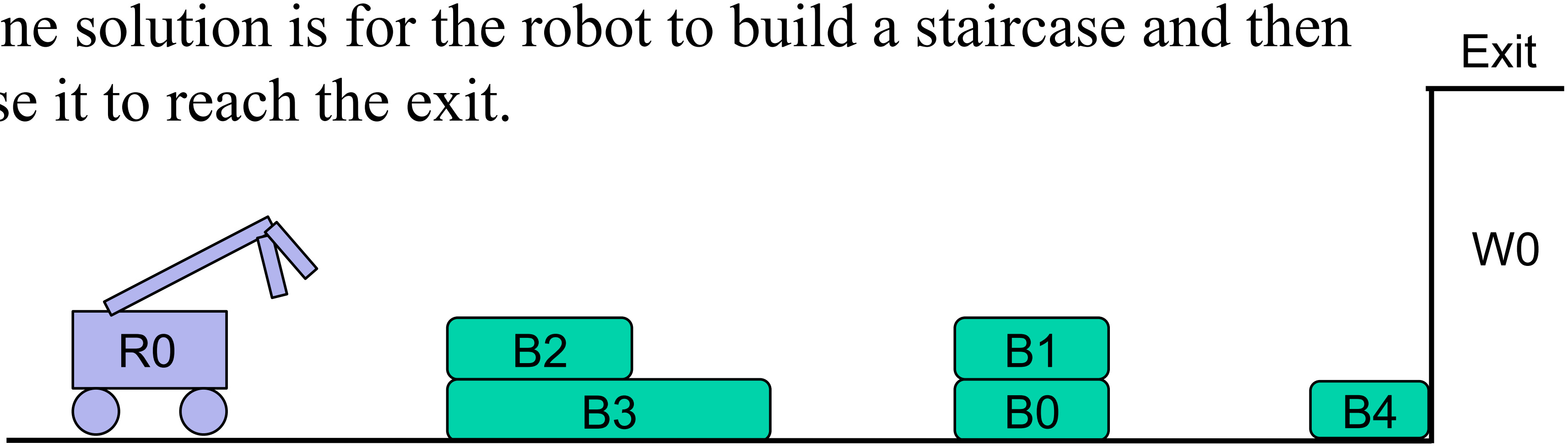
We have explored this in the *MacGyver project*, a collaboration with Mike Stilman and Henrik Christiansen at Georgia Tech.

Building and Using a Staircase

Consider a situation in which a robot wants to exit a room but the exit is too high to reach.

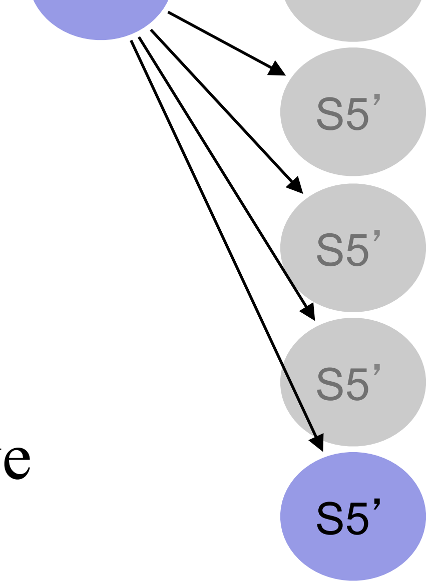
- The robot has the ability to climb stairs, but there is no staircase leading to the exit.
- However, the room contains a number of blocks that the robot's manipulator can stack.

One solution is for the robot to build a staircase and then use it to reach the exit.



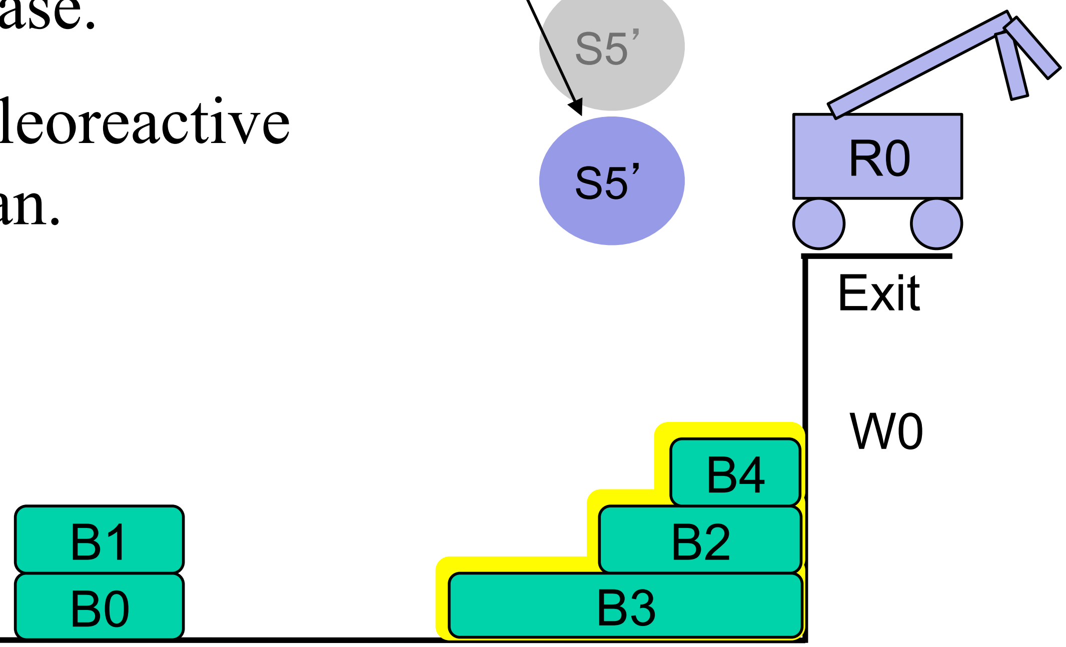
Initial state

Building and Using a Staircase



In this scenario, the extended ICARUS uses its hierarchical skills to create a plan to build and use a staircase.

The architecture then uses teleoreactive execution to carry out the plan.



Goal state (S5')

Extending ICARUS to Create / Use Tools

To support the ability to create and use tools, we have extended the ICARUS architecture to:

- Represent composite objects (e.g., towers, bridges)
- Calculate numeric attributes of such objects (e.g., height, support)
- Predict the numeric effects of environmental actions
- Generate plans that construct composite objects
- Execute these plans to achieve the agent's goals.

Our work has used simulated environments, but the Georgia Tech team has pursued similar ideas with physical robots.

Challenge 4: Adaptive Planning / Execution

The literature on planning and execution literature has reported many different techniques:

- Forward vs. backward search
- Best-first vs. beam vs. greedy search
- Closed-loop vs. open-loop control

Hypothesis: *The most appropriate strategies depend on features of the agent's current situation.*

A fully autonomous system should be able to adapt its strategies to that situation.

Challenge 4: Adaptive Planning / Execution

ICARUS cannot adapt its planning and execution methods, but we are developing a more flexible architecture that:

- Represents strategies are domain-independent control rules
  - *Forward vs. backward search, open vs. closed loop control*
- Encodes problem characteristics along with state and goals
  - *Relative branching factor, reliability of actions*
- Conditions strategic decisions on these characteristics
  - *Search in constrained direction, sense only when unreliable*

The new architecture will support higher-level forms of autonomy than possible previously.

Summary Remarks

We need more research on autonomy in cognitive systems that adapt their internal mechanisms by:

- *Generating goals* at the top level
- *Focusing attention* by prioritizing these goals
- *Creating and using tools* from available materials
- *Adapting strategies* for planning and execution

Together, these should give cognitive systems that are far more *adaptive*, and thus more *autonomous*, than exist now.

End of Presentation