A Cognitive Architecture for Embodied Agents

Pat Langley

Institute for the Study of Learning and Expertise
Palo Alto, California, USA

http://www.isle.org/~langley/

Thanks to D. Choi, G. Iba, T. Konik, N. Li, N. Nejati, D. Shapiro, D. Stracuzzi, and N. Trivedi for their contributions. This talk presents research partly funded by grants from ONR and DARPA, which are not responsible for its contents.

Introductory Remarks

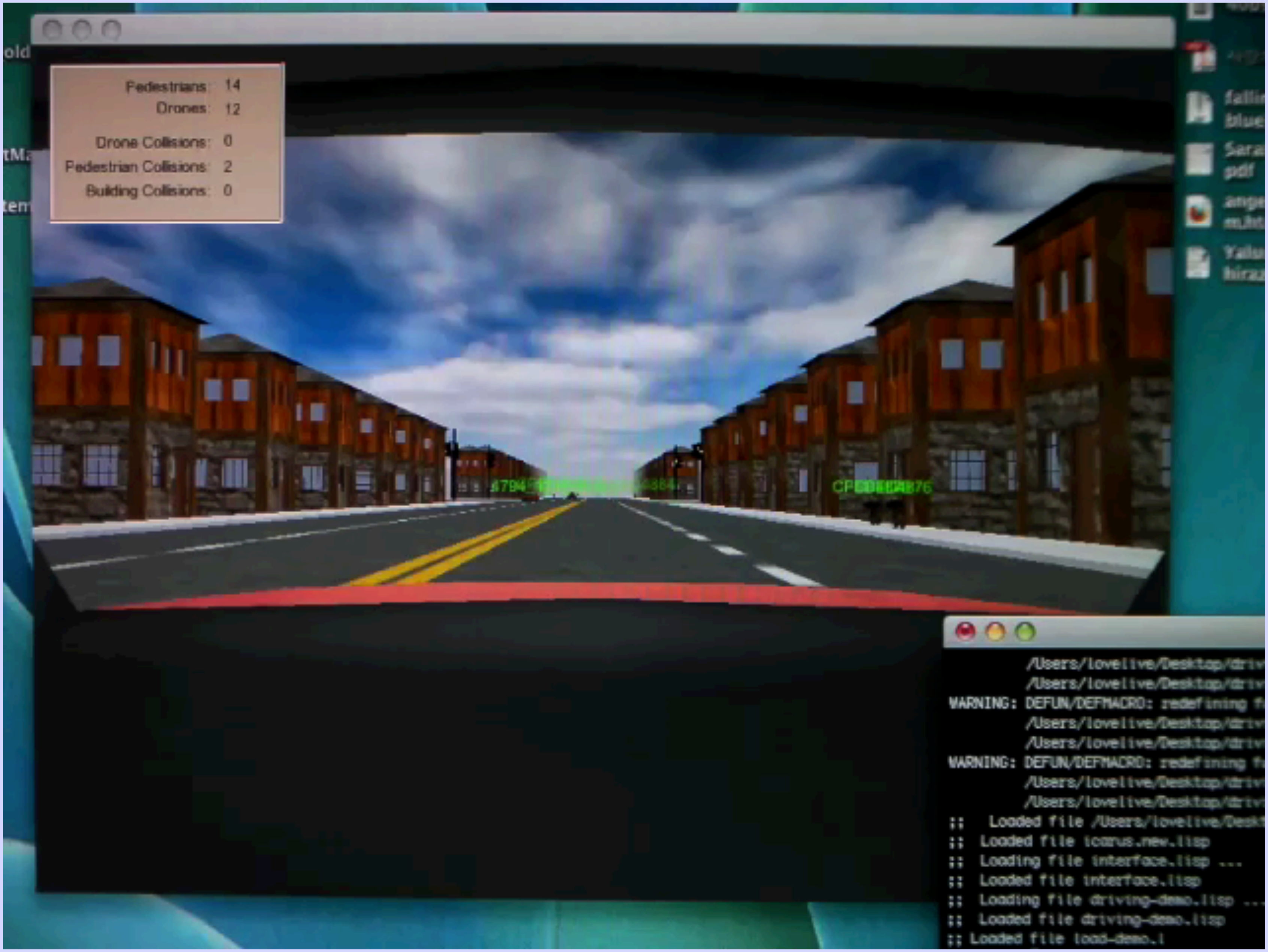
Cognitive Architectures

A *cognitive architecture* (Newell, 1990) is an infrastructure for intelligent systems that:

- makes strong theoretical assumptions about the representations and mechanisms underlying cognition
- incorporates many ideas from psychology about the nature of human cognition
- contains distinct modules, but these access and alter the same memories and representations
- comes with a programming language that eases construction of knowledge-based systems

A cognitive architecture is all about mutual constraints, as it aims to provide a *unified* theory of the mind.

Synthetic Agents for Urban Driving



Outline for the Talk

- Review of the ICARUS cognitive architecture
  - Common and distinctive features
  - Conceptual inference
  - Skill execution
  - Problem solving
  - Skill acquisition
- Synthetic agents developed with ICARUS
- Ongoing architectural extensions
  - Focusing cognitive attention
  - Creating and using tools
  - Adaptive planning and execution

Theory
+
Implementation

The ICARUS Architecture

ICARUS (Langley & Choi, 2006) is a computational theory of the human cognitive architecture that posits:

1. Short-term memories are distinct from long-term stores
2. Memories contain modular elements cast as symbolic structures
3. Long-term structures are accessed through pattern matching
4. Cognitive processing occurs in retrieval/selection/action cycles
5. Cognition involves dynamic composition of mental structures

These assumptions are not novel; it shares them with frameworks like Soar (Laird et al., 1987) and ACT-R (Anderson, 1993).

Distinctive Features of ICARUS

However, ICARUS also makes assumptions that differentiate it from other architectures:

1. Cognition is grounded in perception and action
2. Categories and skills are separate cognitive entities
3. Short-term elements are instances of long-term structures
4. Long-term knowledge is organized in a hierarchical manner
5. Inference and execution are more basic than problem solving

Some of these tenets also appear elsewhere, but only ICARUS combines them into a unified cognitive theory.

Research Goals for ICARUS

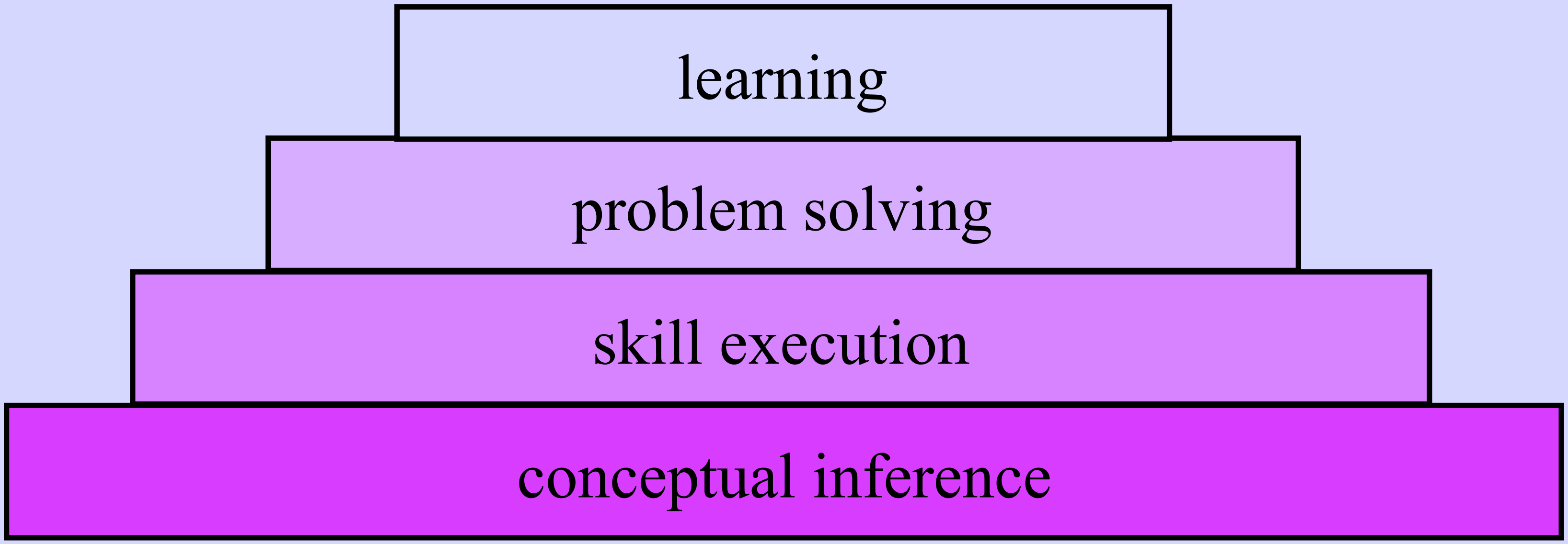
Our current objectives in developing ICARUS are to produce:

- a computational theory of high-level cognition in humans
- that is qualitatively consistent with results from psychology
- that exhibits as many distinct cognitive functions as possible
- that supports creation of intelligent agents in virtual worlds

Modeling quantitative experimental results has its place but can delay achieving broad coverage (Cassimatis et al., 2009).

Cascaded Integration in ICARUS

Like other unified cognitive architectures, ICARUS incorporates a number of distinct modules.



ICARUS adopts a *cascaded* approach to integration in which lower-level modules produce results for higher-level ones.

Conceptual Knowledge and Relational Inference

Theory of Conceptual Inference

Concepts are distinct cognitive entities that humans use to describe their environment:

- Most categories are *grounded* in perception, in that they refer to the physical characteristics of objects or events.
- Many concepts are *relational* in that they describe connections or interactions among objects or events.
- Concepts are organized in a *hierarchy*, with more complex categories defined in terms of simpler structures.
- Everyday conceptual inference is an automatic process that proceeds in a *bottom-up* manner.

ICARUS incorporates these tenets about conceptual structures and their processing.

ICARUS Concepts for Urban Driving

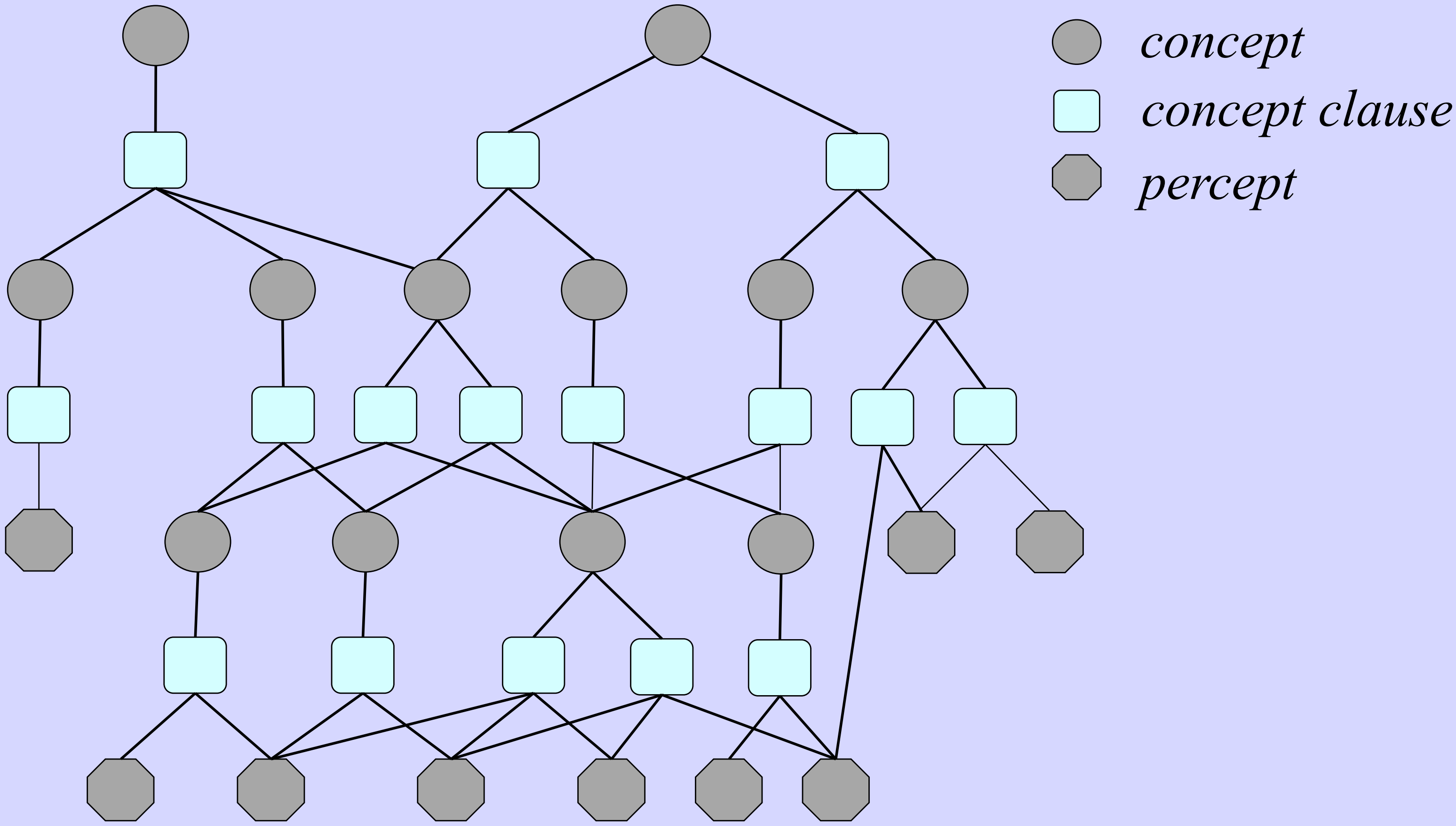
```
((driving-well-in-rightmost-lane ?self ?line1 ?line2)
 :percepts    ((self ?self) (segment ?seg) (line ?line1 segment ?seg))
               (line ?line2 segment ?seg))
 :relations   ((driving-well-in-segment ?self ?seg ?line1 ?line2)
               (not (lane-to-right ?line1 ?line2 ?anyline))))

((driving-well-in-segment ?self ?seg ?line1 ?line2)
 :percepts    ((self ?self) (segment ?seg) (line ?line1 segment ?seg))
               (line ?line2 segment ?seg))
 :relations   ((in-segment ?self ?seg)
               (aligned-and-centered-in-lane ?self ?line1 ?line2)
               (steering-wheel-straight ?self)))

((aligned-and-centered-in-lane ?self ?line1 ?line2)
 :percepts    ( (self ?self segment ?seg)
               (line ?lane1 segment ?seg dist ?dist1 angle ?ang1)
               (line ?lane2 segment ?seg dist ?dist2 angle ?ang2))
 :tests       ((*nearly-equal ?dist1 ?dist2) (*nearly-equal ?ang1 ?ang2))
```

Hierarchical Organization of Concepts

ICARUS organizes conceptual memory in a hierarchical manner.



The same conceptual predicate can appear in multiple clauses to specify disjunctive and recursive concepts.

ICARUS Beliefs and Goals for Urban Driving

Inferred beliefs:

(current-street me A)
(lane-to-right g599 g601)
(last-lane g599)
(under-speed-limit me)
(steering-wheel-not-straight me)
(in-lane me g599)
(on-right-side-in-segment me)
(building-on-left g288)
(building-on-left g427)
(building-on-left g431)
(building-on-right g287)
(increasing-direction me)

(current-segment me g550)
(first-lane g599)
(last-lane g601)
(slow-for-right-turn me)
(centered-in-lane me g550 g599)
(in-segment me g550)
(intersection-behind g550 g522)
(building-on-left g425)
(building-on-left g429)
(building-on-left g433)
(building-on-right g279)
(near-pedestrian me g567)

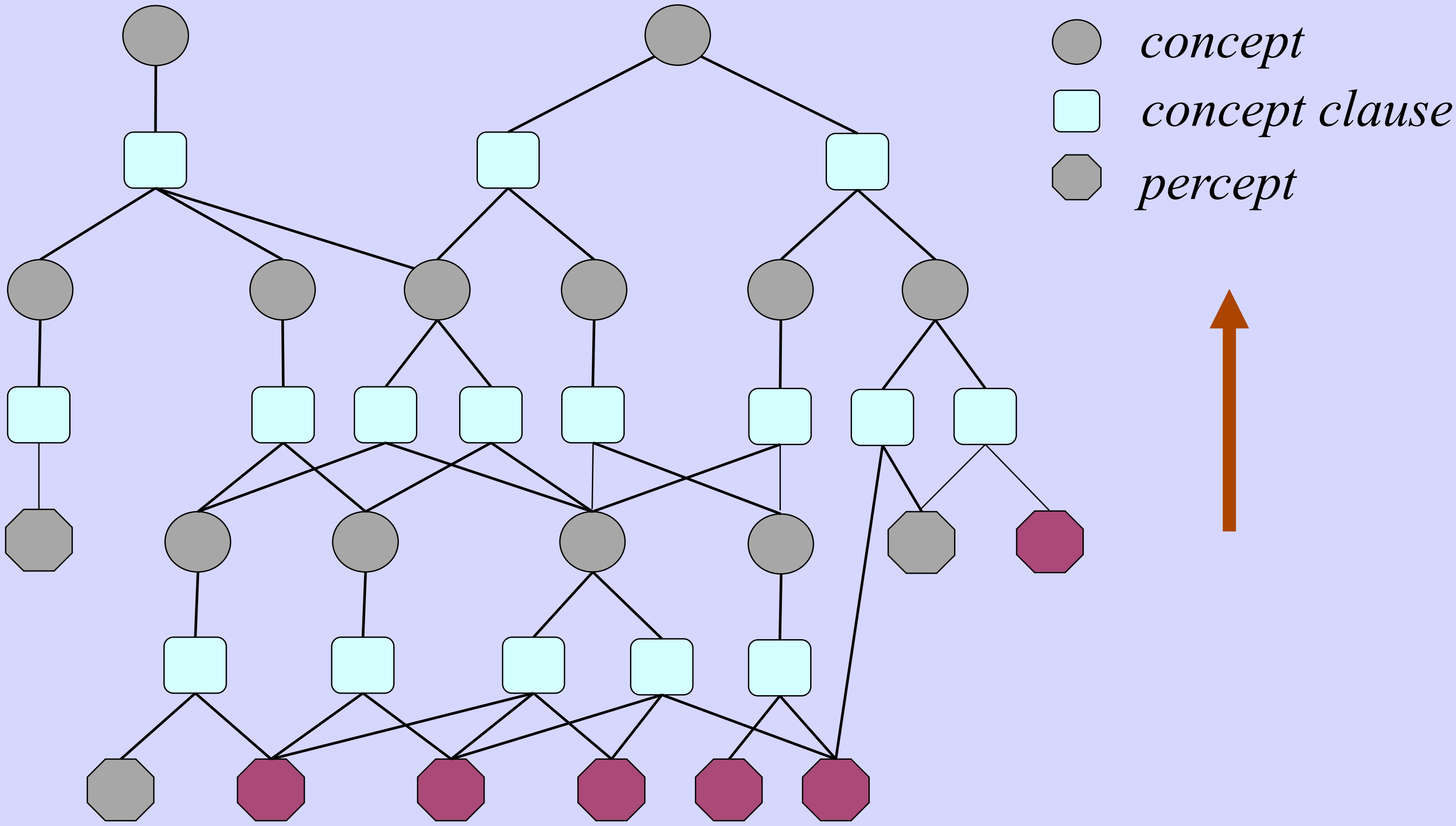
Top-level goals:

(not (near-pedestrian me ?any))
(on-right-side-in-segment me)
(not (over-speed-limit me))

(not (near-vehicle me ?other))
(in-lane me ?segment)
(not (running-red-light me))

Conceptual Inference in ICARUS

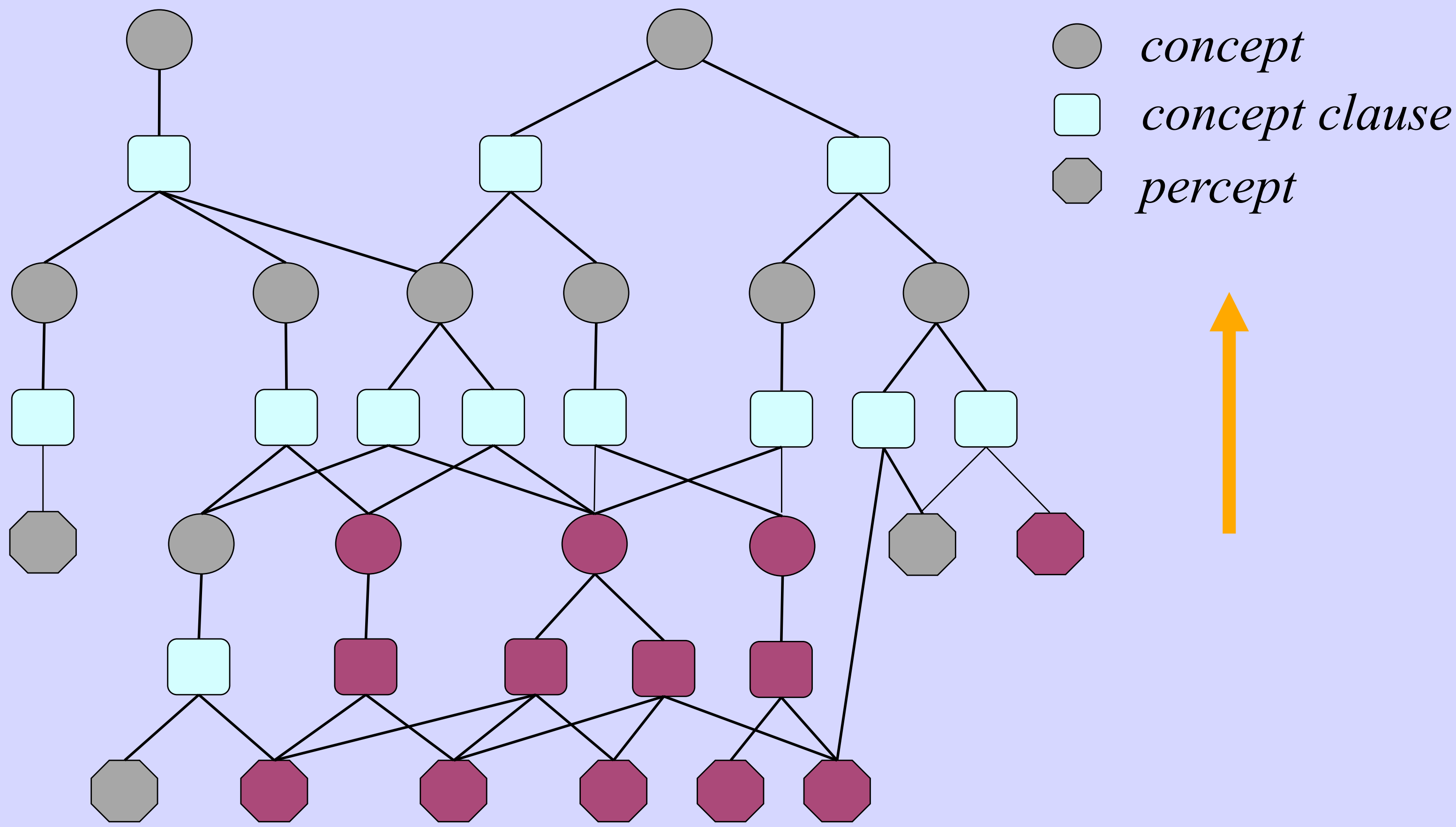
Conceptual inference in ICARUS occurs from the bottom up.



Starting with observed percepts, this process produces high-level beliefs about the current state.

Conceptual Inference in ICARUS

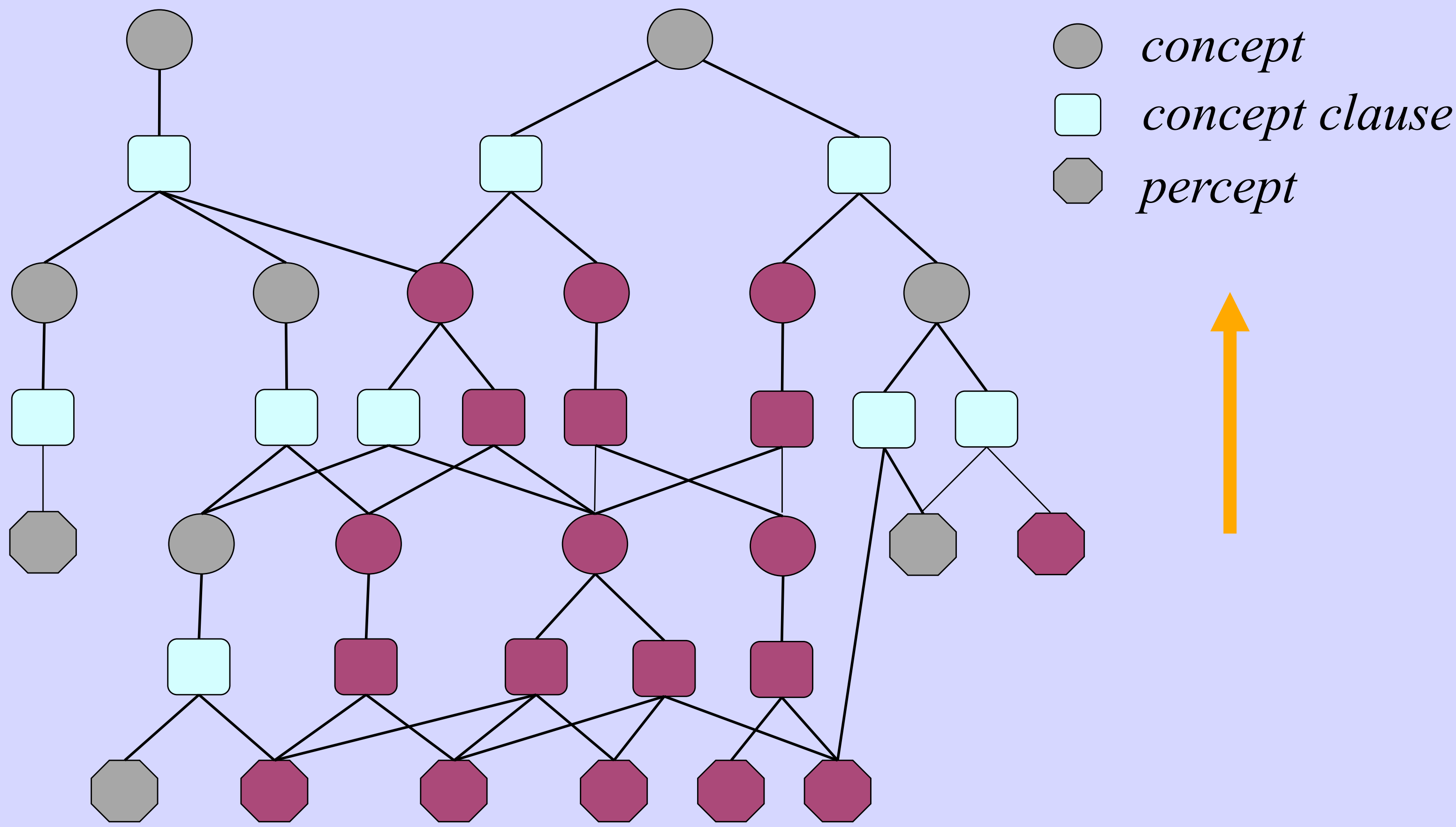
Conceptual inference in ICARUS occurs from the bottom up.



Starting with observed percepts, this process produces high-level beliefs about the current state.

Conceptual Inference in ICARUS

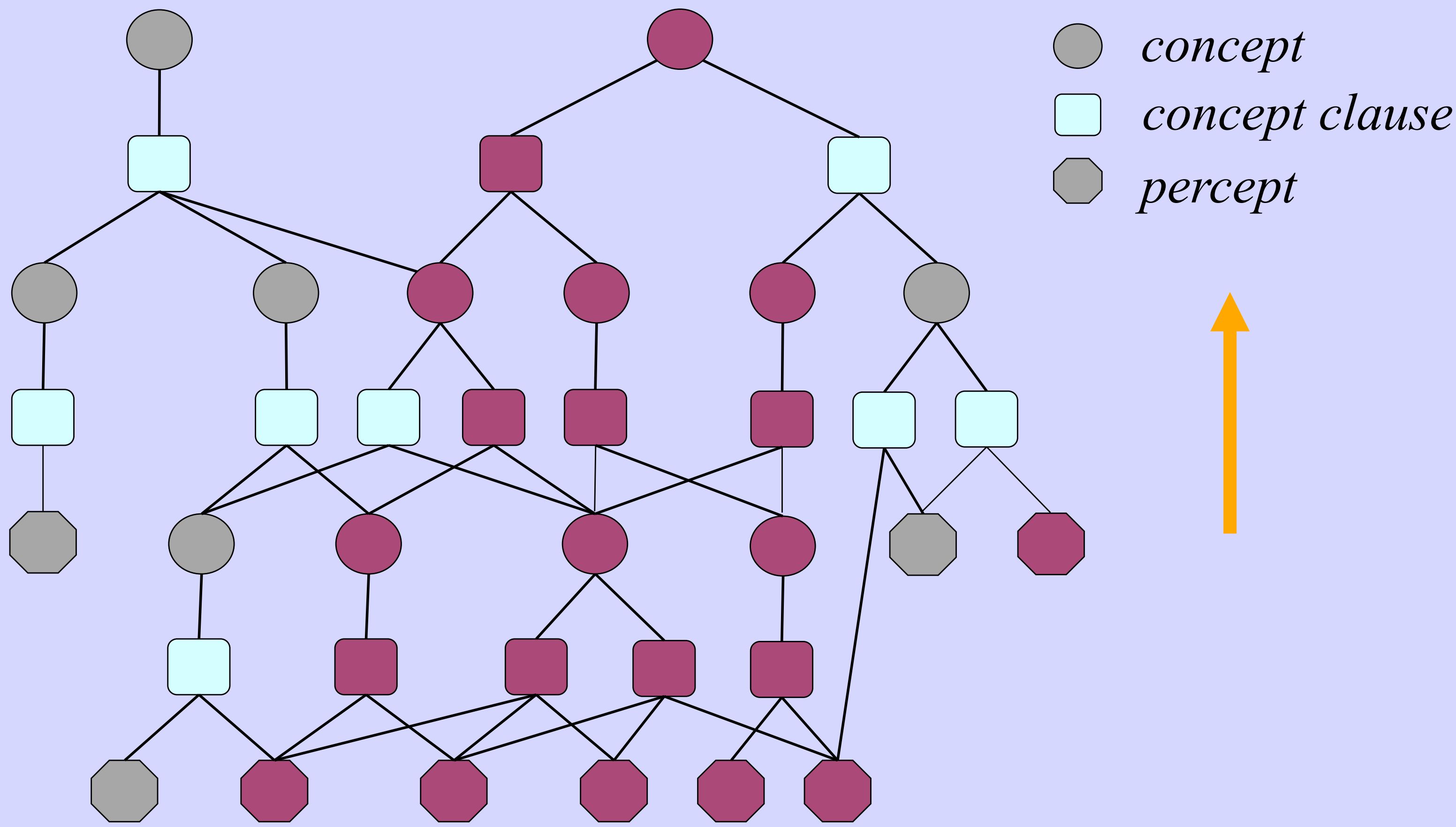
Conceptual inference in ICARUS occurs from the bottom up.



Starting with observed percepts, this process produces high-level beliefs about the current state.

Conceptual Inference in ICARUS

Conceptual inference in ICARUS occurs from the bottom up.



Starting with observed percepts, this process produces high-level beliefs about the current state.

Hierarchical Skills and Teleoreactive Execution

Theory of Skill Execution

Skills are distinct cognitive structures that describe how one interacts with the environment:

- Most human skills are *grounded* in perception (indirectly through concepts) and in action.
- Skills are *relational* in that they describe changes in conceptual structures as a result of their execution.
- Memory for skills is organized as a *hierarchy*, with more complex activities decomposed into simpler ones.
- Skills are *indexed* by the goals they achieve on their successful execution in the environment.
- Execution is *teleoreactive*, i.e., guided by the agent's goals but sensitive to environmental factors.

ICARUS incorporates these assumptions about skill execution.

ICARUS Skills for Urban Driving

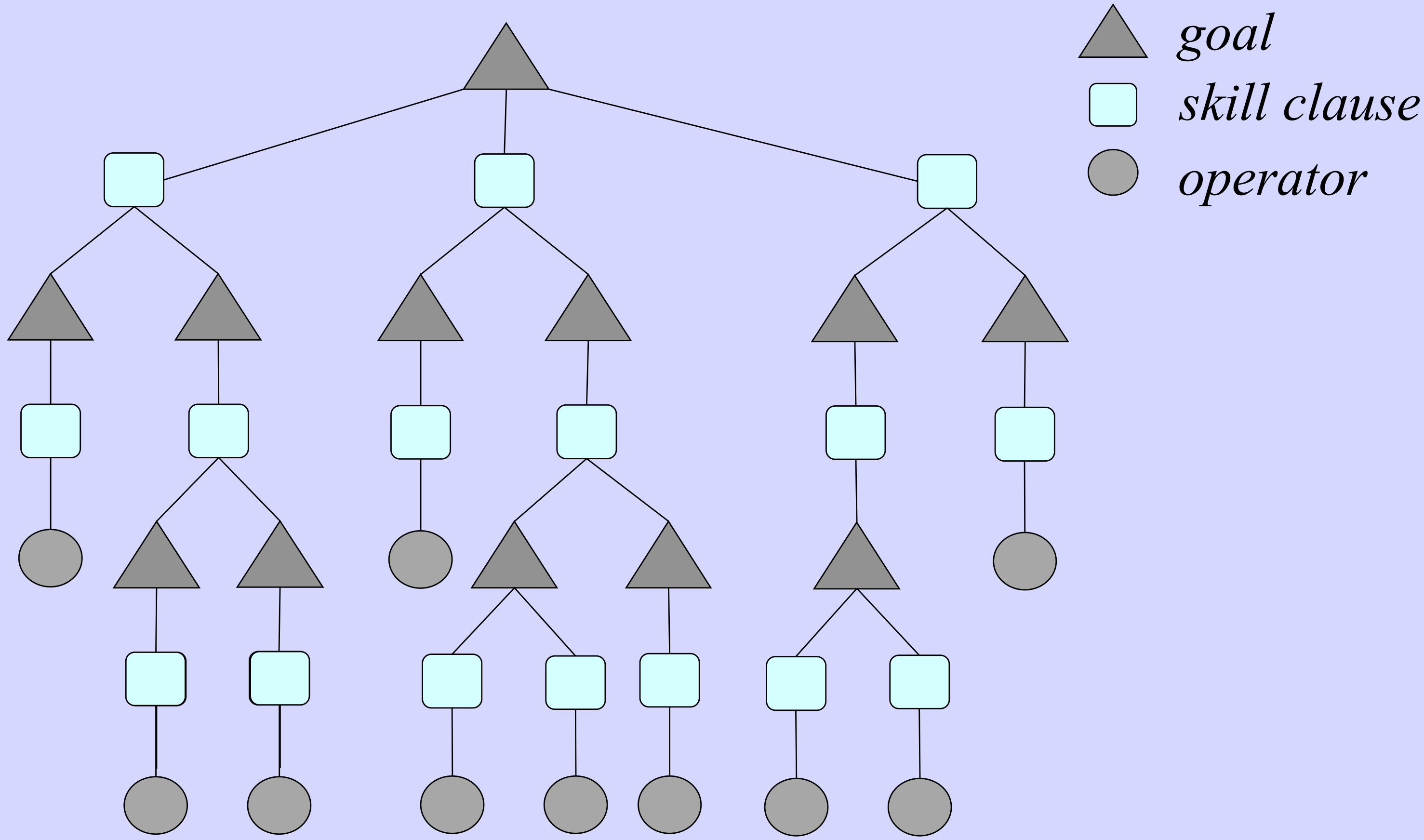
```
((driving-well-in-rightmost-lane ?self ?line1 ?line2)
 :percepts    ((self ?self) (segment ?seg) (line ?line1 segment ?seg))
               (line ?line2 segment ?seg))
 :start       ((not (lane-to-right ?line1 ?line2 ?anyline))
 :subgoals    ((driving-well-in-segment ?self ?seg ?line1 ?line2)))

((driving-well-in-segment ?self ?seg ?line1 ?line2)
 :percepts    ((self ?self) (segment ?seg) (line ?line1 segment ?seg))
               (line ?line2 segment ?seg))
 :start       ((steering-wheel-straight ?self))
 :subgoals    ((in-segment ?self ?seg)
               (aligned-and-centered-in-lane ?self ?line1 ?line2)
               (steering-wheel-straight ?self)))

((aligned-and-centered-in-lane ?self ?line1 ?line2)
 :percepts    ((self ?self))
 :start       ((misaligned-to-left-in-lane ?self ?line1 ?line2))
 :requires    ((not (steering-to-right ?self)))
 :actions     ((*steer 20)))
```

Hierarchical Organization of Skills

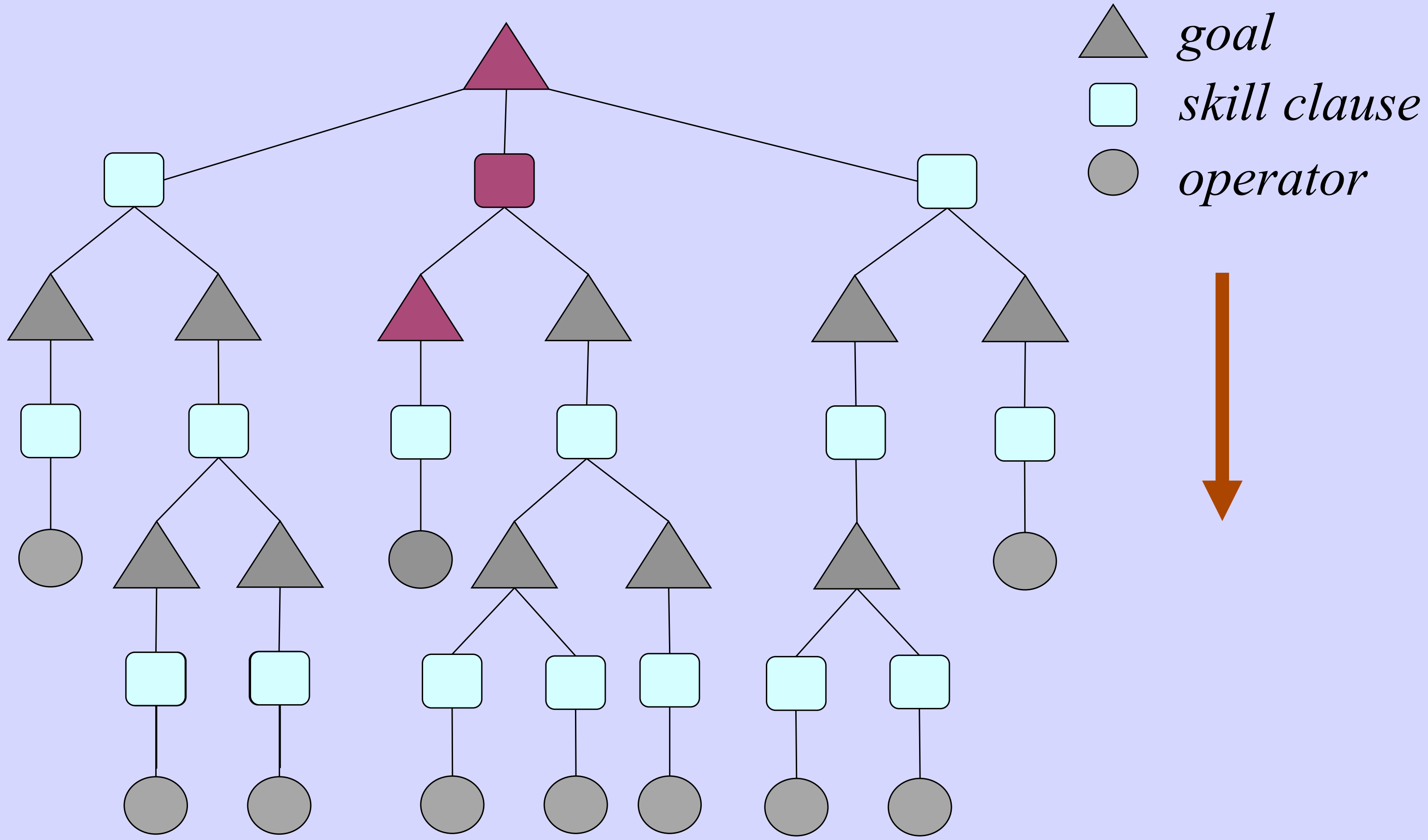
ICARUS organizes skills in a hierarchical manner, which each skill clause indexed by the goal it aims to achieve.



The same goal can index multiple clauses to allow disjunctive, conditional, and recursive procedures.

Skill Execution in ICARUS

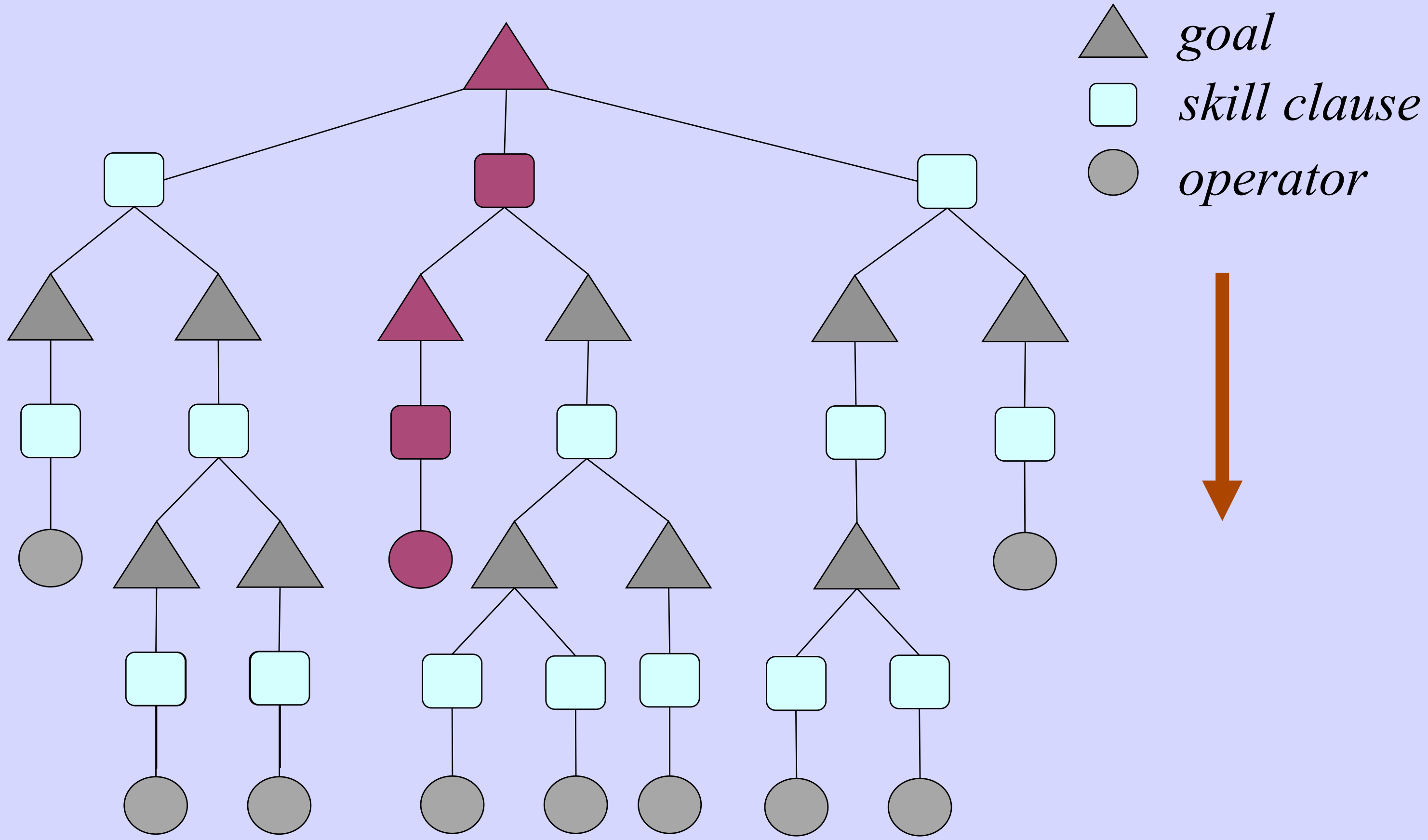
Skill execution occurs from the top down, starting from goals, to find applicable paths through the skill hierarchy.



A skill clause is applicable if its goal is unsatisfied and if its conditions hold, given bindings from above.

Skill Execution in ICARUS

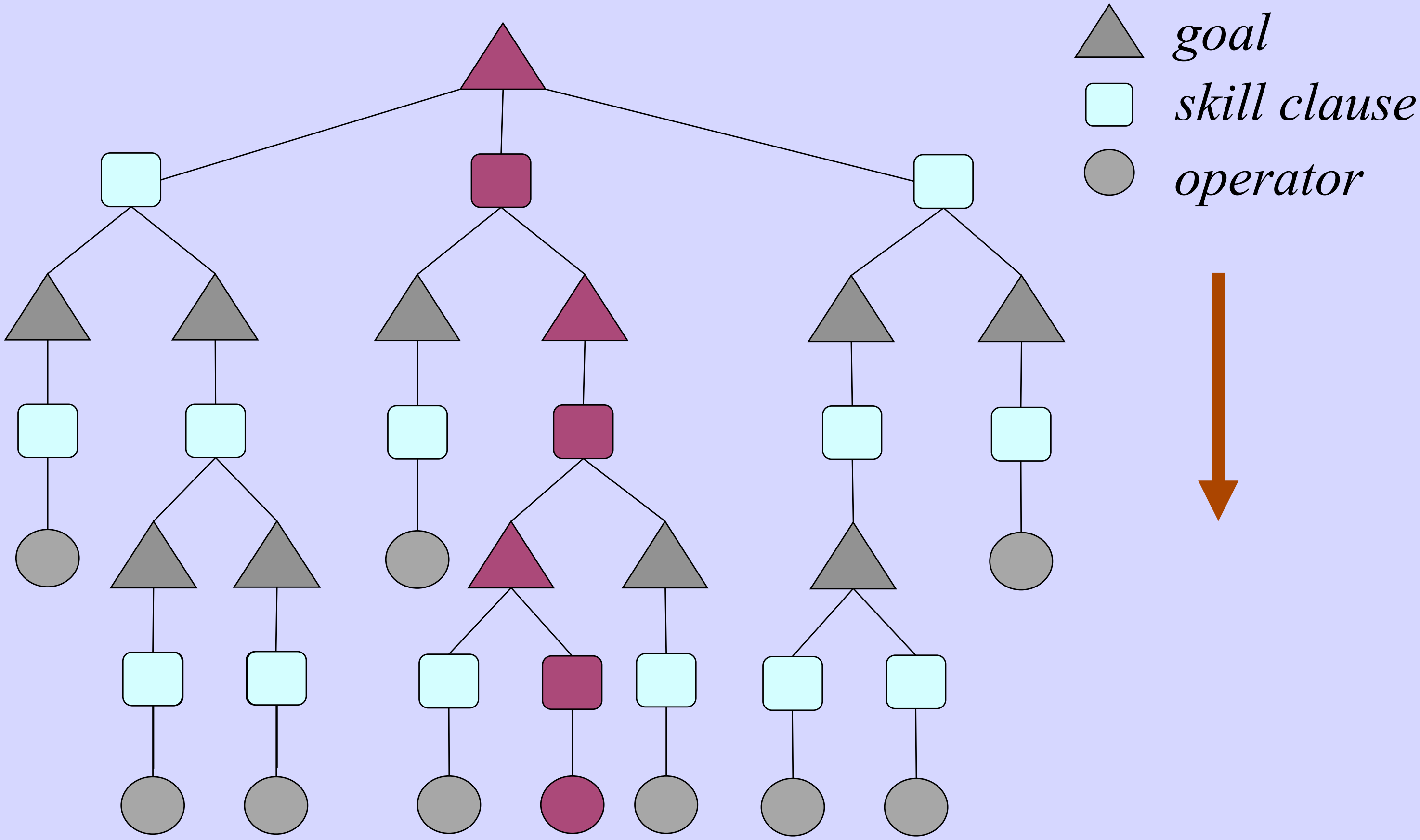
Skill execution occurs from the top down, starting from goals, to find applicable paths through the skill hierarchy.



A skill clause is applicable if its goal is unsatisfied and if its conditions hold, given bindings from above.

Skill Execution in ICARUS

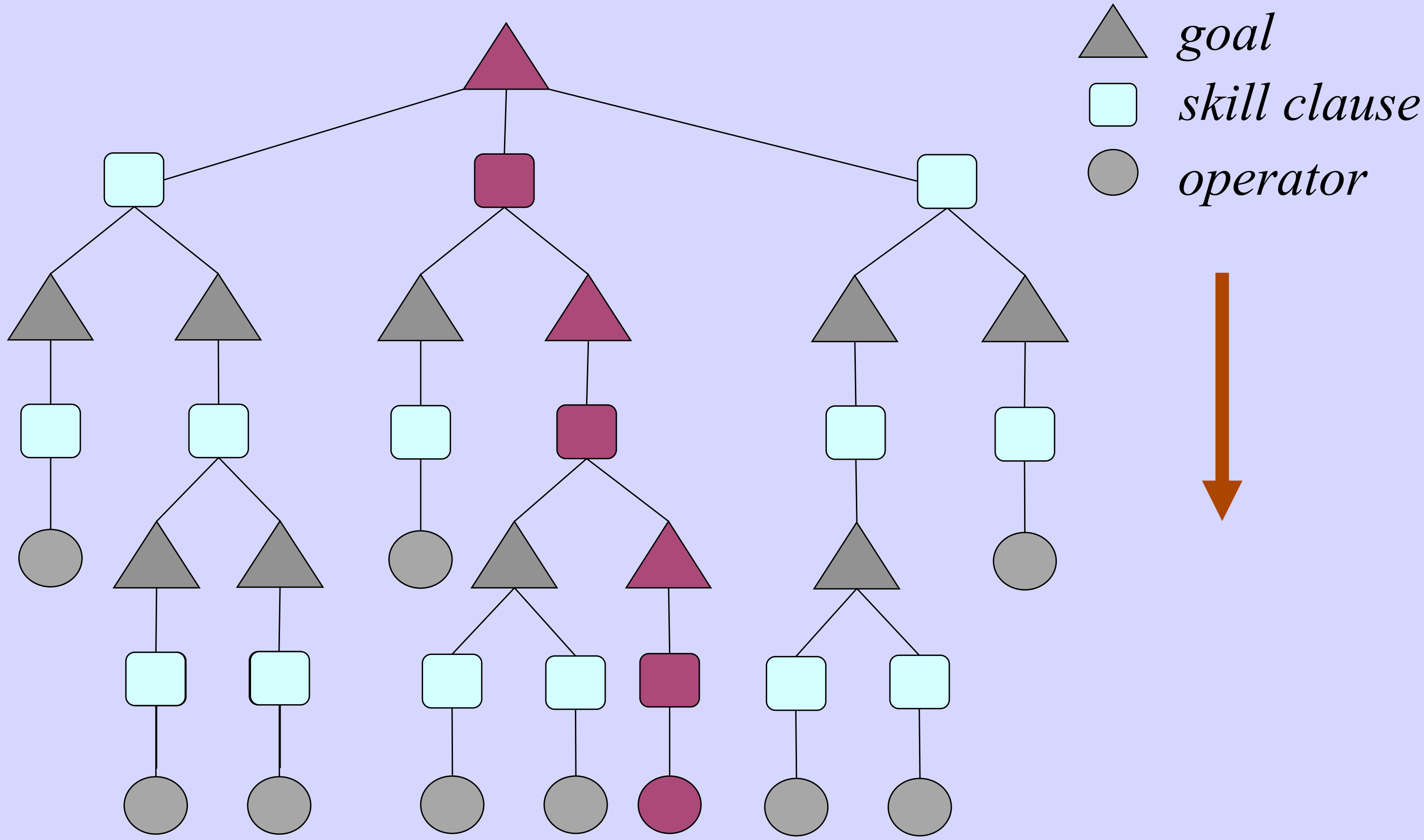
This process repeats on later cycles to produce goal-directed but reactive behavior (Nilsson, 1994).



However, ICARUS prefers to continue ongoing skills when they match, giving it a bias toward persistence over reactivity.

Skill Execution in ICARUS

If events proceed as expected, this iterative process eventually achieves the agent's top-level goal.



At this point, another unsatisfied goal begins to drive behavior, invoking different skills to pursue it.

Problem Solving and Skill Acquisition

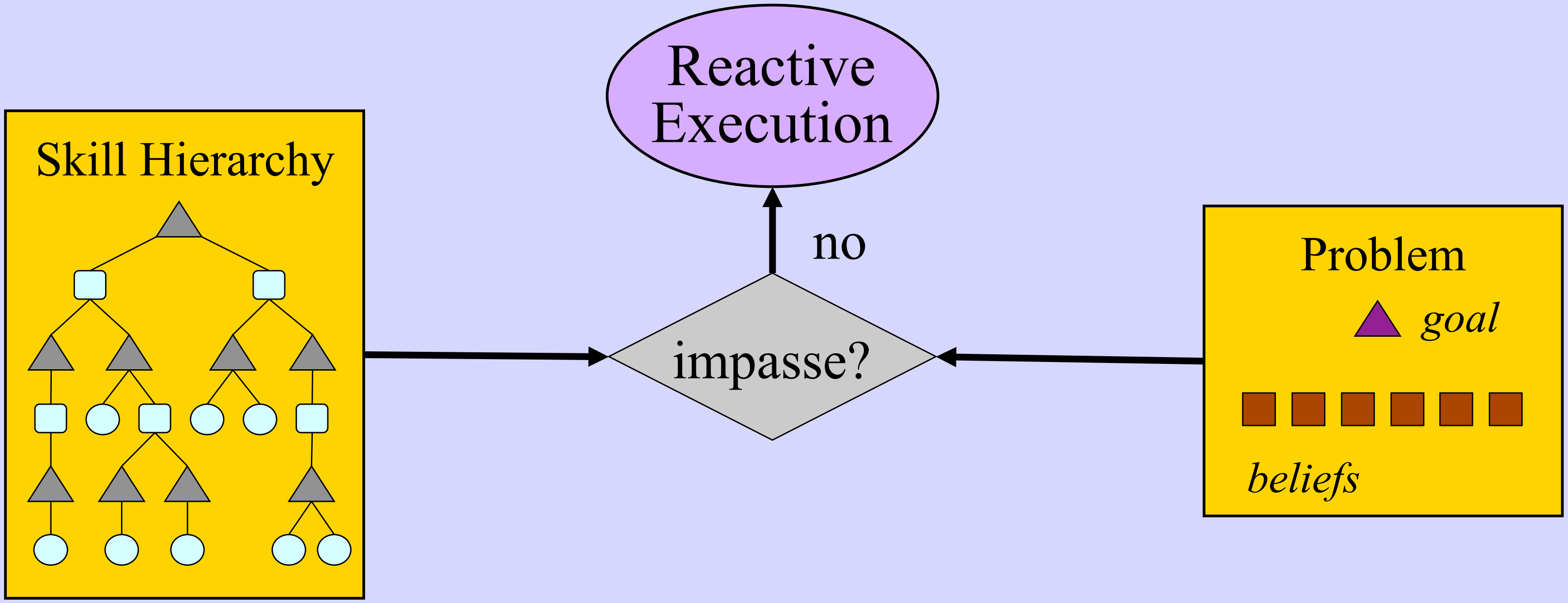
Theory of Problem Solving

Problem solving lets humans achieve goals even on complex, unfamiliar tasks:

1. Problem solving involves *heuristic search* through a *problem space* of possible solutions.
2. This search uses *operators* (skills) to transform *states* (sets of beliefs) into ones that satisfy *goals* (desired beliefs).
3. Humans often use a mix of goal-directed backward chaining and state-driven forward chaining (*means-ends analysis*).
4. Problem solving remains *grounded* in perception and actions, yet often occurs at an *abstract* level of description.
5. Human problem solving often interleaves mental processing with physical execution.

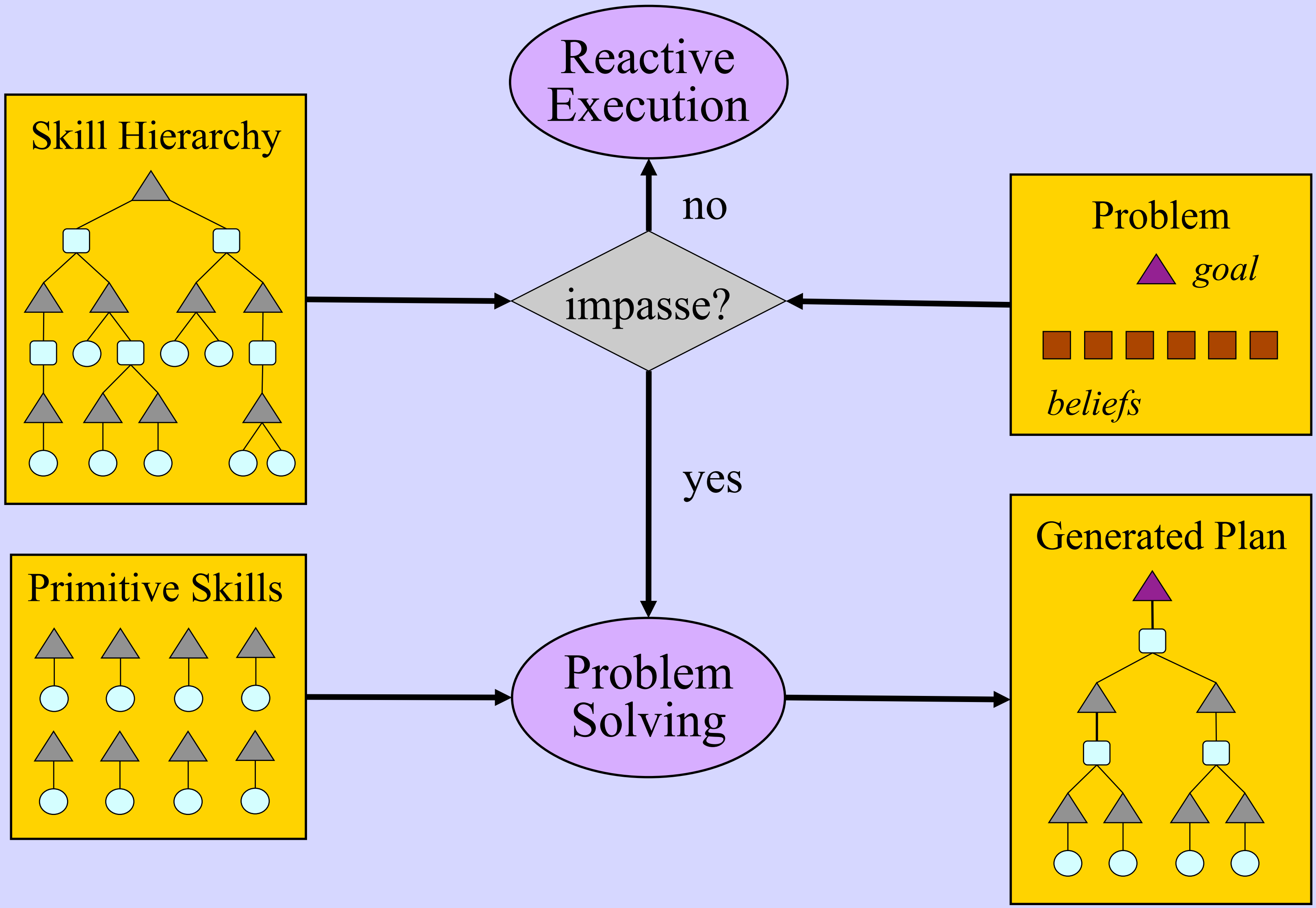
ICARUS adopts these ideas about the nature of problem solving.

Execution and Problem Solving in ICARUS



We have seen that ICARUS invokes teleoreactive execution to achieve goals when relevant hierarchical skills are avalable in memory.

Execution and Problem Solving in ICARUS



When it cannot retrievel relevant skills, ICARUS uses means-ends analysis to chain backward over skills, executing them when they become applicable.

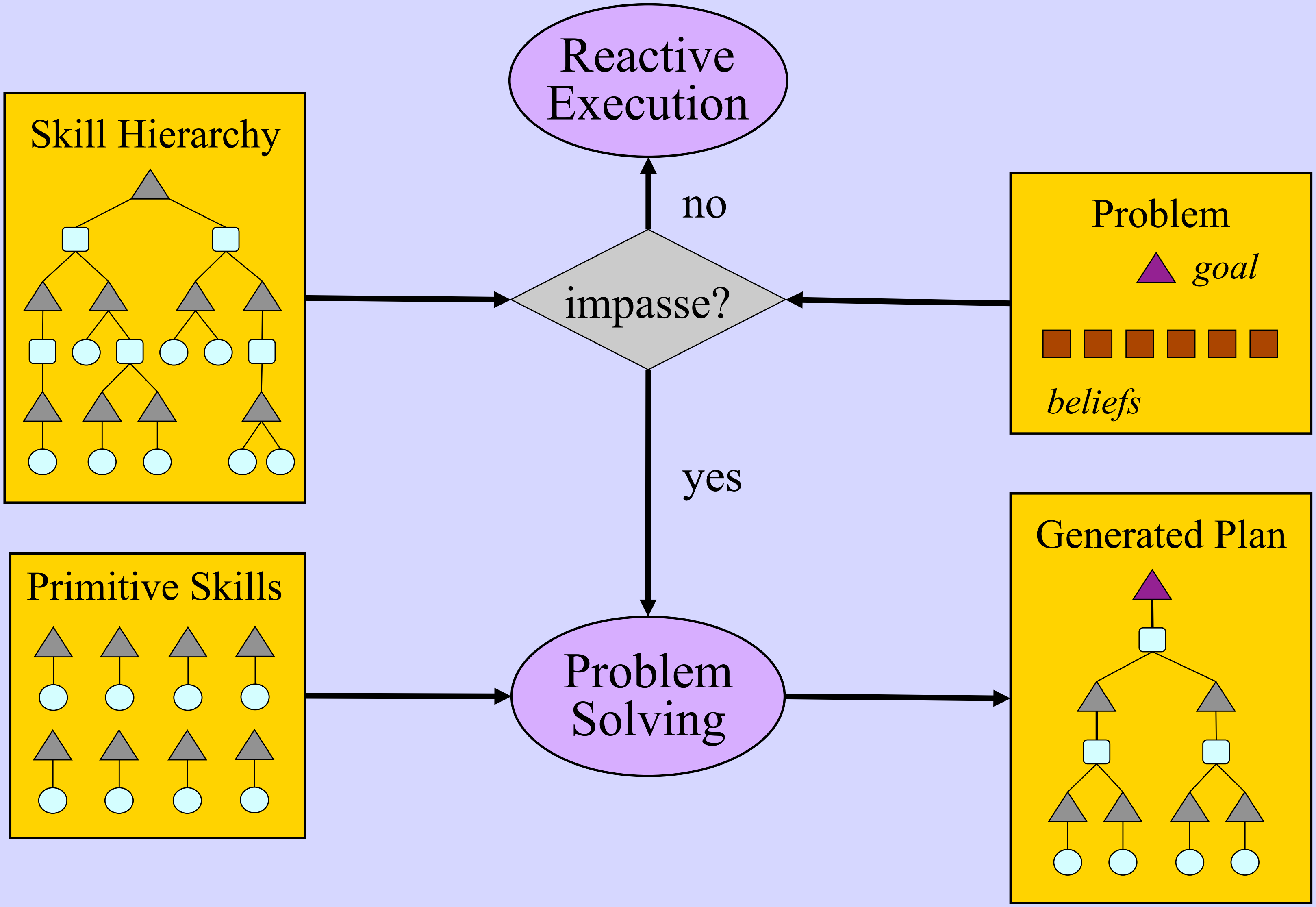
Theory of Skill Acquisition

Skill learning lets humans store the results of experience to improve their future performance:

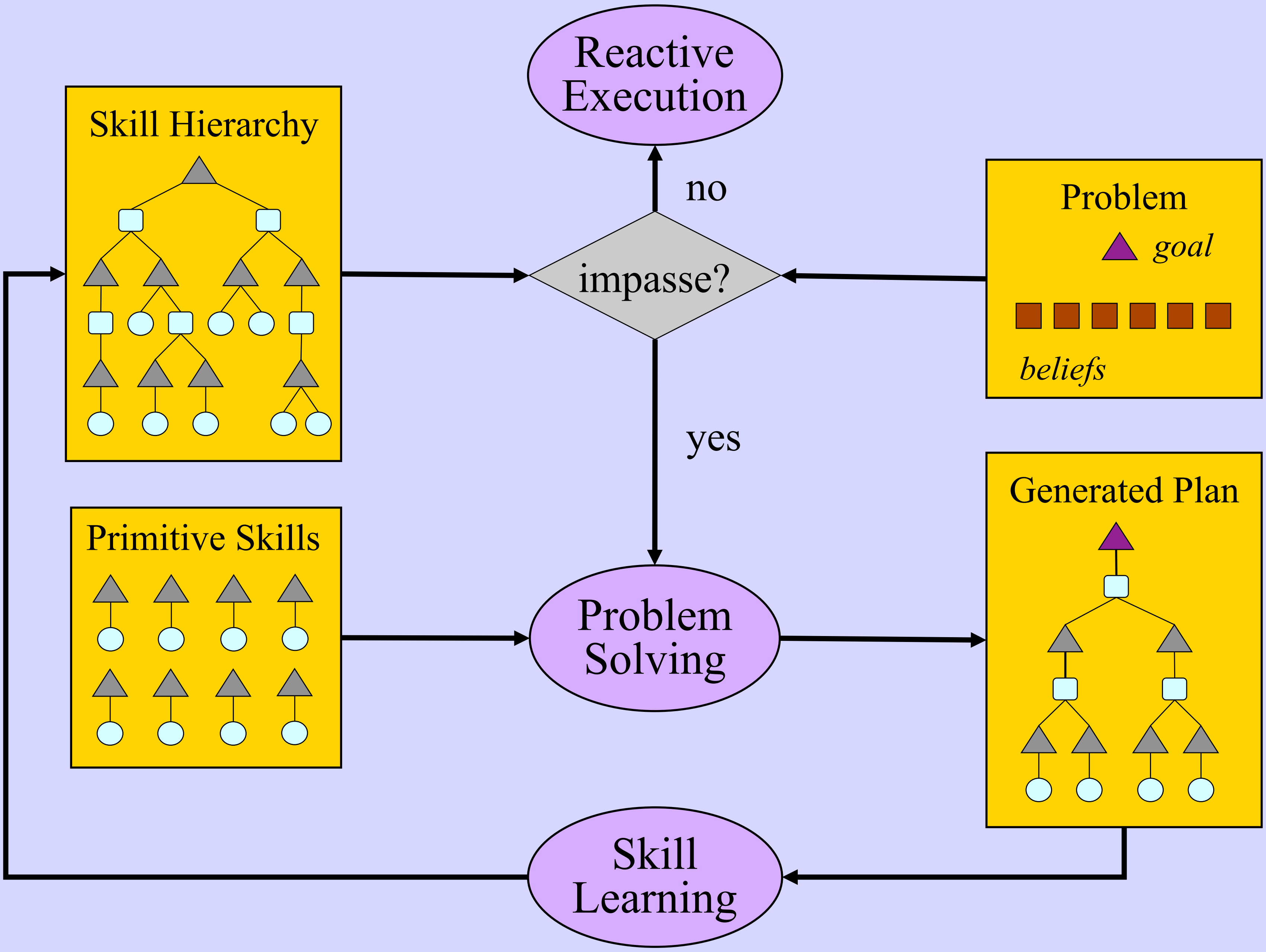
1. Skill acquisition involves monotonic addition to memory of new *symbolic structures*.
2. Learning is driven by *experience* but draws on prior *knowledge*.
3. Skill learning operates in an *incremental* and *cumulative* manner.
4. Acquisition is *interleaved with problem solving*, which provides it with experience.
5. Hierarchical skills are the *generalized traces of successful means-ends analysis*.

ICARUS embodies these five claims about the character of skill acquisition in human cognition.

ICARUS Learns Skills from Problem Solving

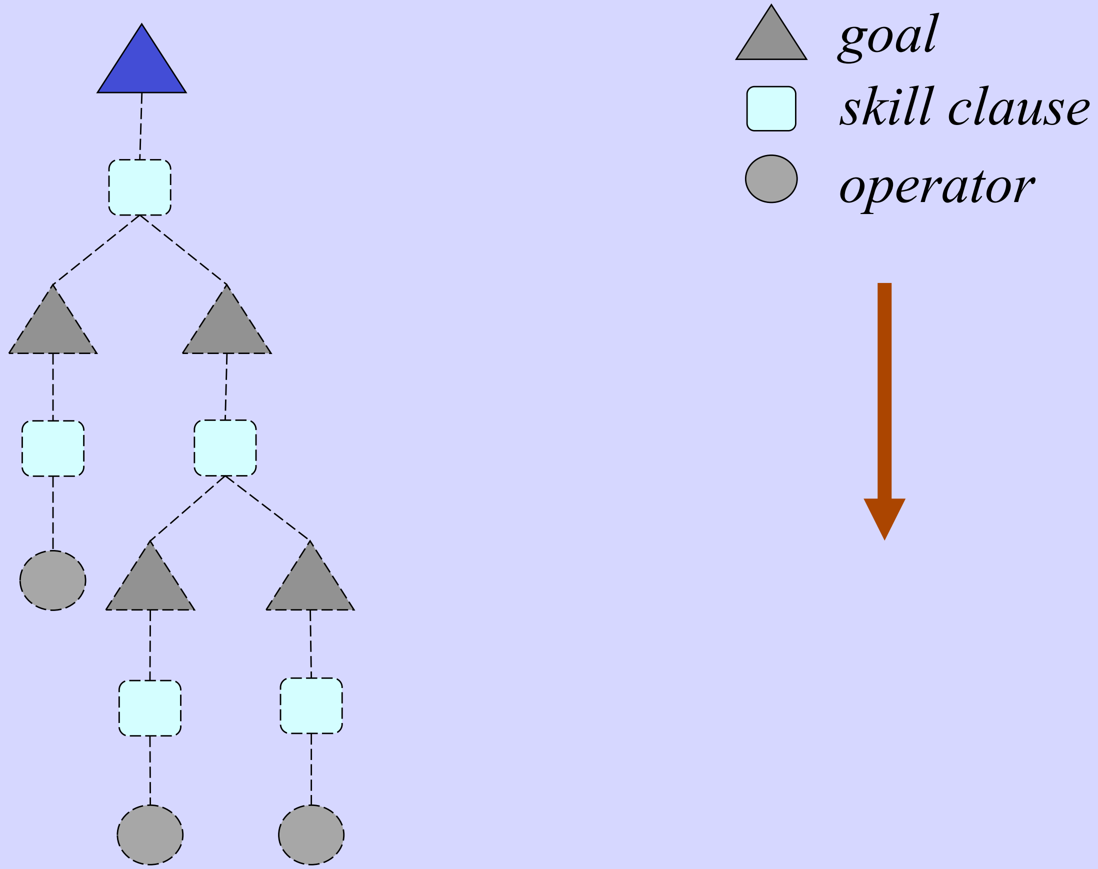


ICARUS Learns Skills from Problem Solving



Skill Learning in ICARUS

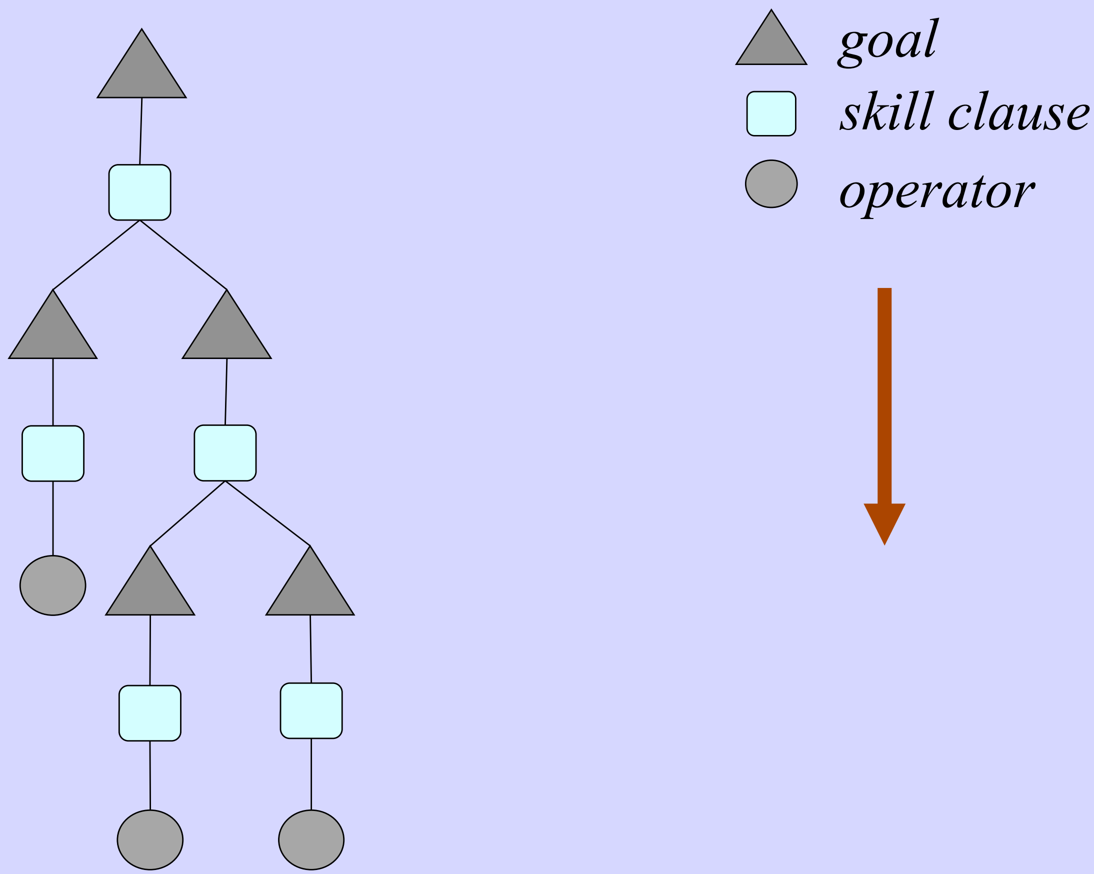
As the architecture carries out means-ends analysis, it retains traces of successful decompositions.



Each trace includes details about the goal, skill, and the initially satisfied and unsatisfied subgoals.

Skill Learning in ICARUS

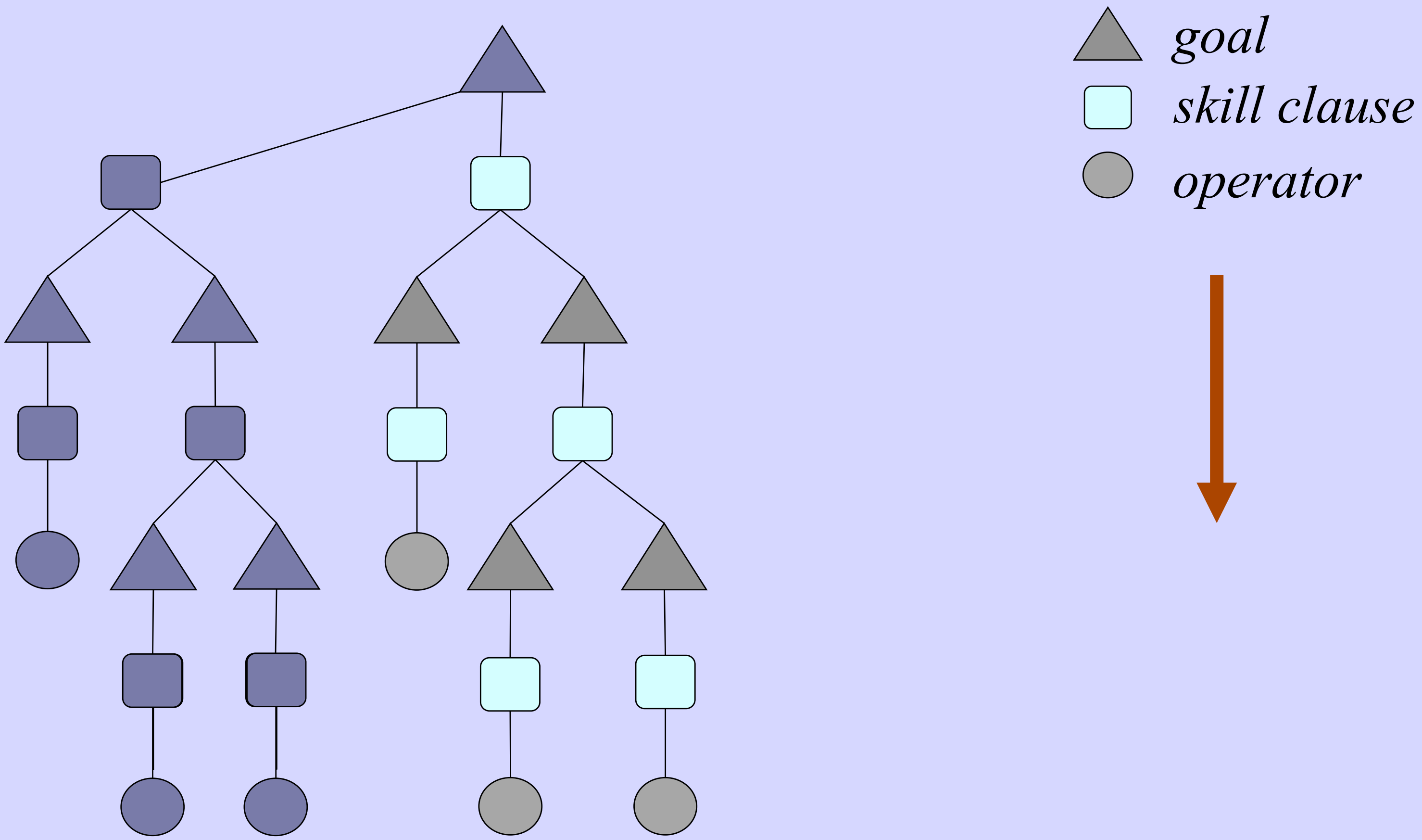
As the system solves achieves each subgoal, it generalizes the associated trace and stores it as a new skill.



Thus, problem solving operates from the top down, but ICARUS acquires skills from the bottom up.

Skill Learning in ICARUS

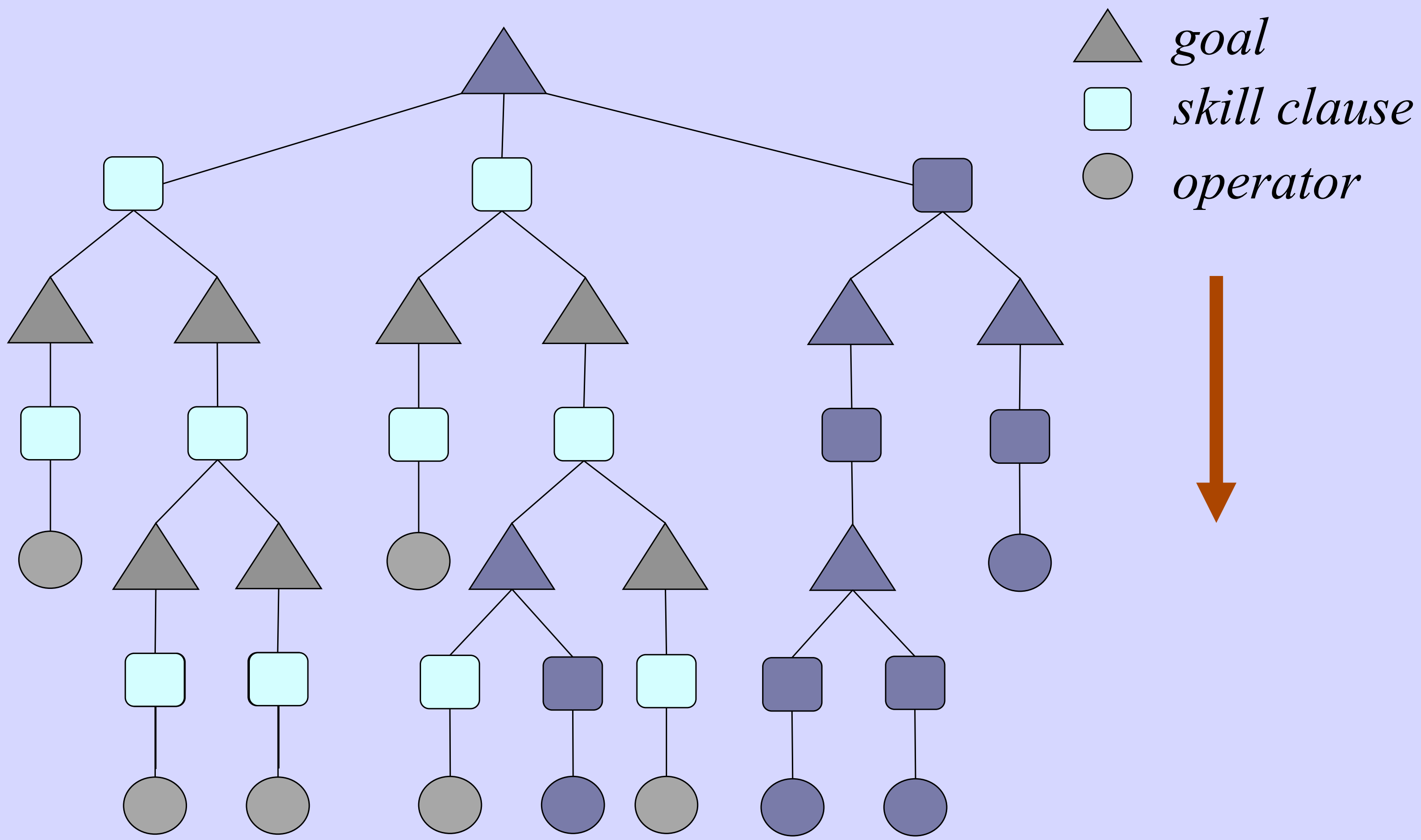
When the architecture achieves new goals, or the same goal in a different way, it gradually expands the hierarchy.



Each newly learned skill is available for use on future problems, thus supporting structural transfer.

Skill Learning in ICARUS

These new acquisitions occur with both top-level goals and with ones internal to the skill hierarchy.



Over time, the architecture acquires a broad set of skills that minimize the need for problem solving.

ICARUS Summary

ICARUS provides a unified theory of the cognitive architecture that supports:

- conceptual inference over grounded relational categories
- goal-directed but reactive execution of hierarchical skills
- means-ends problem solving when routine execution fails
- acquisition of new skills from traces of problem solving

We have used ICARUS to develop synthetic agents for a number of simulated physical environments.

However, each effort has revealed limitations that in turn led to important architectural extensions.

ICARUS Agents for Synthetic Environments

Synthetic Agents for 'Urban Combat'

Urban Combat is a synthetic environment, built on the Quake engine, used in the DARPA Transfer Learning program.

Tasks for agents involved traversing an urban landscape with a variety of obstacles to capture a flag.

Our efforts in Urban Combat led to novel insights about:

- Storing, using, and learning route knowledge
- Learning to overcome obstacles by trial and error
- Supporting different varieties of structural transfer across problems

These abilities appear necessary for robust operation in such settings.

Synthetic Agents for Urban Driving

We have developed an urban driving environment using Garage Games' Torque game engine.

This supports tasks like:

- Aligning car with lane
- Driving at legal speeds
- Changing lanes
- Turning a street corner
- Driving around a block
- Delivering packages

We have demonstrated each of these in ICARUS.

This domain involves a mixture of cognitive and sensori-motor behavior in a constrained but complex and dynamic setting.

Synthetic Agents for Urban Driving

Our early efforts on urban driving motivated two key features of the ICARUS architecture:

- Indexing skills by goals they achieve (Langley & Choi, 2006)
- Multiple top-level goals with priorities (Li & Choi, 2007)

Choi's (2010) work on driving agents motivated other changes to the architecture:

- Long-term memory for generic goals or motives
- Reactive generation of specific short-term goals
- Concurrent execution of multiple skills per cycle

Research on this complex environment led to important new architectural functionalities.

Synthetic Agents for American Football

We have also built ICARUS agents to execute football plays in Rush 2008 (Knexus).

This domain is less complex than driving in some ways.

But it remains highly reactive and depends on spatio-temporal coordination among each team's players.

Moreover, ICARUS has *learned* its football skills from videos of Oregon State University practice games.

Synthetic Agents for American Football

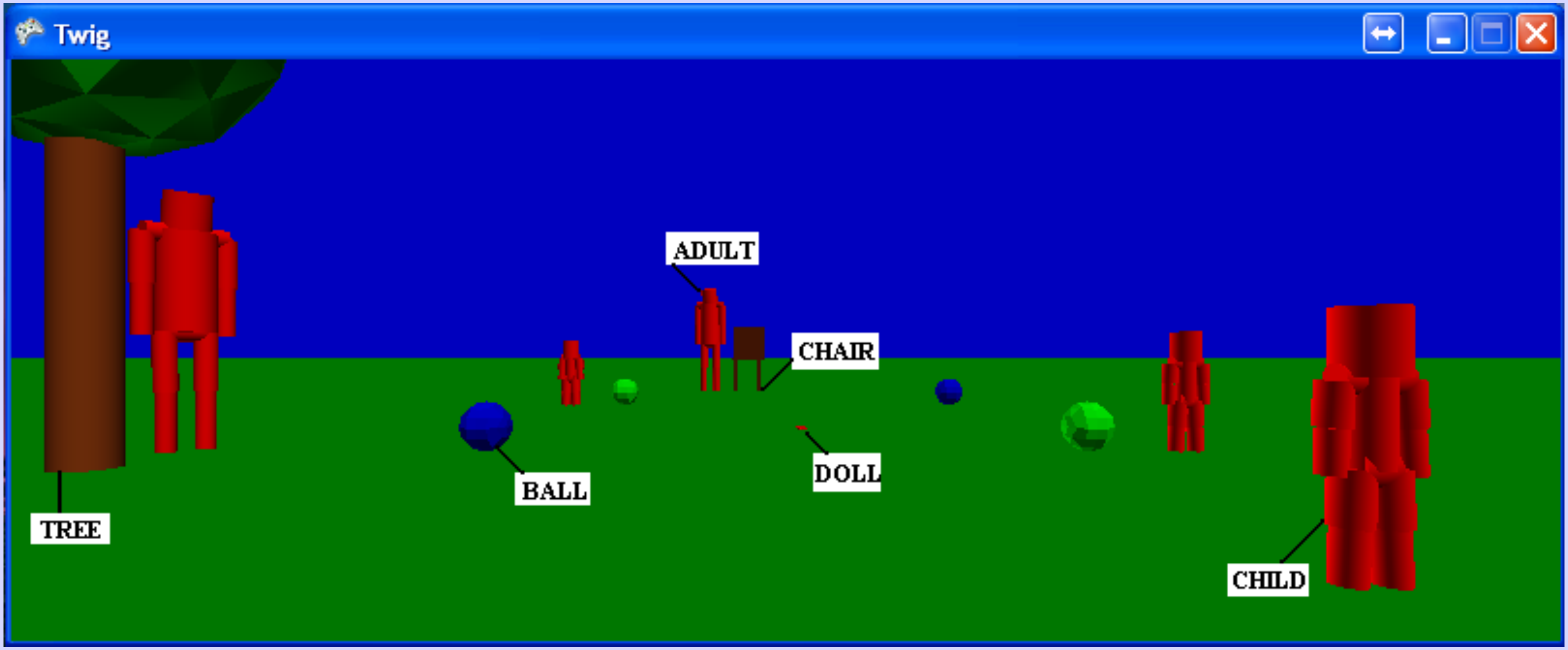
Our efforts on Rush 2008 have motivated additional extensions to the ICARUS architecture:

- Time stamps on beliefs to support episodic traces
- Ability to represent and recognize temporal concepts
- Adapting means-ends analysis to explain observed behavior
- Ability to control multiple agents in an environment

We have used the extended ICARUS to learn hierarchical skills for 20 distinct football plays (Li et al., AIIDE-2009).

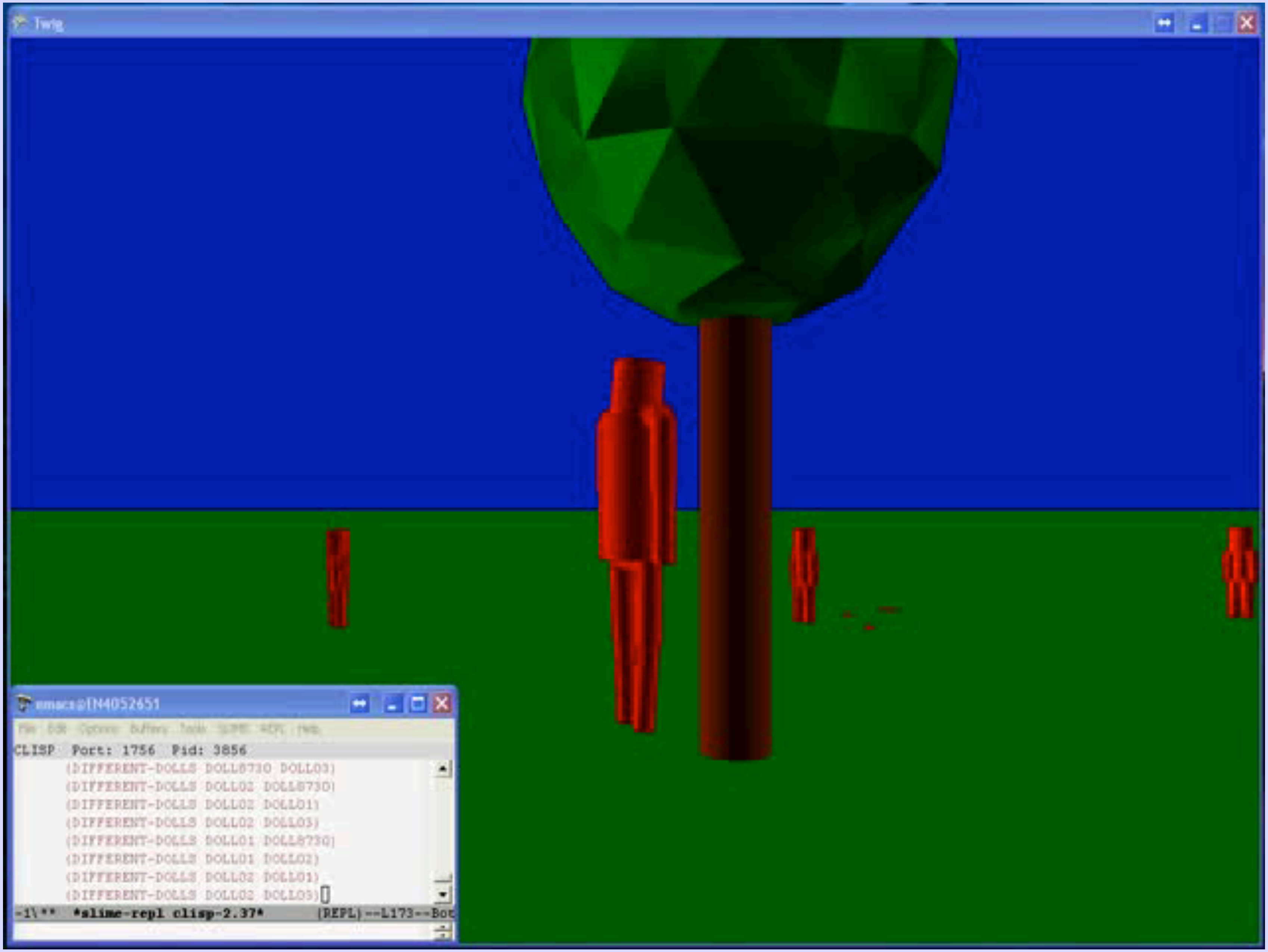
Synthetic Agents for Twig Scenarios

We have also used Horswill's (2008) Twig simulator to develop humanoid ICARUS agents.

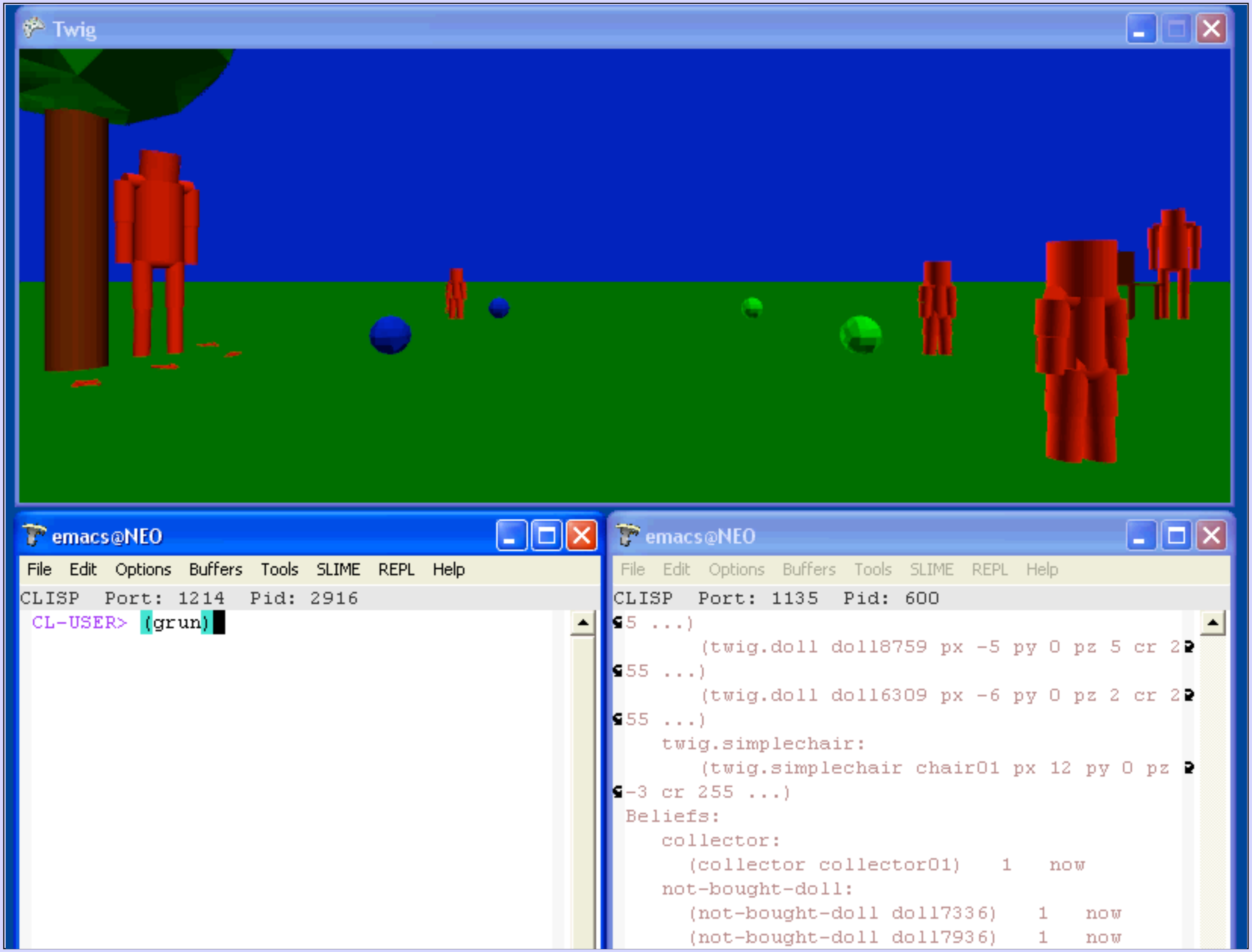


This low-fidelity environment supports a few object types, along with simple reactive behaviors for virtual characters.

Robin Hood in the Twig Environment



A Capitalist Twig Scenario



Challenges in Social Cognition

Twig offers a platform for addressing a number of challenges that arise in social cognition:

- Modal statements about beliefs, goals, intentions of other agents [e.g., (belief me (goal agent2 (holding agent2 doll5)))];
- Flexible inference that supports abductive explanation of others' behaviors (e.g., default assumptions about goals);
- Execution and problem solving that achieves goals for changing others' mental states (e.g., by communication); and
- Representing, reasoning about, and influencing others' emotions (cast as rich cognitive structures).

However, ICARUS also has additional limitations that require research in completely different directions.

Open Challenges and Ongoing Research

Challenge 1: Focusing Cognitive Attention

Classic research on autonomy assumes that the agent pursues at most 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, on a given cycle, focuses cognitive attention 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 traditional notions of *symbolic goals* and numeric *evaluation functions*.

Also, it maps directly onto the psychological idea of *motivation*.

Challenge 2: Tool Creation and Use

Current agents operate in physical environments, but they alter their surroundings in only simple ways.

In contrast, humans can design, create, and use *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.

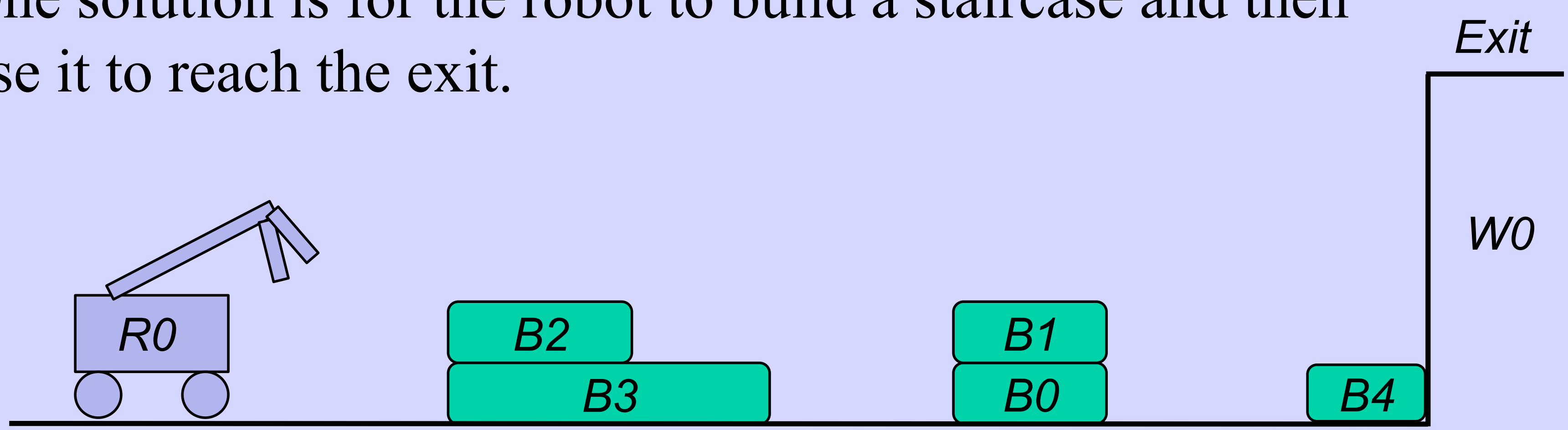
We have explored this in the *MacGyver project*, a collaboration with Mike Stilman at Georgia Tech.

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.

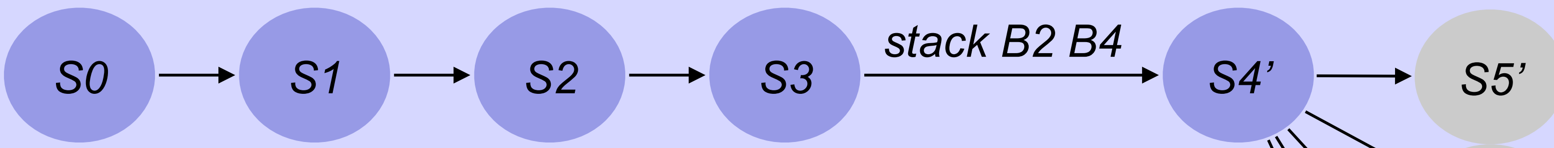
- The robot can climb stairs, but there is no staircase that leads to the exit.
- However, the room contains various blocks that the robot's manipulator can stack.

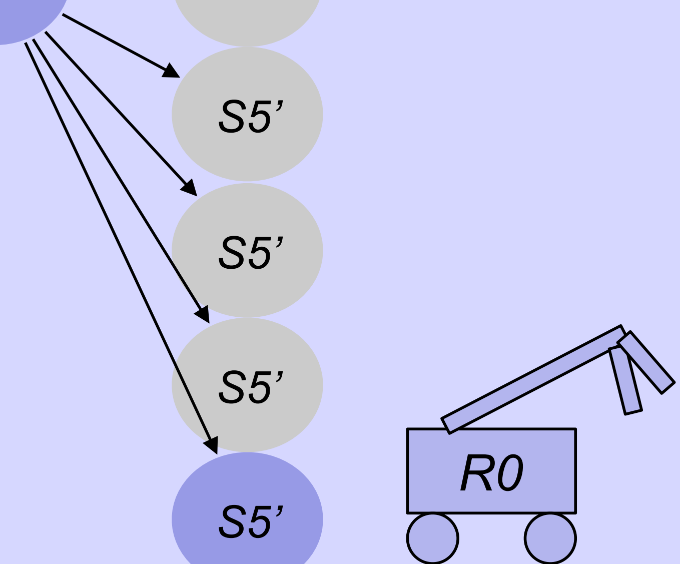
One solution is for the robot to build a staircase and then use it to reach the exit.



Initial state

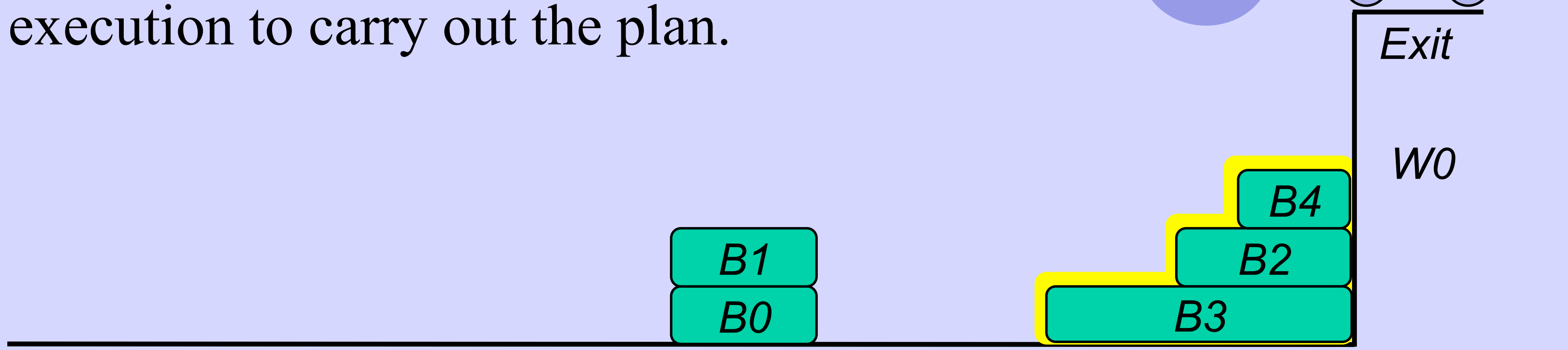
Building and Using a Staircase





In this scenario, the agent uses problem solving to generate a plan for building a staircase using known skills.

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



Goal state (S5’)

Extending ICARUS to Create / Use Tools

To support the creation and use tools, an intelligent agent must be able to:

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

We are developing a new architecture that incorporates these representational and processing abilities.

Challenge 3: Adaptive Planning / Execution

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

- Forward vs. backward search
- Depth-first search vs. iterative sampling
- 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.

Adaptive Planning and Execution

ICARUS cannot adapt its planning and execution methods, but we are devising 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 greater forms of adaptation and autonomy than possible previously.

Concluding Remarks

In this talk, I presented ICARUS, a unified theory of the human cognitive architecture that:

- Grounds high-level cognition in perception and action
- Treats categories and skills as separate cognitive entities
- Organizes skills and concepts in a hierarchical manner
- Combines teleoreactive control with means-ends analysis
- Acquires new skills from successful problem solving

ICARUS combines ideas from many different traditions in a unified account of high-level cognition.

Experience with complex synthetic domains have shown its strengths but also driven theoretical extensions.

End of Presentation