Biological Foundations of Reactive Paradigm

Dr. Divya Udayan J, Ph.D.(Konkuk University, S.Korea)

Department of CSE
Amrita School of Engineering, Amritapuri Campus,
Amrita Vishwa Vidyapeetham
Email: divyaudayanj@am.amrita.edu
Mobile: 9550797705

- This session attempts to set the stage for the **Reactive Paradigm** by recapping influential studies and discoveries and attempting to cast them in light of how they can contribute to robotic intelligence.
- Agency and computational theory

  - explore biological and cognitive sciences for insights in Intelligence.

Agency and Computational theory

One powerful means of conceptualizing the different systems is to think of an abstract intelligent system. Consider some thing we’ll call an ***agent.***

The agent is self-contained and independent.

It has its own “brains” and can interact with the world to make changes or to sense what is happening. It has self-awareness.

Under this definition, a person is an agent.

Likewise, a dog or a cat or a frog is an agent.

More importantly, an intelligent robot would be an agent, even certain kinds of web search engines which continue to look for new items of interest to appear, even after the user has logged off.

Agency is a concept in artificial intelligence that allows researchers to discuss the properties of intelligence without discussing the details of how the intelligence got in the particular agent.

In OOP terms, “agent” is the superclass and the classes of “person” and “robot” are derived from it.

Computational theory

- One helpful way of seeing correspondences is to decide the level at which these entities have something in common. The set of levels of commonality lead to what is often called a ***computational theory*** after David Marr.
- Marr was a neurophysiologist who tried to recast biological vision processes into new techniques for computer vision. The levels in a computational theory can be greatly simplified as:

- **Level 1: Existence proof of what can/should be done.**
- **Level 2: Decomposition of “what” into inputs, outputs, and transformations.**
- **Level 3: How to implement the process.**

Level 1: Existence proof of what can/should be done.

- Suppose a roboticist is interested in building a robot to search for survivors trapped in a building after an earthquake.
- The roboticist might consider animals which seek out humans. As anyone who has been camping knows, mosquitoes are very good at finding people. Mosquitoes provide an existence proof that it is possible for a computationally simple agent to find a human being using heat. At Level 1, agents can share a commonality of purpose or functionality.

Level 2: Decomposition of “what” into inputs, outputs, and transformations.

- This level can be thought of as creating a flow chart of “black boxes.” Each box represents a transformation of an input into an output.

Returning to the example of a mosquito, the roboticist might realize from biology that the mosquito finds humans by homing on the heat of a human (or any warm blooded animal). If the mosquito senses a hot area, it flies toward it. The roboticist can model this process as: input=thermal image, output=steering command.

The “black box” is how the mosquito transforms the input into the output. One good guess might be to take the centroid of the thermal image (the centroid weighted by the heat in each area of the image) and steer to that.

If the hot patch moves, the thermal image will change with the next sensory update, and a new steering command will be generated. This might not be exactly how the mosquito actually steers, but it presents an idea of how a robot could duplicate the functionality. Also notice that by focusing on the process rather than the implementation, a roboticist doesn’t have to worry about mosquitoes flying, while a search and rescue robot might have wheels. At Level 2, agents can exhibit common processes.

Level 3: How to implement the process.

- This level of the computational theory focuses on describing how each transformation, or black box, is implemented.

For example, in a mosquito, the steering commands might be implemented with a special type of neural network, while in a robot, it might be implemented with an algorithm which computes the angle between the centroid of heat and where the robot is currently pointing. Likewise, a researcher interested in thermal sensing might examine the mosquito to see how it is able to detect temperature differences in such a small package; electro-mechanical thermal sensors weigh close to a pound! At Level 3, agents may have little or no commonality in their implementation.

Robotic behaviour

- A *behavior* is a mapping of sensory inputs to a pattern of motor actions which then are used to achieve a task.
- Scientists who study animal behaviors are called *ethologists*.
- Behaviors can be divided into three broad categories:

- ❖ *Reflexive behaviors* are *stimulus-response (S-R)*, such as when your knee is tapped, it jerks upward. Essentially, reflexive behaviors are "hardwired"; neural circuits ensure that the stimulus is directly connected to the response in order to produce the fastest response time.
- ❖ *Reactive behaviors* are learned, and then consolidated to where they can be executed without conscious thought.

Any behavior that involves what is referred to in sports as "muscle memory" is usually a reactive behavior (e.g., riding a bike, skiing).

- ❖ *Conscious behaviors* are deliberative (assembling a robot kit, stringing together previously developed behaviors, etc.).

Reflexive behavior

- *reflexes*: where the response lasts only as long as the stimulus, and the response is proportional to the intensity of the stimulus.
- *taxes*: where the response is to move to a particular orientation. Baby turtles exhibit *tropotaxis*; they are hatched at night and move to the brightest light.
- *fixed-action patterns*: where the response continues for a longer duration than the stimulus.

Coordination and Control of Behaviors

The four ways to acquire a behavior are:

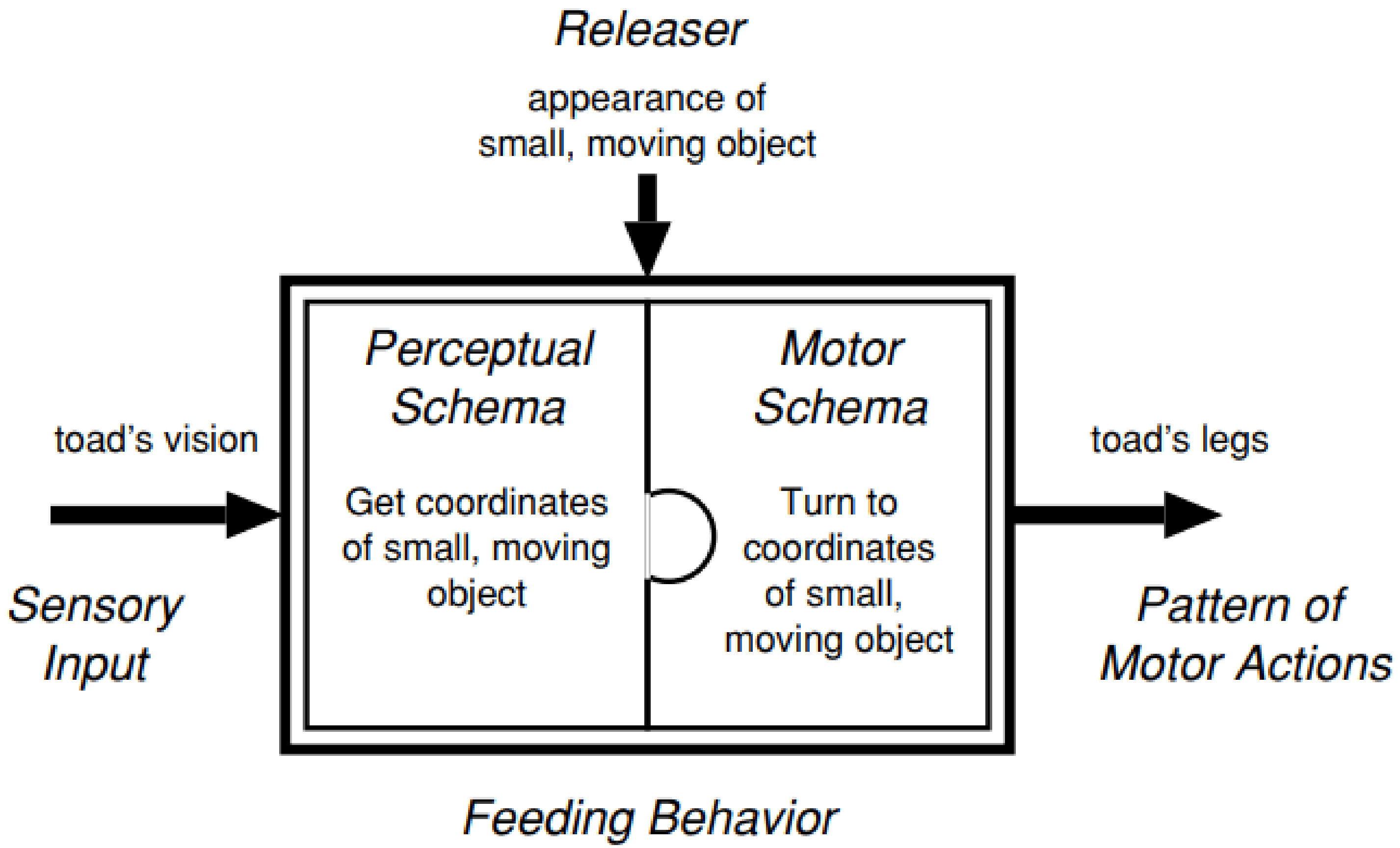
- to be born with a behavior (*innate*).
- to be born with a *sequence of innate behaviors*.
- to be born with behaviors that need some initialization (*innate with memory*).
- to *learn* a set of behaviors.

Behaviors and schema theory

- a *behavior* is a schema which is composed of a motor schema and a perceptual schema. The *motor schema* represents the template for the physical activity, the *perceptual schema* embodies the sensing.
- A behavior takes sensory inputs and produces motor actions as an output.
- A behavior can be represented as a schema, which is essentially an object-oriented programming construct. A behavior is activated by releasers.
- The transformation of sensory inputs into motor action outputs can be divided into two sub-processes: a perceptual schema and a motor schema.

Behavior::Schema

Data	
Methods	perceptual_schema() motor_schema()



Toad's feeding behavior represented as a behavior with schema theory

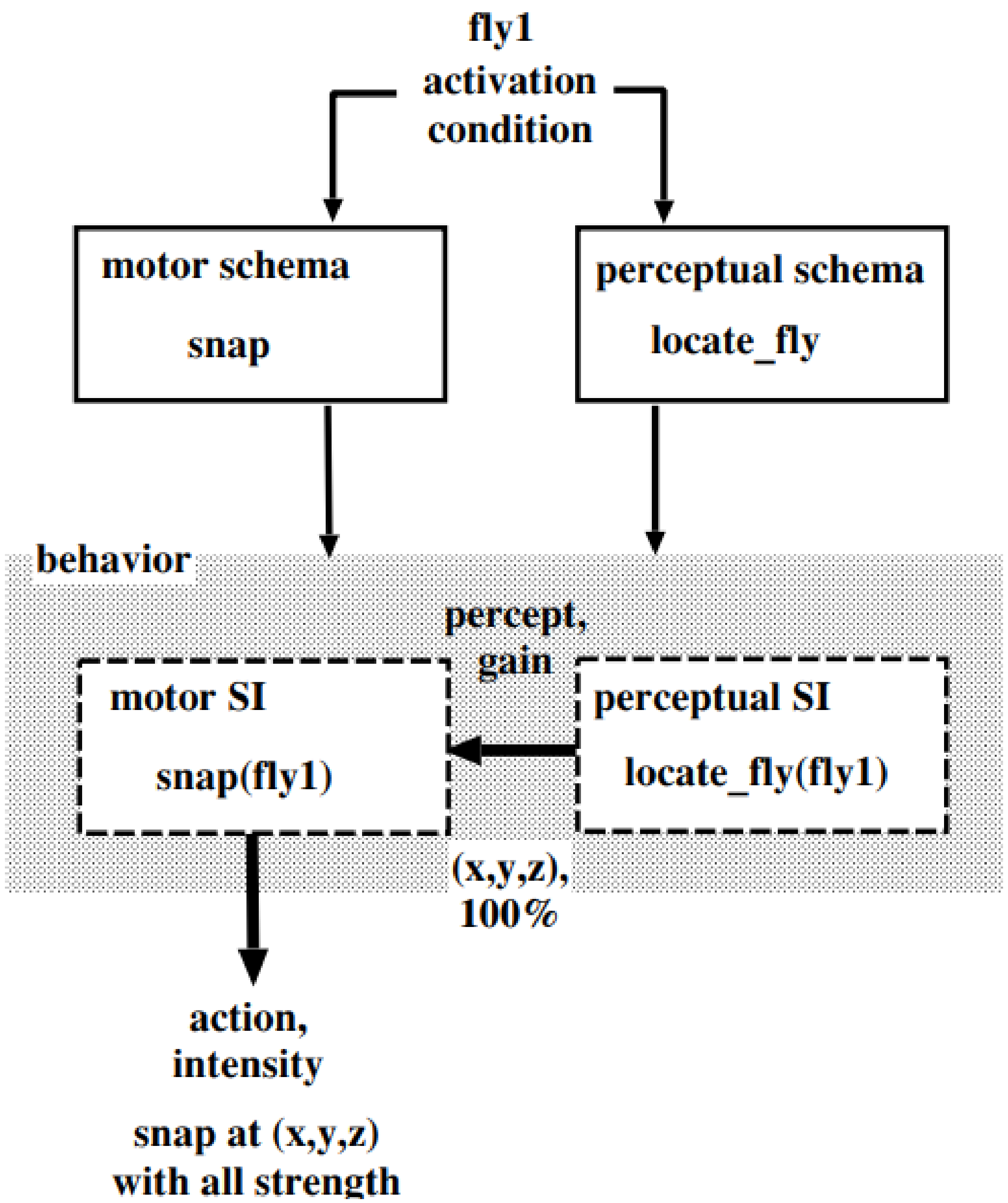


Figure Schema theory of a frog snapping at a fly.

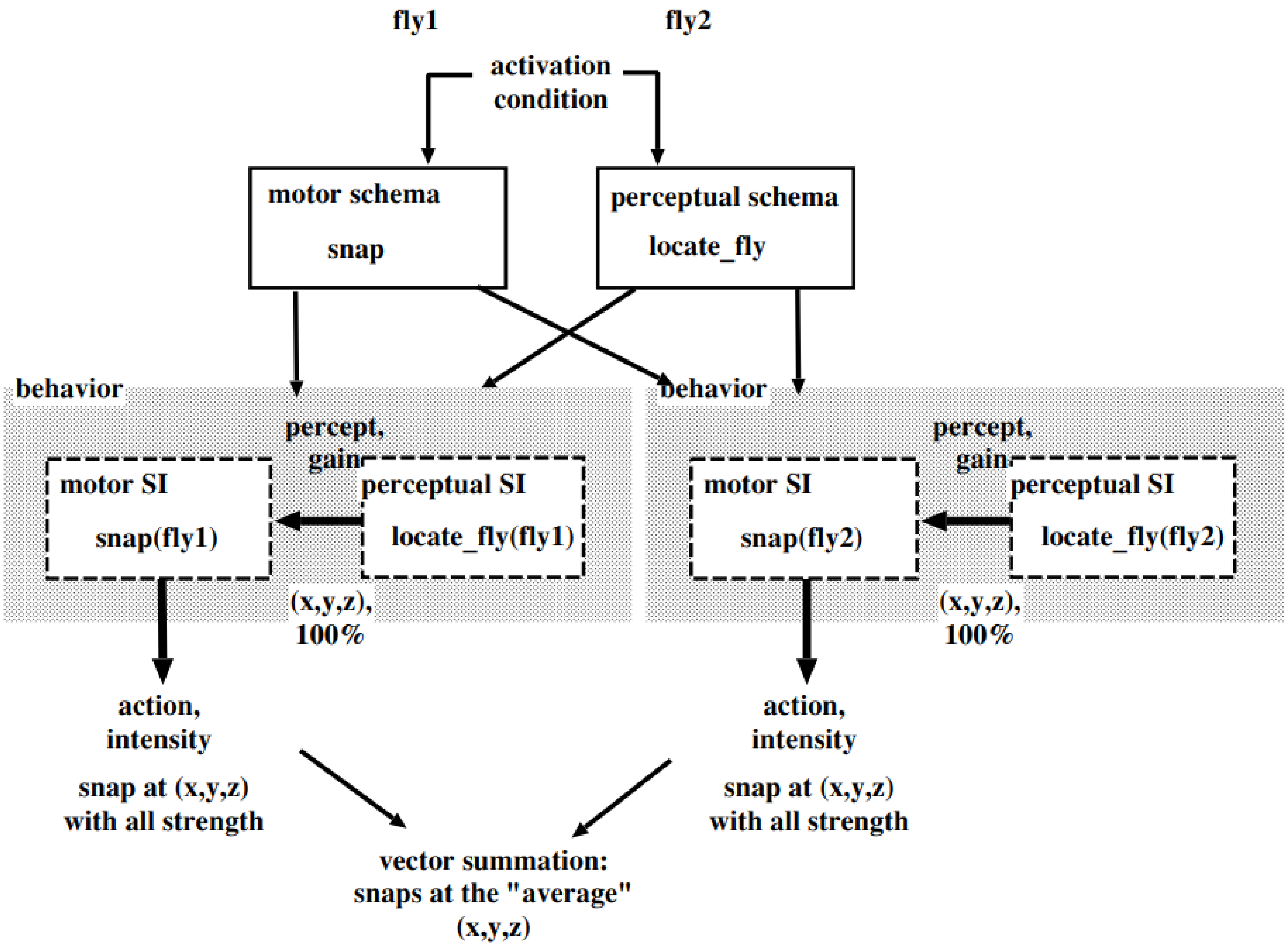


Figure Schema theory of a frog snapping at a fly when presented with two flies equidistant.

```
enum              Releaser={PRESENT, NOT_PRESENT};
Releaser          food, hungry, nursed;
while (TRUE) {
   food = sense();
   hungry = checkStateHunger();
   child = checkStateChild();
   if (hungry==PRESENT)
      searchForFood(); //sets food = PRESENT when done
   if (hungry==PRESENT && food==PRESENT)
      feed(); // sets hungry = NOT_PRESENT when done

  if (hungry== NOT_PRESENT && parent==PRESENT)
     nurse(); // set nursed = PRESENT when done
  if (nursed ==PRESENT)
     sleep();
}
```

Pseudocode

Principles and Issues in Transferring Insights to Robots

- Programs should decompose complex actions into independent behaviors, which tightly couple sensing and acting.
- Behaviors are inherently parallel and distributed.
- In order to simplify control and coordination of behaviors, an agent should rely on straightforward, boolean activation mechanisms (e.g. IRM)
- In order to simplify sensing, perception should filter sensing and consider only what is relevant to the behavior (action-oriented perception).
- Direct perception (affordances) reduces the computational complexity of sensing, and permits actions to occur without memory, inference, or interpretation.
- Behaviors are independent, but the output from one 1) may be combined with another to produce a resultant output, or 2) may serve to inhibit another (competing-cooperating).

Unresolved issues

- How to resolve conflicts between concurrent behaviors?
- When are explicit knowledge representations and memory necessary?
- How to set up and/or learn new sequences of behaviors?