Notes for Meeting 10
Hierarchical Task Networks

## Review of Reactive Control

The simplest approach to support extended agent behavior is to apply decision-making repeatedly.

This approach, known as reactive control, has been widely adopted and works well on narrowly defined tasks.

Many such tasks involve sensori-motor activities that require rapid responses that are conditioned on perception.

However, work in this paradigm ignores higher-level cognition and the tasks that it supports.

## Two Features of Complex Activity

Many human activities have two characteristics that methods for reactive control do not support:

- They have a hierarchical organization that decomposes complex tasks into simpler subtasks; and
- They are inherently sequential, in that these subtasks occur in a particular order.

These features impose both a structure and a continuity to activities that require additional mechanisms.

## Examples of Complex Activities

Consider some everyday activities that illustrate the importance of these features:

- grooming for work in the morning
- cooking dinner for guests
- doing multicolumn addition
- playing music pieces on the piano
- driving home after class

These activities have both hierarchical and sequential characteristics that distinguish them from simple reactive tasks.

## Advantages of Hierarchical Structures

Organizing extended activities in hierarchies has clear benefits, in that it decomposes complex tasks into components that are:

- simple, in that each component requires little content;
- modular, in that one can add or remove components easily;
- composable, in that one can combine these components dynamically;
- reusable, in that one can use a component in different tasks.

Simon (1975) has made functional arguments for the prevalence of hierarchical systems in natural and artificial systems.

## Two Types of Goal

The literature on hierarchical activities often refers to GOALS, but it is important to distinguish between:

- STATE-LIKE goals, which describe desired characteristics of the agent’s environment;
- ACTION-LIKE goals, which refer to activities that the agent wants to carry out.

Both forms of goal are valuable, but they describe different aspects of high-level cognition.

We will generally use "goals" to mean state-like structures and use "intention" to mean actio-like structures.

## The Paradigm of Hierarchical Task Networks

Much AI research on organized activity adopts a framework known as hierarchical task networks.

Long-term knowledge in this framework includes a set of procedures, methods, or skills, each of which specifies:

- a procedure or task name
- conditions for application
- an ordered set of subtasks
- tests for termination

This knowledge imposes a hierarchical, sequential, and conditional structure on behavior.

Procedural content may be complemented by conceptual knowledge that supports inference.

Most work adopts a logic-like representation and relies on pattern matching or unification.

## Test-Operate-Test Units

Miller, Galantar, and Pribram (1960) made the earliest proposal for organizing activity in a hierarchical manner.

Their book, Plans and the Structure of Behavior, outlined structures called Test-Operate-Test (TOTE) units that comprised:

- testing to see if an objective is met
- carrying out an operation to approach the objective
- repeating the test (looping) to see if task is done

They explored the role of such structures in many aspects of human cognition, including memory, problem solving, and communication.

Miller et al. did not implement their theory, but it foreshadowed many systems developed much later.

Uses of Hierarchical Task Networks

One can use hierarchical task networks in a variety of ways, including:

- executing complex but routine activities
- generating plans for future activities
- understanding other agents’ behaviors

Each takes advantage of hierarchical structures to constrain search and make processing tractable.

Examples of HTN Applications

Hierarchical task networks have been used in a variety of challenging applications, including:

- Tac-Air-Soar, which controlled synthetic fighter pilots in military training exercises;
- Bridge Baron (SHOP2), a computer Bridge player that is highly competitive; and
- CIRCA, intended to control insertion the Cassini Saturn orbiter.

Each application made use of the hierarchical knowledge organization, both at construction and at run time.

Some HTN Frameworks

A number of researchers have developed programming environments that incorporate HTN ideas:

- PRS (Georgeff & Lansky, 1987) and SPARC
- CIRCA (Musliner, 1993)
- SHOP (Nau et al., 1998) and SHOP2
- Icarus (Langley & Choi, 2006)

Each of these has been used to construct a variety of knowledge-based systems for complex activities.

Hierarchical Skills in Icarus

Icarus is a cognitive architecture that encodes procedural knowledge as a set of hierarchical skills, each with:

- a head that specifies the name and arguments
- percepts that describe observed objects
- conditions that state relations which must hold for applicability
- an ordered set of subskills or an executable action
- effects that describe the expected changes on completion

Together, a set of Icarus skills forms a hierarchical task network.

Because the syntax is similar to that of Prolog, we also refer to them as a teleoreactive logic program.

Skill Execution in Icarus

The Icarus execution module operates in cycles. Given a top-level skill instance to carry out, it:

- uses concepts to infer a current set of beliefs from percepts
- either
  - creates an intention based on applicable skill clause
  - executes an action if the current intention is primitive
  - returns to the parent intention if the current one is complete

This mechanism operates in a top-down manner that traverses the skill hierarchy across cycles.

The framework supports reactive control but also has a bias toward persistence on the current intention.

One can view Icarus as walking through an AND/OR tree, as do many theorem provers, but doing so over time.

Some Icarus Skills

```
((unstack ?block ?from)
 :percepts   ((block ?block) (block ?from))
 :conditions ((on ?block ?from) (not (on ?other ?block))
              (not (holding ?any)))
 :action     (*grasp-and-lift ?block)
 :effects    ((not (on ?block ?from)) (holding ?block)))

((make-clear ?block)
 :percepts   ((block ?block) (block ?on))
 :conditions ((on ?on ?block) (not (on ?other ?on))
              (not (holding ?any)))
 :subskills  ((unstack ?on ?block) (put-down ?on ?table))
 :effects    ((not (on ?other ?block)) (not (on ?on ?block))))
```

Some Icarus Intentions

```
((make-clear A)  ID: I1
 :bindings   ((?other . C) (?on . B) (?block . A))
 :conditions ((block A) (block B) (on B A) (on C B) (not (holding ?any)))
 :subskills  ((make-clear B) (unstack B A) (put-down B ?table))
 :effects    ((not (on C A)) (not (on B A)))

((make-clear B)  ID: I2  Parent: I1
 :bindings   ((?on . C) (?block . B))
 :conditions ((block B) (block C) (on C B) (not (on ?other C))
              (not (holding ?any)))
 :subskills  ((unstack C B) (put-down C ?table))
 :effects    ((not (on ?other B)) (not (on C B)))
```

Pros and Cons of Hierarchical Task Networks

What advantages do hierarchical task networks offer to AI and cognitive science?

- They provide additional constraints on theories of human cognition.
- They offer an effective means for organizing large knowledge-based systems that support complex activities.
- They remain the most scalable approach to planning because they limit search so well.

However, the framework also come with important disadvantages:

- Some HTN formalisms have a complex syntax that makes them hard to construct and maintain.
- They require domain experts to enter their knowledge manually, which is tedious, slow, and error prone.

HTNs are the analogue of expert systems for intelligent agents that operate over time, and thus inherit their strengths and weaknesses.

Assignments for Meeting 12
Production Systems

Read the articles:

- Young, R. M. (in press). Production systems in cognitive psychology. In N. J. Smelser & P.B. Baltes (eds.), International Encyclopedia of the Social and Behavioral Sciences. Pergamon. [required]

- Wikipedia entry on production systems. [required]

- Examine the third exercise (due 11:59 PM on 3/2/2011) and bring questions about it to class.