This Lecture

- Multi-Level Search
  - BlackBoard Based Problem Solving
  - Hearsay-II Speech Understanding System
Multi-Level vs Hierarchical Search

- Movement patterns among levels from lower to higher and back are not fixed
- Each level is a complete search space
- State (search nodes) held at each of the level do not go away when moving from one level to another
- Operators that modify the search space at one level may use information from multi-levels
Even More Complex Search

- Multi-Level & Bi-Directional
- Non-Monotonic Domain
- Cost of Control
  - Non-uniform and costly with respect to node generation
- Non-uniform cost of operator application
Blackboard Problem Solving Model: Cooperating Experts

A Set of Knowledge Sources (KSs) Incrementally adding knowledge/hypotheses/partial solutions through a shared multi-level structure called the blackboard – think of a group problem-solving process.
Blackboard Structure

- Partitioned into distinct information levels
  - Each level holds a *different representation of the problem space*, with its own primitive elements

- KS decomposition relates naturally to one or a few information levels
  - Localization of KS activity

- Levels form a *loose hierarchical* structure
  - Abstraction of elements of the next lower level
  - An *a priori* framework of a plan for problem solving
  - Analysis/synthesis action between levels
Example BlackBoard System

<table>
<thead>
<tr>
<th>LEVELS</th>
<th>KNOWLEDGE SOURCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE INTERFACE</td>
<td></td>
</tr>
<tr>
<td>PHRASE</td>
<td>Parse, Predict, Contact, Stop</td>
</tr>
<tr>
<td>WORD SEQUENCE</td>
<td>Word-seq, Word-seq-ctl</td>
</tr>
<tr>
<td>WORD</td>
<td>Mow, Verify</td>
</tr>
<tr>
<td>SYLLABLE</td>
<td>Pom</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>Seg</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>Rpol</td>
</tr>
</tbody>
</table>
Blackboard Nodes

- Nodes (partial solutions) exist at particular level and associated with a primitive element
  - Each level has associated with it a vocabulary that defines the range of primitive elements
  - Each node has a set of attributes that can be level-dependent
- Nodes can be related to other nodes at the same or different levels
  - Explicitly through links and Implicitly based on node attributes
- Nodes may represent alternative competing partial solutions
  - Permits direct comparison of alternative search paths
  - Integrated representation of alternative search paths
Implicit linking of Nodes through Time
Explicit Linking of Nodes

- **PHRASAL**
  - ‘question’ (SEQ)
  - ‘modal question’ (SEQ)
  - ‘would’ (SEQ)
  - ‘you’ (OPTION)
  - ‘will’ (SEQ)

- **LEXICAL**
  - ‘you1’ (SEQ)
  - ‘you2’ (SEQ)
  - ‘D’
  - ‘L’
  - ‘J’
  - ‘AX’

- **SURFACE-PHONEMIC**
  - ‘Y’
Blackboard Control

- Application of knowledge is triggered by current state of blackboard (data directed)
- Based on blackboard events:
  - A change to the blackboard (addition, deletion, modification)
  - Non-occurrence of an expected change
- Trigger evaluation of preconditions of relevant KS
- KS whose preconditions are satisfied is instantiated with appropriate context and placed on scheduling queue (agenda, open list)
- Focus of attention mechanism evaluates agenda and chooses for execution KS(s) that are most promising for further system progress
- KS(s) are executed and alter state of blackboard, trigger new blackboard events
Hearsay-II Architecture

Key:
- Program modules
- Databases

Data flow
Control flow
Knowledge Source Structure (KS)

- **Trigger** specifies a set of event predicates that need to be true for KS to be considered for execution.
- **Precondition** specifies a set of state predicates that need to be true for KS to execute.
- **Context** specifies where KS will be applied (KSAR).
- **Obviations** condition specifies a set of state-based predicates that if all true indicate KS/Context is to be removed from agenda.
- **KS action** is arbitrarily complex program.
- **Declarative Information** used for scheduling.
An Example Knowledge Source: Yoke KS *(Hayes-Roth, ‘86)*

**Name:** Yoke-Structures

**Trigger Conditions:**

- $(\text{EVENT-LEVEL-IS} \text{ STRUCTURAL SOLID})$
- $(\text{EVENT-TYPE-IS} \text{ Modify})$
- $(\text{CHANGED-ATTRIBUTE-IS} \text{ APPLIED-CONSTRAINTS})$
- $(\text{SET Possible-Combinations (Get-Possible-Combinations \text{TRIGGER-OBJECT})})$

**Context Variables:**

- $(\text{PS-Anchor Anchoree1 Anchoree2} \text{ Possible Combinations})$

**Preconditions:**

- $(\text{SET Yoking-Info (There-is-Yoking-Info-For Anchoree1 Anchoree2)})$
- $(\text{VALUE Anchoree1 ‘Applied-Constraints})$
- $(\text{VALUE Anchoree2 ‘Applied-Constraints})$

**Obviation Conditions:** NIL

**KS Variables:**

- $(\text{NewLocLabelForAnchoree1 (Generate-LocTableLabel PS-Anchor Anchoree1 1 LENGTH \text{Anchoree1 ‘Legal Orientations})})$
- $(\text{NewLocLabelForAnchoree2 (Generate-LocTableLabel PS-Anchor Anchoree2 2 LENGTH \text{Anchoree2 ‘Legal-Orientations})})$
- $(\text{Descriptor 1(Make-Descriptor-For-Yoke PS-Anchor Anchoree1 Anchoree2)})$
- $(\text{Descriptor2(Make--Descriptor-For-Yoke PS-Anchor Anchoree2 Anchoree1)})$

**Actions:**

- $(1 \text{ T})$
  - $(\text{EXECUTE (SET YokeResult (Yoke-Structures PS-Anchor Anchoree1 Anchoree2 CADAR \text{LAST \text{Anchoree1 ‘Legal-Orientations})})})$
  - $(\text{CADAR \text{LAST \text{Anchoree2 ‘Legal-Orientations})})$
  - $(\text{NewLocLabelForAnchoree1 Descriptor1})$
  - $(\text{NewLocLabelForAnchoree2 Descriptor2})$
  - $(\text{LENGTH Yoking-Info Yoking-Info VanderWaalsCheck?)})$

- $(2 \text{ T})$...
A Yoke-Structures KSAR. Yoke-Structures has been triggered by a modification of helix1’s applied-constraints. This KSAR represents the blackboard context in which helices 2 and 3 have constraints with one another and with helix1. Since both helices have previously identified locations, the KSAR is executable.
Generic Data Interpretation KSs

Synthesize

Level $i$

Level $i-1$

Merge

Eliminate

Extend/Join

Predict

Verify

precondition

data retrieved

new/modified hypothesis

V. Lesser; CS683, F10
Issues in BB Control

- How to decide which of many potential KS instantiations are the most preferred
  - How to compare apples and oranges
  - Different levels and parts of search space

- How to control the potential for combinatorial explosion of hypotheses on the blackboard
  - Overhead significantly increases as large number of partial solutions are placed on BB

- How to decide when the system has an acceptable solution -- search termination criteria
  - Non-monotonic character of search
Hearsay-II Speech Understanding System

Information Retrieval Based on Interpreting Connected Speech

Sample sentences:

“Which abstracts refer to theory of computation?”

“List those articles.”

“What has McCarthy written since 1974?”
Why Connected Speech Understanding is Difficult

- Large search space
  - \( \approx 10^8 \) legal sentences

- Uncertainty and Approximate Knowledge
  - Sensors
  - Acoustic phonetic knowledge

- Knowledge costly to apply

- Difficult to subdivide problem solving

- Interacting constraints
  - Co-articulation phenomenon

- Wide variety of knowledge needs to be applied
Continuous speech blurs word boundaries and changes pronunciations...

*Masking in Time-Domain: Co-Articulation*

How the words look when spoken continuously

How each word would look when spoken in isolation
## Functional Description of the Speech-Understanding KSs

<table>
<thead>
<tr>
<th>Levels</th>
<th>Knowledge Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td></td>
</tr>
<tr>
<td>Phrase</td>
<td>Parse, Semant</td>
</tr>
<tr>
<td>Word Sequence</td>
<td>Word-seq, Predict, Stop</td>
</tr>
<tr>
<td>Word</td>
<td>Maw, Verify, Word-seq-ctl</td>
</tr>
<tr>
<td>Syllable</td>
<td>Pom</td>
</tr>
<tr>
<td>Segment</td>
<td>Seg</td>
</tr>
<tr>
<td>Parameter</td>
<td>Rpol</td>
</tr>
</tbody>
</table>
Hearsay-II Knowledge Sources: Domain and Control

- **Signal acquisition, parameter extraction, segmentation and labeling**
  - SEG: digitizes the signal, measures parameters and produces a labeled segmentation

- **Word spotting**
  - POM: creates syllable-class hypotheses from segments
  - MOW: creates word hypotheses from syllable classes
  - WORD-CTL: controls the number of word hypotheses that MOW creates

- **Phrase-island generation**
  - WORD-SEQ: creates word-sequence hypotheses that represent potential phrases from word hypotheses and weak grammatical knowledge
  - WORD-SEQ-CTL: controls the number of hypotheses that WORD-SEQ creates
  - PARSE: attempts to parse a word sequence and, if successful, creates a phrase hypothesis from it
**Phrase extending**

- PREDICT: predicts all possible words that might syntactically precede or follow a given phrase
- VERIFY: rates the consistency between segment hypotheses and a contiguous word-phrase pair
- CONCAT: creates a phrase hypothesis from a verified contiguous word-phrase pair

**Rating, halting, and interpretation**

- RPOL: rates the credibility of each new or modified hypothesis, using information placed on the hypothesis by other KSs
- STOP: decides to halt processing (detects a complete sentence with a sufficiently high rating, or notes the system has exhausted its available resources) and selects the best phrase hypothesis or set of complementary phrase hypotheses as the output
- SEMANT: generates an unambiguous interpretation for the information-retrieval system which the user has queried
Abstract State Space Through Approximate Knowledge

Approximate $K_1$ by $\tilde{K}_1$

$\rightarrow$ more errors/uncertainty

Correct with $\Delta K_2$

Win if $\text{Cost}(\tilde{K}_1 + \Delta K_2) < \text{Cost}(K_1)$

$K_1 = \text{PARSE}$

$\tilde{K}_1 = \text{WORD-SEQ'}s$ matrix

$\Delta K_2 = \text{PARSE} \text{ applied to sequences}$
Basic Control Cycle

- Scheduler invokes highest-rated KS with specific context
  - Check before running whether precondition still valid

- KS modifies blackboard
  - Focus-of-control database is updated
  - Relevant precondition procedures are notified

- Relevant precondition procedures are evaluated
  - New KS instances are posted on scheduler with context

- Priority of new KS instances are calculated and those old ones are affected by change in control database
Control Strategy

- **Bottom-up processing to word level**
  - Sufficient reliability for opportunistic processing

- **KS as generator functions**
  - Limited generation of alternatives
  - Retriggered to generate additional hypotheses as search stagnates

- **Select sequence of word hypotheses as candidates for phrase hypotheses**

- **Opportunistic search at Phrase Level**
  - Islands-of-reliability
  - Integrate partial phrases coming from different directions
  - Fill out words not bottom-hypothesized
Control Strategy, cont’d

- If search not progressing, retrigger KSs for more hypotheses
  - Implement with control KSs stimulated by agenda

- Search termination
  - Special mode when a spanning hypothesis is constructed of sufficient credibility
  - Use hypotheses to constrain further search
Hearsay-II Trace as a Search Graph

V. Lesser; CS683, F10

A: [ARE]
B: EIGHT
C: SHAW+AND+MARVIN
D: AND+FELDMAN
A-1: [ARE+REDAY]
A-2: [ARE+ANY]
A-2-1: [ARE+ANY]
B: EIGHT
C: SHAW+AND+MARVIN
D: AND+FELDMAN
A-1: [ARE+REDAY]
A-2: [ARE+ANY]
A-2-1: [ARE+ANY]
A-1 and D-1-1: GOAL

[[Diagram of the search graph with nodes and edges labeled with terms and indices.]]
11. KS: PREDICT & VERIFY*

Stimulus: [+ ARE* (phrase)
Action: Predict (from the grammar) 292 words following.
Reject (using the acoustic information) 277 of them. The four highest-rated of the fifteen verified words are
- REDDY (85,26:52),
- ANY* (65,24:49),
- HUGH (55,30:39), and
- YOU (55, 28:39).

12. KS: CONCAT

Stimulus: [+ ARE* (phrase), REDDY (word)
Action: Create phrase: [+ ARE + REDDY(91,0:52)

13. KS: CONCAT*

Stimulus: [+ ARE* (phrase), ANY* (word)
Action: Create phrase: [+ ARE + ANY* (86,0:49)
14. KS:PREDICT & VERIFY*

Stimulus: AND + FELDMAN +] (phrase)
Action: Predict 100 words preceding. Reject 76 of them. The best of the verified 24 (in descending rating order) are
- FEIGENBAUM* (80,72:150),
- WEIZENBAUM (70,72:150),
- ULLMAN (70,116:150),
- NORMAN (70, 108:150), and
- NEWBORN (70, 108:150).

15. KS:PREDICT & VERIFY

Stimulus: EIGHT (phrase)
Action: Predict the word NINE following and verify it (80,52:82).
Predict SEVEN preceding, but reject this because of mismatch with the acoustic segments.
Trace of Hearsay-II, p.6

16. KS: CONCAT*
   Stimulus: FEIGENBAUM* (word), AND + FELDMAN +]* (phrase)
   Action: Create phrase: FEIGENBAUM + AND + FELDMAN +]* (85,72:225)

17. KS: PREDICT & VERIFY*
   Stimulus: FEIGENBAUM + AND + FELDMAN +]* (phrase)
   Action: Predict eight preceding words. Reject one (DISCUSS).
   Find two already on the blackboard:
   - BY* (80,52:72) and
   - ABOUT (75,48:72).
   Verify five others:
   - NOT(75,49:82),
   - ED(75,67:72),
   - CITE(70,49:82),
   - QUOTE(70,49:82),
   - CITES(65,49:82).
Trace of Hearsay-II, *p.10*

25. **KS: CONCAT**

Stimulus: ANY* (word),

BY + FEIGENBAUM + AND + FELDMAN +]* (phrase)

Action: Create phrase:

ANY + BY + FEIGENBAUM + AND + FELDMAN +]* (82,24:225)

[+ ARE + ANY + BY + FEIGENBAUM + AND + FELDMAN +]* (85,0:225) is also created, from

[+ ARE + ANY and BY + FEIGENBAUM + AND + FELDMAN +].

26. **KS: STOP**

Stimulus: [+ ARE + ANY + BY + FEIGENBAUM + AND + FELDMAN +]*

(complete sentence)

Action: Deactivation of several dozen competing hypotheses.
27. **KS:PREDICT & VERIFY**
   Stimulus: [+ ARE + REDDY]

28. **KS:CONCAT**
   Stimulus: [+ARE* (phrase), HUGH (word)]
   Action: Create phrase: [+ HUGH + YOU (83, 0:39)]

29. **KS:CONCAT**
   Stimulus: [+ ARE (phrase), YOU (word)]
   Action: Create phrase: [+ ARE + YOU (83,0:39)]

30. **KS:CONCAT**
    Stimulus: [+ ARE + REDDY (phrase), CITED (word)]
    Action: Create phrase: [+ ARE + REDDY + CITED (80,0:86)]
38. KS:STOP
   Stimulus: Stagnation
   Action: Stop search and accept
   [+ ARE + ANY + BY + FEIGENBAUM + AND + FELDMAN +]*

39. KS:SEMANT*
   Stimulus: Recognized utterance:
   [+ ARE + ANY + BY + FEIGENBAUM + AND + FELDMAN +]*
   Action: Generate an interpretation for the database retrieval system.
Advantages of Blackboards

- Many and diverse sources of knowledge can participate in forming and modifying the emerging solution
  - Linking partial solutions at the same level of abstraction and those at different levels

- Each knowledge source can be implemented using
  - The most appropriate representation of its knowledge
  - The most efficient inference engine for its reasoning
Advantages of Blackboards Continued

- **No a priori commitment to the order of inferencing steps**
  - Bottom-up or Top-down
  - Data-directed or Model/Goal directed

- **Each knowledge source can contribute opportunistically**
  since each has continual access to the current state of the search.
  - The right knowledge can be applied at the right time.
  - Permits Differential diagnosis

- **Control Knowledge can exploit a global view of the emerging set of potential solutions and their relationships**
Next Lecture (Wed Oct 20)

- Sequential Decision Problems
  - Markov Decision Processes (MDP)
  - Partial Orderded MDP (POMDP)

Good Luck on Exam on Monday