Beyond Chunking

John Laird

University of Michigan

23rd Soar Workshop

Architectural Learning

- Automatic & ubiquitous
- Task independent
- Fixed algorithm
- Bounded processing
- No competition with reasoning
- Single experience-based
- Restricted to *active or bounded history of* knowledge

- Examples:
  - Chunking
  - Rule tuning in Act-R

Deliberate/Reflective Learning

- Knowledge engages and controls
- Can be task specific
- Can change with learning
- No fixed bound on processing
- Competes with other processing
- Can draw on multiple examples
- Can bring arbitrary knowledge

- Examples:
  - Task acquisition
  - Learning by instruction
  - Learning by analogy
  - Recovery from incorrect knowledge

Gospel: Learning in Soar

- Chunking is only architectural learning mechanism
- All other learning is reflective and uses chunking

- Why is this good?
  - All other learning in knowledge based and can be controlled
  - Forces us to learn the boundaries of chunking
    - Demonstrated that chunking is complete
  - Avoid *interference* between learning mechanisms
    - When should one learning mechanism be used vs. another
  - Couldn't think of other architectural learning mechanisms that fit the problem space computational model

Indications this Might be a Mistake

- After 20 years, chunking isn't ubiquitous
- Many types of learning are very difficult with just chunking
  - Episodic learning
  - Reward-based learning
  - Concept acquisition

- Seduced by completeness of chunking and Soar
  - Ability to convert problem search => knowledge search
  - Didn't think any other learning was necessary

Why Beyond Chunking?

- *Learning competes with task at hand*
  - Processing is done only to learn, not because to perform the task
  - Chunking requires deliberate processing (operators) to
    - record experiences
    - capture statistical regularities
    - learn new concepts (data chunking)
- Hard to implement, hard to use

Other Architectural Learning Mechanisms?

- Other Sources of Knowledge:
  - Records of past experiences: episodes
  - Statistical correlations of success: reinforcement learning
  - Correlations of facts: semantic learning
  - ??

Requirements

- Automatic, task independent, resource bounded, based on current processing, incremental, …
- Doesn't create structures that slow system down
- Doesn't interfere with chunking
- Must fit into problem space computational model:
  - elaborations, proposals, selection, applications

- We assume all learned knowledge is rules
- How determine the conditions automatically?
  - Complete state?
  - Usage patterns (activation, backtracing, …)?
  - ??