Building Agents Quickly

John E. Laird

Randy Jones, Mike van Lent, and Paul Nielsen

University of Michigan

Soar Workshop 17

June 28, 1997

Motivation

- Large bodies of knowledge are required for high-fidelity behavior.
  - 4,800 rules for FWA / IFOR = medium-fidelity.
- Need to quickly add and modify knowledge.
  - New tactics, new capabilities, new opponents.
- Current approaches are very labor intensive.
  - > 10 person years for TacAir-Soar.
  - Also due to uncertainty about requirements, environment, and interfaces.
- Little prior work on automated procedural knowledge acquisition.

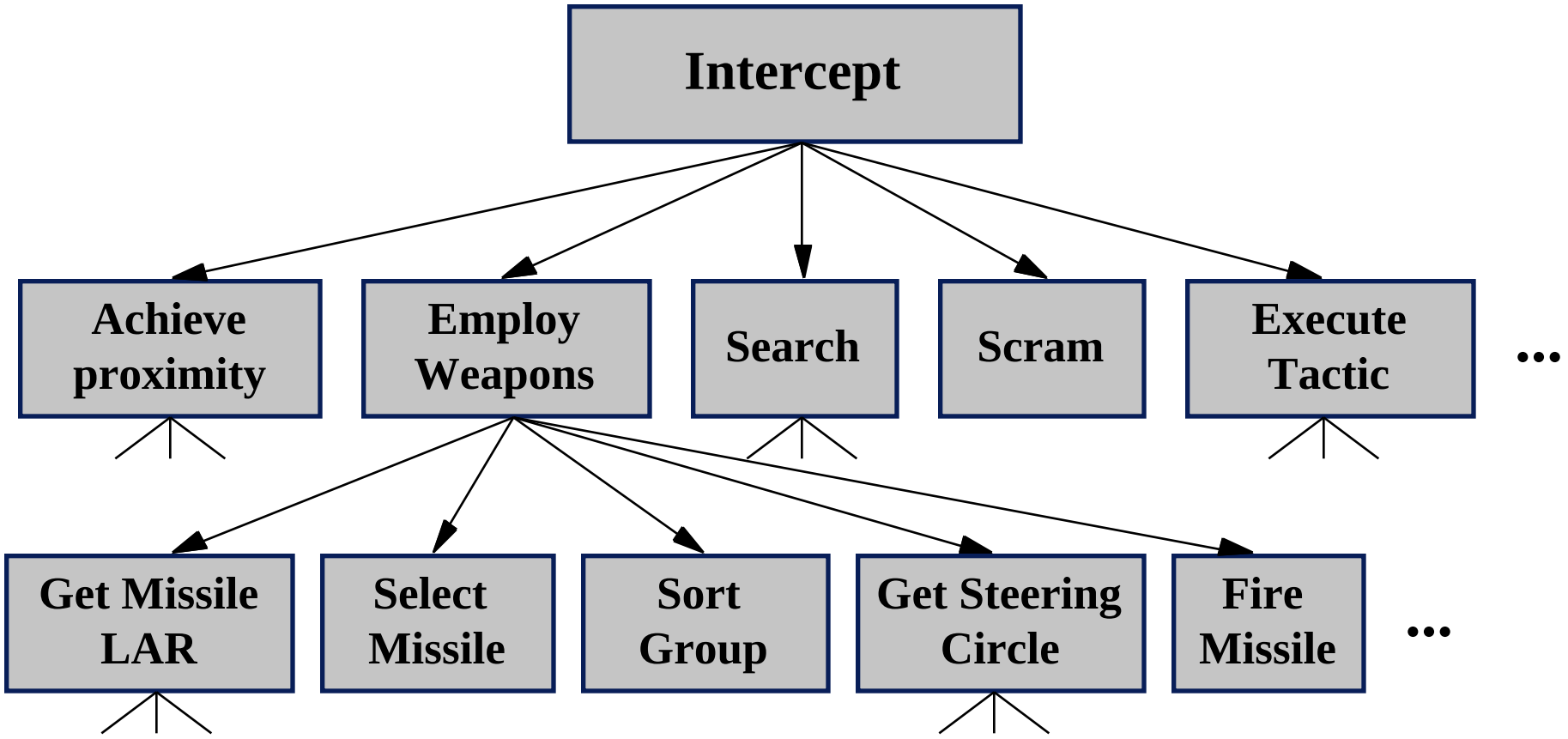
Proposed Approach

- Extract knowledge from a variety of sources:
  - Interviews with expert.
  - Observing behavior of expert.
  - Instruction and critique from expert.
  - Analysis of existing knowledge.
- Use analytical and empirical learning techniques.
- Learn while actively trying to do the task.
  - Provides context for interpreting expert behavior & instruction. Grounds analysis of existing knowledge.

The Domain

- Air-to-air tactics in TacAir-Soar.
- Tactics are intermediate operators.
  - Used to achieve combination of longer term goals such as defend a strike package.
    - Know the higher level goal.
  - Composed of combinations of more primitive operators such as turn to a heading, shoot missile, adjust radar.
    - Don’t need to learn primitive operators for controlling vehicle.
    - Need to learn combinations of the more primitive operators and the parameters for those operators.

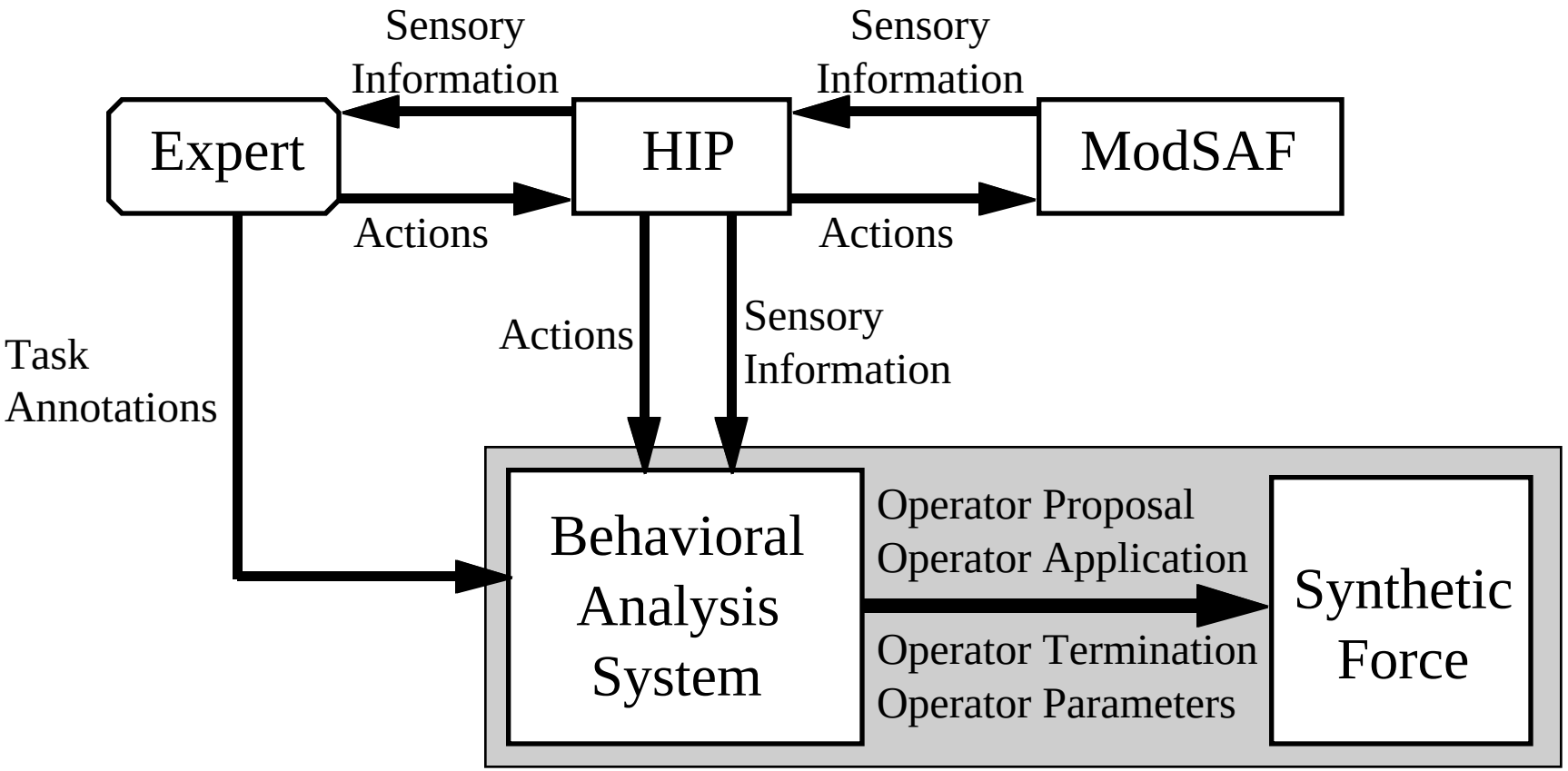
Structure of Tactics



Learning by Observation

- Gather detailed behavioral performance data of a human performing the activity.
  - Tap into interface between human and environment and record available sensory data and actions taken.
- Induce the knowledge that the human is using.
  - “Behavioral Cloning” demonstrated by Sammut et al. for taking-off & landing of a simulated plane.
- Extend by allowing expert to annotate goal / operator selections and terminations.
  - Mike van Lent will discuss this approach in detail.

Behavior Cloning Structure



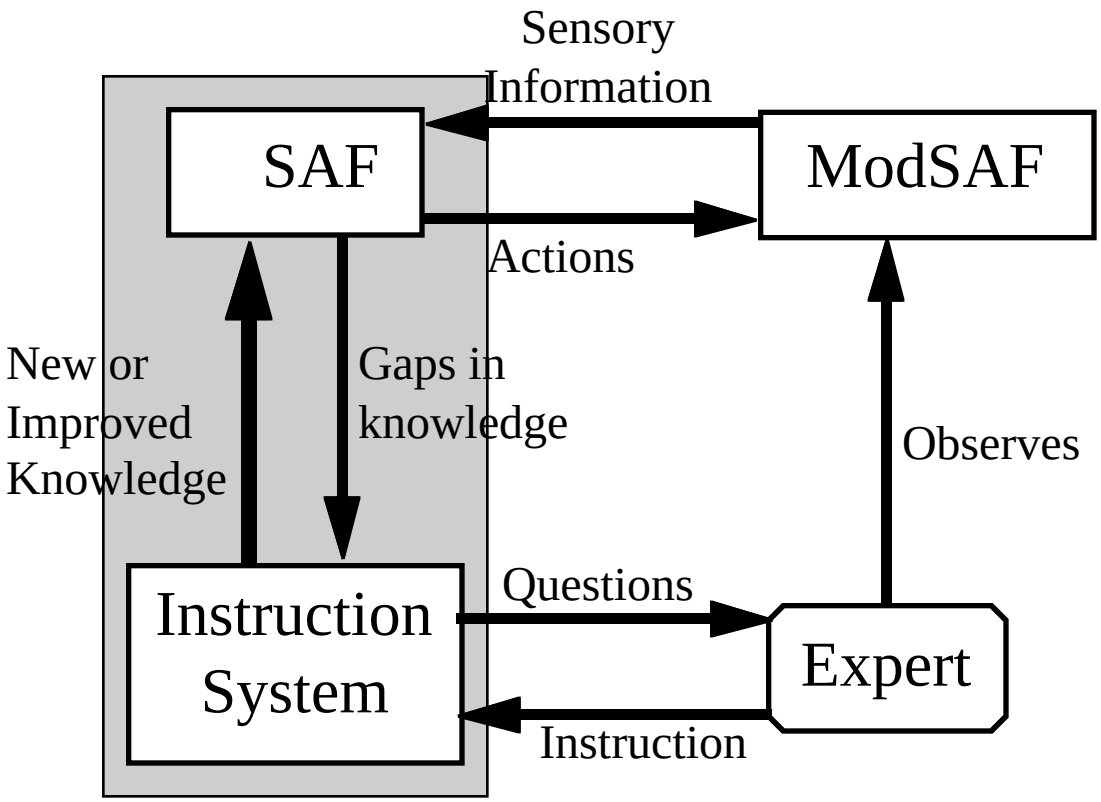
Hypothetical Interaction

- Human breaks off racetrack and heads in general direction of bogey.
- "Starting intercept bogey."
- Turns a few degrees off of collision.
- "Starting turn to achieve lateral separation."
- Turns again.
- "Achieved the lateral separation. Starting turn to maintain lateral separation."
- Turns toward bogey,changes radar modes.
- "Achieved LAR, attempting to get radar lock."

Learning by Instruction

- System attempts the task, but requests instruction from expert when unsure of what to do.
- Instructor tells system next step to perform.
- System attempts to generalize through self-explation. Otherwise, learns by rote.
- Demonstrated in Instructo-Soar for simple robot block manipulation task.
- Supported many types of instruction:
  - Operator proposals, selection, application, termination.
  - Positive, negatives, hypotheticals, hints on features.

Instruction System Structure



Hypothetical Interaction

- How do I do an intercept?
  - First, head toward the bogey.
- OK.
- I'm headed toward the bogey. Now what?
  - Turn away from the bogey 20 degrees.
- Why?
  - It will put you in position for a second shot if the first one misses.
- Huh?

Strengths and Weaknesses

- Strengths
  - Soar drives interaction based on what it needs to learn.
  - Expert needed only when there is a lack of knowledge.
  - Can learn most types of knowledge used by agents.
- Weaknesses
  - Haven't used in dynamic domains.
    - Can't learn "opportunistic" operators.
    - Situation may change before system can ask a question.
  - Doesn't allow expert to interrupt and critique.
    - Assumes prior knowledge is correct.
  - Requires language interface for communication.

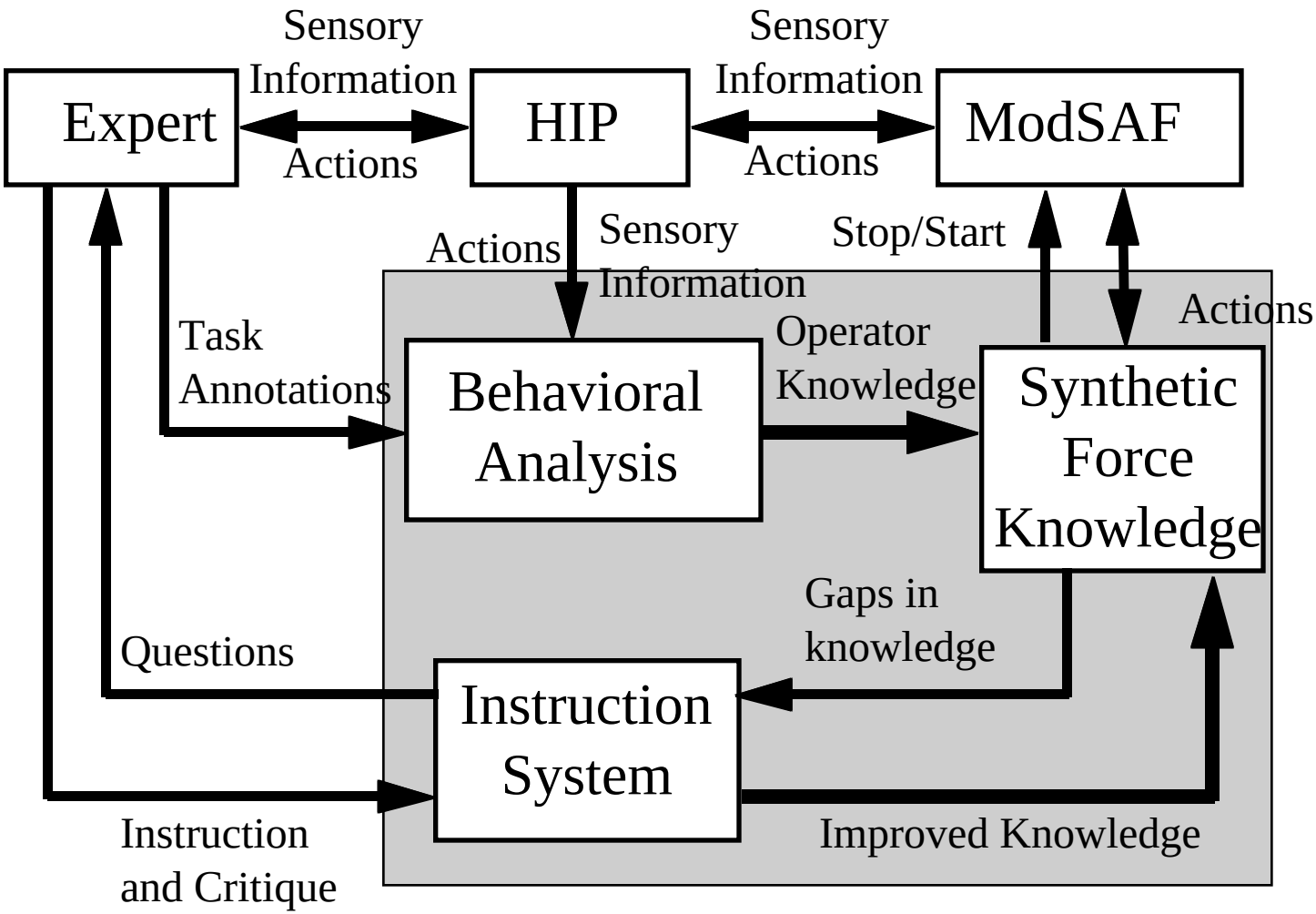
Proposed Extensions

- Allow expert to interrupt system and critique.
  - Will allow it to learn opportunistic operators.
    - “Go beam now!”
  - Will allow it to correct overgeneral knowledge.
    - “Don’t try to climb above a MiG-29.”
- Give control of simulation to Soar.
  - Allow it to stop simulation when it needs extra time for instruction or planning.

Integration of Approaches

- Allow Soar to stop simulation and ask questions during learning by observation.
  - "Why did you do that instead of X?"
  - "Won't that cause this goal to be violated?"
  - "Are we finished trying to achieve LAR?"
  - "How did you know LAR was achieved?"
- Allow human to "jump" in during instruction.
  - "No, no, you are doing it all wrong and I can't explain why. Here is how you do it! Watch me."

System Structure



Nuggets

- Doesn't try to have the computer do too much.
  - No magic discovery of new knowledge.
    - Use human to provide as much knowledge as possible.
  - No new, undiscovered learning approaches.
- Doesn't try to have the humans do too much.
  - Computer analyzes detailed data and knowledge available to it.
- Building on prior successes.
  - TacAir-Soar, Instructo-Soar, IMPROV, SCA.

Coal

- Language interactions.
  - Will probably use a greatly restricted grammar and artificial language.
- Techniques may not scale to complex, dynamic domains.
  - Must build up initial planning knowledge.
- Very ambitious.

Why use Soar?

- Have a task independent framework that defines what needs to be learned for any task.
- Each operator can be learned independently.
  - Don't need to learn tactic / mission / doctrine all at once.
- Each operator part can be learned independently.
  - Can learn proposal from one source and fill in application and termination from another.
- Each operator part can be learned incrementally.
  - Can learn some of the proposals before others.
- Can use prior knowledge when learning.
  - Key component of Instructo-Soar, Improv.