Soar Technology
Thinking *inside* the box.

Learning New Air Combat Tactics With Cascade

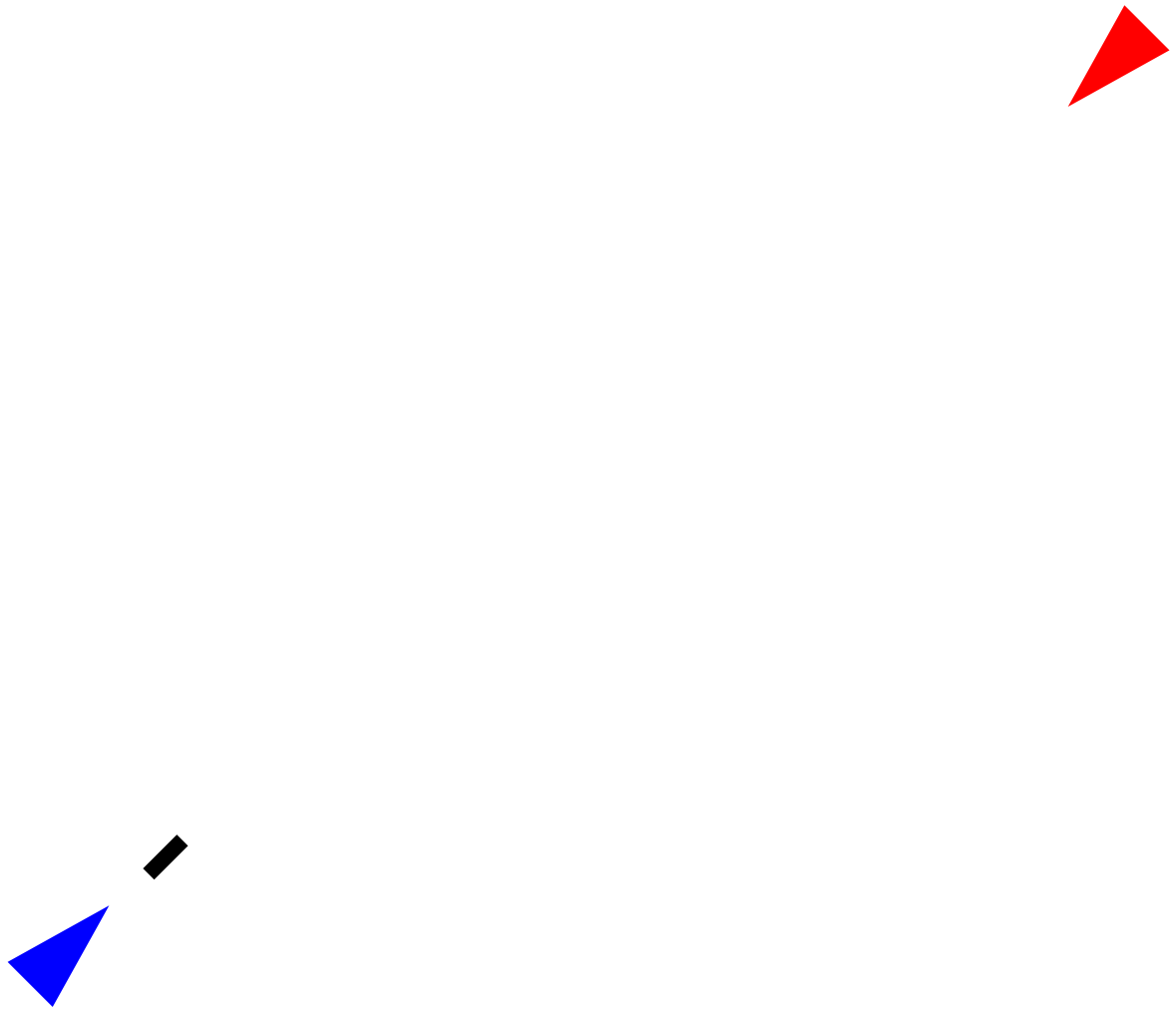
Randolph M. Jones
Keith Knudsen
Laura Hamel

Project Overview

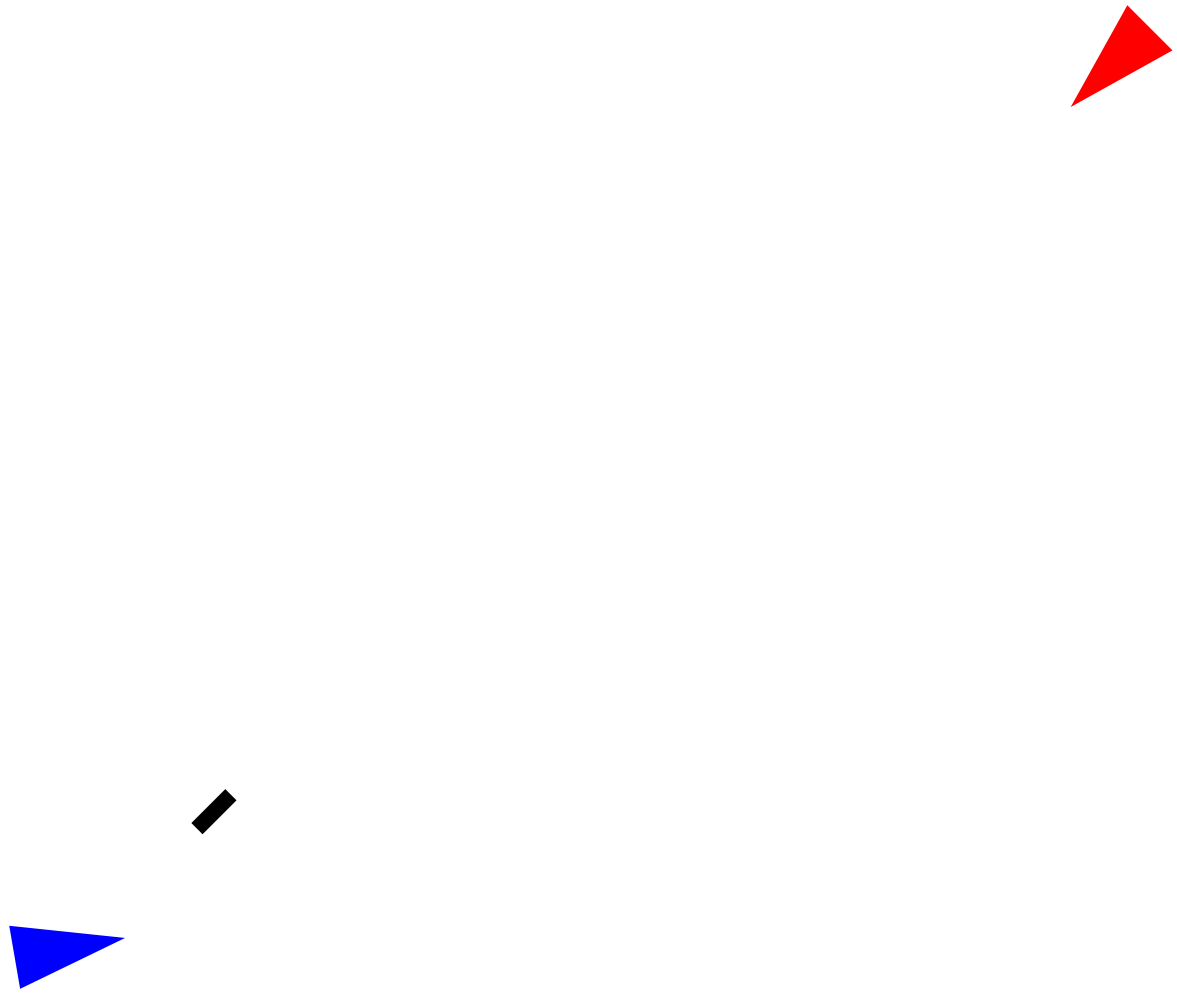
- Goal
  - Rapid Tactics Development Using Existing, Low-Cost Virtual Environments
- Objective System – HBM DEPOT
  - Captures demonstrations of Navy Aviation tactics using a low-cost Delta 3D Navy flight simulator
  - Supplements them with an easy to use diagrammatic representation (pre-loaded with Navy Aviation general domain knowledge) and learning algorithm
  - To generate high-quality human behavior models
  - Suitable for use in any virtual environment where intelligent computer generated forces (CGFs) or non-player characters (NPCs) such as a wingman or sophisticated OPFOR are required

Crank Illustration

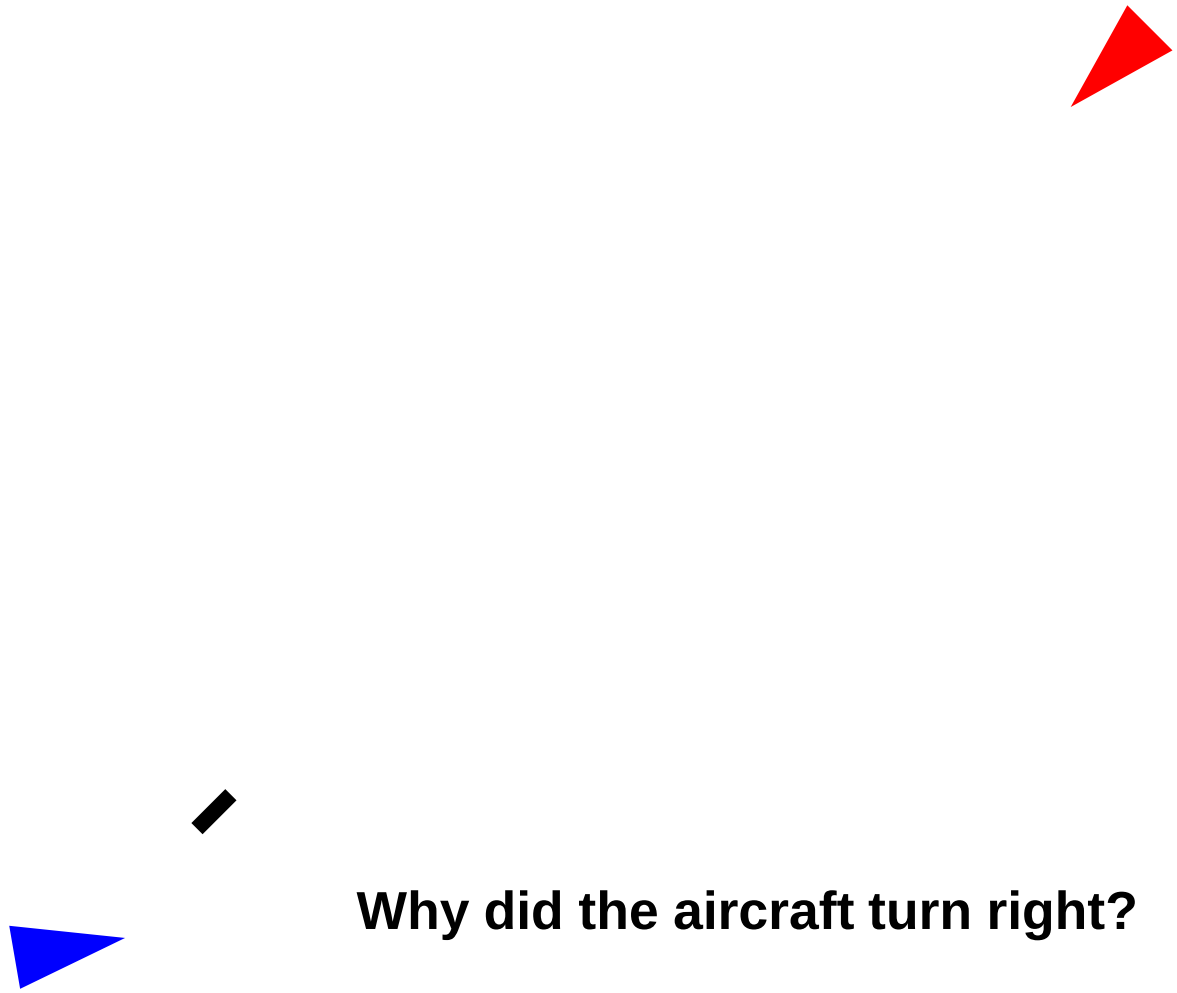
Crank Illustration



Crank Illustration



Crank Illustration

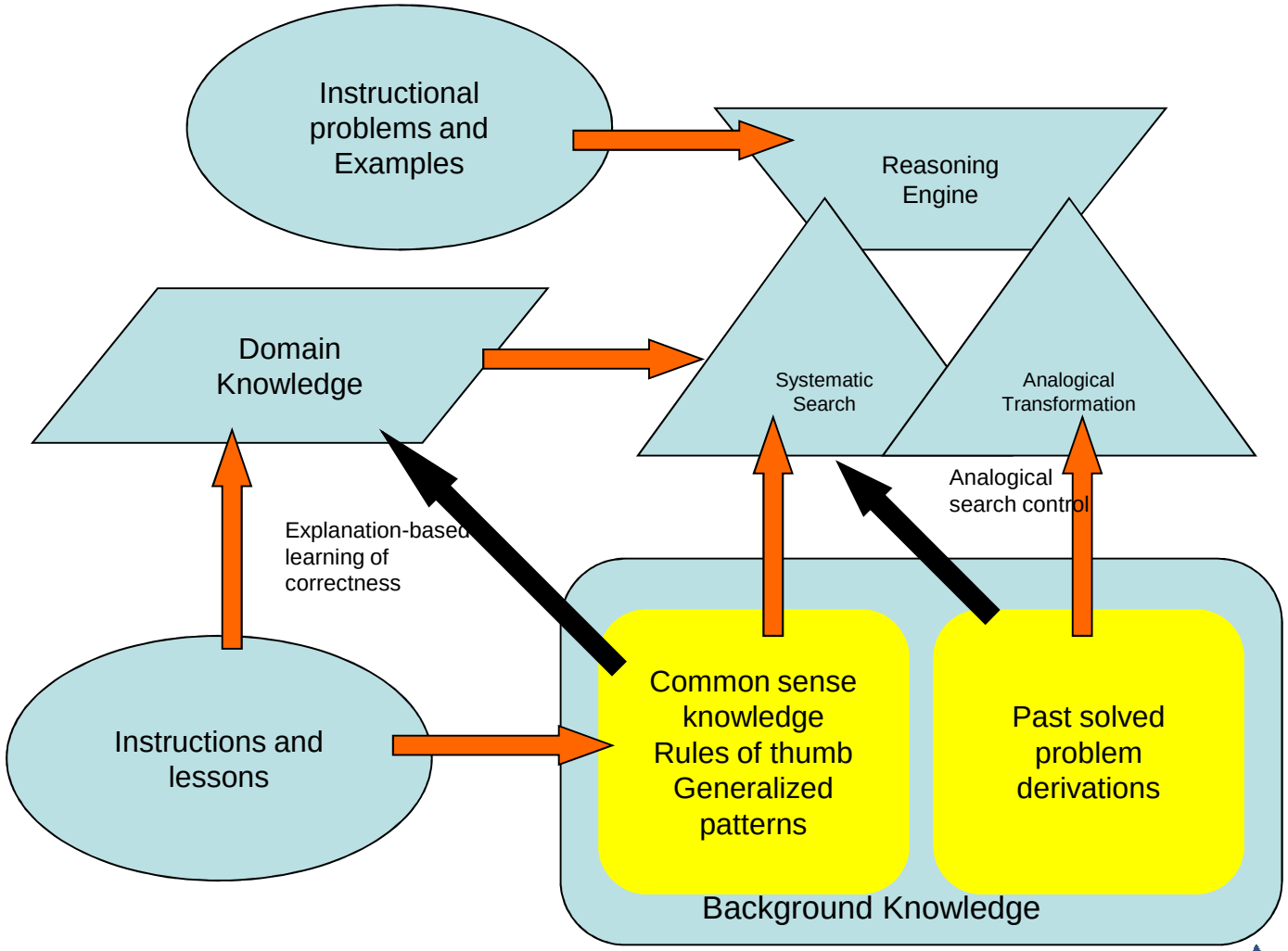


Why did the aircraft turn right?

Learning by explanation

- If this is a behavior the system would already produce in this situation, there is nothing to learn
- If this is a behavior the system would produce in a similar situation, the conditions of the behavior can be generalized
- If this is a behavior the system cannot easily explain, it falls back on general knowledge to produce candidate explanations
  - You must point at something you want to approach
  - You must point away from something you want to avoid
  - A sensor must be pointed at something to sense it
  - Etc.
  - Explanation-based learning of correctness developed by VanLehn, Jones, & Chi, 1991.

Overview of Explanation Approach



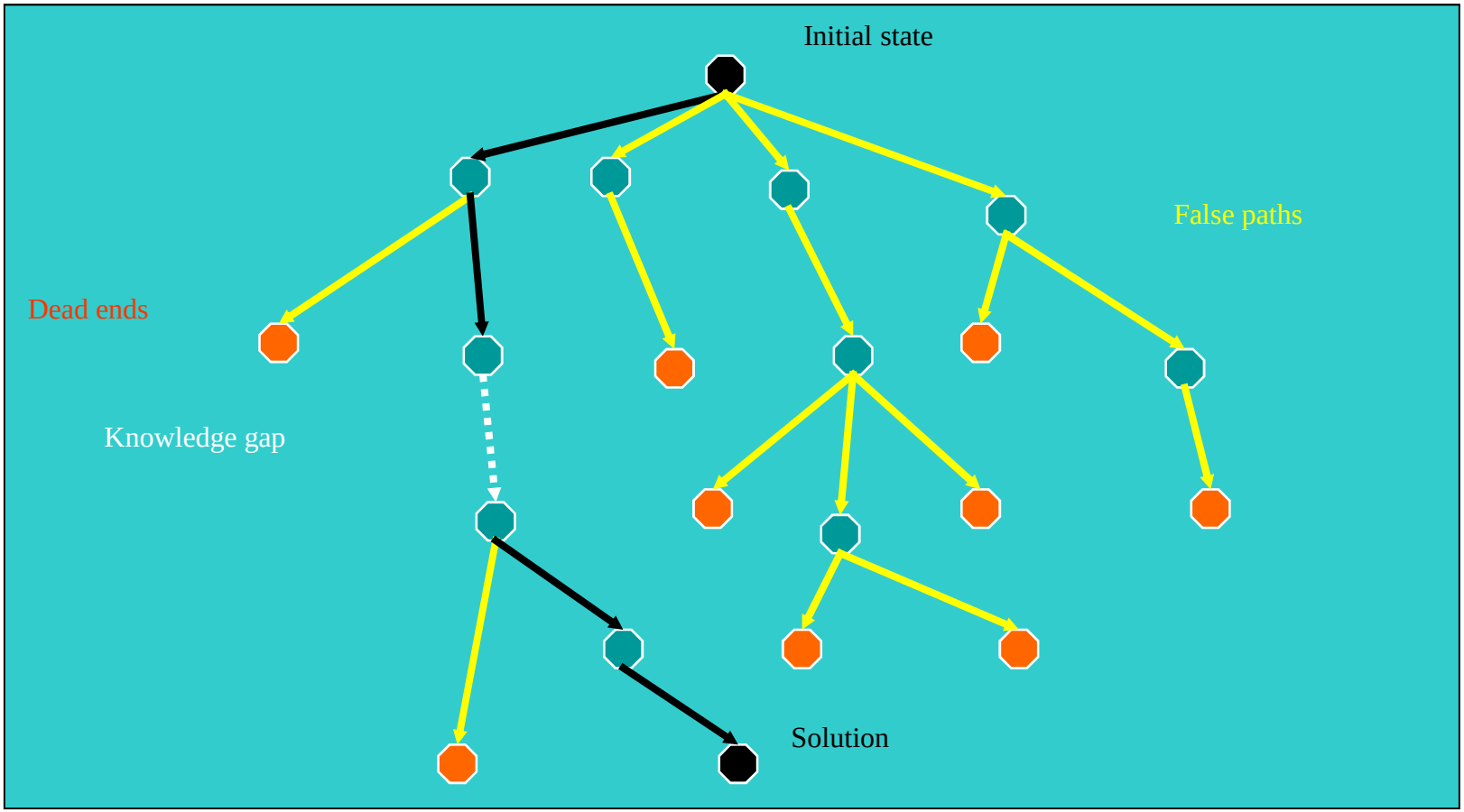
Finding Knowledge Patches

- Potential patches are found by using background knowledge and “rules of thumb” to complete “plausible explanations”
- Multiple candidates can be filtered by a variety of methods, or by asking the user
- Conditions on new knowledge are determined by heuristics to the “best level of generality”, by analogy, or by asking the user

Example Rules of Thumb

- A turn may indicate an approach to a route point
- A turn may indicate avoidance of an active threat
- A turn may indicate a preemptive avoidance of a potential threat
- A turn may indicate an approach to a target
- A turn may indicate an attempt to maintain sensor contact
- A turn may be triggered by the existence/detection of an object
- A turn may be triggered by a range to an object
- A turn may be triggered by the time since some event

Knowledge Patching



Cascade: Doman Knowledge Representation

- If an aircraft is supporting a radar-guided missile against a target, then the desired heading of the aircraft should combine the constraints of maintaining radar contact and approaching the target.

```
constraint(v(f(desiredHeading,A))=v(f(computeHeading,v(
   f(maintainRadarHeading, A, T)),v(f(approachHeading,
   A, T)))), dh=radar_approach) :-
inst(A,aircraft), inst(T,target), goal(supportMissile).
```

- The combined constraints of maintaining radar contact with a target and approaching a target imply coming to a heading equal to the bearing of the target.

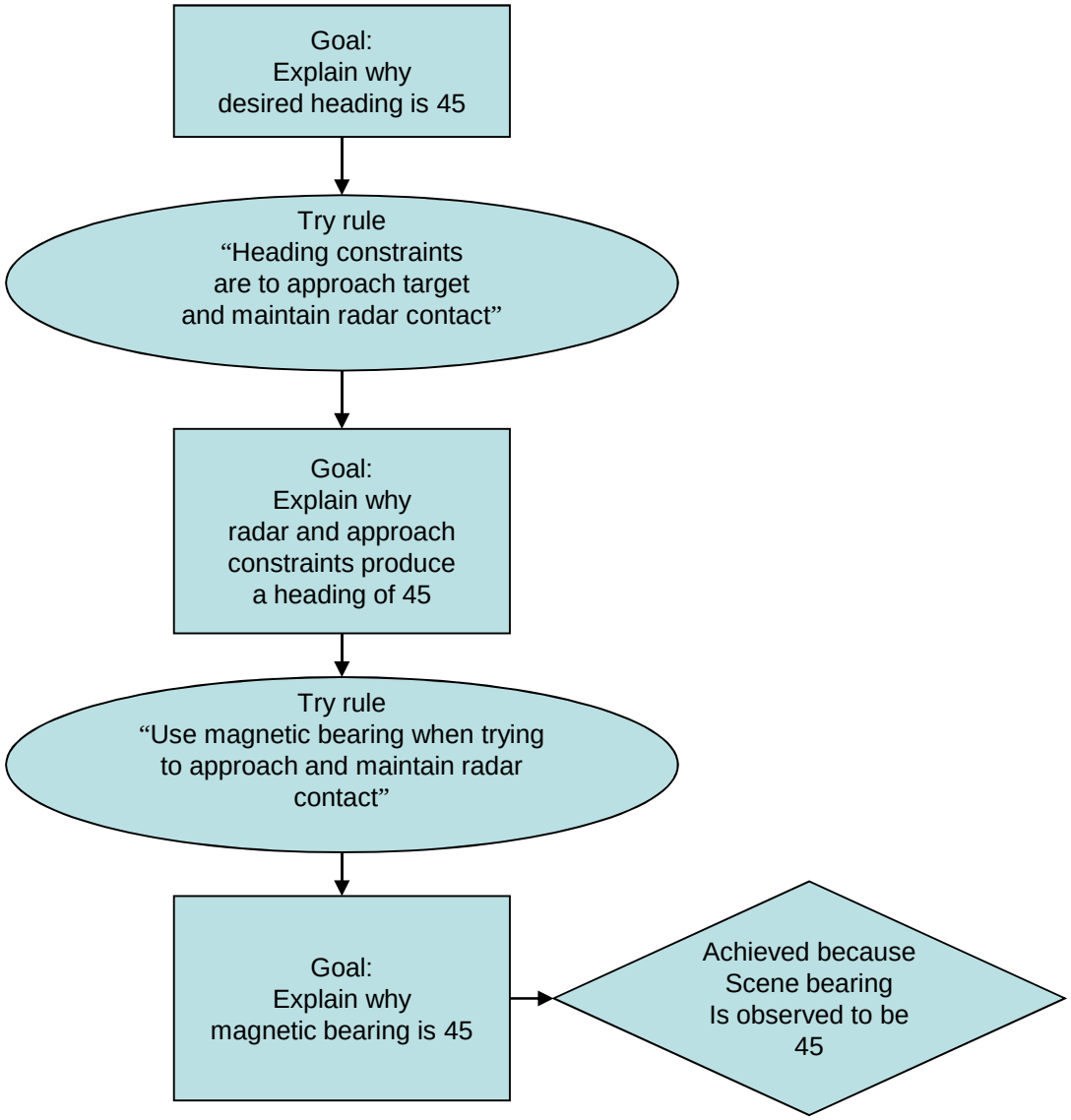
```
constraint(v(f(computeHeading,v(f(maintainRadarHeading,
   A, T)),v(f(approachHeading, A,
   T))))=v(f(bearing,A,T)), radar_approach=bearing) :-
inst(A,aircraft), inst(T,target).
```

Cascade: Rule of Thumb

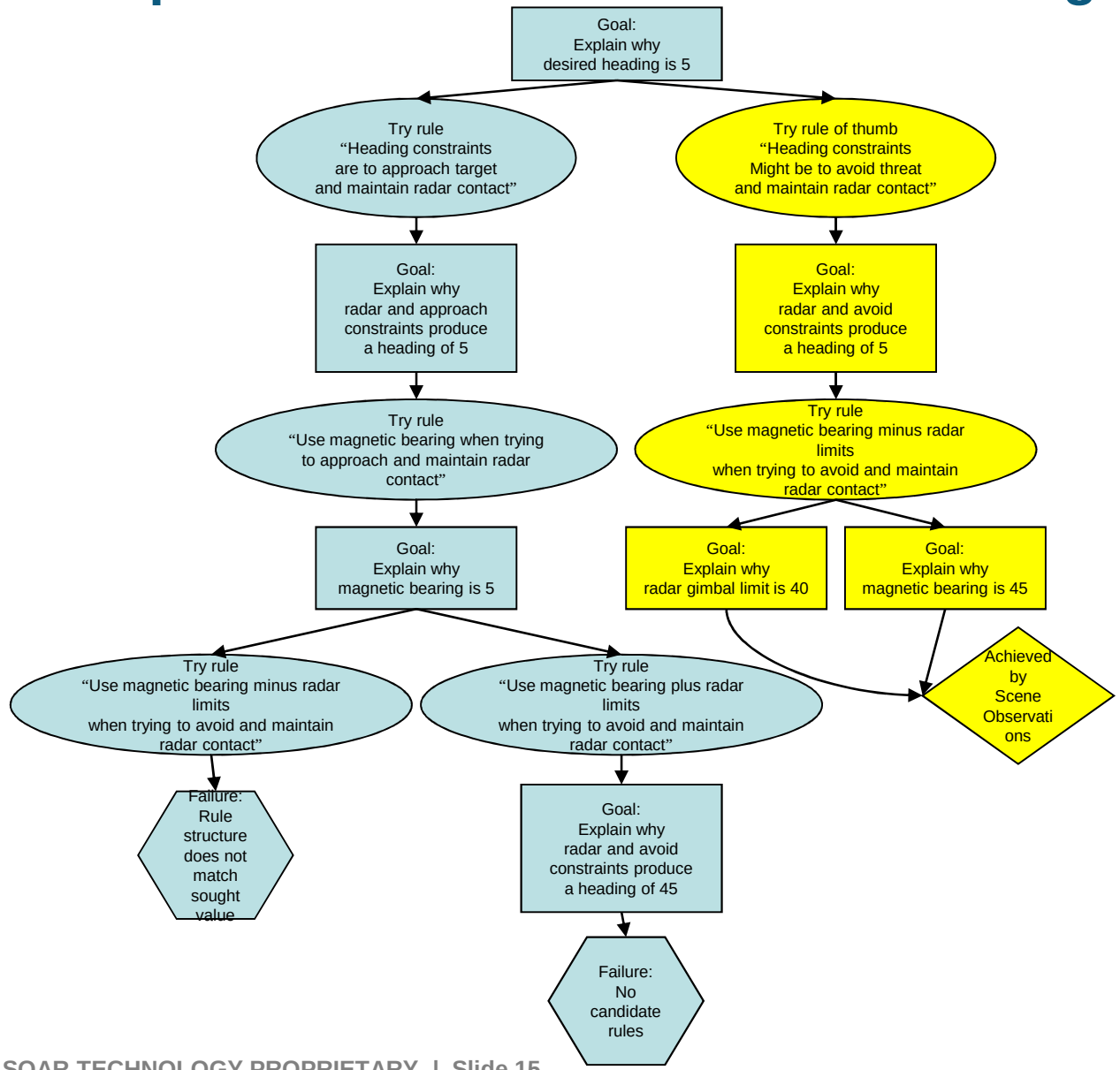
- If the aircraft is attempting to achieve some (unspecified) goal and to execute some (unspecified) tactic, and there is a threat, then one possible action is to avoid the threat while maintaining radar contact with it.

```
og_constraint(v(f(desiredHeading,A))=v(f(computeHeading
   ,v(f(maintainRadarHeading, A, T)),v(f(avoidHeading,
   A, T)))), dh=radar_avoid) :-
inst(A,aircraft), inst(T,threat), goal(G), tactic(X).
```

Cascade: Explanation Generation



Cascade: Explanation Generation with Learning



Results

- Gold
  - Able to apply Cascade to a tactical combat example without changing the Cascade code
  - Cascade-style search for explanations is feasible because there is a relatively small number of “sensible” rules of thumb
- Coal
  - Had to hand-craft tactical knowledge into Cascade’s equation-based representation
  - Some refactoring and generalization of TacAir-Soar code will be necessary to make this work
  - In the long run, we will want more sophisticated explanation searches than currently supported by Cascade