SOAR26: SPatial And Temporal Reasoning (SPAT-R)

Randy Jones – Scientific Advisor
Jack Zaientz – Scientific Advisor
Jens Wessling – Project Manager
Brian Stensrud – Research Scientist
Jonathan Beard, Sean Lisse, David Ray – Research Engineers
What is Spatial/Temporal Reasoning?

- Humans reason about space and time through both *quantitative* and *qualitative* assertions and relationships
  - qualitative
    - “that object is closer to me than this object”
    - “I am inside this room”
    - “that event occurred a long time ago”
  - quantitative
    - “that event happened 6 minutes and 30 seconds ago”
    - “My current position is (23.2, 100.4)”
    - “object B lies 6 meters closer to me than object C”
What is Spatial/Temporal Reasoning? (cont’d)

- Humans use these assertions to make decisions in their environment.

- Spatial and Temporal Reasoning are the processes by which these assertions (or beliefs, in some cases) are derived or calculated.

- To be effective in real-time environments, cognitive agents must also have the capability to generate and use assertions related to space and time.
What is the SPAT-R project?

- SPAT-R is the SPatial And Temporal Reasoning project
- Created as a spinoff of the BINAH project
- Focused on developing re-usable spatial & temporal reasoning capability (as previously defined) for Soar (and other) agents
How can SPAT-R help agents?

- Provides a general spatial, temporal, and spatio-temporal reasoning representation and inference capability
- Permits the leveraging of spatio-temporal reasoning knowledge from one domain application to another
- Forms a common metaphor for spatio-temporal reasoning, lowering the barrier for behavior-developer ramp-up
Why are we working on SPAT-R right now? (Scientific Motivations)

- Spatial and temporal reasoning are current, relevant, and useful

- Plenty of room for new scientific research in this area

- Plenty of opportunities for funded research efforts to extend the SPAT-R appliance
Why is SPAT-R better than the alternative?

- Cognitive rule-based systems are not generally efficient at mathematical calculation
- Majority of mathematical calculation in Soar agents often supports spatial and/or temporal reasoning
- Extensive mathematical calculations can make cognitive agents
  - **Difficult** to develop (*hard to do in Soar*)
  - **Brittle** in execution (*more opportunity for failure*)
  - **Expensive** in computation (*many steps required*)
- SPAT-R appliance moves most calculation out of the Soar agent but makes it easy for a Soar agent to use
  - **Easier** to develop (*it is a re-usable appliance*)
  - **Robust** in execution (*simple representation*)
  - **Efficient** in computation (*complex calculation offloaded*)
What can SPAT-R do for applications?

- Powerful tool for agent development
  - Agents frequently have to reason about space and time as preconditions to actions and decisions
  - The SPAT-R tool will provide developers with a tool to generate these preconditions, so that they can spend more time on domain-specific development

- Isolating spatial reasoning from the rest of the agent has several benefits
  - Supports rigorous engineering while avoiding the Soar kernel’s data overhead
  - Enhances code reuse & portability

- We often confront the question of how to interface symbolic to non-symbolic reasoning & memory

- Our emerging customer base for C³ agents and systems deals a lot with strategy and tactics
What is the current 2006 scope of SPAT-R?

- **Spatial Reasoning (IN SCOPE)**
  - General spatial reasoning capability not available
  - Almost every project needs, but rebuilds from scratch
  - A reusable general purpose

- **Temporal Reasoning (NOT IN SCOPE)**
  - Agent-based temporal reasoning module already investigated on BINAH project

- **Spatio-Temporal Reasoning (NOT IN SCOPE)**
  - Future Work
How does SPAT-R appliance work?

- Agent architecture agnostic spatial operation toolset
- Built on top of SoarTech’s “information management system”
- Accessible to Soar agents as an ATE plugin
- Input/Output-link interface
  - Region definition, operator definition, queries
- Snapshot Query/Response and Persistent Query capabilities
SPAT-R Interface Elements

- Region Definition
  - qualitative spatial region definitions with frame-of-reference, dimensionality, and dimensional projection

- Comparison Operator Definition
  - most frequent operations on regions will be intersection testing “is region X enclosed by region Y?”

- Relationship queries
  - inside/outside, above/below
SPAT-R Region Definition

- **0-D/1-D 'Region' Types**
  - Points, Route Segments, Routes

- **2-D Region Types**
  - Circular regions, polygonal regions, composite 2-D regions

- **3-D Region Types**
  - Spherical regions, cylindrical regions, N-gon regions, hyper-polygonal regions, generic 3-D regions, composite 3-D regions
Comparison Operator Definition

- Comparison Operator Definition
  - most frequent operations on regions will be intersection testing “is region X enclosed by region Y?”
  - intersection tests include region transformation by dimensionality
    - i.e., asking a 2-D question from the reference of a plane whether a point is within a region which was originally defined as a 3-D sphere but is projected as a circle for the purposes of generating the result set

<table>
<thead>
<tr>
<th>X in Y?</th>
<th>X is 0-d</th>
<th>X is 1-d</th>
<th>X is 2-d</th>
<th>X is 3-d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y is 0-d</td>
<td>X == Y</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Y is 1-d</td>
<td>X &lt; Y</td>
<td>X &lt; Y</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Y is 2-d</td>
<td>X &lt; Y</td>
<td>X &lt; Y</td>
<td>X &lt; Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Y is 3-d</td>
<td>X &lt; Y</td>
<td>X &lt; Y</td>
<td>X &lt; Y</td>
<td>X &lt; Y</td>
</tr>
</tbody>
</table>
Relationship Queries

- Domain-specific
  - “is the **aircraft** in my **flight corridor**?”
  - “are any **enemies** on my **left flank**?”

- Composed from operator definitions

- Return those data elements which resolve as “true” (matching the query tests) in a result set
Example Use-Case: “Am I in danger?”

- Regions:
  - SAM site (type: 3-D cylinder)
    - ordered-triple center \((x,y,z)\)
    - scalar value radius
    - orientation unit vector \((<x_0,y_0,z_0>/|<x_0,y_0,z_0>|)\)
    - scalar value height
  - aircraft (type: point)
    - ordered triple \((x,y,z)\)
    - orientation unit vector \((<x_0,y_0,z_0>/|<x_0,y_0,z_0>|)\)

\[^{\text{query}}\]
\[^{\text{comparison}}\]
\[^{\text{operator inside}}\]
\[^{\text{duration persistent-notify}}\]
\[^{\text{subject}}\]
\[^{\text{unique-id point789}}\]
\[^{\text{object}}\]
\[^{\text{unique-id cylinder123}}\]
What kinds of projects could benefit from SPAT-R?

- **Airspace control:**
  - dependent on spatial & temporal assertions

- **Aviator behaviors:**
  - Spatial understanding required in formation or route flying

- **Intelligence analysis:**
  - Spatial & temporal correlation of intelligence hypotheses
  - Spatial and temporal assertions made by the agent can be converted to graphical or textual notations displayed on the GUI

- **C4ISR:**
  - external planning & navigation systems require spatial/temporal assertions

- **Pedagogic training systems:**
  - Director needs real-time awareness of trainee’s position and orientation relative to other objects and NPCs
  - Director is responsible for the proper timing of events in each training scenario
How could SPAT-R appliance help future work?

- Prevents re-invention of the wheel
  - Availability of a re-usable spatial & temporal reasoning capability (as previously defined) for Soar (and other) agents

- Additional feature at low cost
  - Permits the leveraging of spatio-temporal reasoning knowledge from one domain application to another

- Leverages well-considered representations and interfaces
  - Provides a general spatial, temporal, and spatio-temporal reasoning representation and inference capability

- Reduces resource mismatch risk
  - Forms a common metaphor for spatio-temporal reasoning, lowering the barrier for behavior-developer ramp-up
**Nuggets & Coal**

**NUGGETS:**
- We noted that an architectural change to Soar permitting direct access to arbitrary memory locations for designating input/output would be TREMENDOUSLY useful
- We were able to leverage a number of existing Soar development tools (ATE, etc)
- Many projects at SoarTech eager to use
- First implementation to be finished in June

**COAL:**
- Known performance and scaling issues

**Coal for non-Soartech folks:**
- Requires ATE
- Not currently available
Backup Slides
What have we accomplished?

- IR&D project initially approved - late November 2005
- Outreach requirement gathering from stakeholders - December 2005
- Project planning & “Internal 1st Stage Customer” (ISAT) defined –December 2005/January 2006
- Additional scientific & engineering resources brought on board - February 2006
- Created socialtext repository for team work products – February 2006
- Gathering results of initial scientific investigations - late February / early March 2006
- Appliance functionality & interface design – April/May 2006
- Initial Draft Implementation – May/June 2006