Standard Task-Actor Protocol
Vladislav “Dan” Veksler
Participant pool → Task logic → model
Participant pool

Task logic

Model

Task ↔ Actor

<table>
<thead>
<tr>
<th>Loc</th>
<th>Att</th>
<th>Kind</th>
<th>Value</th>
<th>Color</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>(25 25)</td>
<td>NEW TEXT</td>
<td>&quot;textBox1&quot;</td>
<td>BLACK</td>
<td>VISUAL-LOCATION0</td>
<td></td>
</tr>
<tr>
<td>(35 55)</td>
<td>NEW BUTTON</td>
<td>&quot;button1&quot;</td>
<td>BLACK</td>
<td>VISUAL-LOCATION1</td>
<td></td>
</tr>
<tr>
<td>(55 55)</td>
<td>NEW BUTTON</td>
<td>&quot;button2&quot;</td>
<td>BLACK</td>
<td>VISUAL-LOCATION2</td>
<td></td>
</tr>
<tr>
<td>Loc</td>
<td>Att</td>
<td>Kind</td>
<td>Value</td>
<td>Color</td>
<td>ID</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
<td>----------------</td>
</tr>
<tr>
<td>(25,25)</td>
<td>NEW TEXT</td>
<td>(textBox1)</td>
<td>BLACK</td>
<td>VISUAL-LOCATION0</td>
<td></td>
</tr>
<tr>
<td>(35,55)</td>
<td>NEW BUTTON</td>
<td>button1</td>
<td>BLACK</td>
<td>VISUAL-LOCATION1</td>
<td></td>
</tr>
<tr>
<td>(55,55)</td>
<td>NEW BUTTON</td>
<td>button2</td>
<td>BLACK</td>
<td>VISUAL-LOCATION2</td>
<td></td>
</tr>
</tbody>
</table>

Task A logic

Task B logic

Task C logic

ACT-R model

Py-IBL model

Soar model

Sigma model
Does the model scale to other tasks?
Does the task scale to other models?

- How much work is needed to interface a given model with a new task-environment?
- Is a given task-environment suitable for cross-framework modeling simulations, as well as human empirical studies?
- Will different models and human actors all be presented with similar information?
Creating a standard task-actor protocol

- Save Resources
- Task/model Reuse
- Scientific Replication

*Restriction

Why?

- Standard Task-Actor Protocol
  - ACT-R
  - Py-IBL
  - Soar
  - Sigma

Participant pool

* Task A logic
* Task B logic
* Task C logic
STAP
(Simple Task-Actor Protocol)

- Worked on as a part of Robotics and Network Science Collaborative Task Alliances

https://github.com/vdv7/stap
# Demo

<table>
<thead>
<tr>
<th></th>
<th>Shepard, Hovland, Jenkins (1961)</th>
<th>IED-tactical (hearts &amp; minds)</th>
<th>SS-RICS robot navigation task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Participant</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ACT-R (lisp)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>IBL (python)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

[https://github.com/vdv7/stap](https://github.com/vdv7/stap)
STAP v1.0

- Meant to allow
  - Hierarchical state description and vector graphics
  - Varying types of actions (e.g., click, hold-down, type)
  - Varying types of feedback (e.g., success/fail, temp reward, long-term score)
  - FTRT simulations
  - Seamless playback & 3rd party observation
  - Backwards compatibility
  - Auto-genera instructions

How?

```json
{
  "<handshake>>
  "s": "<state>>,
  "t": "<title info>>,
  "a": "<allowed actions>>,
  "r": "<reward>>,
  "$": "<score>>
}
```

https://github.com/vdv7/stap
STAP v1.0

- Resource savings, Reuse, Replication
  - Don’t worry about gui development (just pick a template)
  - Write task logic once, serve it to varying devices and computational models
- Standard logging/playback
- Online studies or FTRT local simulations

```json
{
  "s": "<<state>>",
  "t": "<<title info>>",
  "a": "<<allowed actions>>",
  "r": "<<reward>>",
  "$": "<<score>>"
}
```

https://github.com/vdv7/stap
Questions?

{  
  "s": <<state>>,  
  "t": <<title info>>,  
  "a": <<allowed actions>>,  
  "r": <<reward>>,  
  "$": <<score>>
}

https://github.com/vdv7/stap