Collaborators:
Norbou Buchler (ARL)
Troy D. Kelley (ARL)
Christian Lebiere (CMU)
Don Morrison (CMU)
Cleotilde Gonzalez (CMU)
Michael Yu (CMU)
Marshall Scott Poole (UIUC)
Alex Yahja (UIUC)
Tarek Abdelzaher (UIUC)
Eric Avery (ARL)
Sean McGhee (ARL)
model A

model B

model C

ACT-R

TASK 1

TASK 2

TASK 3
Task development

- How easy will it be to compare model data with humans?
- How easy will it be to compare model with other models?
- How easy will it be for somebody else to run their participants or models on your task?
Performance Comparison

- Ideally, model performance is always contrasted with
  - Behavioral data
  - Many other computational models of varying frameworks
  - Across multiple tasks

- In reality, this is a very high hanging fruit
**Task ↔ Actors**

**Participant pool**

```
<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>“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>
```
Participant pool

Amazon Mechanical Turk

Volunteer Science

Task ▸ Actors

ACT-R

Soar

Sigma

PyIBL

...
Cross-framework comparison costs become prohibitive

- Combinatorial explosion

Tasks ↔ Actors
To sum up....

- GUI ≠ API
  - Graphical task interfaces designed for humans are rarely machine-readable
  - Task simulations designed for computational agents are rarely human-readable
- \( API_1 \neq API_2 \)
  - Each task simulation employs its own API
  - API designed for one agent framework is rarely readable by another

practical problem → theoretical implications
Imagine a world where...

- you can connect your model to someone else's plug-and-play task software
- you can grab someone else's simulation and replay it in your lab without any non-standard software
- you can replicate someone else's experiment results without hassle
- you can connect different cognitive systems to your task without hassle
The solution

1. A functional-essence approach to task development
The solution

1. A functional-essence approach to task development
   - separation of function from style
Function VS Style

- **Function**
  - two numbers
  - textbox
  - submit button

- **Style**
  - spacing and colors
  - font-type and size
  - button style

```html
<html>
<head><style>
input[type="submit"] {background-color:#aaccff;border-radius:5px;}
</style></head>
<body>
<table width=100% height=100%><tr valign=middle><td align=center>
<form style="font-family:ttahoma;font-size:18px;"
  5 + 5 = <input>
  <input type=submit>
</form>
</td></tr></table>
</body></html>
```
Functional-essence approach to task development

- **Function**
  - 8x8 table with alternating squares
  - 6 different recognizable piece types

- **Style**
  - size and colors of the board and squares
  - images representing each piece
The solution

1. A functional-essence approach to task development
2. A standard & simple API for function-only interaction with most task types
Simple Task-Actor Protocol (STAP)

http://vdv7.github.io/stap/

Collaborators:
Norbou Buchler (ARL)
Troy D. Kelley (ARL)
Christian Lebiere (CMU)
Don Morrison (CMU)
Cleotilde Gonzalez (CMU)
Michael Yu (CMU)
Marshall Scott Poole (UIUC)
Alex Yahja (UIUC)
Tarek Abdelzaher (UIUC)
Eric Avery (ARL)
Sean McGhee (ARL)

Simple Task-Actor Protocol (STAP)

- **Subset of the JSON protocol**
  - JSON modules in most programming languages

- **Task interactions**
  - Vector graphics and animation
  - Varying types of actions (e.g., click, hold-down, type)

- **Time**
  - Faster-than-real-time simulations
  - Slower-than-real-time simulations

- **Goal** and task description
  - Machine-readable
  - Auto-generated human instructions

Hello World

```
print("Hello World!")
```

STAP

(task logic -> "Hello World!" -> API)

API to virtual display

(computational participant)

API to GUI

human participant

(add-text-to-exp-window "Hello World!")

Hello World!
STAP Style Templates

Hello World!
Click a button

Button 1  Button 2

Hello World!
Click a button:

[Button 1]  [Button 2]

task logic

STAP

API to GUI

human participant
STAP for ACT-R

- Don Morrison's (CMU)
STAP interaction/playback

- RedForce task videos
  - ACT-R model
  - PyIBL agent
  - human participant
STAP task examples


Instructions
- Produce the shortest possible circuit that connects each circle on the map.
- Click a white circle to add it to the circuit.
- Click a blue circle to remove it from the circuit.

Map

---

Trial 2 of 32

Is this object a Greeble or a Groble?

- Groble
- Greeble

Instructions

Press the "Start Timer" button to start a timer. Press "Stop Timer" when you think the ball reaches the RED line.
STAP task examples

- http://vdv7.github.io/stap/
STAP task examples

Robot navigation: human vs instance-based learning agent
Serialized task-architecture interactions

- model A
- model B
- model C

ACT-R

CL-STAP

- TASK 1 (python)
- TASK 2 (JAVA)
- TASK 3 (LISP)
Advancing the science

- Pace
  - experiments are easier to develop (no GUI development, API only)
  - simulations are easier to run (one-time cost to set up framework for API)
- Persistence
  - enable connection to multiple tasks
- Scale
  - separation of model/architecture from task enables scaling up of task and number of connected models (24 million?)
Computational Cognition Competitions

- Competitions and Grand Challenges
  - primary means to motivate and galvanize the research community to solve ambitious scientific and engineering challenges

- Developing a new task for a competition is not trivial
  - API, documentation, distribution/connection framework, human data

- No reuse of task interfacing between competitions
  - Increases engineering costs for participants
  - Reduces researcher buy-in
Other approaches

- web-apps (HTML5)
- visual (pixel-by-pixel)
- real-world (robotics)
- physical simulations (virtual world API's)
- task domain –specific API's (e.g. VGML)
- architecture-specific API's (e.g. JNI)
QUESTIONS?

http://vdv7.github.io/stap