

# JSON Network Interface (JNI) to ACT-R Interfacing ACT-R with External Environments

Mike Schoelles and Ryan Hope

# ACT-R WS 2013

# Motivation

---

- ACT-R should be able to interface with complex, dynamic task environments/simulations written in any programming language on any operating system



# Brief History

---

- 2000 - CMIMS - Ritter, Baxter, Jones and Young
  - UNIX domain sockets
- 2003 - Symposium Mike Byrne
  - Many different approaches
  - X-plane/UDP -Byrne
- 2009 - Space Fortress - Destefano
  - D-BUS (IPC)
- 2010 - Hello Java! -Buttner
  - Text strings over TCP/IP
- 2011 - X-Plane -Schoelles



# Interface Variations

---

- Devices/Experiments/Games/Simulations
- Written in many languages
  - Lisp, Java, Python, Matlab, Java Script, C #, C++
- Run on different Operating Systems
  - MAC, Window, Linux
- Different machine configurations
  - same machine
  - two machines

# POII

---

- Language?
- OS?
- Multiple machines?

# Requirements

---

- A communication interface
- A common data format
- Support all CMU versions of ACT -R
  - support EMMA and PAAV
- Different levels of Users
  - ACT-R
  - Programming

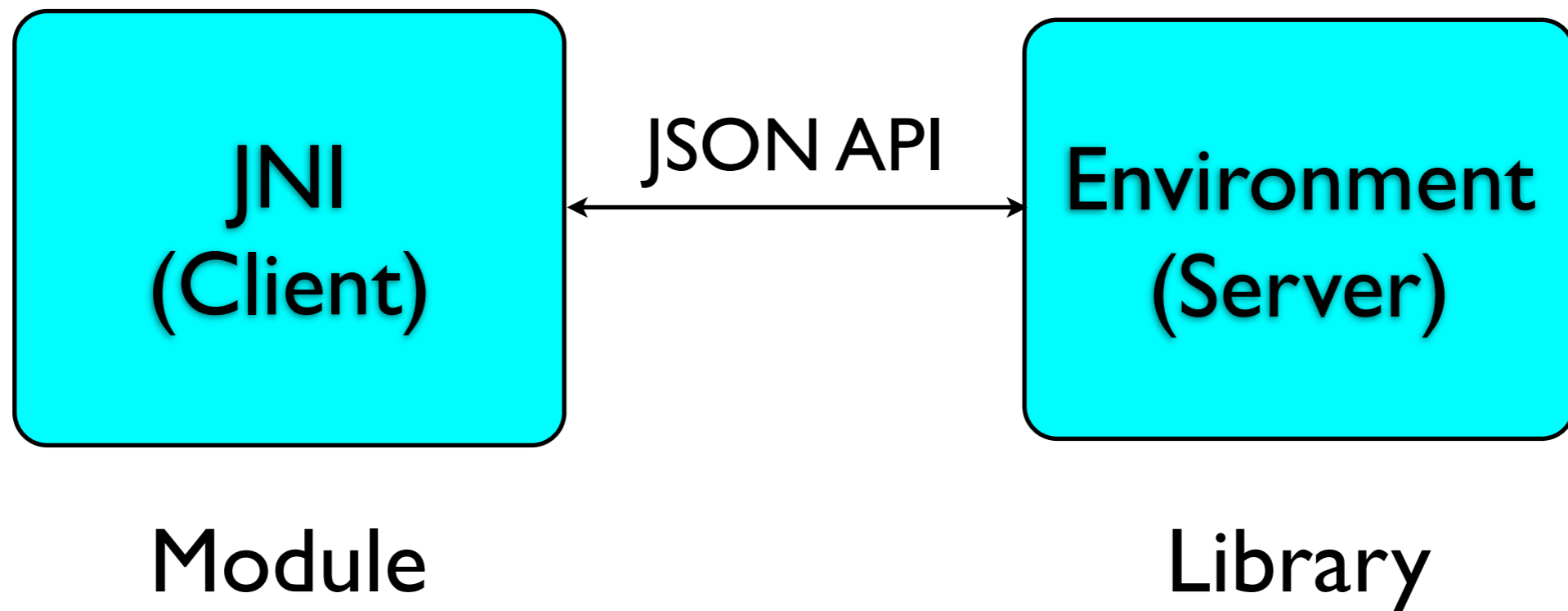
# JSON Network Interface

---

- Java Script Object Notation
- TCP/IP
- Library of Language Interfaces (Git repository)

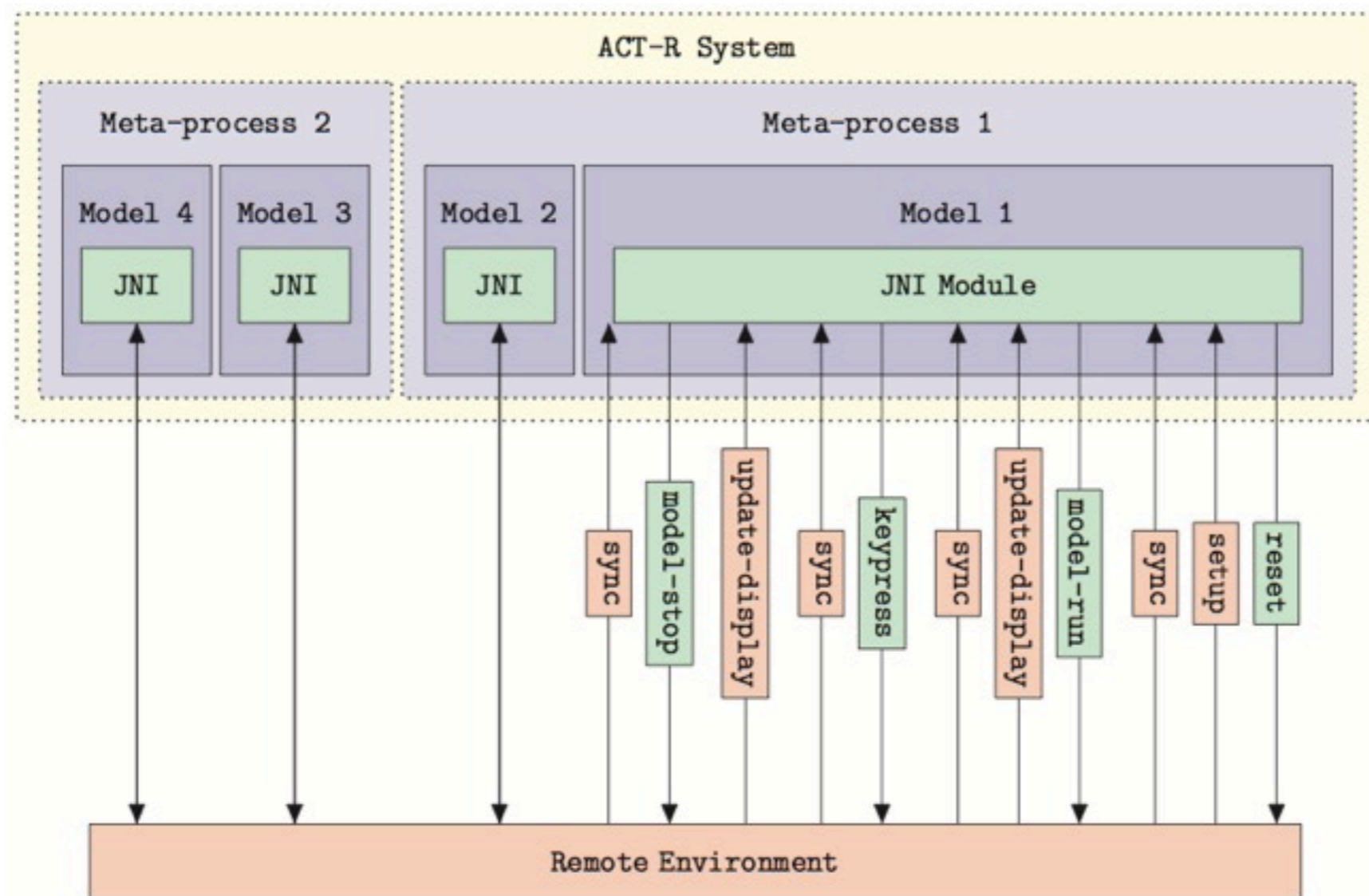
# Design

---





# Design - multimodels



# JSON

---

- JSON (JavaScript Object Notation) is a lightweight data-interchange format
- easy for humans to read and write
- easy for machines to parse and generate
- built on two structures, objects and arrays

Example:

```
{"keyA": [1, 2], "keyB": false, "keyC": true, "keyD": null}
```

# The JSON API

---

- `{"model":<modelID>,"method":<command>,"params":<params>}`\r\n
- JNI Commands:
  - keypress
  - mousemotion.mouseclick
  - speak
  - reset, model-run, model-stop
  - set-mp-time
  - gaze-loc, attention-loc



# JSON API

---

- Environment Commands:
  - sync
  - setup
  - update-display
  - trigger-reward, trigger-event
  - set-cursor-loc
  - new-digit-sound, new-tone-sound, new-word-sound and new-other-sound

# Example: update-display command

---

```
{  
  "model": "myModel",  
  "method": "update-display",  
  "params": {  
    "visual-location-chunks": [  
      {"isa": "visual-location",  
       "slots": {"screen-x": 100,  
                 "screen-y": 200}},  
      {"isa": "visual-location",  
       "slots": {"screen-x": 300,  
                 "screen-y": 400}},  
    ],  
    "visual-object-chunks": [  
      [{"isa": "visual-object",  
        "slots": {"value": "hello"}},  
       {"isa": "visual-object",  
        "slots": {"value": ":world"}}],  
    "clear": true  
  }  
}
```

# Configuring the JNI Module

---

- (sgp :jni-hostname "localhost" :jni-port 5454)

# Time Synchronization

---

- Asynchronous - JNI does not wait for acknowledgement from environment when command is sent
  - dynamic environment, real-time
- Synchronous - JNI waits for acknowledgement from environment when command is sent
  - static environments, faster than real-time
- Time lock - JNI sends time signal periodically to environment
  - sends meta-process time to environment to drive a clock
  - dynamic environments, faster than real-time
- All modes support stepping through a model

# High Level Libraries

---

- Available:
  - `actr6 jni` - A Python/Twisted for communication with the JNI module, available in the Python Package Index (PyPi)
- In Progress:
  - Java



# Where to get JNI

---

- <https://github.com/RyanHope/json-network-interface>
- Everything is on GitHub:
  - Source code
  - Links to binaries
  - Installation instructions
  - Documentation
  - Examples
  - Issue Tracker (For bugs & feature requests)

---

**Thank You**  
**Questions?**