Evaluating Methods for Long-Term HRI Studies Under Resource Constraints

Status Update, Summer 2012

Nate Derbinsky

Motivation

Goal: long-term HRI research

Systems that support **high levels of robot autonomy** and **user trust** while supporting real-time interaction for days-weeks of continual operation

Challenge: experimental-resource constraints

- Robot time sharing
- Study participants

Focus: Memory

Prior work has shown that long-term memory has the potential to make robotic companions more capable and believable

<u>Open Issues</u>

- When/what to encode
- Useful knowledge for learning/retrieval
- Efficient and scalable algorithms

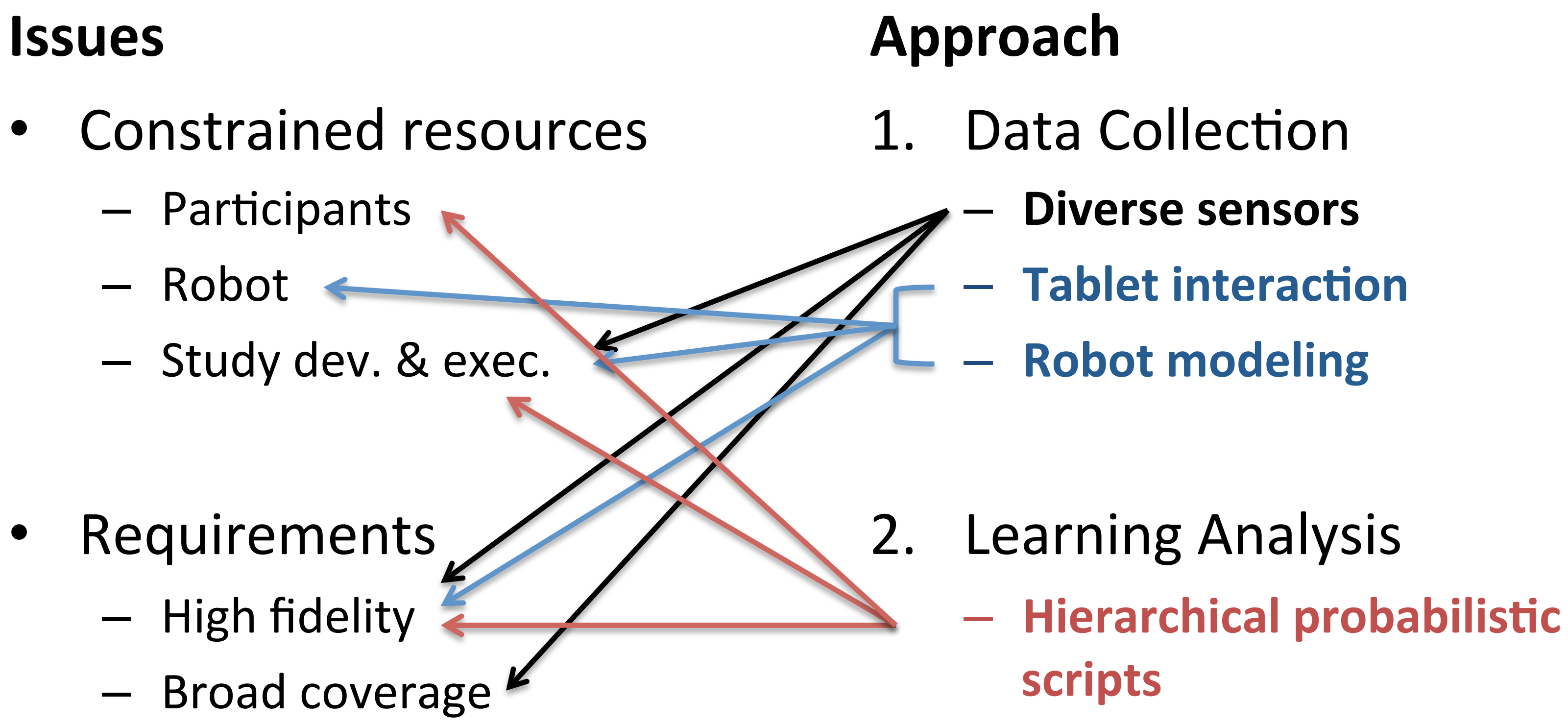
Goal: Long-Term HRI Dataset

Data Collection. Long-term episodic traces of HRI in UH Robot House.

Learning Analysis

- General properties
  - Data size, rate, patterns, …
- Task-relevant learning opportunities
  - User context, preferences, …

Methodology



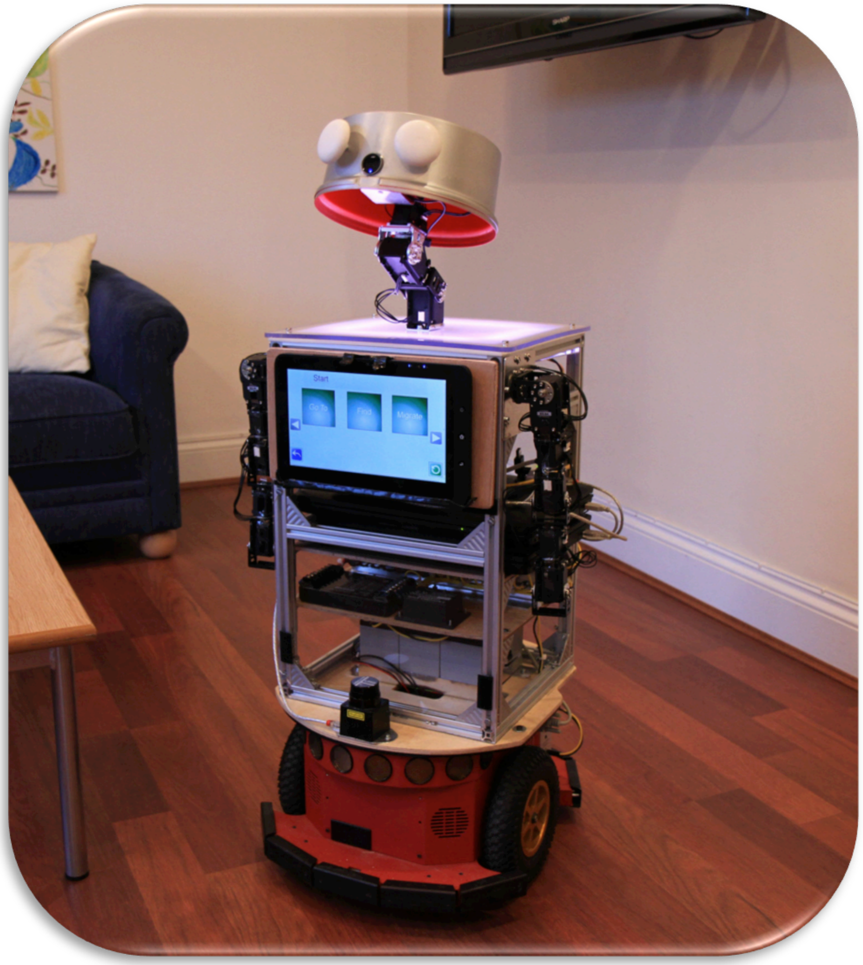
Issues

- Constrained resources
  - Participants
  - Robot
  - Study dev. & exec.
- Requirements
  - High fidelity
  - Broad coverage

Approach

1. Data Collection
   - **Diverse sensors**
   - **Tablet interaction**
   - **Robot modeling**
2. Learning Analysis
   - **Hierarchical probabilistic scripts**

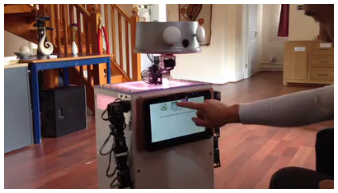
Sensing Environment: UH Robot House



- Types of events
  - Electricity/water fluctuations
  - Discrete (e.g. drawer open/close, seat occupation)
- Event += time, place
  - User not directly sensed
- Logged centrally to MySQL

“Robot” Interaction

1. Gather timing data for Sunflower actions (movement, open/close drawer)



2. Extract parameters (assume $\mathcal{N}$)
3. Simulate and log actions via tablet

Sunflower Data

Methodology

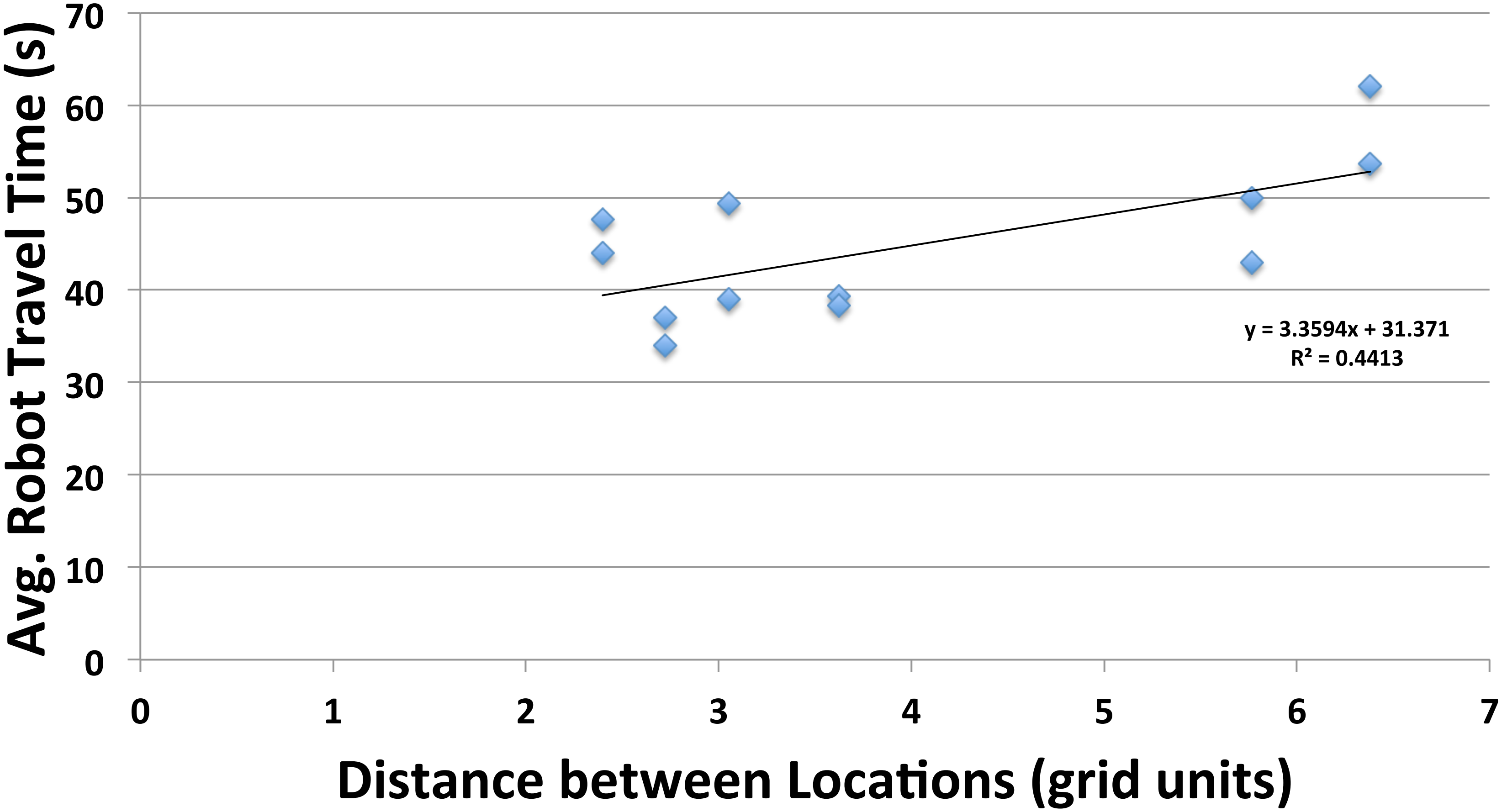
- 3 trials/operation
  - Drawer (2): open, close
  - Move (12): 4 locations, 2 directions
- Data: time (s) from command input to action conclusion, rounded

Lessons Learned

- Movement time and variance were highly dependent upon starting/ending location
- Drawer variance was negligible

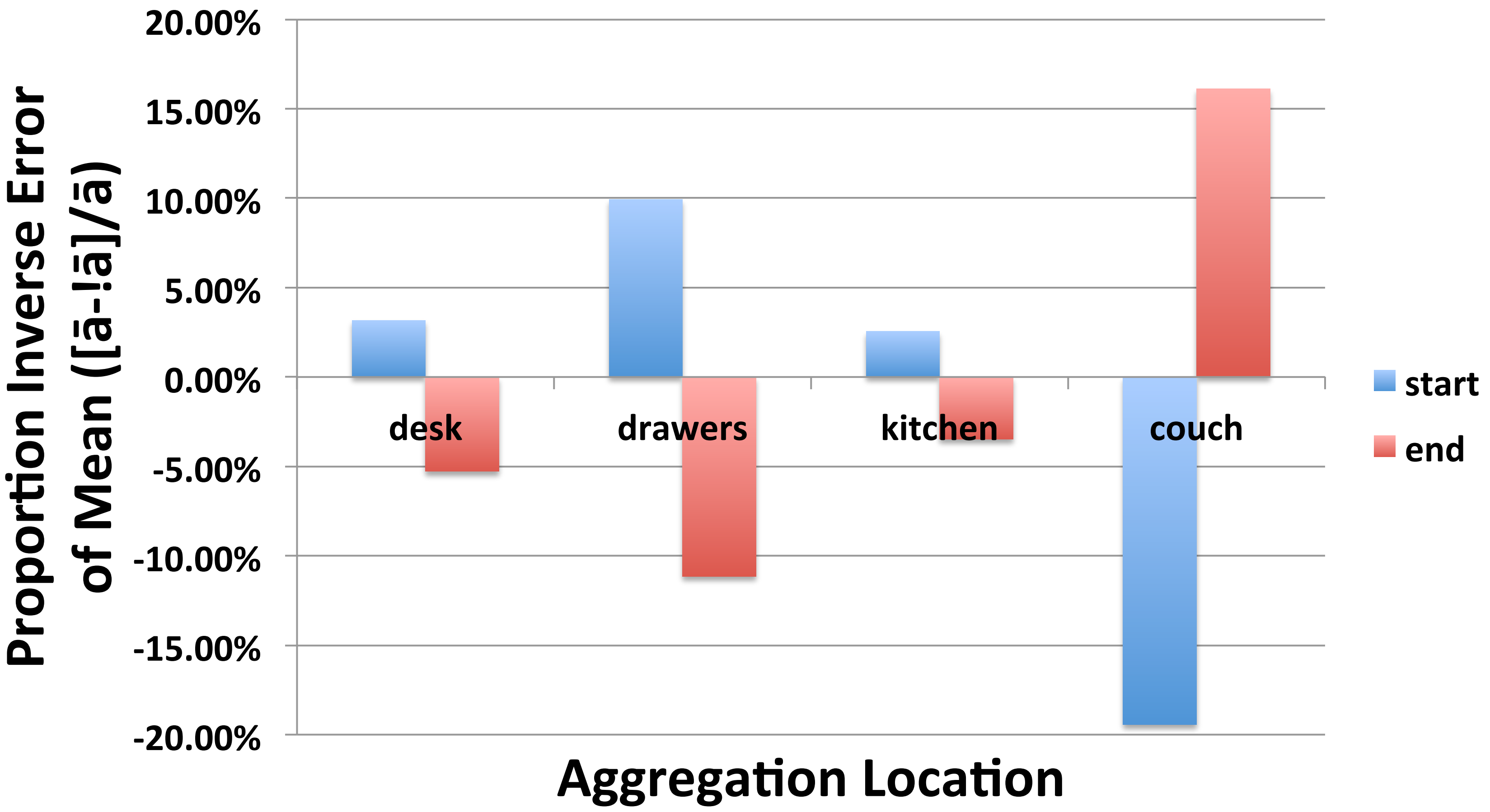
Sunflower Movement Data

Cannot Approximate from Distance



Sunflower Movement Data

Must Measure Both Directions



Sunflower Model

Future Work

- High variance -> need many more trials

  Still a useful methodology under resource constraint?

- Simplistic independence assumptions -> need to evaluate plausibility w.r.t. additional complexities

  e.g. human positioning/movement, drawer load

- Quantitative analysis of fidelity

  e.g. can a standard classifier tell the difference between actual time and model prediction?

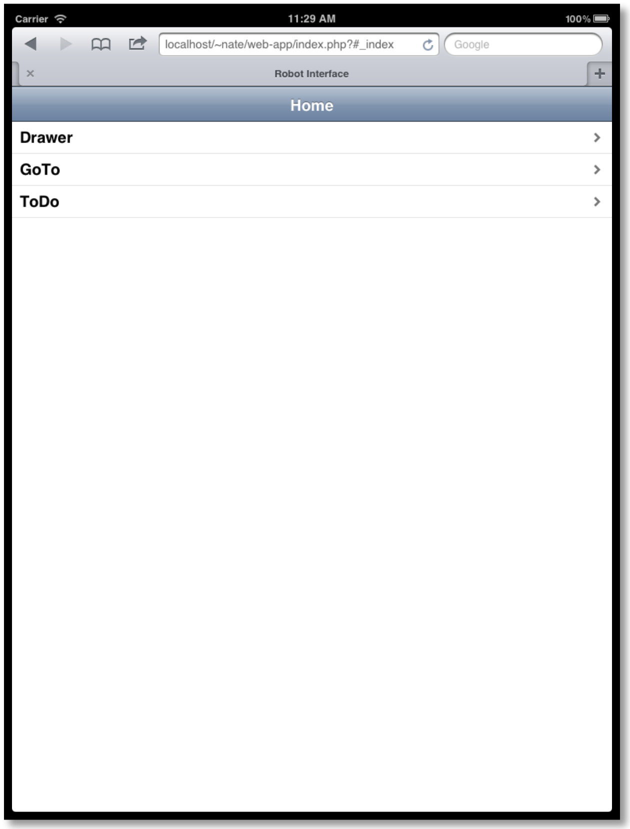
Tablet Interface

- Web-based (HTML, CSS, JS, PHP)
  - Cross platform/device
  - Touch optimized (via iUI ~ iOS)
  - Easily extensible

- Logs events to MySQL
  - Draws on models for event data & interaction

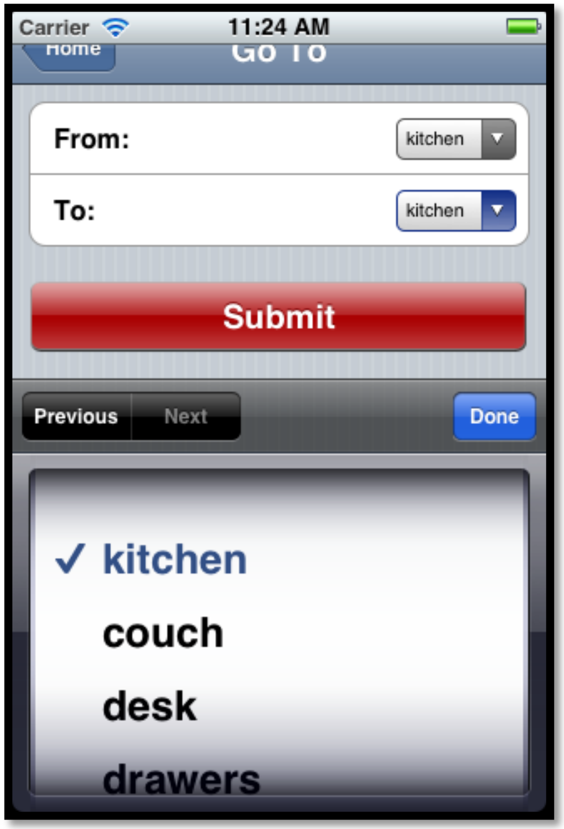
Tablet Interface

Home



Tablet Interface

Event Entry

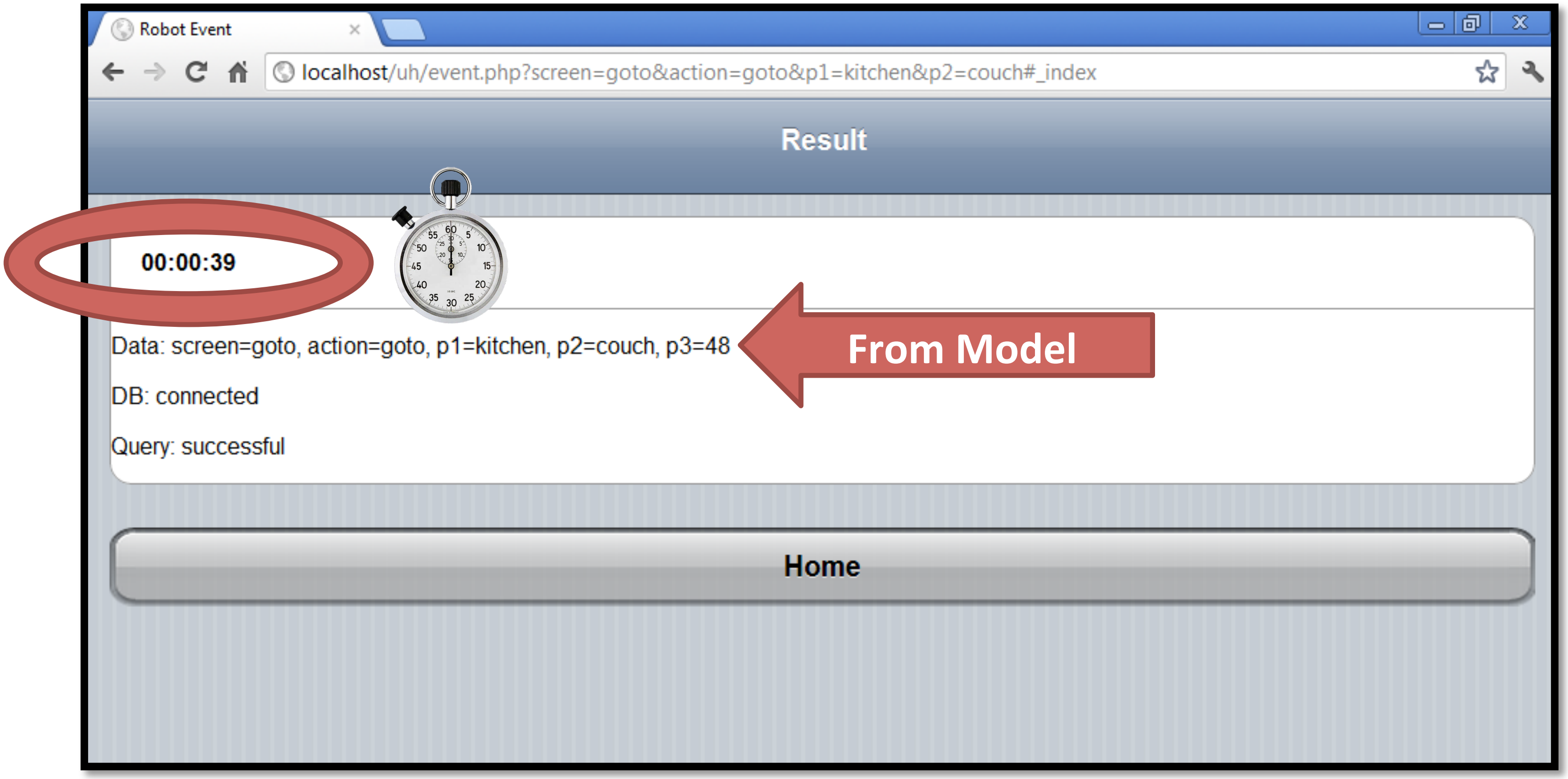


From: kitchen

To: kitchen

Tablet Interface

Event Execution & Logging



localhost/uh/event.php?screen=goto&action=goto&p1=kitchen&p2=couch#_index

00:00:39

Data: screen=goto, action=goto, p1=kitchen, p2=couch, p3=48

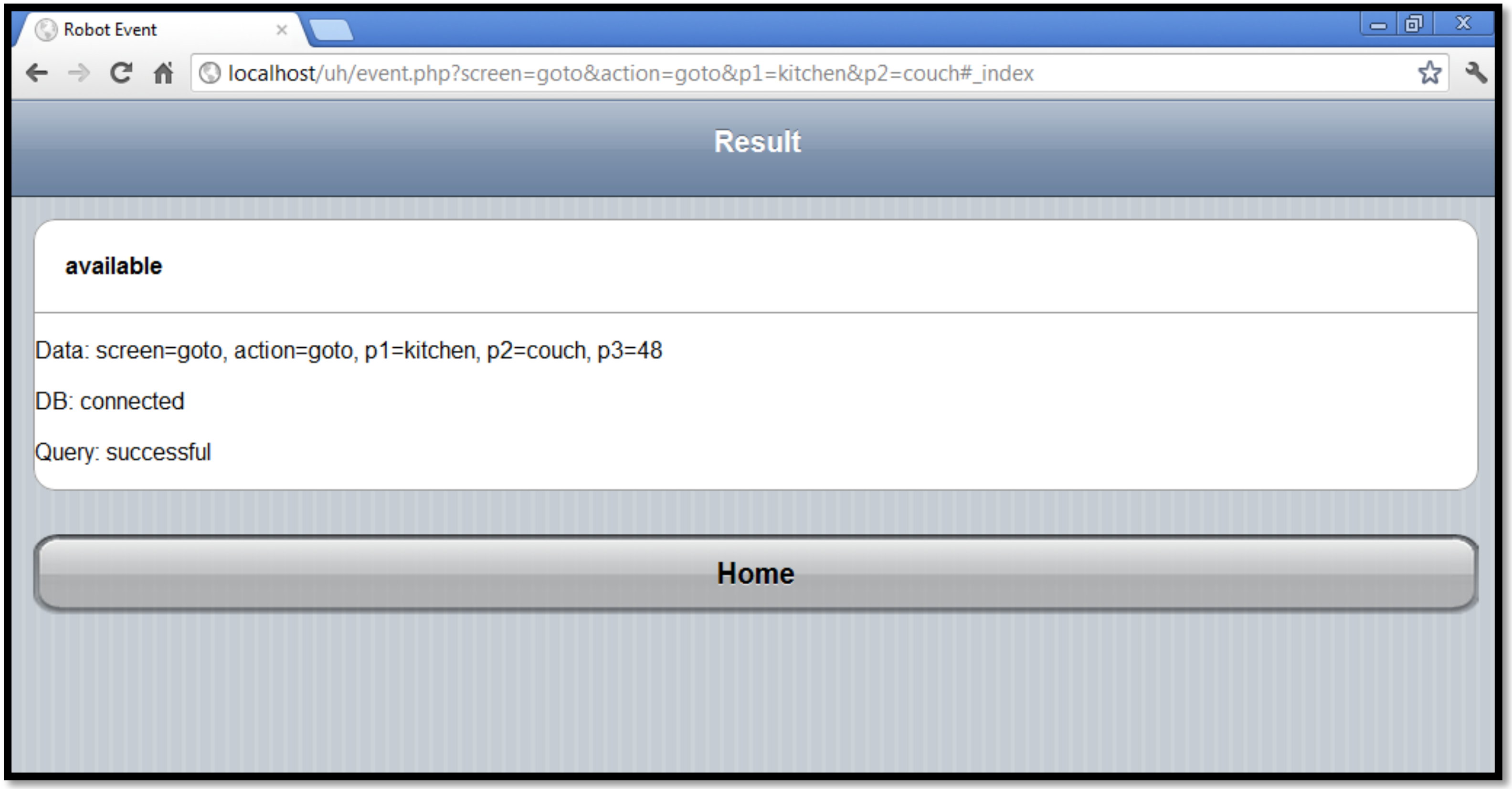
DB: connected

Query: successful

From Model

Tablet Interface

Robot Modeling



```
localhost/uh/event.php?screen=goto&action=goto&p1=kitchen&p2=couch#_index
```

Result

available

Data: screen=goto, action=goto, p1=kitchen, p2=couch, p3=48

DB: connected

Query: successful

Tablet Interface

Issues for Future Work

- Tablet lag/hang, resistive touchscreen, & mouse-centric rendering -> errors (e.g. fat finger, duplicate submission)
  - Consider better hardware (esp. screen)
  - Consider action confirmations ("Are you sure?")
- User needs to remember the events to trigger (e.g. open then close when robot transports)
  - Consider batch operations
- User needs to perform actions -> inaccurate timing of other actions
  - Evaluate impact
  - Consider human actor

Script Generation

Problem: how to generate scripts that are…

- **Hierarchical**. Describe HRI activities at various levels of abstraction
- **Probabilistic**. Incorporate flexible, yet structured stochasticity
  - Flexible: describes complex environmental regularities
  - Structured: assists in evaluating later learning
- **Scalable**. Can be easily and reliably manufactured and reproduced for multiple participants over longitudinal studies

Hierarchical Probabilistic Scripts

- Framework to…
  - describe script structure
  - visualize!
  - generate consistent scripts
- Instantiated for **eat**, **work**, **clean**, **relax**
  - Un+observable variables
  - Action sets and sequences
  - Controlled RNG seeds

- Features
  - Acyclic variable dep.'s
  - Flexible CPTs
  - Arbitrary depth
- Components
  - Soar: structure -> script
  - Java: describe & visualize, run Soar

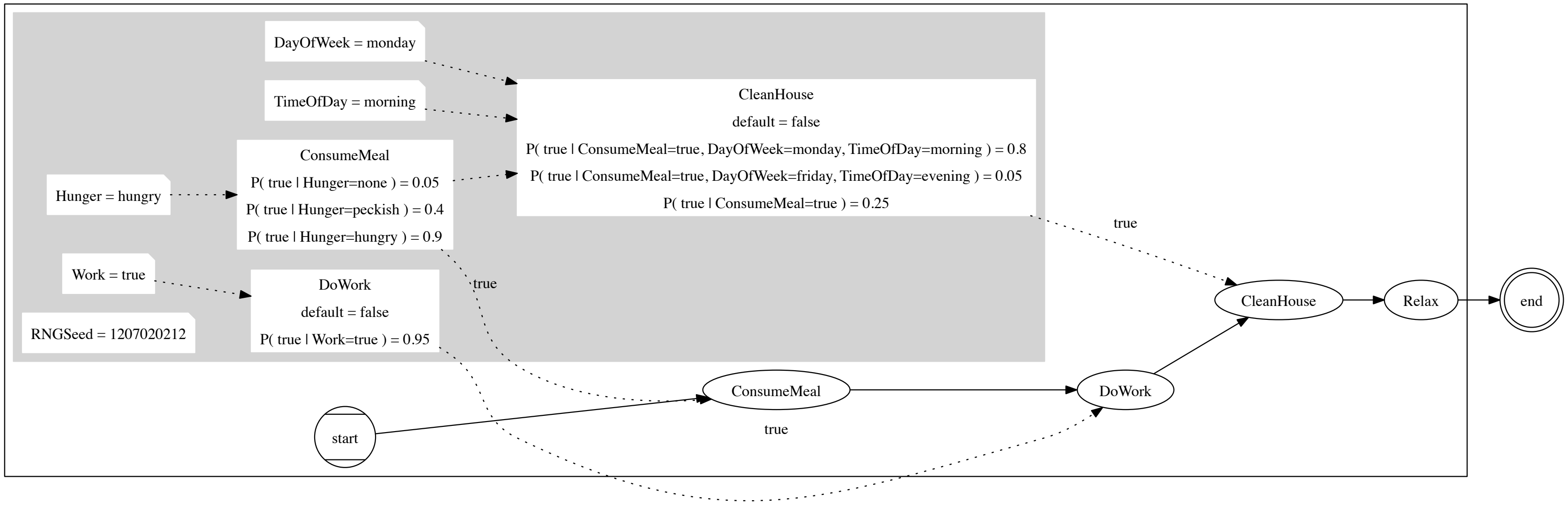
Example Template Description

```
HPPlan myPlan = new HPPlan(PlanType.seq,

 new HPPVariable[]{ vDayOfWeek, vTimeOfDay, vHunger, vWork,
                    vConsumeMeal, vDoWork, vCleanHouse, vSeed },

 new HPPAction[]{

  new HPPAction("ConsumeMeal",
   new HPPConstraint[]{ new HPPBinaryConstraint(vConsumeMeal, true) }, myMealPlan),

  new HPPAction("DoWork",
   new HPPConstraint[]{ new HPPBinaryConstraint(vDoWork, true) }, myWorkPlan),

  new HPPAction("CleanHouse",
   new HPPConstraint[]{ new HPPBinaryConstraint(vCleanHouse, true) }, myCleanPlan),

  new HPPAction("Relax", new HPPConstraint[]{}, myRelaxPlan),

 }

);
```

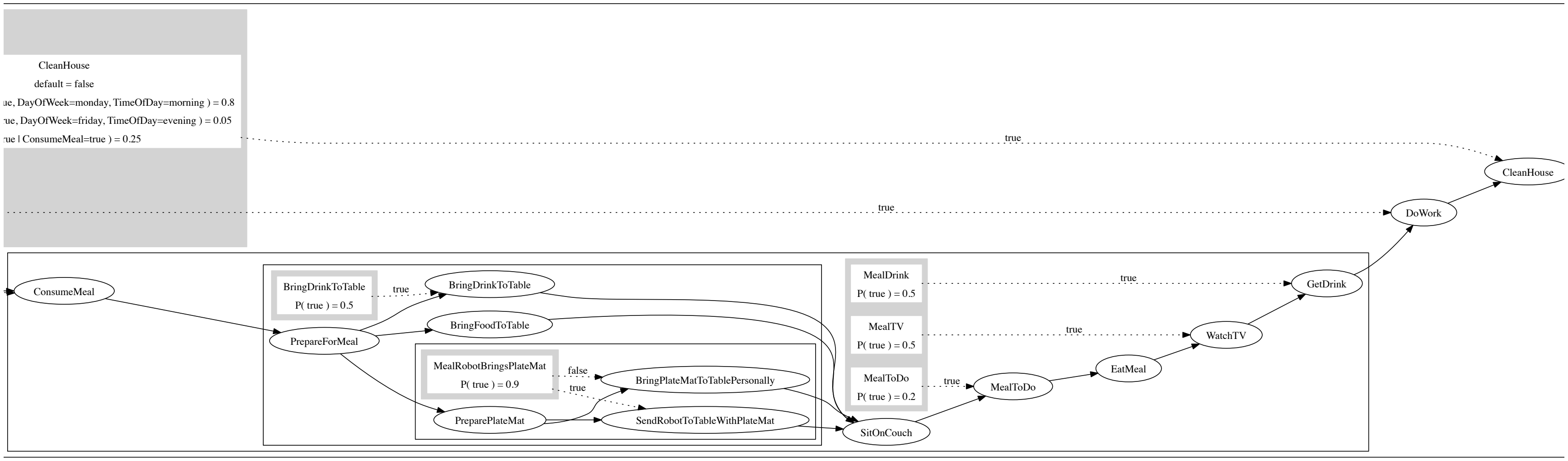
Example Variable Description

```
HPPVariable vConsumeMeal = new HPPBinaryVariable("ConsumeMeal", null, new HPPCPTBinaryEntry[] {
 new HPPCPTBinaryEntry(new HPPConstraint[]{ new HPPConstraint(vHunger, Hunger.none.toString()) }, true, 0.05),
 new HPPCPTBinaryEntry(new HPPConstraint[]{ new HPPConstraint(vHunger, Hunger.peckish.toString()) }, true, 0.4),
 new HPPCPTBinaryEntry(new HPPConstraint[]{ new HPPConstraint(vHunger, Hunger.hungry.toString()) }, true, 0.9),
});

HPPVariable vDoWork = new HPPBinaryVariable("DoWork", false, new HPPCPTBinaryEntry [] {
 new HPPCPTBinaryEntry(new HPPConstraint[]{ new HPPBinaryConstraint(vWork, true) }, true, 0.95),
});

HPPVariable vCleanHouse = new HPPBinaryVariable("CleanHouse", false, new HPPCPTEntry[] {
 new HPPCPTBinaryEntry(new HPPCPTBinaryEntry[]{
   new HPPBinaryConstraint(vConsumeMeal, true),
   new HPPConstraint(vDayOfWeek, DayOfWeek.monday.toString()),
   new HPPConstraint(vTimeOfDay, TimeOfDay.morning.toString())
 }, true, 0.80),

 new HPPCPTBinaryEntry(new HPPCPTBinaryEntry[]{
   new HPPBinaryConstraint(vConsumeMeal, true),
   new HPPConstraint(vDayOfWeek, DayOfWeek.friday.toString()),
   new HPPConstraint(vTimeOfDay, TimeOfDay.evening.toString())
 }, true, 0.05),

 new HPPCPTBinaryEntry(new HPPCPTBinaryEntry[]{ new HPPBinaryConstraint(vConsumeMeal, true) }, true, 0.25),
});
```

Example Template Visualization (1)


DayOfWeek = monday
TimeOfDay = morning
CleanHouse
default = false
P(true | ConsumeMeal=true, DayOfWeek=monday, TimeOfDay=morning) = 0.8
P(true | ConsumeMeal=true, DayOfWeek=friday, TimeOfDay=evening) = 0.05
P(true | ConsumeMeal=true) = 0.25
ConsumeMeal
P(true | Hunger=none) = 0.05
P(true | Hunger=peckish) = 0.4
P(true | Hunger=hungry) = 0.9
Hunger = hungry
Work = true
DoWork
default = false
P(true | Work=true) = 0.95
RNGSeed = 1207020212
true
true
CleanHouse
Relax
end
ConsumeMeal
DoWork
true
start

Example Template Visualization (2)



Example Script Output

```
Script Input> year=2012 month=7 day=5 time=morning hunger=peckish work=true trial=3

Resulting KB:
  Hunger=peckish
  RelaxationToDo=true
  RelaxationActivity=read
  Work=true
  WorkToDo=true
  DoWork=true
  ConsumeMeal=true
  MealRobotBringsPlateMat=true
  CleanHouse=false
  WorkMusic=true
  DayOfWeek=thursday
  RNGSeed=1207050113
  MealDrink=true
  TimeOfDay=morning
  WorkLocation=couch
  BringDrinkToTable=true
  MealToDo=false
  MealTV=false
  WorkDuration=ten
  RelaxationPosition=lie

Resulting Script:
  start -> SendRobotToTableWithPlateMat -> BringFoodToTable -> BringDrinkToTable -> SitOnCouch -> EatMeal -> GetDrink -> TurnOnMusic -> WorkToDo ->
WorkForTenMinutes -> GetPeriodicalFromDrawers -> LieOnCouch -> RelaxToDo -> Read
```

Robot House Data

- Integration test/proof of concept
  - – 1 day, 1 subject (me)
  - – 3 parameter settings, 2 trials each
    - Rushed (e.g. did not "work for 6 minutes")
    - Noisy (e.g. ongoing Robot House experiments, my mistakes, tablet issues)

- Data
  - – Script-generation parameters
  - – MySQL dumps: sensor log, tablet events

Future Work

- Plan-structure GUI (undergrad)
- Data Collection (grad)
  - Solidify script template
  - Multiple participants, many days/week, many weeks
- Data analysis (grad)
  - Learning algorithms for sequences, regularities, causal & hierarchical induction

Thanks :)

Questions?