Enabling Trust with Behavior Metamodels

Scott Wallace
WSU Vancouver
Challenges

- Software assistants are increasingly a part of everyday life
- What will constrain the use of these assistants?
  - Technology? Psychology?
Technology as a Constraint

- The most obvious constraint on tomorrow’s intelligent assistants
  - “we aren’t doing that yet because we don’t know how”
  - “…because we don’t have computers/sensors/algorithms/etc that are precise/fast enough”
- Focus of most AI research
Psychology as a Constraint

- Less obvious, less explored possibility
- Perhaps we aren’t willing to turn all tasks over to computerized assistants...
  - How do engineers weigh the risks and benefits of the technology they develop?
  - How do end users determine when and what technology to adopt?
Psychological Constraints

- Potential concerns:
  - Will this project/invention be safe for society?
  - Will it be a useful tool?

- Approach:
  - Validation / Testing
    - Did we make what we intended to?
The end user...

- Potential concerns:
  - Will this project/invention be safe?
  - Will it be a useful to me?

- Needs:
  - Marketing?
  - Trust
Trust is a critical factor in developing human-human and human-computer relationships.

We can design systems so as to help facilitate trust.

Trust seems most important for end-users, especially early adopters, but the underlying components of trust will also benefit developers.
Trust

- Examined three models of trust
  - Recently cited / multi-disciplinary
  - Developed from models with longer history

- Based on this survey, four common properties can be identified
  - Understandability, predictability, similarity, liability
Understandability/Predictability

- Based on reputation of other party
- Based on knowledge of other party’s behavior
  - Knowledge based trust (Ratnasignham)
  - Cognitive Trust (Lewis & Weigert)
  - Habitus (Misztal via Fahrenholtz, Bartlet)

- Important for end-users and developers
Between Humans & Computers

- An explanation of why software aids may make mistakes can increase trust

- Systems that can justify their actions engender greater trust
Similarity

- Can the parties in the trust relationship find common ground?

- Empathy, common values (Lewis & Weigert)
- Solidarity, familial associations (Misztal via Fahrenholtz, Bartlet)
Between Humans & Computers

- Users find programs more credible when they are considered part of same group as user (e.g., company).

- Agents that use a conversational strategy that is consistent with the users’ behavior engender more trust.
Liability

- Deterrence based trust (Ratnasignham)
  - Early form of trust
  - Supported by threat of punishment

- Emotional trust (Lewis & Weigert)
  - Entering trust relationship causes bond
  - Breaking bond causes pain/wrath
Metamodels: Enabling Trust

- Metamodels are high-level descriptions of the agent’s behavior
  - They are easy to create
  - “Easy” to understand
  - Consistent with agent’s behavior
A Hierarchical Representation

- Similar to Finite State Machine, AND/OR Tree
- Describes potential sequences of behavior
How Metamodels May Aid Trust

- Understandability
  - Illustrates reasoning path leading to a state
- Predictability
  - Illustrates reasoning path extending from a state
- Similarity
  - Agent’s reasoning process may map to user’s
- Liability
  - ???
Exploring Understandability…

- How can we see if metamodels improve understandability?
  - By watching developers find bugs in a program.

- Begin with an existing Soar agent performing a simple goal-directed task: “Correct Behavior”.
  - Create two more agents based on this original: “A”, “B”.
  - New agent’s behave somewhat differently.
  - Participants observe correct and flawed behavior.
  - Do metamodels help find behavioral differences?
Three Agent Programs

- Original Agent ("Correct Behavior")
  - This is the specification of how to behave
  - Serves identical role to a human expert we may want to emulate
  - 4 distinct behaviors
- Flawed agent "A"
  - Occasionally pursues inappropriate goals
  - 12 distinct behaviors, 4 are consistent with specification
- Flawed agent "B"
  - Occasionally replaces one goal for another (inappropriately)
  - 8 distinct behaviors, 4 are consistent with specification
Participants’ Task

- Five participants – all experienced with Soar
- Each participant looks for bugs in “A” and “B”
  - On one agent users were **aided** by metamodels
  - On other agent users were **unaided**
- In **aided** task, users get metamodel of specification, and metamodel of flawed agent
- In **unaided** task, users get behavior sequences (observations) from specification
- In **both** tasks, users must identify error verbally, then proceed to fix it
Finding & Fixing Error

<table>
<thead>
<tr>
<th>Participant</th>
<th>Time To Correct Behavior (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aided by HBR</td>
</tr>
<tr>
<td>2</td>
<td>Unaided by HBR</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Scott A. Wallace

AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants.
March, 2007
Finding Error

Time To Identify Error (min.)

Aided by HBR
Unaided by HBR

Participant

Scott A. Wallace

AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants.
March, 2007
Fixing

![Graph showing time to fix errors for participants aided and unaided by HBR.](image)
Conclusions

- Multi-resolution models of behavior may be valuable tools for helping both developers and end users.

- A key challenge is to abstract away the correct features.

- As we vary the level of abstraction there should be a cost benefit curve associated with interpreting the model can we quantify this?