Software-Enabled Solutions for Human-on-the-Loop Autonomous Systems

Rebekka Wohlrab
Humans as the ones that do what a system can’t do

Humans as maintainers

Humans as bystanders

Human as input-givers

Human as collaborators

Challenges of human-machine collaboration

• Humans and machines speak different languages
• Complexity of the problem space with many relevant qualities (safety, security, performance, cost, reliability, ...)
• Uncertainty
• Trust
• ...


Human-on-the-loop autonomous systems

What does the human do?

What does the human want the robot to do?

What does the robot do?

What does the robot want the human to do?

How can multiple human users find a consensus on what the robot should do?

This Photo by Unknown Author is licensed under CC BY-SA-NC
Human-on-the-loop autonomous systems

What does the human do?
What does the human want the robot to do?
Preference elicitation [2, 3]
What does the robot want the human to do?
What does the robot do?
How can multiple human users find a consensus on what the robot should do?

Preference elicitation [2, 3]
Negotiation [2]
Monitoring [4]
Tradeoff Explainability [1]


This Photo by Unknown Author is licensed under CC BY-SA-NC
Qualities in automated mission planning

- **Problems:**
  - Autonomous systems often do not allow for changing priorities of objectives at runtime
  - Tradeoffs tend to be opaque

Options:

1. **Option 1:**
   - Travel time
   - Safety
   - Privacy

2. **Option 2:**
   - Travel time
   - Safety
   - Privacy

3. **Option 3:**
   - Travel time
   - Safety
   - Privacy

How to select a plan?
• Using a planner
• It needs an optimization function
• Need to indicate weights/priorities for different objectives
  • Safety: rather unimportant (0.1)
  • Privacy: rather unimportant (0.1)
  • Travel time: very important (0.8)

utility(plan) = 0.8·utility_travel_time(plan) + 0.1·utility_safety(plan) + 0.1·utility_privacy(plan)

• What should those weights/priorities be?
• How do they impact the generated plans?
Markov Decision Process

mdp

formula goal = rLoc=6;
label "end" = rLoc=6 & !computeGo & barrier;

module module_1
  rLoc : [0..6] init 1;
  // moveTo
  [moveTo_LP_L2_RP_] rLoc=1 -> 0.0:(rLoc'=1) + 1.0:(rLoc'=2);
  [moveTo_LP_L4_RP_] rLoc=2 -> 0.2:(rLoc'=2) + 0.8:(rLoc'=4);
  //...
endmodule

rewards "travelTime"

[moveTo_LP_L2_RP_] rSpeed=0 & rLoc=1 : 1.0;
[moveTo_LP_L4_RP_] rSpeed=1 & rLoc=2 : 0.8;
endrewards

\[ \text{multi}(R_c^{\min} [C], P_{\geq 1} [F(\nothing) \land G(\neg\nothing)], R_{\leq 5}^{ct} [C]) \]
Generating data...

<table>
<thead>
<tr>
<th>priority of travel time</th>
<th>priority of safety</th>
<th>priority of privacy</th>
<th>cost of travel time</th>
<th>cost of safety</th>
<th>cost of privacy</th>
<th>number of steps</th>
<th>decision at Location L1</th>
<th>decision at Location L2</th>
<th>decision at Location L3</th>
<th>decision at Location L4</th>
<th>decision at Location L5</th>
<th>decision at Location L6</th>
<th>decision at Location L7</th>
<th>decision at Location L8</th>
<th>decision at Location L9</th>
<th>decision at Location L10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>50</td>
<td>100</td>
<td>10</td>
<td>25</td>
<td>L2</td>
<td>L4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>0.9</td>
<td>50</td>
<td>100</td>
<td>10</td>
<td>25</td>
<td>L2</td>
<td>L4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0.8</td>
<td>50</td>
<td>100</td>
<td>10</td>
<td>25</td>
<td>L2</td>
<td>L4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0</td>
<td>0.7</td>
<td>45</td>
<td>96</td>
<td>20</td>
<td>22</td>
<td>L3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Principal Component Analysis (PCA)

- Strongly correlated
- Negatively correlated

Priority of privacy

Priority of safety

Priority of travel time

Less important than the other priority variables
### Generating data...

<table>
<thead>
<tr>
<th>priority of travel time</th>
<th>priority of safety</th>
<th>priority of privacy</th>
<th>cost of travel time</th>
<th>cost of safety</th>
<th>cost of privacy</th>
<th>number of steps</th>
<th>decision at Location L1</th>
<th>decision at Location L2</th>
<th>decision at Location L3</th>
<th>decision at Location L4</th>
<th>decision at Location L6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>50</td>
<td>100</td>
<td>10</td>
<td>25</td>
<td>L2</td>
<td>L4</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>0.9</td>
<td>50</td>
<td>100</td>
<td>10</td>
<td>25</td>
<td>L2</td>
<td>L4</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0.8</td>
<td>50</td>
<td>100</td>
<td>10</td>
<td>25</td>
<td>L2</td>
<td>L4</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.3</td>
<td>0</td>
<td>0.7</td>
<td>45</td>
<td>96</td>
<td>20</td>
<td>22</td>
<td>L3</td>
<td>...</td>
<td>L6</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Other application domains for tradeoff explanations

• Software architecture
• Smart manufacturing
• Cloud-based systems design (thesis with Volvo Cars)


Qualities in automated mission planning

- Problems:
  - Autonomous systems often do not allow for changing priorities of objectives at runtime
  - Tradeoffs tend to be opaque

Option 1:
- Travel time
- Safety
- Privacy

Option 2:
- Travel time
- Safety
- Privacy

Option 3:
- Travel time
- Safety
- Privacy

Self-adaptive systems

• A self-adaptive system is a system that can handle changes and uncertainties in its environment, the system itself, and its goals autonomously.

Architecture: Elicitation and Explanation Framework

### Participants

<table>
<thead>
<tr>
<th>Part.</th>
<th>Occupation</th>
<th>Experience with technical jects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Engineering manager</td>
<td>3-5 yrs.</td>
</tr>
<tr>
<td>2</td>
<td>Graduating software development student</td>
<td>1-2 yrs.</td>
</tr>
<tr>
<td>3</td>
<td>Graduating software engineering student</td>
<td>1-2 yrs.</td>
</tr>
<tr>
<td>4</td>
<td>Software developer</td>
<td>6+ yrs.</td>
</tr>
<tr>
<td>5</td>
<td>Backend developer</td>
<td>3-5 yrs.</td>
</tr>
<tr>
<td>6</td>
<td>Cloud engineer/architect</td>
<td>6+ yrs.</td>
</tr>
<tr>
<td>7</td>
<td>UX-design student</td>
<td>0 yrs.</td>
</tr>
<tr>
<td>8</td>
<td>Consultant manager</td>
<td>6+ yrs.</td>
</tr>
<tr>
<td>9</td>
<td>Software architect</td>
<td>6+ yrs.</td>
</tr>
<tr>
<td>10</td>
<td>Product owner</td>
<td>6+ yrs.</td>
</tr>
</tbody>
</table>
Travel Time is the highest weight. However, the optimal path for these weights is not the fastest path. The optimal path is 1.33 times as safe and 0.54 times as intrusive, while only being 0.94 times as fast.

The optimal path is the same for the following attributes: [[Privacy, Cost Function, Safety]]

Important nodes in the graph are: [L8, L17, L25, L31]. These nodes acts like hubs, where the optimal paths for the chosen attributes either diverge or converge.
Method for Utility Function Definition

- Based on the Analytic Hierarchy Process (AHP) – pairwise comparison of quality attributes

\[ U(m) = 0.8 \cdot \text{safety}(m) + 0.1 \cdot \text{duration}(m) + 0.1 \cdot \text{privacy}(m) \]


Create an AHP matrix

• Pairwise comparison of qualities
  • Extremely preferred 9
  • Very strongly preferred 7
  • Strongly preferred 5
  • Moderately preferred 3
  • Equally preferred 1

• Creation of a reciprocal matrix A

• Normalized principal eigenvector of the matrix A represents the relative priorities of the qualities

\[ U(m) = 0.799 \cdot \text{safety}(m) + 0.105 \cdot \text{duration}(m) + 0.096 \cdot \text{energy}(m) \]

<table>
<thead>
<tr>
<th></th>
<th>Travel time</th>
<th>Safety</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Travel time</td>
<td>1</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Safety</td>
<td>1/9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Privacy</td>
<td>1/7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Principal eigenvalue: \( \lambda_{\text{max}} \approx 3.01 \)

Corresponding normalized eigenvector: (0.799, 0.105, 0.096)
Optimized for cost function

Travel Time Cost: 8.05
Safety Cost: 6.00
Privacy Cost: 7.00

Travel Time is the highest weight. However, the optimal path for these weights is not the fastest path. The optimal path is 1.33 times as safe and 0.54 times as intrusive, while only being 0.94 times as fast.

The optimal path is the same for the following attributes: [[Privacy, Cost Function, Safety]]

Important nodes in the graph are: [L8, L17, L25, L31]. These nodes acts like hubs, where the optimal paths for the chosen attributes either diverge or converge.
Evaluation

- Listing costs
- Visualizing optimal path
- Textual contrastive explanation
- Comparing paths
- Tooltips of overlapping paths
- Description of important nodes
- Description of equal paths
- Tooltips of costs

Response:
- very unhelpful
- unhelpful
- neutral
- helpful
- very helpful
What’s next?

1) When should systems provide what kinds of explanations? (in runtime robot mission planning, when cybersecurity attacks happen, when developers make design decisions, ...)

Monitoring [4]
Explainability [1]

What does the human want the robot to do?

Preference elicitation [2, 3]

2) How can systems elicit humans’ preferences in different contexts?

3) When and how should robots initiate collaborative tasks?

4) How can systems deal with conflicting requirements?

What does the robot want the human to do?

How can multiple human users find a consensus on what the robot should do?

What does the robot do?

What does the human do?

What does the human want the robot to do?
ShiftLeft – a WASP NEST project
Thank you!
Rebekka Wohlrab
wohlrab@chalmers.se