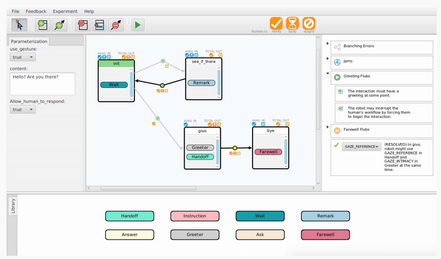
Computational Tools for Human-Robot Interaction Design

David Porfirio[1], Maya Cakmak[2], **Allison Sauppé[3]**, Aws Albarghouthi[1], Bilge Mutlu[1]
(1) University of Wisconsin–Madison, (2) University of Washington, (3) University of Wisconsin–La Crosse

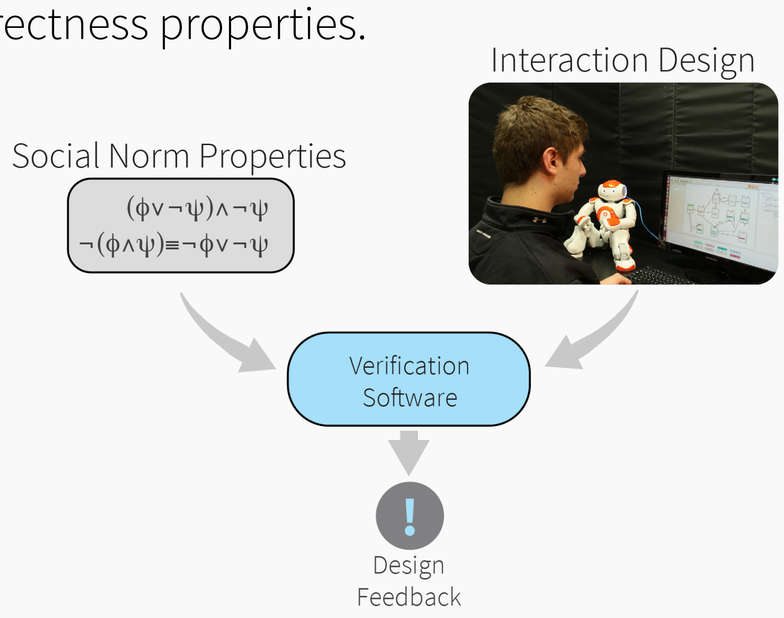
We have developed a series of **software tools** to assist with the **design of human-robot interactions** with social robots. Each of these tools recognizes that interaction designers have **varying levels of expertise** in both programming and the domain of the interaction that is being designed. Our tools rely on concepts from **programming languages** to provide assistance to designers when programming human-robot interactions.

Verifying Social Norms

We designed a visual programming environment, RoVer, which enables interaction designers to visually program human-robot interactions. RoVer employs formal verification to ensure that designed interactions satisfy a set of social norm correctness properties.



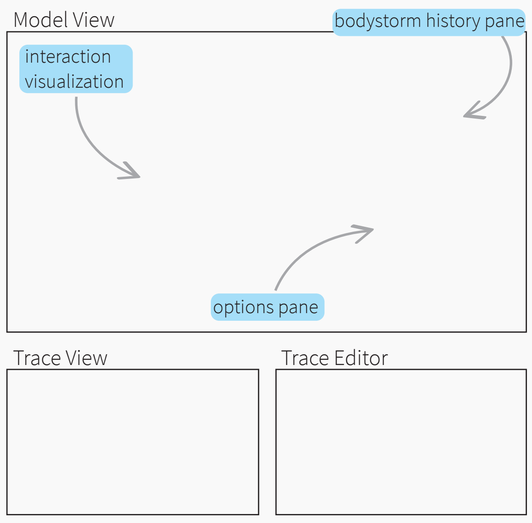
The RoVer interface.



The pipeline through which RoVer provides feedback to designers.

We found that designers were able to more easily identify and understand social norm violations when using assistance from formal verification.

Capturing Designer Intent

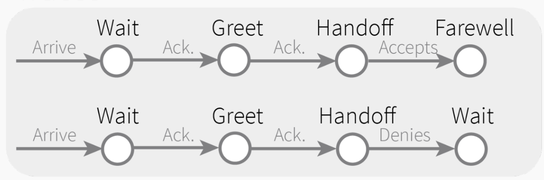


The Synthé interface.

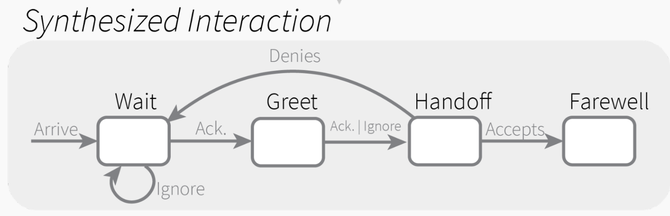
Designers role-play an interaction between a human and a robot.

We designed another programming environment, Synthé, which records design teams role-playing an interaction between a human and a robot, and synthesizes an interaction based on their performances.

Traces



Synthesized Interaction

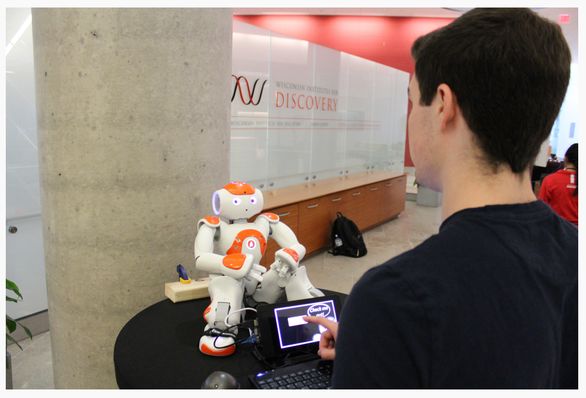


An example of how program synthesis works within Synthé.

Synthé represents designer performances as execution traces. We found Synthé to be effective in solving interaction programs that accept user execution traces.

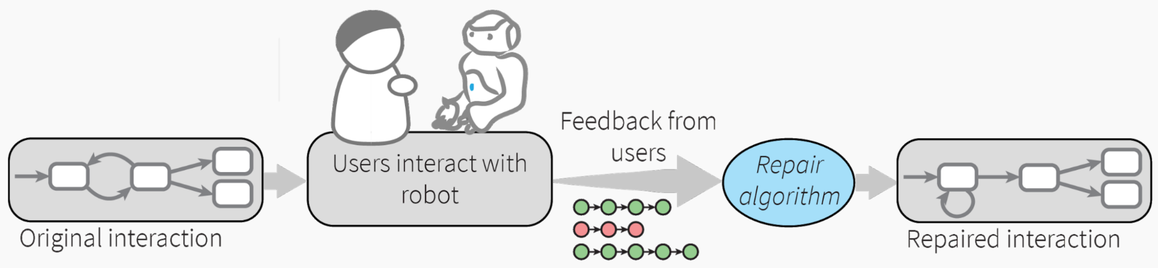
Automatically Transforming Interactions

We designed and implemented a system for transforming robot programs based on feedback from a social context.



An example setup for collecting feedback from users.

In our system, end-users experience and rate execution traces of an interaction design.



A cycle of collecting user feedback and transforming the interaction.

In our system, end-users experience and rate execution traces of an interaction program. A repair algorithm then transforms the interaction program structure such that poorly-rated traces are excluded, while positively-rated traces are included. Our evaluation shows limiited evidence that this system improves user experience over time.

2020 National Robotics Initiative (NRI) Principal Investigators' Meeting
FEBRUARY 27 - 28, 2020 | ARLINGTON, VIRGINIA

Award ID #s: 1651129
1925043