artificial intelligence research koblenz

Probabilistic Models for the Verification of Human-Computer Interactions

www.uni-koblenz.de/ag-ki

Introduction

The user of an application is not always aware of all possible consequences of her interaction potential. **Due to incorrect interaction, time and money is lost**, or even humans injured. The consideration of user errors and their overall impact on the system should form an important part of an analysis of a system's usability, safety, and security.

The goal of our technique is to proof properties of the interaction, i.e., to formally prove that all possible ways of HCI in a given scenario conform to the requirements.

The basis of our work is the (non-probabilistic) method of Beckert and Beuster (2006) for the formalization, analysis, and verification of user interfaces. It is based on GOMS, which is a well-established (non-formal) user modeling method.

... to answer questions such as ...

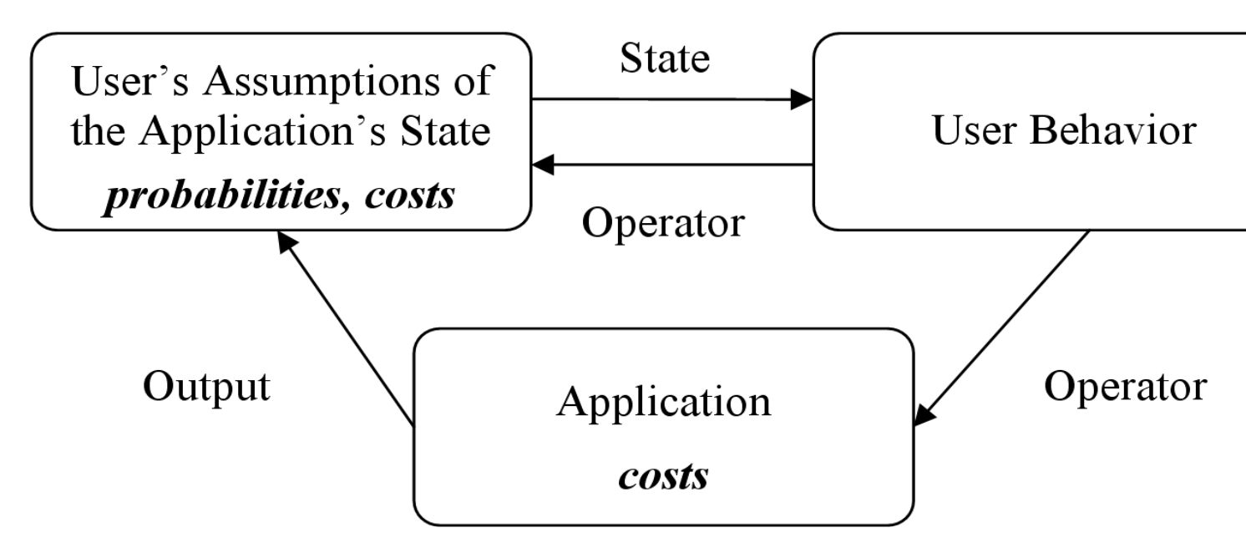
Our Method

Assumption: A user can only interact correctly with a system if she always **correctly interprets the system's state**, including the internal state and relevant data – in Linear Temporal Logic:

$$G((a_0 \leftrightarrow c_0) \wedge (a_1 \leftrightarrow c_1) \wedge \ldots \wedge (a_n \leftrightarrow c_n))$$

where $a_0,\ldots,a_n$ are the critical properties of the application, and $c_0,\ldots,c_n$ are the user's assumptions about whether theses properties hold or not.

HCI model: In order to apply automated reasoning, the following components are needed:
(1) a formal GOMS model describing the **user behavior** and its corresponding IOLTS L_{goms},
(2) a component representing the user's **assumptions of the software's state** and its corresponding probabilistic IOLTS L_{user}, and
(3) a component representing the **application** itself and its corresponding IOLTS $L_{application}$.



The mutual composition of the three components provides the **complete model of the interaction**, making complete formal modeling possible:

$$L_{interaction} = (L_{user} \mid\mid L_{goms}) \mid\mid L_{application}$$

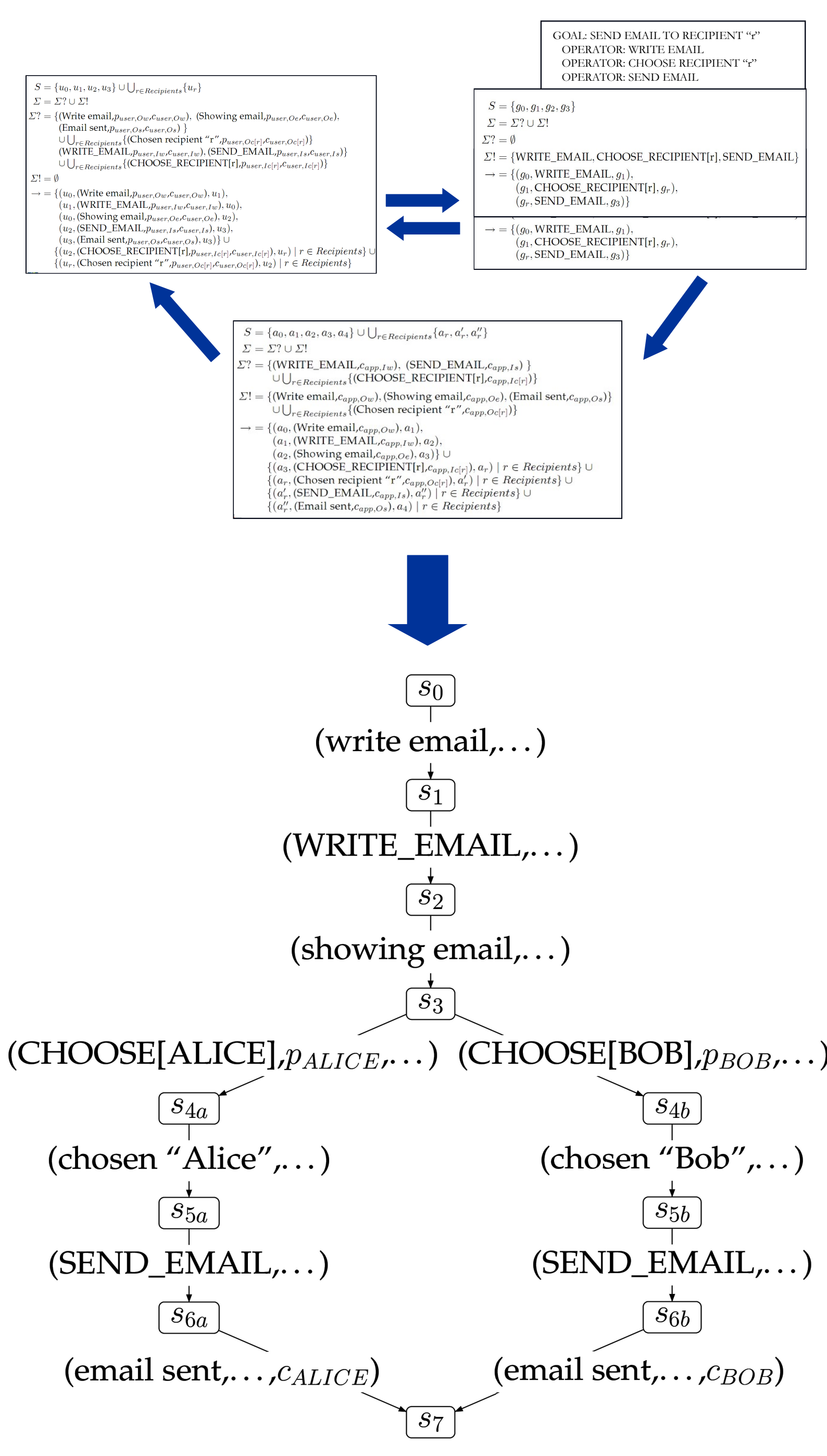
Input Output Labeled Transition Systems (IOLTS) are used to model the components of the interaction. An IOLTS is a tuple $L = (S; \Sigma; s_0; \rightarrow)$ where S is a set of states, $s_0 \in S$ is an initial state, $\rightarrow \subseteq S \times \Sigma \times S$ is a transition relation, and a set of labels $\Sigma = \Sigma? \cup \Sigma! \cup \Sigma I$. We call $\Sigma?$ the input alphabet, $\Sigma!$ the output alphabet, and ΣI the internal alphabet.

"What is the probability that the user will unintentionally send confidential information to unauthorized recipients?"

Mailing Example

Scenario: A user interacts with an email program. She intends to write a confidential email and send it to Alice. However, with a certain probability, she can **choose Bob, which will result in high costs**.

Possible situations:
(1) the user accidentally selects Bob, **notices** her mistake, and **corrects** her mistake,
(2) she selects Bob again, but **does not notice** her mistake, and **sends the email to Bob**, and
(3) she selects Bob, **notices** it, fails to change the addressee, and **sends the email to Bob**.



Some properties in PCTL and natural language:

P$_{\geq 0.95}$ [$\Diamond$ `sentTo(Alice)`]
"the email is sent to Alice in at least 95% of the interactions"

P$_{\leq 0.20}$ [$\Diamond$ `(chosen(Bob)` $\wedge$
`(chosen(Bob)` **U** `sentTo(Bob)))`]
"with a probability of at most 0.20, Bob is selected as the addressee and the error is not corrected"

R$_{=?}$ [**F** `email_sent`]
"the average total cost for sending an email"
(can be computed using PRISM's support for costs)

"How much time does a user on average need to send an email?"

Different User Models

A way to lower the probability of sending the email to Bob would be to introduce additional dialogue boxes that would require the user to confirm her selection.

Adjusting the probabilities:

Fixed, "accidental execution" $p(a, S) = 0.8$

Different levels of the confirmation's complexity

$$p(a, S) = \begin{cases} 0.7 & \text{if } S.confirmationType = simple \\ 0.9 & \text{if } S.confirmationType = complex \end{cases}$$

Learning user

$$p(a, S) = \begin{cases} 0.6 \quad \text{and} \quad \{S.learned := true\} & \text{if } \neg S.learned \\ 0.8 \quad \text{and} \quad \{S.learned := true\} & \text{if } S.learned \end{cases}$$

Inattentive user

$$p(a, S) = \frac{1}{S.inattentionLevel + 2} + 0.4 \quad \text{and} \quad \{S.inattentionLevel{+}{+}\}$$

If the user is modeled to react **reasonably** (versus **inattentively**) to a confirmation, it can be proven that the probability to send the email to Bob is lowered.

Conclusion

In this paper, we have introduced a method for the formalization of probabilistic models of HCI.

It provides the options to
(1) **model probabilistic behavior** of users and applications, and
(2) **add costs** to the steps of the interaction.

This allows at to
(1) **ask quantitative questions**, such as "what is the probability that the user will unintentionally send confidential information to unauthorized recipients", and
(2) **verify the corresponding properties**.

Our method can
(1) help to develop user interfaces by avoiding the trap of having to do human error analysis at the latest stages in the design process
(2) help to "falsify" a designer's assumptions of human performance. By setting up several scenarios, the analyst is able to discover the impact of alternative designs on the expected costs of HCI.

Thus, **formal modeling and an examination of the expected costs can together contribute to the design of user interfaces**, which have to be robust to the error-prone behavior of humans.

artificial intelligence research koblenz
UNIVERSITÄT KOBLENZ · LANDAU

Bernhard Beckert, Markus Wagner
Department of Computer Science, University of Koblenz, Germany
beckert@uni-koblenz.de, wagnermar@uni-koblenz.de