An Approach to Generating Human-computer Interfaces from Task Models

Matthew L. Bolton, Ph.D.
Department of System and Industrial Engineering
State University of New York at Buffalo

Samaneh Ebrahimi
Department of Mechanical and Industrial Engineering
University of Illinois at Chicago

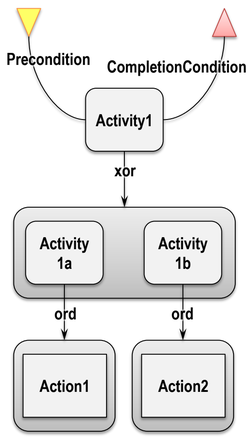
Meng Li
Department of System and Industrial Engineering
State University of New York at Buffalo

Introduction

- Human-computer interaction (HCI) can occur in situations unanticipated by designers and thus not always be usable or support operator tasks
- The proposed method uses existing work, which shows that formal methods and L* machine learning can be used to analyze and design robust HCI, to automatically generate interface designs from task models guaranteed to usably support operator task goals

Task Model

- Human task behavior is represented using the Enhanced Operator Function Model (EOFM)
- Human behavior is captured as a hierarchy of goal-directed activities and actions
- Strategic knowledge describes when activities are relevant
- Decomposition operators define the ordinal relationships between activities and actions

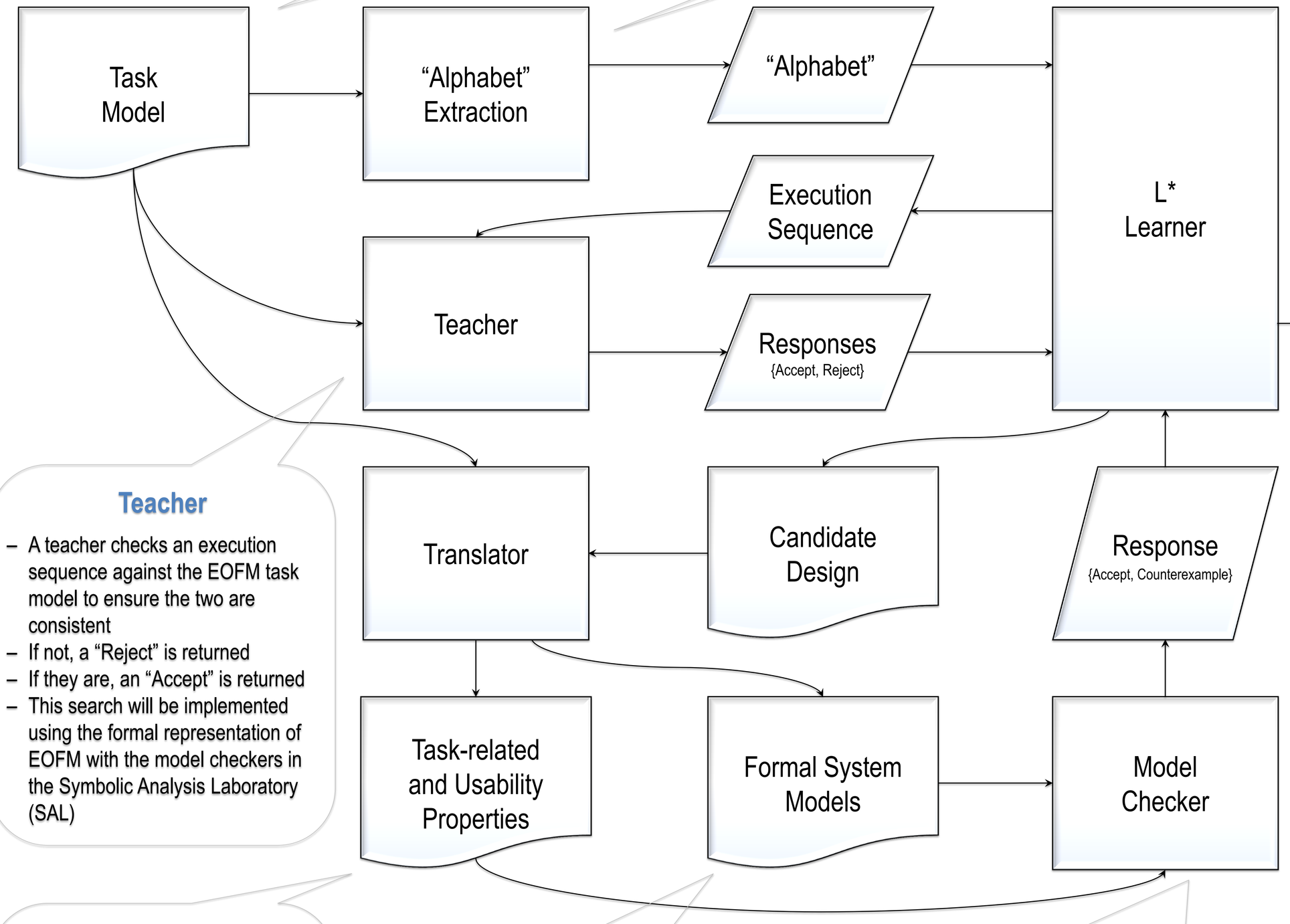


"Alphabet" Extraction

- Task models are parsed to find the events that trigger changes in the human-computer interface:
  - Human actions
  - System conditions from task strategic knowledge
- These are treated as characters in the alphabet accepted by a finite state machine representing the interface design

L* Learner

- An L* algorithm iteratively learns a finite state automata representation of an interface design by issuing queries and producing candidate designs that are examined by other processes in the approach
- Queries represent execution sequences ("strings" of "alphabet" characters)
- The L* Learner receives inputs indicating if a produced execution sequence is valid or not
- Candidate interface designs represent learned interfaces consistent with previous inputs
- The L* Learner receives inputs indicating if the candidate is valid or not and, if not, an illustrative unacceptable execution sequence (a counterexample)



Teacher

- A teacher checks an execution sequence against the EOFM task model to ensure the two are consistent
- If not, a "Reject" is returned
- If they are, an "Accept" is returned
- This search will be implemented using the formal representation of EOFM with the model checkers in the Symbolic Analysis Laboratory (SAL)

Interface Design

- The produced finite state machine will represent a human-computer interface design guaranteed to always support the human behavior in the task model and adhere to the checked usability properties
- The formal model can then be used in the design, implementation, and testing of the actual human-computer interface

Task-related and Usability Properties

- A translator uses the design and task model to generate LTL specification properties for checking the candidate design
- Task-related specification properties that assert desirable properties of task execution are generated from the EOFM task models
- Usability property patterns and the candidate design are used to create usability specifications

Formal System Models

- Two formal system models are created
- Model 1 represents the human operator interacting with the candidate design with the behavior in the task model
- Model 2 represents the candidate interface design's behavior independently of the human behavior in the task model

Model Checker

- A model checker (SAL) is used to evaluate the acceptability of the candidate design
- It does this by checking the two formal system models against the generated specification properties
- Model 1 is checked against the task-related properties
- Model 2 is checked against all other usability properties
- If a verification fails, the counterexample illustrating why the interface is unacceptable is returned
- If all of the specifications are verified, "Accept" is returned

Conclusions and Future Work

- The implementation of this approach is currently underway
- If successful, the approach has the potential to improve the usability of human-computer interfaces and encourage user-centered design
- The implemented approach will be tested and validated using artificial examples as well as a PCA pump application

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1429910.