

State Event Models for the Formal Analysis of Human-Machine Interactions

Sébastien Combéfis¹ Dimitra Giannakopoulou²
Charles Pecheur¹

¹Université catholique de Louvain (UCL)
ICT, Electronics and Applied Mathematics Institute (ICTEAM)

²NASA Ames Research Center (ARC)

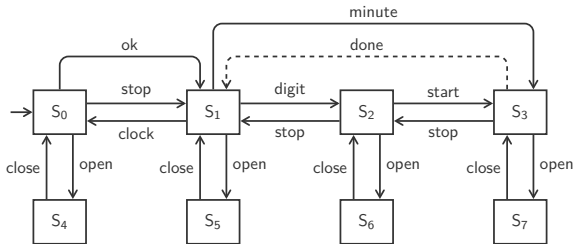
March 26, 2014

Introduction

- Automated formal analysis techniques for HMI systems
- Detection of potential automation surprises
- Conformance relation between actual system and mental model according to which it is operated

Formal Modelling

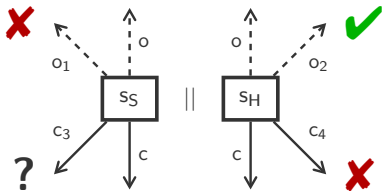
- HMI-LTS extends LTS with inputs and outputs:
 - **Commands** executed by the user
 - **Observations** executed by the system and observed by the user
 - **Internal actions** invisible to the user



Interaction Model

■ Interaction:

- Represented with the **synchronous parallel composition**



■ Bad situations:

- A command missing on the system model (c_4)
- An observation missing on the mental model (o_1)

Full-control property

- **Full-control property** captures safe interaction
- During the **interaction** between a user and a system:
 - The user must know exactly the possible commands...
 - ...and at least all the possible observations

\mathcal{H} fc \mathcal{S} if and only if :

$\forall \sigma \in \mathcal{L}^*$ such that $s_S \in (s_{0_S} \text{ after } \sigma)$ and $s_H \in (s_{0_H} \text{ after } \sigma)$:

$$A^c(s_S) = A^c(s_H) \quad \text{and} \quad A^o(s_S) \subseteq A^o(s_H)$$

Generation Problem

- **Goal:** Given the model of a system, **automatically** generate a **minimal full-control** conceptual model
- **Motivation:**
 - Extract the minimal behaviour of the system, so that it can be controlled **without surprise**
 - Help to build **artifacts**: manuals, procedures, trainings, ...
 - If such abstraction does not exist, provide feedback to help **redesigning** the system

ADEPT toolset

- Automatic Design and Evaluation Prototyping Toolset
- Java-based tool
- Support designers in early prototyping phases of automation interfaces

Autopilot ADEPT model I

The screenshot displays the Eclipse IDE interface for the ADEPT model. The main window is titled "User Interface Editor" and shows a cockpit simulation with various instruments and controls. The left pane is the "System Browser" showing a tree view of the model's components, including "lateralSystemTable". The bottom pane is the "Navigator" showing a file tree for the project "777".

The "User Interface Editor" window shows a cockpit simulation with the following elements:

- Top Panel:** Displays various flight parameters and controls, including "IAS" (250), "HDG" (180), "V/S" (5000), and "ALTITUDE" (5000). It also includes buttons for "AT", "FLCH", "AIR DEGRADE", "HOLD", "VSPFA", and "HOLD".
- Center Panel:** Features a central instrument display showing a heading scale from 140 to 360 degrees, with a current heading of 180 degrees. Below the heading scale is a vertical scale for altitude, ranging from 4800 to 5200 feet, with a current altitude of 5000 feet.
- Right Panel:** Contains a "Simulation Control Panel" with buttons for "FLY", "CLB Init", and "CRZ Init".

The "System Browser" window shows a tree view of the model's components, including:

- Root
- topLogicTable
- TopContainer
- aircraftAutomationObjects
- Lateral
- lateral360CorrectionS
- lateralFeedbackTable
- lateralHdgTrkUnitsSys
- lateralInterf...e.outputState
- lateralNavigationSyste
- lateralSystemTable
- lateralTargetFeedback
- lateralTargetSystemTz
- acBankAngle
- currentLateralMode
- hdgTrkUnits
- lateralAutoflightIntent
- lateralAutoflightRequ
- lateralDirection
- lateralFlightplanLegM
- lateralFlightplanLegTy
- lateralFlightplanNavig
- lateralFlightplanTarge
- lateralInterf...e.outputState

The "Navigator" window shows a file tree for the project "777", including:

- 777
- settings
- bin
- dot
- images
- jar
- sgb

Autopilot ADEPT model II

	0	1
airspeedFeedbackTable		
INPUTS		
<input type="checkbox"/> airspeedSystemTable.outputState		
Maintain Airspeed Target	•	
Capture Airspeed Target	•	
Hold Current Airspeed	•	
Protect Airspeed Target		•
OUTPUTS		
<input type="checkbox"/> pfdAirspeedTape.currentValue		
<input checked="" type="checkbox"/> indicatedAirspeed	•	•
<input type="checkbox"/> cautionLabel.background		
255, 204, 0		•
<input type="checkbox"/> autothrottleModeFailureBar.opaque		
False	•	
True		
<input type="checkbox"/> pitchModeFailureBar.opaque		
False	•	
True		
<input type="checkbox"/> pfdAirspeedTape.preSelectedTarget		
<input checked="" type="checkbox"/> selectedSpeed Target		•
<input type="checkbox"/> pfdAirspeedTape.selectedTarget		
<input checked="" type="checkbox"/> selectedSpeed Target		•

State Event Models

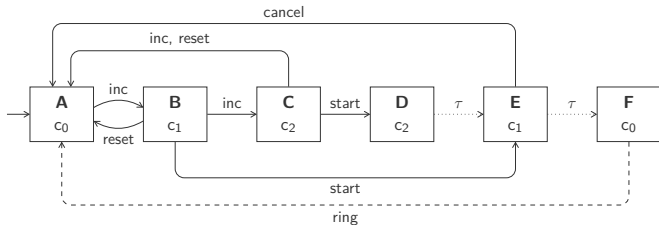
- **ADEPT models** combine state with transition information
- A **state** is made of n variables x_i ranging over domains D_i
- Only some state-variable are **visible**



HMI-LTS are enriched with state-values

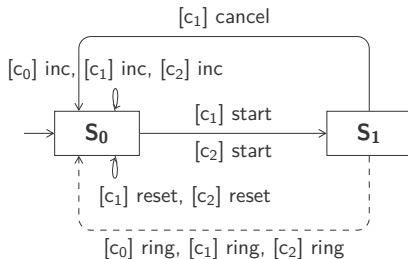
HMI State-Valued System Model

- Each state s is associated with a **state-value** $\mathcal{O}(s)$
- **Two kinds of observations** are possible in a system



HMI State-Valued Mental Model

- Transition are **guarded** with a state-value
- A transition will be executed if the guard is satisfied in the current state of the system



Enriched models to HMI-LTS

- System model



- Mental model



- The transformation preserves the developed algorithms

Conclusion

- An enriched model for system and mental model
- Translation from ADEPT models (to be automated)
- Reverse translation from HMI-LTS to ADEPT to be done