Formal Analysis of Human-Automation Interaction

Sébastien Combéfis

Université catholique de Louvain, Belgium
Computer Science and Engineering Department
Sebastien.Combefis@uclouvain.be

Supervisor(s): Pr. Charles Pecheur, Université catholique de Louvain, Belgium
Keywords: formal methods, model-checking, human computer interaction, system design, labelled transition system, equivalence relation

Abstract. Human-automation interaction analysis has extensively been studied by researchers in psychology, human factors, ergonomics and systems safety. Recently researchers began using formal methods to model and analyze such interactions. Formal methods with its solid basis in mathematics can bring a lot to this field. This work aims at leveraging and adopting formal verification techniques such as model-checking to the analysis and design of complex systems, where the interaction between the system and human plays an important role.

1 Research Area – Main Themes

There are more and more large and complex systems involving both humans and machines interacting together. System failures occur due to a bad design of the machine or due to the human incorrectly operating the machine. But several system failures have happened due to an inappropriate interaction between the operator and the machine. A well-known class of problems is known as *automation surprises*, that occur when the system behaves differently than its operator expects. For example, the user may not be able to drive the system in the operating mode he wants or he may not know enough of the machine's current state to properly determine or control its future behaviour. A concrete example is a cruise-control system. The user must be able to predict if the system is active or not and how it will evolve in response to an action like pressing the gas pedal or braking. Automation surprise can lead to *mode confusion* [6, 10] and sometimes to critical failure, as testified by real accidents [7, 8, 2].

Analysis of human-automation interaction is a field that has extensively been studied by researchers in psychology, human factors and ergonomics. But by the mid 1980s, researchers began using formal methods to analyze theses interactions. This work wants to study formally human-automation interaction. Formal methods can bring a new and challenging way to analyze and reason about such interactions which can help systems designers for the analysis and design of complex systems involving interactions with human.

Different problems might be asked in the analysis of human-machine interaction. The first kind of problems is linked to *verification* of some properties on the interaction, such as : *"May a system exhibit potential mode confusion for its operator ?"* or *"No matter in which state the machine is, can the operator always drive the machine into some recover state ?"*. See for example Rushby [9] or Campos et al. [1] who have dealt with this kind of problems using model-checking [5]. Another kind of problems is linked to *generation* of some elements that help in a correct interaction, such as procedures and recovery sequences [4] or user interface [2, 3].

2 Directions of the work

This work is currently in its very beginning and the initial phase deals with the problem of automatic user interface generation for mode monitoring [3]. The machine is simply viewed as a set of discrete states among which transitions can happen resulting in a change of the machine's state. Moreover, the user may want to distinguish some operating modes — user-relevant sets of states — which are called *mode* [6]. For example, the autopilot of a plane can be set in altitude hold or vertical climb mode. The machine is modelled as a labelled transition system whose states are partitioned according to the modes. The user wants to operate the machine and be able to monitor its modes, that is always being aware of the mode in which the machine currently is, and the mode the machine will transition into in response to a user action on the machine.

The easiest way for the user to monitor the modes of the machine is to know exactly the model of the machine, assuming he is able to observe all the transitions that can occur in the system. But this model can be too large for a human to remember it and more generally, there are transitions that the user cannot control nor observe. The problem addressed in the first part of this work is how to find a reduced version of the machine model, so that if the user knows exactly this reduced model, he can operate the machine and monitor its modes. This problem can be seen as a problem of partitioning the states of the machine with respect to an equivalence relation. That is the direction currently followed in this work.

3 Results

Some first results have been obtained in the formalization of the problem. This section presents these results and then discusses some interesting issues. As already mentioned, the machine is modelled as a labelled transition system whose states are partitioned according to the modes of the machine. The transitions of the machine can be classified into three categories :

1. **External actions** are controlled by the user ;
2. **Internal actions** are triggered by the machine's internal dynamics or by the environment and are observed by the user ;

3. **Hidden actions** are not controlled nor observed by the user.

Following control theory, external actions are controllable and observable actions, internal actions are uncontrollable but observable actions and hidden actions are uncontrollable and unobservable actions.

Let's take as an example a simple digital alarm-clock just showing the hour and minute. Figure 1 shows the machine model of this simple digital alarm-clock. The machine has four states and seven transitions. External actions are showed as plain arrows, internal actions as dashed arrows and the τ actions are the hidden ones.

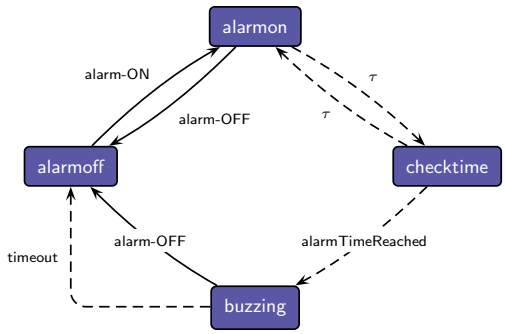

Fig. 1. Machine model of a simple digital alarm-clock. There are two external actions (alarm-ON and alarm-OFF), two internal actions (alarmTimeReached and timeout) and hidden actions. The initial state is alarmoff.

The alarm-ON and alarm-OFF actions are external, the user triggers them with buttons on the machine. The alarmTimeReached and timeout actions are internal. The former is triggered when the alarm time is reached and can be observed by the user through the buzzer sound. The latter is triggered when the alarm-clock is buzzing and a certain time elapsed. The user observes it by noticing that the buzzer stops. Then, there are two hidden transitions to enter and leave the checktime state in which the current time is compared to the alarm-time. The user cannot control nor observe these actions.

Let's transition to the vehicle transmission system example from Degani [3]. Figure 2 shows the machine model of this system. It has 8 states partitioned into 3 modes and 20 transitions. The user can trigger the push-up and pull-down actions by using the gear lever while the up and down actions are internal actions occurring within the system based on throttle, engine and speed values. They are just observable by the user. Note that this transmission system is quite tricky, when the system is in the LOW mode and the user push-up the gear lever, the system can either transition into the MEDIUM mode or in the HIGH mode regarding it is in the low-1 or low-2 state or in the low-3 state.

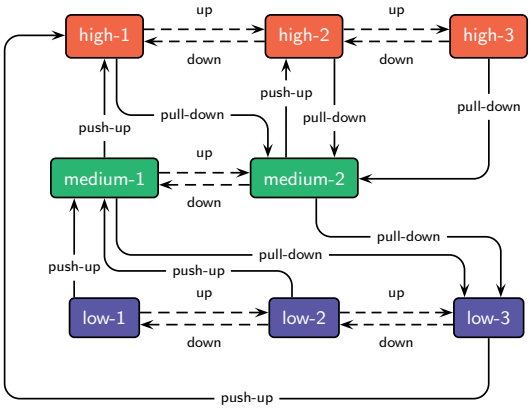

Fig. 2. Machine model of the vehicle transmission system. The states are partitioned into three modes (LOW, MEDIUM and HIGH levels). There are two external actions (push-up and pull-down) and two internal actions (up and down). The initial state of the machine is low-1.

What we are interested in is to find a reduced version of this model such that the user can monitor the three modes. A first naive try consists in the most reduced model with only three states, one for each mode as showed on figure 3(b). This reduced model is not correct. Indeed the user cannot properly anticipate the mode of the machine when the reduced model is in the low state because of a non-determinism on an external action (push-up).

Thus, a correct reduced model must distinguish the states low-1 and low-2 from the state low-3 of the machine. Figure 3(b) shows the most reduced model that is deterministic on external transitions. But once again, this reduced model is not suitable. This time, the issue is a little more subtle. There is a non-determinism with the internal action up on the low-B state but it does not matter since no matter which of the two up actions is followed, the machine stays in the same mode.

An issue may arise when considering a sequence of actions and this issue can be highlighted by running the reduced model in parallel with the machine model. Here is a resulting sequence of states from the machine and the reduced model in response to the sequence of action $\langle$ up, push-up $\rangle$:

$$(\text{low-1}, \text{low-B}) \xrightarrow{\text{up}} (\text{low-2}, \text{low-C}) \xrightarrow{\text{push}-\text{up}} (\text{medium-1}, \text{high})$$

The last couple is not acceptable, that is clearly a mode confusion problem since the machine model and the reduced model are not in the same mode. In fact to obtain the smaller reduced model that allows mode monitoring, the three states of the LOW mode must be separated.

In fact, whether the reduced model of figure 3(b) can be acceptable or not depends on what is considered as an acceptable model. Whenever the reduced

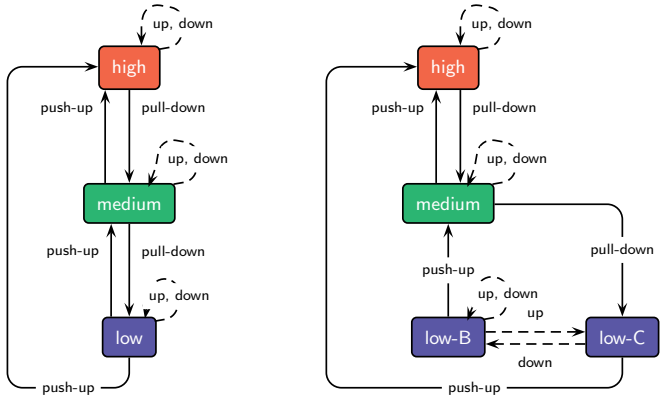

Fig. 3. Two possible reduced version of the machine model of figure 2. (a) Most reduced model whose states are simply the three modes. (b) Most reduced model deterministic on external transitions.

model is in the low-B state and an up action occurs, the user observes it and he can predict the next mode. Furthermore, he can predict the effect of pushing-up the gear lever, but he has to check first on the user interface in which state of the reduced model he is. There is thus a difference between being able just to predict the next mode in response to a punctual action or in response to a sequence of actions.

What is currently done in this work is to define some equivalence relations on the state of the machine model in order to get reduced models. These different equivalences have to be compared with respect to properties on the reduced model and the user interface linked to it and algorithms to compute them.

Acknowledgements This work is supported by project MoVES under the Interuniversity Attraction Poles Programme — Belgian State — Belgian Science Policy.

References

1. José Creissac Campos, Michael D. Harrison, and Karsten Loer. Verifying user interface behaviour with model checking. In *Proceedings of the 2nd International Workshop on Verification and Validation of Enterprise Information Systems*, pages 87–96, 2004.
2. Asaf Degani. *Taming HAL: Designing Interfaces Beyond 2001.* Palgrave Macmillan, January 2004.
3. Michael Heymann and Asaf Degani. Formal analysis and automatic generation of user interfaces: Approach, methodology, and an algorithm. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(2):311–330, April 2007.

4. Michael Heymann, Asaf Degani, and Immanuel Barshi. Generating procedures and recovery sequences : A formal approach. In *Proceedings of the 14th International Symposium on Aviation Psychology*, 2007.
5. Edmund M. Clarke Jr, Orna Grumberg, and Doron A. Peled. *Model checking*. The MIT Press, January 1999.
6. Nancy G. Leveson, L. Denise Pinnel, Sean David Sandys, Shuichi Koga, and Jon Damon Reese. Analyzing software specifications for mode confusion potential. In *Workshop on Human Error and System Development*, pages 132–146, 1997.
7. Nancy G. Leveson and Clark Savage Turner. Investigation of the therac-25 accidents. *IEEE Computer*, 26(7):18–41, July 1993.
8. Everett Palmer. Oops, it didn't arm. - a case study of two automation surprises. In *Proceedings of the 8th International Symposium on Aviation Psychology*, pages 227–232, 1996.
9. John Rushby. Using model checking to help discover mode confusions and other automation surprises. *Reliability Engineering and System Safety*, 75(2):167–177, February 2002.
10. Nadine B. Starter and Dadiv D. Woods. How in the world did we ever get into that mode ? Mode error and awareness in supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):5–19, March 1995.