International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Analysis Model of Identifying Erroneous Human Behaviour in Coma Patient Recognition System

Manisha Umak[1], U. A. Jogalekar[2]

[1]Smt. Kashibai Navale College of Engineering, Vadgaon(BK), Pune, Maharashtra, India

[2]Professor, Smt. Kashibai Navale College of Engineering, Vadgaon(BK), Pune, Maharashtra, India

Abstract: ***Complex safety critical applications like Automated Medication systems for Coma patients involve both human operators and automated devices. In this environment, human operator can make some error in operation of systems which will affect the safety of patients. In this project, we propose a verification mechanism which automatically generated the erroneous behaviors of humans and test it on the formal models of the system to evaluate the security. With this verification system, human errors can be identified and system can be made more robust to failures.***

Keywords: Model Checker, Task analysis, System Safety, Human Automation Interaction(HAI), Coma, Coma Patient Monitoring, EOFM (Enhanced operator Function Model).

1. Introduction

Many of the interactive systems are safety critical in the sense that action which user performed may have consequences that will compromise safety (i.e. cause damage). An important aspect of the engineering that interactive computer systems is to provide appropriate processes and proof that systems are designed to satisfy the various types of requirements that have been established to reduce the risk of products causing such harm.

This paper addresses the engineering problem which describes a formal technique that supports proof that user interface related safety requirements are fully satisfied in a specified interactive systems design. The medical domain is chosen to demonstrate and evaluate the approach. In many countries, medical equipment undergoes a degree of scrutiny prior to being placed on the market. This scrutiny is required by regulators to provide confidence that the device is safe and fit for purpose.

Wherever the human operator doesn't follow the normative procedures for interacting with a system, is commonly related to failures in Human Automation interaction surroundings. Human operator will create following mistakes.

- Omission of activities.
- Erroneous repetition of activity.
- Erroneous execution of activity.

Human–automation interaction (HAI) is particularly important to the operation of safety-critical systems. Erroneous human behaviour, where the human operator does not follow the normative procedures for interacting with a system, is often associated with failures. Formal methods are a set of languages and techniques for the modelling, specification, and verification of systems. A formal model describes a system as a set of variables and transitions between variable values (states). Task Analytic behaviour models (products of a cognitive task analysis) can be included in the formal system model and formal verification can be used to evaluate the impact of the modelled behaviour (which can be normative or erroneous) on system safety.

We will model the Coma Patient Medication Automation system as input for the project work. In the project base paper, the medication automation for general patients is given; we will customize this model for Coma patient.

2. Literature Survey

2.1 Formal Verification to Evaluate Human Automation Interaction

Failures in complicated systems controlled by human operators can be hard to anticipate because of unexpected interactions between the elements that compose the system, including human-automation interaction (HAI). HAI analysis would benefit from various number of techniques that support investigating the possible combinations of systems conditions and HAIs that might result in failures. Formal verification is a powerful technique used to mathematically prove that an appropriately scaled model of a system does or does not exhibit desirable properties. This paper discuss how formal verification has been used to evaluate HAI.

It has been used to evaluate human-automation interfaces for usability properties and to find potential mode confusion. It has also been used to evaluate system safety properties in light of formally modeled task analytic human behavior. While capable of providing insights into problems associated with HAI, formal verification does not scale as well as other techniques such as simulation. However, advances in formal verification continue to address this problem, and approaches that allow it to complement more traditional analysis methods can potentially avoid this limitation.

2.2 A Model for Types and Levels of Automation

Technical developments in computer hardware and software now make it possible to introduce automation into virtually all aspects of human-machine systems. Given these technical capabilities, which system functions should be automated and to what extent? We outline a model for types and levels of automation that provides framework and an objective basis

for making such choices. Appropriate selection is important because automation does not merely supplant but changes human activity and can impose new coordination demands on the human operator. We propose that automation can be applied to four broad classes of functions:

1) information acquisition; 2) information analysis; 3) decision and action selection; and 4) action implementation. Within each of these types, automation can be applied across a continuum of levels from low to high, i.e., from fully manual to fully automatic. A particular system can involve automation of all four types at different levels. The human performance consequences of particular types and levels of automation constitute primary evaluative criteria for automation design using our model.

2.3 Using Analytic Models to Visualize Model Checker Counterexamples

Model checking is a type of automated formal verification that searches a system model's entire state space in order to mathematically prove that the system does or does not meet desired properties. An output of most model checkers is a counterexample: an execution traces illustrating exactly how a specification was violated. In most analysis environments, this output is a list of the model variables and their values at each step in the execution trace.

They have developed a language for modeling human task behavior and an automated method which translates instantiated models into a formal system model implemented in the language of the Symbolic Analysis Laboratory (SAL). This allows us to use model checking formal verification to evaluate human-automation interaction. In this paper present an operational concept and design showing how our task modeling visual notation and system modeling architecture can be exploited to visualize counterexamples produced by SAL.

2.3.1 Enhanced Operator Function Model

There are several task analytic modeling paradigms such as Operator Function Model, Concur Task Trees (CTTs), hazard networks, User Action Notation (UAN), and several varieties of Goals, Operators, Methods, and Selection rules (GOMS). Collectively these techniques encompass the following features: Activities can be modeled as sequences of actions or as part of more complex hierarchies of activities which can ultimately decompose into atomic actions: Observable human actions (such as pushing a button, turning a wheel, or flipping a switch) are supported by all techniques but some also support human cognitive or perceptual actions (such as remembering a number or noticing an alarm).

2.4 Architecture and Development Environment of a Knowledge-Based Monitor that Facilitate Incremental Knowledge-Based Development

Being able to incrementally define and test knowledge bases for intelligent systems is desirable. However, as more knowledge is added, the knowledge engineer must ensure that unwanted interactions between the existing and additional knowledge do not occur. One knowledge-based monitoring system, Hazard Monitor (HM), provides the ability to add knowledge incrementally.

HM's architecture includes tailorable components that allow the knowledge engineer to eliminate unwanted knowledge interactions. HM also includes knowledge-base development tools to facilitate initial and incremental knowledge-base development. This paper describes HM's architecture and knowledge structures plus its knowledge-base development tools that facilitate the knowledge engineering process.

2.5 Toward a Multi-Method Approach to Formalizing Human-Automation Interaction and Human-Human

Breakdowns in complex systems often occur as a result of system elements interacting in ways unanticipated by analysts or designers. The use of task behaviour as part of a larger, formal system model is potentially useful for analyzing such problems because it allows the ramifications of different human behaviors to be verified in relation to other aspects of the system. A component of task behaviour largely overlooked to date is the role of human-human interaction, particularly human-human communication in complex human-computer systems. We are developing a multi-method approach based on extending the Enhanced Operator Function Model language to address human agent communications (EOFMC). This approach includes analyses via theorem proving and future support for model checking linked through the EOFMC top level XML description.

2.6 Generating Phenotypical erroneous Human Behavior to Evaluate Human Automation Interaction Using Model Checking

Model-driven design and analysis techniques provide engineers with formal methods tools and techniques capable of evaluating how human behavior can contribute to system failures. This paper presents a novel method for automatically generating task analytic models encompassing both normative and erroneous human behavior from normative task models. The generated erroneous behavior is capable of replicating Hollnagel's zero-order phenotypes of erroneous action for omissions, jumps, repetitions, and intrusions. Multiple Phenotypical acts can occur in sequence, thus allowing for the generation of higher order phenotypes.

The task behavior model pattern capable of generating erroneous behavior can be integrated into a formal system model so that system safety properties can be formally verified with a model checker. This allows analysts to prove that a human-automation interactive system (as represented by the model) will or will not satisfy safety properties with both normative and generated erroneous human behavior. We present benchmarks related to the size of the state space and verification time of models to show how the erroneous human behavior generation process scales.

3. Proposed Work

Erroneous human behavior and avoid that transaction. It also replaces that value with its proper value but yet it does not

provide the proper solution that will solve in proposed system with some modification. We are going to apply this proposed solution on coma medication system data. Existing system drawback that will analyze in given proposed system:

a) **Scalability:** A significant increase within the state space size and verification times was discovered between the normative human behavior models. Despite this advantage, enhancements in measurability would still increase the pertinence of the tactic.
b) **Does not identify higher order failure:** It does not identify complicated erroneous human behavior. In this paper defined the modification that will help to identify complicated human erroneous activity.
c) **Does not Provide deadlock detection system:** Exiting does not find out the deadlock in the system state that will also cover in the presented proposed system However, our new method is capable of generating these types of higher order failures without considering all of the complex combinations of extraneous actions that would be required to generate similarly ordered erroneous behaviors using the technique from. While the method presented in this paper could be used to explore.

Figure1 shows the overview for the functions defined in coma patient recognition system. In fig.1, input is passing the EOFM (Enhanced operator function model) language used for model the operator as input/output system. Inputs may come from several sources including: the human device interface, mission goals, environment, and other human operators. Output variables are human actions. The system consists of 3 modules

➢ FSM Parser: This module will create the FSM (Finite State Model) model from the EOFM language specification which consists of state and transition.

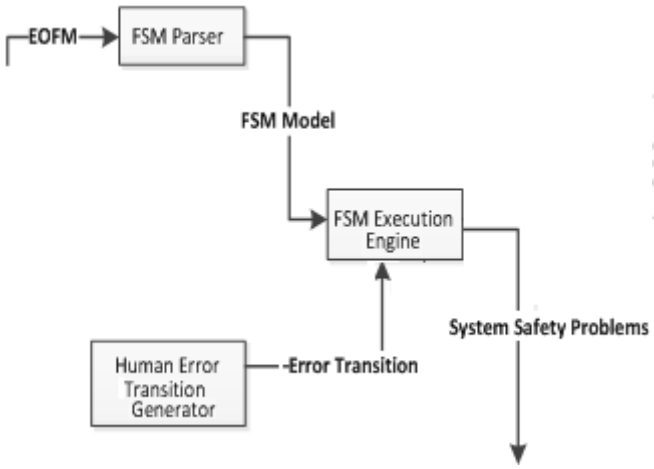

Figure 1: System Architecture

- FSM Execution Engine: This module will execute the FSM state machine and generate all possible output states and also summarizes the number of safe state and number of unsafe states. higher order erroneous behaviors based on intentional failures.
- Human Error Transition Generator: This module will generate the different combination of erroneous human behaviour.
- Final outcome shows the system is in safe state or not and if system is not in safe state then we translate that value and convert unsafe state into the safe state.

4. Result Discussion

The model implementation perform to determine if the error generation process as implemented would generate the desired erroneous transition of activity execution state and evaluate how the error generation process impacted model complexity. For the model analysis, a simple instantiated EOFM was constructed in which single activity (aParent) which decomposes into a single action (a) with an ordered (ord) decomposition operator. This translated EOFM interacts with simple human operator actions.

The human operator also receives Boolean inputs (precondition, repeat condition, and completion condition) from the human device interface model which give (aParent) access to simulated pre, repeat and completion conditions that can change between true and if false at each step in a system model execution. Because model complexity should vary based on the maximum number of erroneous transitions (KMax), this was varied in these tests in order to obtain metrics of complexity (number of states) and verification time. The implementation provides the following outcome for the coma patient system:

- The erroneous states in the coma medication system.
- All possible human errors.
- The safeguard for the erroneous states.

5. Conclusion

In this paper, we have presented generating erroneous behavior in coma patient recognition system where formal model and model checker used to detect and evaluate erroneous type of behavior. This focuses on the EOFM model that helps to evaluate erroneous human behavior by using transition diagram and they provide reasonable confidence that the EOFM to SAL translation process is adhering to the formal semantics.

References

[1] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu, “Using formal verification to evaluate human–automation interaction: A review,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 43, no. 3, pp. 488–503, May 2013.
[2] R. Parasuraman, T. Sheridan, and C. Wickens, “A model for types and levels of human interaction with automation,” IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 30, no. 3, pp. 286–297, May 2000.
[3] M. L. Bolton and E. J. Bass, “Using task analytic models to visualize model checker counterexamples,” in Proc. IEEE Int. Conf. Syst. Man,Cybern., Oct. 2010, pp. 2069–2074.
[4] E. J. Bass, S. T. Ernst-Fortin, R. L. Small, and J. Hogans, “Architecture and development environment of a

knowledge-based monitor that facilitate incremental knowledge-base development," IEEE Trans. Syst.,Man, Cybern. A, Syst. Humans, vol. 34, no. 4, pp. 441–449, Jul. 2004.

[5] E. J. Bass, M. L. Bolton, K. Feigh, D. Griffith, E. Gunter, W. Mansky, and J. Rushby, "Toward a multi-method approach to formalizing human automation interaction and human–human communications," in *Proc*. IEEE Int. Conf. Syst., Man, Cybern., Oct. 2011, pp. 1817–1824.

[6] L. De Moura, S. Owre, and N. Shankar, "The SAL language manual," Computer Science Laboratory, SRI International, Menlo Park, Tech. Rep. CSL-01-01, 2003.

[7] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu, "Generating phenotypical erroneous human behavior to evaluate human–automation interaction using model checking," Int. J. Human-Comput. Stud., vol. 70, no. 11, pp. 888–906, 2012. 5, pp. 961–976, Sep. 2011