Extending the Boundaries of Model-Based Development to Account for Errors

Sandra Basnyat
LIIHS–IRIT, University Paul Sabatier
118 route de Narbonne, 31062
Toulouse Cedex 4
+33 561 55 74 04
basnyat@irit.fr

Rémi Bastide
LIIHS–IRIT, University Paul Sabatier
118 route de Narbonne, 31062
Toulouse Cedex 4
+33 561 55 74 05
bastide@irit.fr

Philippe Palanque
LIIHS–IRIT, University Paul Sabatier
118 route de Narbonne, 31062
Toulouse Cedex 4
+33 561 55 69 65
palanque@irit.fr

ABSTRACT

This paper presents an approach for relating informed task models and system models in the domain of safety critical interactive systems. The models, which are usually developed from an error-free perspective, have been informed to account for erroneous user behavior. We believe this perspective extends the general boundaries of model based development (MBD), by taking into account additional information relating to previous experiences of failure, for the design of safer safety-critical interactive systems. We illustrate our ideas using a fatal mining accident case study.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human factors, human information processing and software psychology.

General Terms

Design, Reliability, Security, Human Factors, Verification.

Keywords

Model-based development, safety-critical interactive systems, human errors, system errors, barriers.

1. INTRODUCTION

Model-based development (MBD) is a developing trend in the domain of software engineering [13] advocating the specification and design of software systems from declarative models [22]. It relies on the use of explicit models and provides the ability to represent and simulate diverse abstract views that together make up a 'system', without the need to fulfill its implementation. This can of course save time and money and more importantly, allows designers to identify problems early in the development process.

In the domain of Safety-Critical Interactive Systems (SCIS), extending the boundaries with respect to what is typically taken into account and represented in various design models we believe, can help us to design safer, SCIS. More specifically, additional data, such as human 'errors', human behavior, past experiences such as incidents and accidents, known protection systems such as human-related procedures and technically related barriers that can provide a very important source of data for improving interfaces used in SCIS.

Task analysis and modeling is a central element to user centered design approaches. For this reason a lot of work has been devoted to it and to its integration in the development process of interactive systems. A task model is a representation of user tasks often involving some form of interaction with a system influenced by its contextual environment. Users perform tasks, which are structured sets of activities in order to achieve higher-level goals. Tasks can be further decomposed corresponding to lower level sub goals. This notion of decomposition naturally results in tree-like structures and thus a hierarchical representation of the model.

Task modelling is usually performed with an error-free perspective. Errors are usually dealt with during the software development phase (when the system supporting the user tasks is developed) by systematically investigating possible erroneous user input. This results in an under-specification of system responses to erroneous user behaviour. However, human error plays a major role in the occurrence of accidents in safety-critical systems such as in aviation, railways systems, or nuclear power plants [17]. Unfortunately, practice shows that reaching the necessary exhaustivity in task modeling is very difficult in terms of availability of resources and economy. These aspects drive people responsible for task analysis and modeling to focus on most frequent and standard activities, thus leaving the infrequent or erroneous ones unconsidered. However, this is precisely where the emphasis should be placed in order to deal efficiently with error tolerance.

In the domain of SCIS, formal methods are widely used. Organizations are willing to pay the additional costs of formal methods in order to improve the global safety of their systems. Before a system is implemented, formal model-based approaches allow designers to apply techniques to validate their models and verify properties. Such techniques include model checking, formal verification and performance evaluation. However, only recently we have been witnessing the use of highly interactive user interfaces in this application domain. There is now a growing need to improve methodological support of formal techniques to account for errors and for instance to accommodate to for direct manipulation and multimodality.

A complementary way to enhance a system's safety is to take into account information from previous real life cases. One such usually available and particularly pertinent source is the outcome of an incident or accident investigation. Designs of any nature can be improved by taking into account previous experiences, both positive and negative. However, for a safety-critical system, previous incidents and accidents are interesting factors as they are clearly what we wish to avoid. In this paper we present an approach exploiting incident and accident investigation techniques to support the design of a safety-critical system.

The human operator of a SCIS will have safety-critical tasks that are crucial in maintaining system safety. The task of the operator is often hard to integrate in system design. A further way to increase safety in SCIS, is to consider barriers, both human-related and system-related and use them to inform models within MBD. We define barriers as the combination of technical, human and organizational measures that prevent or protect against an adverse effect. [26].

In this paper we briefly present two previous approaches that we have been working on to tackle the incorporation of error-related behavior and events in MBD to inform the task model and the system model. The first approach uses "task patterns" for taking into account in an efficient and systematic way both standard and erroneous user behaviour in task modelling. The second approach looks at incident and accident investigation techniques to inform a system model. Throughout the approaches, we use formal modelling techniques.

This defensive approach to MBD will minimise the adverse affects of erroneous events by designing the system to deal with such events. This could be by simply designing the system to return to a safe state after a problematic event has taken place.

We present our approaches on a given fatal mining accident case study. The case study is briefly presented in the following section to give the reader an understanding of the models presented in later sections. We provide an overview of our previous work on extending the boundaries of task modeling and system modeling, (sections 3 and 4 respectively). We discuss in section 5 ways of relating the task model and system model to ensure coherence between models for a more error-tolerant design. This is indeed the main focus of this paper. We conclude by discussing our further work including the desire to incorporate a different type of human-computer interaction error, namely, mode confusion for extending the boundaries in MBD to account for errors.

2. The Case Study

The case study we have chosen to illustrate our ideas is a fatal US mining accident [28]. We focus our attention on the waste fuel delivery system of its cement plant and the interactions between operators and the system. Due to space constraints, we cannot fully describe the system or the accident, however see [2] for an in depth discussion. It is not necessary to fully understand the accident in order to follow this paper.

3. Extending the Boundaries of Task Modeling

We have previously worked on extending the general approach to MBD by providing an approach for taking into account errors that can occur while interacting with a system. Our basic idea is to bring together, in a unifying framework, three essential principles of the User Centered Development (UCD) process for more reliable interactive software, namely task analysis and modeling, formal description of the system and human error categorization. We incorporate additional data; including information represented using incident and accident investigation techniques, and socio-technical barriers using the Safety Modeling Language (SML) [24].

In this section, we briefly present our approach for extending a single-user task model to account for errors.

As stated in the introduction, when modeling user behavior, an error-free perspective is usually employed. Task modeling as yet, does not allow for the description, representation and analysis of unexpected eventualities that may occur including human error. Attempting to do so usually results in combinatory explosion of the models.

We propose to use task patterns as a way of dealing exhaustively with potential user errors. The patterns of tasks have been modeled and can be directly reused within a task model in order to represent potential deviations of user behavior.

Since the task model influences the system design it is important to understand how to manage and overcome possible errors. Figure 1 presents a task model using the ConcurentTaskTrees notation for the worker of the waste fuel delivery system of the cement plant. Further models for the Kiln Control Operator, the Supervisor, and a task model representing cooperative tasks between users can be found in [3].

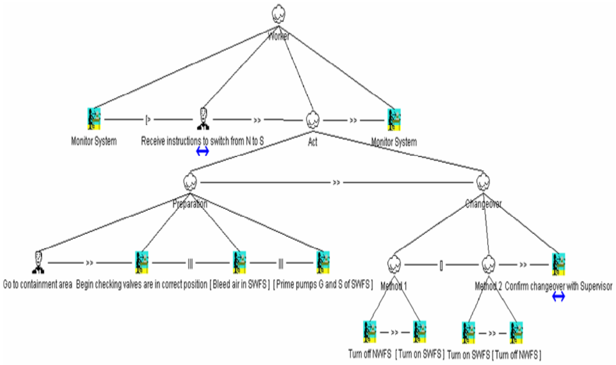

Figure 1 Cement Plant Worker Task Model

3.1.1 Task Patterns

Task patterns for interactive systems design is a relatively new concept with the aim of solving design problems using existing knowledge of previously identified patterns and solutions. Task patterns were first introduced by Paternò [21] as reusable structures for task models. The patterns were described as hierarchical structured task fragments that can be reused to successively build the task model.

We suggest that task patterns can be used as a means of incorporating erroneous behaviour into a standard task model as a method for making explicit such errors. This is fully explained in [17]. To summarise however, we analyse the subtasks of a task model, using an exhaustive classification of human errors (described in 3.1.2). We are then able to group task specific errors together according to the affect it would have on the system. Combinations of these generalised errors form small task models that can be 'plugged in' to replace the subtask.

3.1.2 Human Error

It has been claimed that up to 80% of all aviation accidents are attributed to human 'error' [12]. Although the term "human error" appears very controversial, theories of human errors such as Rasmussen's [23] SRK, Hollnagel's [10] Phenotypes and Genotypes and Norman's [16] classification of slips can be considered widely acceptable. Using the above mentioned classifications and based fundamentally on the SRK theory, we

Hazard Name	Brief Description
Water Hammer Effect	Caused by rebound waves created in a pipe full of liquid when a valve is closed too quickly.
PLC Not Connected	The new PLC was not connected to the F-System.
Valves Bled While System in Operation	The valves should not be bled while the system is in operation
Motors Running Dry	Occurs when motors are turned on but fuel is not in pipes
Fuel Spreading	Occurs when the valves are bled while motors and fuel are running
Gathering Air	Air can gather in a pump if the valve was not bled before start-up of the system.

Table 1. Summary of Hazardous States Identified

have produced Human Error Reference Tables (HERTs) for analysing potential human error in task models.

The benefit of producing such reference tables enables the exact identification of very precise types of 'error' while analyzing human behavior associated to particular subtasks of a task model. Thus for each subtask of a task model, it is possible to determine, by means of elimination, which human errors are applicable and what the impact of the 'error' will be on the task in hand. For an in-depth discussion of the applying the HERTs to a sub-task, see [17]. The aim of these reference tables is not to guarantee a comprehensive and exhaustive identification of every possible eventuality but to provide investigators with systematic ways of exploring likely and reoccurring user deviations.

4. Extending the Boundaries of System Modeling

To model systems for this research, we will be using the PetShop [15] environment and the Interactive Cooperative Objects (ICO) [4] language. PetShop is dedicated to editing, simulating and analysing ICOs specifications. ICOs are an object-oriented, Petri net-based formalism dedicated to the modelling of interactive systems. This language is dedicated to the construction of highly interactive distributed applications. It is to be used by expert programmers skilled in formal description techniques, object-oriented approaches, distributed and interactive systems [18].

The WFDS has been modelled using the ICO notation to deliver a relatively high level of detail of the behaviour of the system. We have modelled each individual component of the plant as a Petri net i.e the pumps, the grinders, the kilns etc and have connected these Petri nets according the fuel flow within the plant to make explicit the description of all conditions of the components.

4.1 Incident & Accident Investigation Techniques

The system model has been extended with respect to what is typically taken into account, to include hazardous events. To do this, we applied safety cases to the accident report. Safety cases comprise of safety requirements and objectives, the safety argument, and the supporting evidence. Goal Structuring Notation (GSN) is a graphical argumentation notation, which represents the components of a safety argument, and their relationships. When applied to an accident the way proposed [2] we are able to identify not only technical and human failures, but understand more about the deeper problems that led to these failures taking place, organizational and so on.

We promote the use of system modeling to prove that a given incident or accident cannot be triggered in the new system design. We are not claiming that the mishap will not recur in the real system, but simply in the model describing the behavior of the system. This model is an abstraction of the real system assuming that system engineers will use that model for designing the improved system.

5. Relating task model and system model extensions to account for error

In this section, we discuss ways of relating the two previous studied techniques for accounting for error in the task model and system model.

When analyzing the 'bleed air in SWFS' optional subtask of the worker task model illustrated in Figure 1 using the HERTs, we exposed exhaustive lists of possible erroneous events. It was therefore often necessary to continue the analysis by grouping potential interaction problems into more generic types. In our case study, the analysis identified six different forms of problem. These included difficulties in the bleed task, a failure to begin bleeding the system, a failure to completely bleed the system, a failure to close the bleeding valve, bleeding the system too early and bleeding the system too late. Of these, a failure to completely bleed the system seems the most likely to have led to the water hammer effect observed in the immediate interval before the accident. The fourth of these task problems could also lead to a hazard with fuel leaking on the operator. The six identified erroneous events can be combined using various CTT temporal operators, resulting in small task patterns. The erroneous task patterns can be 'plugged in' to the existing task model to replace the analyzed sub-task.

These task models (including representations of possible user errors) can then be tested over a system model in order to verify that the system under development is able to tolerate such erroneous user behavior.

Table 1 highlights the hazardous states that were identified following the safety case analysis and hazards that were identified from the accident report itself.

The results of the safety case analysis highlight complementary findings to those found in the task model analysis. We again found issues relating to the use of valves, i.e., bleeding valves while the pumps were in operation.

We can relate the extended task model (using the results of the HERTs analysis and corresponding task patterns) and extended system model (using the results of the safety case analysis) by means of system model informing. The identified system-related hazards and human-related erroneous behavior can be modeled as transitions and places in the Petri net system model. It must be noted that not all hazardous events can be modeled in the system model. Hazards relating to lack of training, lack of supervision etc must be modeled elsewhere. Figure 2 illustrates part of the North Pump-G Valve component of the complete informed system model of the cement plant. We cannot show the complete pump component or the complete system model due to space constraints. See [2] for further illustrations.

The model in Figure 2 includes two hazards, 1) when the North Pump-G is spreading fuel (due to its valve being open, while the fuel is flowing and when there is no longer air in the pipes), and 2) when the North Pump-G is gathering air.

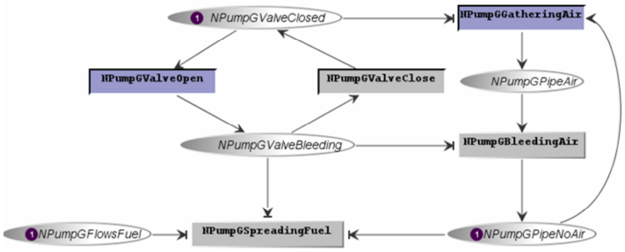

Figure 2. North Pump-G Valve Including Erroneous Events

The example shows the modeling of erroneous human-system interaction behavior (bleeding pipes while pumps in operation) and of a system-related erroneous event (pipes gathering air).

5.1 Scenario to Support New System Validation

We can apply scenarios to identify whether our modeled system is able to tolerate erroneous events, particularly the events that led to the accident being analyzed, and ultimately, to design the system so that it can return to a stable state if an error has occurred. There are several sources of scenarios. Scenarios can be extracted using the extended task model. CTTe facilitates this process as the tool can be used to simulate different scenarios. We can also extract scenarios from the accident report, using techniques such as Events and Causal Factors Analysis (ECFA). On the system side, we can simulate all possible scenarios modeled in the Petri net. The marking tree technique allows us to view all possible paths of interaction. The marking tree of a Petri net provided with an initial marking, explicitly details the set of reachable states from this initial marking, as well as the sequences of transitions needed to reach those states.

By applying such development techniques, the validation of the safety-critical system appears earlier in the design process than in more classic techniques. A formal validation occurs through the use of verification techniques and analysis tools. This validation is possible due to the algebraic foundation of the Petri nets.

6. CONCLUSIONS

This paper has provided the foundations for developing a framework for extending the boundaries of MBD with respect to what is generally accounted for in models. Taking into account human-related and system-related errors early in the development cycle is particularly important in the design of safety-critical interactive systems. Our approaches have focused on the task model and system model due to their importance in User Centered Design approaches. For instance, task analysis and modelling provide support for capturing and representing user activities [8] while interacting with a system, while the system model allows us to simulate the impact that user interactions will have on the system, in particular, erroneous interactions.

The aims of using the two MBAs presented are twofold, to improve the quality of SCIS and enable reuse. As described, these aims are achieved by accounting for error in the design process by analyzing subtasks of a task model to determine potential paths of deviation, by using incident and accident reports to determine potential human-computer interaction and system hazards, and by developing error-tolerant task patterns applicable to other designs. It is not yet clear however, if the approach will accelerate the development process or reduce costs, due to the multidisciplinary nature of the approach. The process of maintaining information with respect to errors identified from the task models and incident and accident investigation, when informing the system model is systematic. The errors (human and system related) described using natural language are listed in a table. They are then transformed into Petri net format, states (places) and events (transitions) in order to map the errors onto the system model. For further information, see [3].

6.1 Further Work

A further way to protect against accidents and erroneous events is to implement barriers. These can be human-related or system-related barriers. We are currently working on this idea. See [25] for the use of barriers to inform model based system design.

Although we have applied our approaches to a mining case study, our work targets interactive safety critical systems, such as the new generation of interactive cockpits compliant with ARINC 661 [1] specification, that we have previously worked on [14] and made available in Airbus A380 or Boeing 787. Indeed, for such applications the system model represents the *actual* behavior of the interactive application.

To date, we have considered human 'errors', system hazards and socio-technical barriers to inform the task and system model. Mode confusion or "automation surprise" [19] are further pertinent sources of accidents, particularly in automated aircraft systems. In [6], 184 aircraft incidents and accidents involving mode awareness are listed. In the complexity of modern, computerized systems, the current system state can come as an unpleasant shock to even an experienced user [11]. This relates to non-deterministic interfaces, which are those where 'the same user action can lead to more than one outcome'[7]. We would like to explore this type of 'error' and integrate them into our approaches and eventually a framework.

Further more, the Human-Factors Analysis and Classification System (HFACS) tool developed by [27] aims at answering the question "Why do aircraft crash?". They use a classification system, based on four levels of human failure that are broken down, somewhat like our HERTs. We would like to incorporate the HFACS tool (targeted specifically at aircraft disasters) with HERTs (a more general error classification system) to perform an in-depth error analysis of an automated system accident. The results can then be used to inform a system model represented using Petri nets.

7. ACKNOWLEDGMENTS

This work was supported by the EU funded ADVISES Research Training Network, GR/N006R02527

http://www.cs.york.ac.uk/hci/ADVISES/.

8. REFERENCES

[1] ARINC 661 Cockpit Display System Interfaces to User Systems. Arinc Specification 661. April 22, 2002. Prepared by Airlines Electronic Engineering Committee.

[2] Basnyat, S., Chozos, N., Johnson, C., Palanque, P. (2005) Incident and Accident Investigation Techniques to Inform Model-Based Design of Safety-Critical Interactive Systems. 12th International workshop on Design, Specification and Verification of Interactive Systems. 13-15 July 2005. Newcastle upon Tyne, England.

[3] Basnyat, S., Chozos, N., Johnson, C., Palanque, P. (2005) Redesigning an Interactive Safety-Critical System to Prevent an Accident from Reoccurring. 24th European Annual Conference on Human Decision Making and Manual Control. (EAM) Organised by the Institute of Communication and Computer Systems. 17-19 October 2005. Athens, Greece.

[4] Bastide, Rémi; Sy, Ousmane; Palanque, Philippe, and Navarre, David. Formal specification of CORBA services: experience and lessons learned**.** ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'2000). ACM Press; 2000.

[5] Breedvelt, I., Paternò, F., Sereriins, C. Reusable Structures in Task Models, Proceedings Design, Specification, Verification of Interactive Systems. Springer Verlag, pp.251-265 (1997).

[6] Dan Hughes and Michael Dornheim, Automated Cockpits: Who's in Charge?, Aviation Week & Space Technology, January 30-February 6, 1995

[7] Degani, A. (2004) Taming HAL: Designing user interfaces beyond 2001 Palgrave Macmillan, 2004. 312 pages. ISBN: 031229574X

[8] Diaper, D., Stanton, N. The Handbook of Task Analysis for Human-Computer Interaction. Lawrence Erlbaum Associates. (2003)

[9] Fields, B., Paternò, F., Santoro, C (1999). Analysing User Deviations in Interactive Safety-Critical Applications, Proc. of DSV-IS '99, pp. 189-204, Springer-Verlag

[10] Hollnagel, E., The Phenotype of Erroneous Actions: Implications for HCI Design. In: Weir, G.R.S. and Alty, J.L., (Eds.), Human-Computer Interaction and Complex Systems, Academic Press. (1991)

[11] Hourizi, R., and Johnson, P. (2001) Unmasking Mode Error: A New Application of Task Knowledge Principles to The Knowledge Gaps in Cockpit Design. Interact'01, The EighthIFIP Conference on Human Computer Interaction, Tokyo, July 9-14th.

[12] Johnson C.W. Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting. University of Glasgow Press, Glasgow, Scotland, October 2003. ISBN 0-85261-784-4. (2003)

[13] MDA Guide version 1.0.1, OMG Document number omg/2003-06-01, http://www.omg.org/cgi-bin/doc?omg/03-06-01

[14] Navarre, D., Palanque, P. and Bastide, R., 2004, A Formal Description Technique for the Behavioural Description of Interactive Applications Compliant with ARINC 661 Specification. HCI-Aero'04 Toulouse, France, 29 September-1st October 2004

[15] Navarre, D., Palanque, P., Bastide, R., 2003, A Tool-Supported Design Framework for Safety Critical Interactive Systems in Interacting with computers, Elsevier, Vol. 15/3, pp 309-328. 2003

[16] Norman D.A. The design of everyday things. New York: Currency-Doubleday, 1988.

[17] Palanque, P., Basnyat, S. (2004) Task Patterns for Taking into Account in an Efficient and Systematic Way Both Standard and Abnormal User Behaviour. IFIP 13.5 Working Conference on Human Error, Safety and Systems Development Toulouse (HESSD) Toulouse, France, August 22-27, 2004.

[18] Palanque, P and Bastide, R. UAHCI 2003. User-Centered Point of View to End-User Development. Universal Access for Human-Computer Interaction Heracklion, Crete, June 2003.

[19] Palmer, E, 1995, "Oops It Didn't Arm", A Case Study Of Two Automation Surprises, Proceedings Of The Eighth International Symposium On Aviation Psychology

[20] Paternò F. and Santoro C. (2002). Preventing user errors by systematic analysis of deviations from the system task model. International Journal Human-Computer Studies, Elsevier Science, Vol.56, N.2, pp. pp. 225-245, 2002.

[21] Paternò, F. Model Based Design and Evaluation of Interactive Applications. Springer Verlag, Berlin (1999)

[22] Puerta, A.R. Supporting User-Centered Design of Adaptive User Interfaces Via Interface Models. First Annual Workshop On Real-Time Intelligent User Interfaces For Decision Support And Information Visualization, San Francisco, January 1998

[23] Rasmussen, J. Skills, rules, knowledge: Signals, signs, and symbols and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics, 13(3):257-267 (1983)Risukhin, V (2001). Controlling Pilot Error Automation McGraw-Hill Professional 318 pages ISBN: 0071373209

[24] Schupp, B.A., A.R. Hale, and H.J. Pasman. Optimal Integration of Safety in Complex System Design Using the Safety Modelling Language; Probabilistic Safety Assessment and Management (PSAM 7): 2004.Springer: p. 116-21.

[25] Schupp, B., Basnyat, S., Palanque, P., Wright, P. (2006) A Barrier-Approach to Inform Model-Based Design of Safety-Critical Interactive Systems 9th International Symposium of the ISSA Research Section Design process and human factors integration: Optimising company performances 1-3 March 2006 Nice, France

[26] Schupp, B., Smith, S., Wright, P., Goossens, L. Integrating human factors in the design of safety critical systems - A barrier based approach, Human Error, Safety and Systems Development (HESSD 2004), W. Johnson and P. Palanque (Eds.), Vol. 152 (2004) 285-300, Springer.

[27] Shappell, S.A. and Wiegmann, The Human Factors Analysis and Classification System – HFACS. DOT/FAA/AM-00/7, 2000

[28] United States Department Of Labor Mine Safety And Health Administration Report Of Investigation Surface Area Of Underground Coal Mine Fatal Exploding Pressure Vessel Accident January 28, 2002 At Island Creek Coal Company Vp 8 (I.D. 44-03795)