Using PVS to investigate incidents through the lens of distributed cognition

Paolo Masci*, Huayi Huang, Paul Curzon, and Michael D. Harrison

Queen Mary University of London
Mile End, London, United Kingdom
{paolo.masci,huayih,paul.curzon,michael.harrison}@eecs.qmul.ac.uk

Abstract. A systematic tool-based method is outlined that raises questions about the circumstances surrounding an incident: why it happened and what went wrong. The approach offers a practical and systematic way to apply a distributed cognition perspective to incident investigations, focusing on how available information resources (or the lack of them) may shape user action, rather than just on causal chains. This perspective supports a deeper understanding of the more systemic causes of incidents. The analysis is based on a higher order-logic model describing how information resources may have influenced the actions of those involved in the incident. The PVS theorem proving system is used to identify situations where available resources may afford unsafe user actions. The method is illustrated using a healthcare case study.

Keywords: Theorem proving, incident analysis, socio-technical system.

1 Introduction and motivation

We explore whether automated reasoning tools, like PVS [10], informed by a distributed cognition perspective can lead to a cost effective approach that can help investigators improve their awareness about the circumstances surrounding an incident. *Distributed cognition* [5] explains how people within a socio-technical system use information resources to support their actions and achieve their goals. These information resources may be external (on pieces of paper, signs, computers) or internal (in the head). Understanding how they are deployed and transformed as people perform actions helps to understand the socio-techinical system and what might have led to an incident.

We illustrate our proposed method with a medical incident example. It is based on a comprehensive investigation report [2]. The analysis demonstrates that additional, and potentially error-inducing conditions not envisaged in the original report can be identified.

Contribution. We: (i) demonstrate how a distributed cognition perspective could help investigators understand the circumstances surrounding an incident.

* Corresponding author.

In particular we focus on how the availability of internal and external information resources (or the lack of them) may shape user action. (ii) illustrate how the built-in PVS type-checking mechanism can be used to challenge investigators about their reconstruction of facts.

2 The proposed approach for incident investigation

We explore how an automated reasoning tool like PVS can be used systematically to help investigators understand the contributing factors of an incident by (i) making explicit their conjectures about the availability and use of resources; (ii) supporting an exploration of the validity of the logical argument about how resources are used; (iii) challenging the validity of possible recommendations aimed at avoiding the recurrence of such incidents.

Distributed cognition for incident investigation. We propose a constructive method to incident investigation informed by distributed cognition. In particular this perspective suggests focussing on information resources and their transformation. The method therefore involves the following steps: (1) modeling information resources used by those involved in the incident (e.g., infusion rate printed on a medication order); (2) modeling how information resources propagate within the system (e.g., how a medication order is entered into the pharmacy information system); (3) formulating and verifying conjectures about how resources were used (e.g., were relevant resources available at critical moments to relevant actors) and facts about the prescribed use of information resources (e.g., according to procedures and regulations).

Related work. Our approach is not intended to replace existing accident analysis methods. Rather it can be used in a complementary way to further improve the investigators' awareness about the circumstances surrounding an incident, enhancing the final recommendations. A variety of techniques have been proposed for conducting incident analysis. Johnson's substantial and systematic review of the topic covers many of the more mature techniques [7]. Using formal descriptions of incidents is not a new idea. For example, Ladkin's Why-Because analysis [8] uses formal proofs to verify the correctness and completeness of the causal argument hypothesised by the investigator. Petri Nets have also been used effectively to describe the path towards an incident. A comprehensive overview of formal methods for incident investigation can be found in [6]. Leveson [9], Hollnagel [4] and others critique these approaches because they are largely based on event chains and because inappropriate classifications can bias the analysis. Leveson's STAMP approach aims to overcome some of the perceived deficiencies enabling an exploration of how *constraints* are propagated systemically and contribute to the circumstances of the incident.

3 Illustrative example

Our example is based on a comprehensive accident report concerning an intravenous infusion pump [2]. Documented incidents with a range of infusion pumps

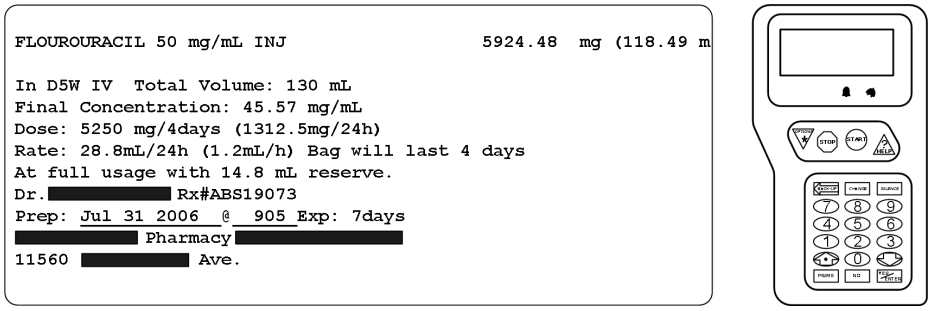

Fig. 1. Reproduction of the label and of the pump used in the incident [2].

show that the wrong drug, or the wrong volume at the wrong rate, may have disastrous consequences for the patient to whom the infusion was being administered [3]. In this case a pump that delivered a drug dose over a period of time to treat a patient in an oncology out-patients unit was programmed incorrectly. The events relevant to this incident included the prescription of the medication at the pharmacy; transferring the prescription to the out-patients unit; one nurse using the label attached to the drug bag to program the infusion pump; another nurse cross-checking, and commencement of the infusion process.

We explored the circumstances surrounding the incident by producing a PVS higher-order logic model, then used PVS methodically to explore the facts and events. Questions raised by the analysis that cannot be answered through the report highlight aspects that may have warranted further investigation. Our complete PVS specification is available at [1]. We focus here for the purposes of illustration on one part of the incident.

3.1 Modeling information resources

Resource identification allows the analyst to externalize facts about the information that is available to the actors. Each resource is modeled using a different PVS `datatype`. The PVS *predicate subtyping* language mechanism which restricts the domain of already defined data-types is used extensively in our specifications. When using expressions with subtypes, PVS automatically generates proof obligations. They identify type correctness conditions (TCCs) to ensure the valid use of the type. By this means issues in the incident may be highlighted.

We start to identify information resources through the "initial understanding" of the incident described in the report: a nurse mistakenly programmed the infusion pump with the wrong rate (28.8 mL/h instead of 1.2 mL/h).

The report notes that a label attached to the drug bag was used to program the pump. This printed label specified: unit of delivery, concentration, rate, and volume to be infused (see Figure 1). Further details are in the incident report [2]. The label provides information resources. It is modelled using a record type `[# a: A, b: B, ... #]`. Each field represents a distinct information resource.

```
label_th: THEORY BEGIN
```

```
  drug_name_type: TYPE = { fluorouracil, cisplatin, %... }
  rate_type: DATATYPE BEGIN mL_Xh(val: real, unit: nat): mL_Xh? END rate_type
  % ...
  bag_label_type: TYPE =
[# drug_name : drug_name_type,
     % ...
    rate_mL_24h  : rate_type,
    rate_mL_h : rate_type, % ... #]
END label_th
```

The label specification models the multiple fields contained in the bag label. If different resources can be specified with the same type, then such resources are potentially either replicated or have compatible content (e.g., in terms of values and/or units) but different meaning. Either case could lead to confusion. Checking such type matches can thus reveal potential issues that may warrant further investigation. In the incident being analyzed, the bag label contained information resources (rate, dose, among others) specified multiple times in different formats. According to the report, this seemed to be the direct cause of the incident. One field on the label was used incorrectly in preference to another: *"The calculated rate (28.8 mL/h) was observed to match a number on the pharmacy label."* ([2], page 13).

The pump contains information resources including the displays, labels that may have been attached to the pump, and audible alarms. Similarly to the bag label, the pump can be modelled as a record type, `pump_type` (not shown here, available from [1]). The predicate subtype used in each field reflects the constraints imposed by the pump on such information resource.

3.2 Modeling transformations of information resources

Transformations are modeled as functions over resources. PVS generates proof obligations to ensure correct use of types. Discharging a proof obligation challenges the investigator's reconstruction of events and facts. Modeling the transformations helps the investigators to be clear about relations that hold among resources. Building a specification that correctly type-checks in the presence of these transformations can therefore help identify when and in what form resources are needed. An example transformation is the use of the information resources printed on the bag label by the nurse to enter the rate into the pump. Consider the information resource "rate". The constraints imposed by the bag label can be naturally modelled as a PVS datatype (`rate_type`, defined in theory `label_th`) with constructor `mL_Xh(val: real, unit: nat)`. The pump rate, on the other hand, is simply a non-negative real number below a maximum value (`rate_type`, defined as `{ x: nonneg_real | x <= max_rate }` in theory `pump_th`). The transformation function is:

```
enter_rate(rate: label_th.rate_type): pump_th.rate_type = val(rate)
```

PVS generates a proof obligation to ensure the correct use of types:

```
enter_rate_TCC: OBLIGATION
    FORALL (rate: label_th.rate_type): val(rate) >= 0 AND val(rate) <= max_rate;
```

In order to discharge this proof obligation, we need to show that the label rate ranges over values that can be entered in the pump — the pump rate is a bounded real number. This proof obligation, with the available information, cannot be discharged — the rate specified on the bag label is unbounded. Although mathematically trivial it highlights implications for the incident investigation that are potentially significant. In fact it raises the question: What are the constraints on the rate value printed on the label? If answers are not available then this may suggest a weakness in the system and a potential for unsafe workarounds. The proof obligation also stimulates further investigation about rate value bounds: What is the procedure in practice when a nurse has to program a pump and the label indicates values that cannot be entered? These issues were not covered in the incident report [2].

3.3 Conjectures about the use of information resources

Conjectures about the actual or prescribed use of information resources can be formulated as predicates over resources. They can be embedded in the specification of information resources – PVS then systematically generates proof obligations that ensure the conjectures hold.

One significant aspect of the incident was the safe limit of administration for the drug. A reasonable conjecture is that the resources available to the nurse provided appropriate information about safe infusion rates. The predicate subtype for the infusion rate in the label is {`r: rate_type | safe_rate?(r, drug_name)`}, where `drug_name` is another information resource provided by the label (PVS allows the specification of dependent subtypes). Instantiating the label (see Figure 1) automatically generates the proof obligation:

```
fluorouracil_bag_label_TCC: OBLIGATION safe_rate?(mL_Xh(28.8, 24), fluorouracil);
```

Given available information resources, this proof obligation cannot be discharged. Neither the label nor the pump provides information about safe limits. A bag label reporting safe limits could have helped the nurses or the patient catch the mistake, e.g., while reviewing the therapy parameters — recognition and pattern matching over recall from memory. Similarly, a pump with safeguards would have prompted a warning and, thus, could have helped catch the mistake. This seems to be a real problem in the incident: *"The calculation was not validated with a mental approximation"* ([2], page 18). A similar issue due to the propagation of information resources from the medication order to the computerised physician order entry can be highlighted with this approach. This issue is not explicitly covered in the incident report [2], though the report points out that *"a miscalculation occurred when the pharmacist initially reviewed the order in the clinic"* ([2], page 33).

4 Conclusions

This brief illustration indicates that applying a distributed cognition perspective to incident analysis can lead to insight that would help guide an incident inves-

tigator. Missing insight could of course just mean that this particular report was weak rather than our method useful. However, we argue that the method found issues beyond that related to direct causes of the particular incident. Insight can also relate to other issues that could lead to future mishaps. A traditional causal analysis method as used does not aim to highlight such issues. They would only be found through craft skill not the method. Our technique shares with STAMP [9] the notion that incident analysis is about discovering systemic failures rather than focusing on causal chains. In future work we do however need to carry out more case studies to further explore the benefits of our approach.

We showed that a relatively simple use of a theorem prover can support this analysis. In the illustration sub-typing alone was used to raise issues and questions and it was not necessary for the analyst to formulate theorems. The analyst just models the incident using PVS to frame their understanding. The tool automatically produces the obligations and proof attempts, demonstrating the satisfaction or otherwise of predefined constraints. As more information is uncovered and modeled further proof obligations raise issues that may warrant further investigation or lead to further recommendations.

Acknowledgments

This work is supported by EPSRC (EP/G059063/1). Michael Harrison had support from Newcastle University. Dr. Astrid Mayer of the UCL Research Department of Cancer Biology, checked some details of the incident report.

References

1. Fluorouracil incident in PVS, Dec 2011. `http://tinyurl.com/PVS-fluorouracil`.
2. ISMP Canada. Fluorouracil incident root cause analysis report. `http://www.ismp-canada.org/download/reports/FluorouracilIncidentMay2007.pdf`.
3. J. Zhang *et. al.* Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics*, 36, 2003.
4. E. Hollnagel. *Barriers and accident prevention.* Ashgate, Aldershot, UK, 2004.
5. E. Hutchins. *Cognition in the Wild.* The MIT Press, new edition, 1995.
6. C. Johnson and C.M. Holloway. A survey of logic formalisms to support mishap analysis. *Reliability Engineering & System Safety*, 80(3):271–291, 2003.
7. C.W. Johnson. *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting.* University of Glasgow Press, Glasgow, Scotland, 2003.
8. P. Ladkin, B. Sieker, and J. Sanders. *Safety of Computer-Based Systems.* Springer-Verlag, Heidelberg and London. draft version from July 27, 2011.
9. N. Leveson. A new accident model for engineering safer systems. *Safety Science*, pages 237–270, 2004.
10. S. Owre, J.M. Rushby, and N. Shankar. PVS: A Prototype Verification System. In *CADE '92*, Lecture Notes in Artificial Intelligence. Springer–Verlag.