Reuse in Hazard Analysis: Identification and Support

Shamus P. Smith and Michael D. Harrison

The Dependability Interdisciplinary Research Collaboration,
Department of Computer Science,
University of York, York YO10 5DD,
United Kingdom.
{Shamus.Smith, Michael.Harrison}@cs.york.ac.uk

Abstract. This paper investigates the nature of hazard analysis reuse over two case studies. Initially reuse in an existing safety argument is described. Argument structures within the hazard analysis are identified and the amount of verbatim reuse examined. A second study is concerned with how reuse changes as a result of tool support. In contrast to the first case, the defined arguments are more diverse - reuse has occurred but is less verbatim in nature. Tool supported argument adaptation has aided the customisation of the reused arguments.

1 Introduction

Descriptive dependability arguments[1] have become a standard part of the process of determining the dependability of a system. At the centre of this demonstration process is the use of techniques for systematic hazard analysis. Hazard identification, classification and mitigation techniques establish that either hazards can be avoided or that they will not affect the dependability of the system. To aid this process, descriptive arguments are commonly produced to mitigate the *perceived* severity of hazards.

In such a process there are two main requirements that need to be fulfilled, that the analysis has (1) sufficient rigour and (2) sufficient coverage. Our confidence in the rigour of a safety case, of which a hazard analysis is a component, is directly linked to the confidence we have in the hazard analysis itself. This confidence will be reinforced by objective evidence of coverage and depth of the analysis - that there are no unexpected adverse consequences within a safety-critical system. In recognition of these issues, a range of methods have been developed to support systematic hazard analysis, for example, Hazard and Operability Studies (HAZOP) [11], Failure Modes and Effect Analysis (FMEA) [6] and THEA (Technique for Human Error Assessment) [14].

Methods such as these commonly involve significant personnel effort and time commitment. Such analysis also generates large amounts of documentation, best

[1] We consider descriptive arguments as informal arguments in contrast to more quantitative, numeric arguments.

S. Anderson et al. (Eds.): SAFECOMP 2003, LNCS 2788, pp. 382–395, 2003.

supported by appropriate tools. The tools that exist typically aim for productivity and can inadvertently promote 'ad hoc' levels of analysis reuse. Verbatim copying and pasting is common practice and there is a risk that such reuse might be used inappropriately and inconsistently [10,16]. This is particularly the case when portions of a hazard analysis are reused without examining the associated descriptive arguments. In summary current tools aim to obtain sufficient coverage but at the cost of rigour that in this situation is obviously unacceptable.

In what follows two case studies are examined. Firstly, an investigation into reuse in a hazard analysis used as part of an existing safety argument is described. Verbatim reuse[2] is used as a measure to determine the frequency of actual reuse in practice. Secondly, tool supported reuse is demonstrated and the nature of the resulting reuse examined in a hazard analysis carried out by the authors on a proposed system.

2 Reuse in Practice: DUST-EXPERT

2.1 Introduction

For understandable reasons it is rare to find complete examples of hazard analysis in the open literature. It is therefore difficult to verify reuse practices within real world cases. However informal discussions with experts in safety-critical systems seem to indicate that reuse is common within industry based hazard analysis. These views appear to be consistent with the results of the following analysis.

2.2 The Domain

DUST-EXPERT is an application that advises on the safe design and operation of manufacturing plants subject to dust explosions. Dust explosion reduction strategies are suggested by the tool which employs a user-extensible database that captures properties of dust and construction materials [5]. Because of concerns about the consequences of wrong advice a safety case argument was developed [5]. Part of this argument involves a hazard analysis utilising the HAZOP technique.

HAZOP is described as a technique of *imaginative anticipation* of hazards and operation problems [15, pg43]. It is a systematic technique that attempts to consider events in a system or process exhaustively. A full description of the method is not relevant to the argument of this paper and the reader is directed to [11]. Suffice to say that a key feature is the way that implicit descriptive arguments are defined, how these arguments can be structured and the extent of their reuse, particularly verbatim reuse. Figure 1 shows a fragment of the software HAZOP for DUST-EXPERT. Verbatim reuse can be seen at references *h 16* and *h 17*.

The HAZOP argument leg of the DUST-EXPERT safety case involves the identification and mitigation of hazards. This part of the analysis contains 334

[2] Verbatim reuse is reuse without modifications [9, pg7].

Hazop Ref.	Item	Guide word	Cause	Consequence/ Implication	Indication/ Protection	Question/ Recommendation	
h 14	**CHANGEVALUE**	No Action	Windows limitations	User types and nothing happens. *No hazard* provided user notices	Addressed by testing	r14	(a)—ensure tests are included to cover changing variables in subwindows (b)—Investigate e.g. colour change when value registered internally
h 15	**CHANGEVALUE**	More Action	Windows limitations	New value bound to several internal variables	Addressed by testing and possibly internal validation	r15	(a)—ensure tests are included to cover this case. (b)—Investigate redisplaying whole screen when individual value updated.
h 16	**CHANGEVALUE**	Less Action		As h 14			
h 17	**CHANGEVALUE**	As well as Action		As h 15			

Fig. 1. Fragment of software HAZOP.

individual HAZOP rows. In order to perform the analysis, descriptive arguments for the HAZOP rows were transformed into a XML[3] structure that faithfully preserves the meaning of the original analysis. An example argument corresponding to the HAZOP reference *h 15* in Figure 1 is shown in Figure 2.

For the descriptive arguments described in this paper the consequence elements are elicited from the *Consequence/Implication* column of the HAZOP and the claim elements are elicited from the *Indication/Protection* and *Question/Recommendation* columns of the HAZOP (for example see Figures 1 and 2). The structure of the arguments in this form is that the claims *support* the mitigation of the consequence. Arguments of this type are used to reduce, or mitigate, the perceived severity of hazardous consequences.

```
CONSEQUENCE_MITIGATION
REF: h15
CONSEQUENCE
----CORRUPT_SYSTEM_DATA: New value bound to several internal variables, hazard
SUPPORT
----CLAIM
------TESTING_CLAIM: Can be picked up in testing
SUPPORT
-----CLAIM
-------SYSTEM_CLAIM: Detected by internal validation
```

Fig. 2. Example descriptive argument.

[3] There is a vast array of texts on XML (Extensible Markup Language) including [12].

2.3 Analysis

Given this example of HAZOP in practice, it is possible to investigate verbatim reuse in the HAZOP data through propagated arguments. Arguments consist of two types: *consequence mitigation* arguments describe how an undesirable consequence can be mitigated by some claim(s) over an environment, for example a claim that appropriate test cases will show that a consequence will not happen; *no meaning* arguments arise when items in an environment cannot be considered meaningfully with HAZOP deviation keywords, for example *more action*, *less action* and *no action*. In this case study, there were 265 consequence mitigation arguments and 69 no meaning arguments. For this analysis only the consequence mitigation arguments have been considered relevant.

To search the XML structure several filtering algorithms were developed to identify interesting features and patterns over the arguments. Arguments in this case study are tree structures with nodes for consequences and support claims. When building the XML argument trees, the data in each node was examined to generate a general classification tag for each node. For example in Figure 2 *CORRUPT_SYSTEM_DATA* is a general consequence tag and *TESTING_CLAIM* is a general claim tag. The tags were used in conjunction with the natural structure of the argument trees, e.g. breadth and depth, to represent a "meta" structure for comparing the arguments. The developed algorithms focused on the general tags assigned to each consequence, and the similarity of data within the arguments. In the DUST-EXPERT domain, five consequence tags were identified.

- *Input failure* consequences involved problems with the user inputting information into the system, for example "user types and nothing happens".
- *Redundancy* consequences typically involve the duplication of system features, for example "several identical help screens appear".
- *Output failures* occur when expected output from the system is not observed by the user, for example "help screen does not appear".
- *System failure* consequences occur when internal system events cause undesirable results, for example, "system spontaneously changes password or adds user".
- *Corrupt system data* consequences are when the DUST-EXPERT database or internal variables have been corrupted via some event, for example, "new values are bound to several internal variables".

These general consequence tags were used to group the consequence descriptions into common themes, thus aiding both the transformation of the arguments into a consistent XML structure and the reuse mechanism. There is no implication here that an exhaustive set of consequence tags are discovered, only that these were the tags that were encountered during the analysis.

Row one of Table 1 shows the number of arguments with each consequence tag. Two filtering results are reported here.

- The first filter identified the amount of verbatim, copy-and-paste, reuse over the arguments. The algorithm produced a list of the arguments with unique

Table 1. Reuse within consequence tags over argument data.

Consequence	Input Failure	Redundancy	Output Failure	System Failure	Corrupt System Data
1) Total arguments	65	18	105	9	68
2) Unique args (Reuse)	42 (35%)	18 (0%)	85 (19%)	8 (11%)	59 (13%)

structure, by physical depth, breadth and tag labels, and unique data. Over the 265 consequence mitigations, 212 are unique data arguments while the remaining 53 occurrences are verbatim reuse. In this example therefore 20% of the arguments have been reused in a verbatim fashion.

– A second filter was applied over the five consequence tags in order to provide a list of the unique arguments for each consequence. Row two of Table 1 shows the number of unique arguments, by data, for each consequence tag followed by the implied percentage of verbatim reuse.

The total amount of verbatim reuse in this example is 20%. This is a considerable amount of potentially unjustified or inconsistently reused analysis. Such reuse will have a direct bearing on any confidence issues in terms of the validity of the analysis. Typically confidence in the analysis is subjectively determined by the regulator's or auditor's confidence in the skill of the analyst. As hazard analysis is a lengthy process and can involve hundreds of lines it seems possible that confidence may be misplaced. Ad hoc approaches to reuse can propagate inconsistencies that can undermine the confidence in the hazard analysis itself and in any associated dependability arguments. It is therefore essential to support the process with a structured approach hereby clarifying how reuse is to be applied. The next section explores these issues and the application of tool support within the context of a second case study.

3 Supported Reuse: Mammography

3.1 Introduction

The analysis in Section 2 and informal discussions indicate that reuse within hazard analysis is common, but that ad hoc application may render an argument unsafe. Due to the systematic nature of hazard analysis techniques one solution is the integration of tool support. Tool support may give the analyst the ability to reflect efficiently on particular examples of reuse. In [16] a mechanism for systematic argument reuse was proposed. A prototype tool [17] to support this mechanism has been developed by the authors and applied to the following case study. The tool provides a platform for documenting a HAZOP style hazard analysis and enables the construction and reuse of consequence mitigation arguments. The motivation for the tool has been to enable the authors to investigate the application of reuse within a constructed case. Hence details of the tools development, evaluation and detailed use are out of the scope of this paper.

As with the study in Section 2 the reuse is applied to the arguments line-by-line with the prototype tool prompting the user with reuse candidates. A specific case is presented to illustrate and explore the approach, namely the hazard analysis of a computer-aided detection tool (CADT) for mammography.

3.2 The Domain

The UK Breast Screening Program is a national service that involves a number of screening clinics, each with two or more radiologists. Initial screening tests are by mammography, where one or more X-ray films (mammograms) are taken by a radiographer. Each mammogram is then examined for evidence of abnormality by two experienced radiologists [8]. A decision is then made on whether to recall a patient for further tests because there is suspicion of cancer [2]. Over the screening process it is desirable to achieve a low number of false positives (FPs), so that fewer women are recalled for further tests unnecessarily, and a high true positive (TP) rate, so that few cancers will be missed [8]. Unfortunately the radiologists' task is a difficult one because the small number of cancers is hidden among a large number of normal cases. Also the use of two experienced radiologists, for *double readings*, makes this process labour intensive.

A solution that is being explored is the use of computer-based image analysis techniques to enable a single radiologist to achieve performance that is equivalent or similar to that achieved by double readings [3,8]. Computer-aided detection systems can provide radiologists with a useful "second opinion" [18]. The case study in this section involves the introduction of a CADT as an aid in screening mammograms. When a CADT is used, the radiologist initially views the mammogram and records a recall decision. Then the CADT marks a digitised version of the X-ray film with "prompts" that the radiologist should examine. A final decision on a patient's recall is then taken by the human radiologist based on the original decision and the examination of the marked-up X-ray. A summary of this process can be seen in Figure 3 (from [2]).

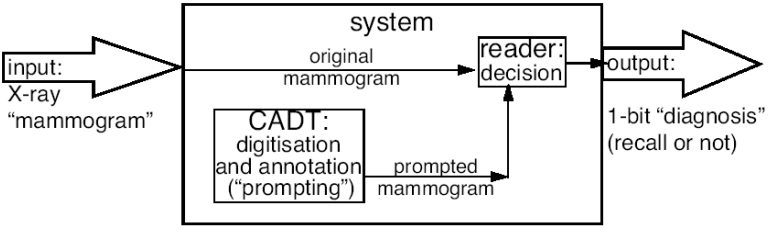

Fig. 3. Model for person using computerised aid for reading mammograms in breast screening.

A system based on the model shown in Figure 3 has been investigated to identify the undesirable consequences, for example an incorrect recall decision,

that may arise. The activities and tasks that make up the diagnosis process form the basis of the analysis (see Figure 4). The argument for safe use involves a number of argument legs covering three main activities namely (i) human analysis of the X-ray, (ii) CADT analysis of the X-ray and (iii) the recall decision by the human based on a review of their original analysis and the CADT analysis.

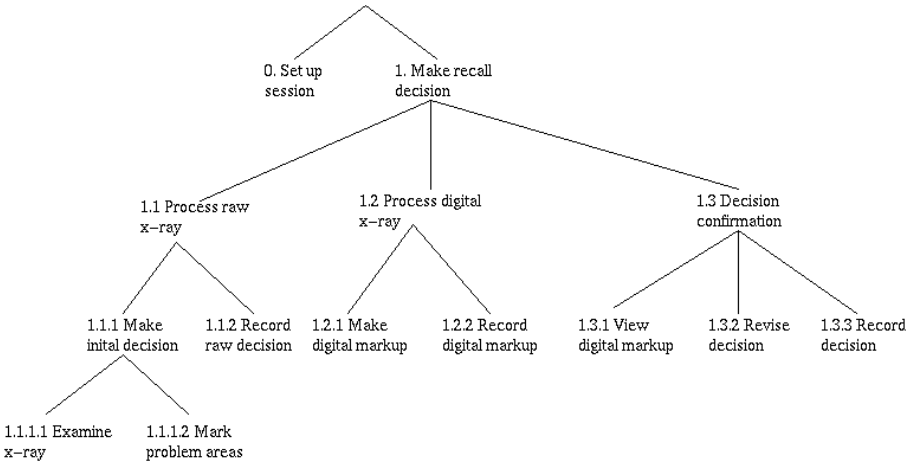

Fig. 4. Activities in the recall decision making process.

3.3 Reuse and Tool Support

When investigating the introduction of new technology the construction of a safety case is common. For this domain a safety case would consist of several elements including reliability analysis for the marking of the digital mammogram, the CADT performance and the consequences of human-error. However, for this paper one element of the safety case analysis will be considered, namely hazards and consequences in the diagnosis process as defined in the overall system model (see Figure 3).

A hazard analysis for the system was completed by a team including the authors using a line-by-line approach for reusing analysis components. The identified consequences in the current case were individually matched against consequence examples already defined in the current domain, i.e. from a library of already defined consequence arguments. This can be illustrated as follows.

Suppose there was a case in a current domain with an activity A_1 that resulted in outcome O_1 where the outcome could be mitigated by argument M_1 and verified by evidence E_1. If a new case also has a (A_1, O_1) pairing, it may be possible to mitigate the new pairing with either the (M_1, E_1) pair of mitigation claim and evidence (see Case 2 in Table 2) or an adaptation of the pairing appropriate to the current situation.

Table 2. Reuse examples.

Case 1				Case 2			
Activity	Outcome	Arg	Evidence	Activity	Outcome	Arg	Evidence
A_1	O_1	M_1	E_1	A_1	O_1	$M_1?$	$E_1?$
A_2	O_2	M_2	E_2				
A_3	O_2	M_3	E_3	A_4	O_2	$M_2?M_3?$	

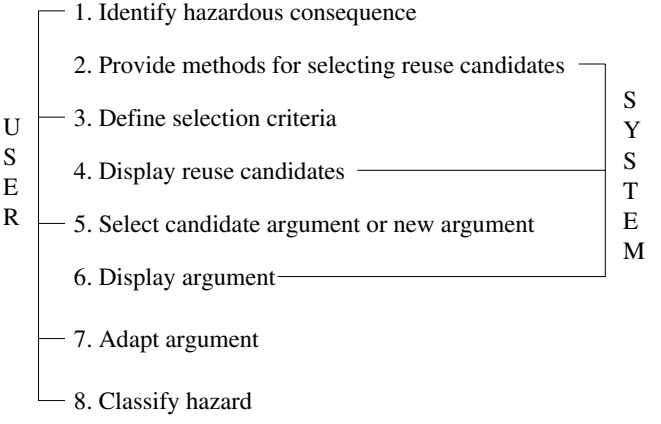

Fig. 5. Argument reuse process for hazard mitigation and classification.

Alternatively, if multiple reuse candidates are identified, expert judgement would be required to select the most appropriate candidate. For example consider that in Table 2 activities A_2 and A_3 in Case 1 both share outcome O_2. If either of these arguments were to be reused for Case 2 activity A_4, which argument would be reused, M_2 or M_3?

The approach attempts to highlight potential candidates for reuse from the previous arguments and transfer the mitigation arguments and associated claim and evidence reasoning to the current case. However, candidate selection is only half of the approach. After selection, a reused candidate is typically adapted. Adaptation involves customising existing elements in the analysis and the addition of new argument and evidence elements. This process is context sensitive and requires expert domain knowledge.

A method to support this process has been developed [17] and includes steps for the identification of hazardous consequences, the definition of selection criteria to search for possible reusable arguments and the selection of reuse candidates or the definition of a new argument form. The new argument (either from a reuse candidate or a new argument template) must then be adapted to meet the specifics of the current analysis row. Finally a judgement on the nature of the hazard or consequence, i.e. whether it has been completely mitigated or not, is produced. An overview of the method can be seen in Figure 5 where the major tasks, both user and system, are identified.

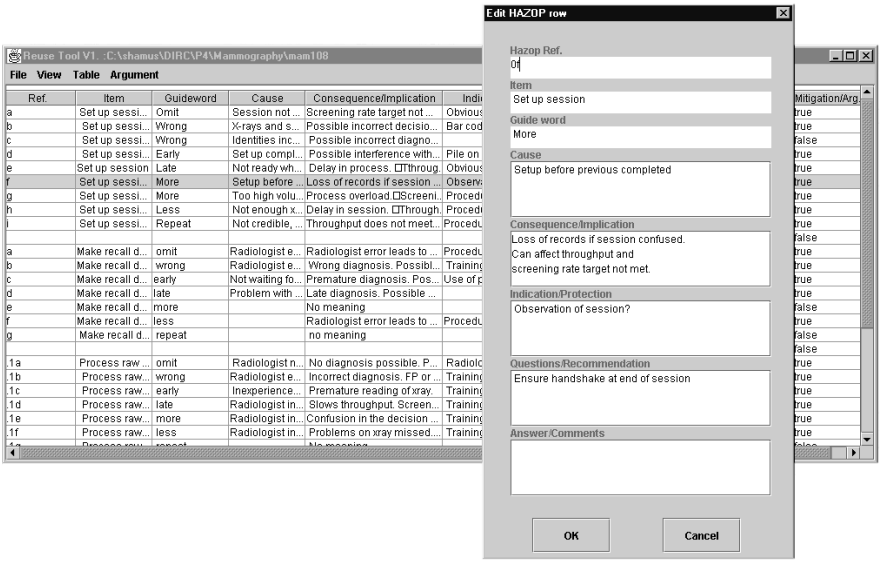

Fig. 6. Editor for collating hazard data.

A prototype tool [17] has been developed to support this process. The tool aids both the gathering of hazard documentation (see Figure 6) and the selection and adaptation of reuse candidates. The tool automates the matching process between arguments to find suitable candidates for reuse either by keywords or via consequence and/or claim tag matching. The matching process compares arguments based on a notion of structural similarity [4,13] over argument structure and data elements.

Figure 7 shows a selection of arguments presented as candidates for possible reuse after a keyword search. Multiple reuse candidates are commonly identified for each query and the final selection for reuse and the adaptation is left to the domain expert/tool user. As not all searches will provide an appropriate candidate for reuse, the tool also allows arguments to be defined as new argument forms.

Having completed one analysis the significant question is how tool support affects the natural occurrence of reuse as described in Section 2. There are a number of ways in which reuse may have been altered.

- The tool may produce a bias toward more verbatim reuse. Users may skip the argument adaptation step and leave the reused arguments in their initial form with the same argument structure and data.
- By prompting the user to select and adapt arguments from previous examples, a greater amount of artificial argument diversity may result. For instance more varied argument forms may be defined as users trivially adapt a reused argument. An example from the mammography case study can be seen in Figures 8 and 9. The Figure 8 argument has been reused in the

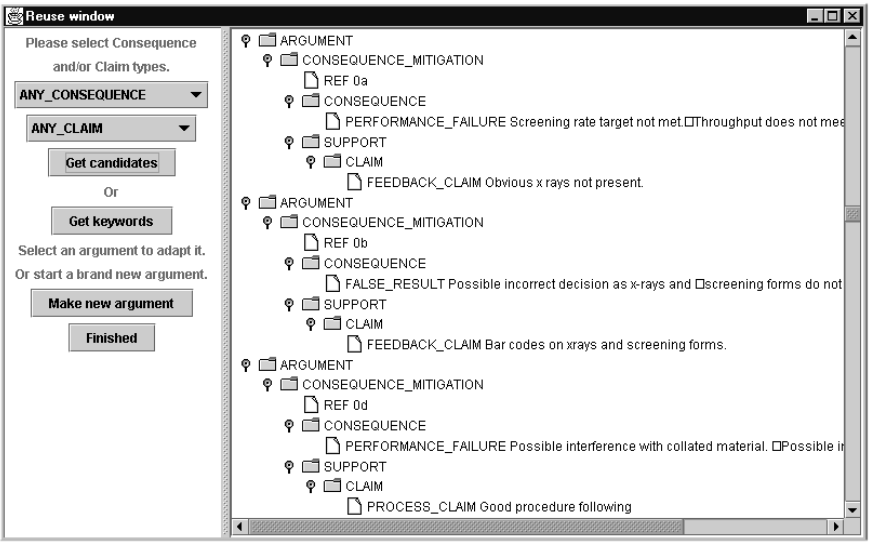

Fig. 7. Presenting argument reuse candidates.

Figure 9 argument by matching the consequence tag, in this case *OUTPUT_FAILURE*. The consequence data in the second argument (Figure 9) has been adapted while the structure and claims of the argument itself are unchanged. Thus, although the structure remains the same, a unique, by data, argument has been defined.

– Users may try to adapt every instance of reuse to form new argument forms. This may have the advantage of customising the fit of the arguments to the current situation but may not be cost effective due in part to time considerations. Also producing large libraries of unique arguments increases the searching cost for selecting reuse candidates.

REF: 1.2.1e
OUTPUT_FAILURE: Additional markups of features. Possible FP
mitigated by
---TRAINING_CLAIM: Large training set for examples for Neural Net
mitigated by
---DIVERSITY_CLAIM: Use of diverse marking algorithms
mitigated by
---TESTING_CLAIM: Test cases meet expert option and actual outcomes
----------**supported by**
----------TESTING_CLAIM: Coverage by test cases

Fig. 8. Original mammography argument.

REF: 1.2.1g
OUTPUT_FAILURE: Confusion in the decision making due to multiple markups
mitigated by
---TRAINING_CLAIM: Large training set for examples for Neural Net
mitigated by
---DIVERSITY_CLAIM: Use of diverse marking algorithms
mitigated by
---TESTING_CLAIM: Test cases meet expert option and actual outcomes
----------**supported by**
----------TESTING_CLAIM: Coverage by test cases

Fig. 9. Adaptation of consequence data after reuse.

The following section presents an analysis of the application of reuse within a constructed case in comparison to the raw data case described in Section 2 and the issues noted above.

3.4 Analysis

A total of 61 consequence mitigation arguments were defined for the activities shown in Figure 4. The support tool was used to transform the arguments into a XML structure and for this domain eight consequence tags were identified: *input failure*, *performance failure*, *output failure*, *false result*, *corrupt system data*, *record loss*, *CADT failure* and *false positive*. Row one of Table 3 shows the number of each consequence tag over the arguments.

Table 3. Reuse within consequence tags.

Consequence	Input Failure	Performance Failure	Output Failure	False Result	Corrupt System Data	Record Loss	CADT Failure	False Positive
1) Total args	1	17	21	8	1	8	4	1
2) Unique args (Reuse)	1 (0%)	16 (6%)	18 (14%)	7 (13%)	1 (0%)	8 (0%)	4 (0%)	1 (0%)
3) Adapted args (Reuse)	0 (0%)	12 (71%)	14 (67%)	5 (63%)	0 (0%)	4 (50%)	2 (50%)	0 (0%)

The algorithms described in Section 2.3 were used in the mammography system analysis to identify the amount of verbatim reuse over the arguments. The first algorithm produced a list of arguments with unique structure, by physical depth, breadth and tag labels, and unique data. Over the 61 consequence mitigations there are 56 unique arguments making 5 occurrences of verbatim reuse. Thus in this case, 8% of the arguments have been reused in a verbatim fashion. The spread of these arguments can be seen in row two of Table 3.

For each new argument, the tool provides a list of candidates for reuse that have been matched either on the basis of consequence tags or keyword matching (see Figure 7). The user then adapts the new argument, on-the-fly, to one of

these candidates. The reuse mechanism simplifies the adaptation process producing more unique argument forms and a smaller number of arguments with verbatim data. Although these adapted arguments are unique by data, they are still a product of the reuse mechanism. Unfortunately, they can not be identified using the algorithms applied in Section 2.3. Therefore the data and structural elements of the arguments were manually examined to determine whether any similarities between arguments were a product of the reuse process. The complete set of arguments was separated into three groups, (i) arguments produced with verbatim reuse, (ii) arguments produced via reuse and adaptation and (iii) new argument forms.

Over the 61 consequence mitigations there are 37 arguments that had been adapted via the reuse process. This results in 61% of the arguments being adapted in this case. The spread of these arguments over the consequence tags can be seen in row three of Table 3.

As with the case in Section 2 there has been a significant amount of reuse over the hazard analysis. However in this case the reuse has been coupled with argument adaptation. This process provides the benefits of allowing consistency to be maintained between reused arguments while customising the arguments to the context of the current analysis.

4 Conclusions

Descriptive arguments are a standard part of the process of determining the dependability of any system. Such arguments are typically at the core of hazard analysis techniques that contribute to the construction of safety cases. Unfortunately hazard analysis is a time consuming and labour intensive process and hence reuse of analysis components is common. Reuse of analysis also results in the reuse of the associated descriptive arguments. However, when dependability issues are concerned, inappropriate reuse can lead to misleading levels of confidence in the final analysis. This is clearly undesirable in areas such as safety-critical systems.

This paper has described the nature of reuse over two case studies. Reuse issues in a hazard analysis used in partial support of a safety case for an advisory system have been described. Argument structures within the hazard analysis were constructed and the amount of verbatim reuse examined. A reuse support tool aided the analysis of a new proposed system for providing automated support in mammography. In contrast to the first case, the defined arguments are more diverse - reuse has occurred but is less verbatim in nature. Tool support has promoted active reflection by the analyst on the arguments to be reused and has resulted in increased argument adaptation. This has the advantage of a better fit for the reused arguments to the current situation.

However, one issue of concern with tool support is that bias may be incorporated into the reuse process. For example, new forms of arguments may be ignored in preferance to arguments suggested by the tool. Currently, this issue is the responsibility of the user who applies expert judgement in the argument

construction and adaptation process. We are investigating if the process supported by the tool promotes undue bias towards certain kinds of arguments and whether tool support could be configured to avoid bias.

Another issue is the cost of the reuse process. There will be costs associated with both the organisation of the raw data into argument structures and the ease of the final reuse. Also there is the overhead of identifying appropriate reuse arguments. Such issues must be balanced against any proposed benefits. However, issues of cost and benefit typically require some form of measure to allow realistic predictions to be made. We are currently investigating a notion of confidence (and confidence in the worth of an argument) as such a measure to demonstrate that argument reuse will lead to improved arguments and consequently improved confidence in the arguments.

Acknowledgements. This work was supported in part by the UK EPSRC DIRC project [7], Grant GR/N13999. The authors are grateful to Adelard [1] for providing the DUST-EXPERT safety case, and Eugenio Alberdi and Andrey Povyakalo who provided helpful feedback on a field test of the prototype tool in the mammography domain.

References

1. Adelard. Dependability and safety consultants. http://www.adelard.com [last access 6/06/03].
2. Eugenio Alberdi, Andrey Povyakalo, and Lorenzo Strigini. "Diversity modelling" of computer aided diagnosis in breast screening. DIRC workshop, November 2002, London.
   http://www.csr.city.ac.uk/people/lorenzo.strigini/ls.papers/2003_CADT/ [last access 6/06/03].
3. Caroline R. M. Boggis and Susan M. Astley. Computer-assisted mammographic imaging. *Breast Cancer Research*, 2(6):392–395, 2000.
4. Katy Börner. Structural similarity as guidance in case-based design. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topic in Case-Based Reasoning*, volume 837 of *Lecture Notes in Artificial Intelligence*, pages 197–208. Springer-Verlag, Berlin, 1993.
5. Tim Clement, Ian Cottam, Peter Froome, and Claire Jones. The development of a commercial "shrink-wrapped application" to safety integrity level 2: The DUST-EXPERT^{TM} story. In Massimo Felici, Karama Kanoun, and Alberto Pasquini, editors, *18th International Conference on Computer Safety, Reliability, and Security (SAFECOMP 1999)*, volume 1698 of *Lecture Notes in Computer Science (LNCS)*, pages 216–225, Toulouse, France, 1999. Berlin: Springer.
6. B. S. Dhillon. Failure modes and effects analysis – bibliography. *Microelectronics and Reliability*, 32(5):719–731, 1992.
7. DIRC. Interdisciplinary research collaboration on dependability of computer-based systems. http://www.dirc.org.uk [last access 6/06/03].

8. Mark Hartswood and Rob Proctor. Computer-aided mammography: A case study of error management in a skilled decision-making task. In Chris Johnson, editor, *Proceedings of the first workshop on Human Error and Clinical Systems (HECS'99)*. University of Glasgow, April 1999. Glasgow Accident Analysis Group Technical Report G99–1.
9. Santhi Karunanithi and James M. Bieman. Measuring software reuse in object oriented systems and ada software. Technical Report CS-93-125, Department of Computer Science, Colorado State University, October 1993.
10. Tim P. Kelly. *Arguing Safety – A Systematic Approach to Managing Safety Cases*. PhD thesis, Department of Computer Science, The University of York, 1999.
11. Trevor Kletz. *Hazop and Hazan: Identifying and Assessing Process Industrial Hazards*. Institution of Chemical Engineers, third edition, 1992. ISBN 0-85295-285-6.
12. William J. Pardi. *XML in Action: Web Technology*. IT Professional. Microsoft Press, Redmond, Washington, 1999.
13. Enric Plaza. Cases as terms: A feature term approach to the structured representation of cases. In *First International Conference on Case-based Reasoning (ICCBR-95)*, pages 265–276, 1995.
14. Steven Pocock, Michael Harrison, Peter Wright, and Paul Johnson. THEA – a technique for human error assessment early in design. In Michitaka Hirose, editor, *Human-Computer Interaction: INTERACT'01*, pages 247–254. IOS Press, 2001.
15. David. J. Pumfrey. *The Principled Design of Computer System Safety Analysis*. PhD thesis, Department of Computer Science, The University of York, 2000.
16. Shamus P. Smith and Michael D. Harrison. Improving hazard classification through the reuse of descriptive arguments. In Cristina Gacek, editor, *Software Reuse: Methods, Techniques, and Tools*, volume 2319 of *Lecture Notes in Computer Science (LNCS)*, pages 255–268, Berlin Heidelberg New York, 2002. Springer.
17. Shamus P. Smith and Michael D. Harrison. Supporting reuse in hazard analysis. DIRC workshop, November 2002, London.
http://www.cs.york.ac.york/∼shamus/papers/smithdirc02.pdf
[last access 6/06/03].
18. Bin Zheng, Ratan Shah, Luisa Wallance, Christiane Hakim, Marie A. Ganott, and David Gur. Computer-aided detection in mammography: An assessment of performance on current and prior images. *Academic Radiology*, 9(11):1245–1250, November 2002. AUR.