Safety-Critical Systems: Open Assessment 2003-2004

Prof. Chris Johnson,
University of Glasgow, Glasgow, G12 8QQ. Scotland.
johnson@dcs.gla.ac.uk, http://www.dcs.gla.ac.uk/∼johnson

1 Introduction

A significant part of this course is devoted to an analysis of the IEC61508 standard. You can obtain a more detailed overview of this standard from:

`http://www.iec.ch/zone/fsafety/fsafety_entry.htm`

IEC61508 has a number of flaws but it remains the most important standard for the development of programmable systems across the majority of UK industries. Part of the reason for this is that it is recommended by the UK Health and Safety Executive (HSE).

IEC61508 is a development standard; it is intended to support the design, implementation and maintenance of programmable systems in the safety-critical industries. However, the HSE recently sponsored a project to determine whether this standard could also be used to support the analysis of incidents and accidents. There were several reasons for this. There are no agreed standards for how to analyse adverse events involving programmable systems. It was argued that much could be gained if we could trace the causes of any failures back to problems in the development activities that are recommended by the IEC61508 standard. Also, if a failure could not be traced back to problems in the implementation of the development standard then this could be used to arge for revisions of that standard. The intial reports from the HSE project are available from the following web site:

`http://www.dcs.gla.ac.uk/ johnson/hse/`

In particular, the three case study documents show how the proposed approach can be applied to several different incidents. Your task is to develop a web-based tool to assist in the application of these analytical techniques. The intention is that you should discover a little more about the 61508 standard and also more about the reasons why safety-critical programmable systems actually fail 'in the real world'.

2 Your Task

There are two parts to this exercise. The first is to devise a tool to assist in the analysis of programmable system failures using the IEC61508 standard. The second is to evaluate the usefulness of your tool against one or more case studies with a group of potential end-users.

2.1 Tool development

You are free to use any implementation techniques during the development of your analytical tools. These could simply focus on the development of an html version of the flow chart's described in the technical report mentioned above. For instance, client side image maps could lead the user from the high level questions down to lower-level elements of the

IEC61508 taxonomy. Alternatively, you might re-use existing AWT/Swing code on the web to develop a simple diagram editor for the ECF drawings. Finally, if you feel that the proposed approach is inadequate or weak in particular areas then you may choose to develop it or ammend it in various ways. For example, the existing flow-chart provides a poor means of analysing human factors failures. Alternative flow-charts are available in documents such as:

`http://www.hse.gov.uk/research/rrhtm/rr081.htm`

If you extended the flow chart in this way then you should talk to me about your ideas as soon as possible to check that they would be sufficient to obtain a good mark. Once you have built your tool to support the analysis of incidents and accidents using the 61508 standard, you must conduct an informal or formal evaluation to assess the effectiveness of your work.

2.2 Evaluation

The initial HSE project was focussed on a limited number of case studies and some consultations with potential end-user companies. There has been almost no work to determine whether a range of different users can exploit the approach to come up with similar findings about the same incident. It also seems likely that some incidents involving the failure of programmable systems will be more difficult to analyse using the 61508 scheme than others. You should, therefore, conduct a formal or informal evaluation to explore one or more of these issues. For example, you could give a group of users the same incident to analyse using your tool and then compare the findings that they reached. Alternatively, you could compare the findings arrived at using the tool with those reached using the existing paper-based approach. You could be to compare a tool-based flowchart with a paper-based ECF and so on. Another approach would be to look at the use of the tool with several different icnidents.

You should pay attention to the experimental design that is used in the evaluation, either referring to the IS3 course:

`http://www.dcs.gla.ac.uk/ johnson/teaching/is3`

Or directly by asking me before conducting the study. In particular, you will need to identify potential case studies remebering that the others on this course will be familliar with the case studies that are presented on the web sites cited in this question.

3 Transferable Skills

As mentioned, this exercise will provide a first-hand introduction to two different technical problems in safety-critical systems development. The first is the challenge of understanding a complex development standard. The second is to understand how to analyse the causes of complex, programmable system failures. Both of these areas are the subject of continuing research as regulatory and investigatory organisations are only just beginning to understand the problems posed by new generations of computer-related systems. Hence many of the skills provided by this assessed exercise are in very scarce supply.

4 Assessment Criteria and Submission Details

This exercise is degree assessed. It contributes 30% to the total marks associated with this course. The body of the report should not exceed fifteen A4 pages. The report must be printed out and must be submitted in a secure binder (i.e., one that will keep the pages together and in the correct order). It must include:

- A title page containing your student as well as your contact details (email address etc);
- A table of contents and appropriate page numbers;
- A section on the tool that you developed.
- A section on the evaluation method that you used.
- A results sections.
- Conclusions.

In addition to the fifteen pages associated with the body of the report, you may also include appendices. These should contain the listing of any code used during the study together with suitable acknowledgements for the source of code that has been borrowed from other programmers. ¡P¿ The report should be handed in at the start of the lecture on Friday 28th November 2003. Extensions will only be granted in exceptions circumstances and they should be requested prior to the deadline. Extensions for medical reasons should be reported as soon as possible and should be supported by forms from a medical practitioner. Extensions for equipment failures may be granted provided that you let me know as soon as they occur; so that I can make sure they get fixed as soon as possible. Please make sure that you keep back-up copies of all of your work towards this exercise. The following marking scheme will be applied:

- 15 for the method;
- 10 for the results;
- 15 for the conclusion;
- 10 for the technical documentation.

All solutions must be the work of the individual submitting the exercise. If any code or design ideas are borrowed from course notes, books or other students then those sources MUST be clearly acknowledged. All questions about this exercise should be addressed to Chris Johnson.