Xday, XX May 2006.

9.30 am - 11.15am

University of Glasgow

DEGREES OF BEng, BSc, MA, MA (SOCIAL SCIENCES).

COMPUTING SCIENCE - SINGLE AND COMBINED HONOURS
ELECTRONIC AND SOFTWARE ENGINEERING - HONOURS
SOFTWARE ENGINEERING - HONOURS

SAFETY-CRITICAL SYSTEMS DEVELOPMENT

Answer 3 of the 4 questions.

Sample Solutions

1.

a) The Airbus A380 cabin air pressure control is a highly safety-critical system. It must ensure the comfort and well-being of passengers and crew across a range of altitudes and was developed to the DO-178B standard, ‘level A’. At this level, the failure of any software is assumed to have potentially catastrophic consequences for flight operations.

Explain how risk assessments can be used to justify the assignment of ‘level A’ criticality to the A380 cabin air pressure control system.

[5 marks]

[unseen problem/seen problem]
In the course, we have discussed a number of standards that can be used to support the development of safety-critical software including DO-178B and IEC61508. In the answer to this question, I am looking for some appreciation that it is the impact on the equipment under control that determines the level of adverse consequences from any software failure (2 marks). I would also expect the usual formulations of the risk equation including consequence and likelihood assessments from any hazards that arise from the failure of the software (2 marks). I would also expect some reference to the difficulty of assessing risk from any subsystem within such a complex application domain and of validating risk assessments in this area (2 marks).

b) In order to meet the DO-178B ‘level A’ requirements, software testing must have complete coverage across the three different dimensions listed below:

- Statement Coverage; every statement in the program should be invoked at least once during testing.

- Decision Coverage; every entry and exit in the program should be tested at least once and each decision in the program should be examined at least once across all possible values.

- Modified Condition Decision Coverage; every entry and exit point in the program should be invoked at least once, every decision in the program should be tested for all possible outcomes at least once and each condition in a decision should be shown to independently affect the outcome.

Briefly explain the management techniques that must support the manual analysis of code to ensure coverage of ‘level A’ software verification under DO-178B. (Hint: consider the potential problems of conducting tests across each of these three dimensions and whether it is possible to satisfy all of these requirements in the general case).

[5 marks]

[unseen problem/seen problem]

The focus of this question is not directly on the difficulty of satisfying these requirements but on the difficulty of managing the verification process (1 mark). In the course we have looked at examples of the documentation that was developed within a NASA project assessing the output of different contractors developing the same piece of code under the DO-178b standard and some reference could be made to these documents and to the need for a document management system to help record the suite of tests that must be performed to meet the demands of the three forms of coverage (2 marks). The obvious point is that with anything but a trivial program there will be many hundreds of thousands of test cases, for instance with MCDC testing, and so we need some way of keeping track of the testing process to document and prove the coverage that is required (2 marks). We have also mentioned in passing the use of automated verification environments – noting both the strengths and limitations of some of these systems (2 marks). There is also a tie-in with the final part of this question so some students might gain additional marks by noting these links (1 mark).

c) Part of the A380 cabin pressure control system has been developed using a code generator. This system enables developers to use more abstract, mathematically based languages using environments such as Matlab, Simulink and Stateflow to create the high-level design for a system. The code generator will then help to derive executable software based on these designs.

Explain why the use of code generators can reduce the burdens associated with the verification of 'level A' software in DO-178B and explain the potential dangers of using a code generator.

[10 marks]

[unseen problem]

Code generators enable designers to construct the architecture of a system and to specify abstract properties of an implementation without considering all of the details that may have to be introduced into the final code (2 marks). This provides important benefits because these abstract expressions of a design are likely to be more tractable and hence more amenable to verification than the eventual implementation (2 marks). For example, the use of mathematically based environments such as Matlab and Stateflow provide important tools for the formal analysis of abstract designs (2 marks).

The limitations stem from the refinement and reification that must take place in order to move from the abstract implementation to the final code (2 marks). For example, the transformation to final code must preserve the properties that were verified on the abstract representation. In the general case, it is a non-trivial task to ensure that this is the case because the refinement process typically introduces additional details that may not be easy or possible to express in the abstract language (2 marks).

There are some particular problems with the formal analysis of safety-critical systems that must be considered when using these approaches. Simply providing a verification framework does not address the problems of identifying and formulating the theorems that might preserve the safety of an application process (1 mark). The proof of a theorem on an abstract representation may also create a false sense of confidence in the behavior of an implementation, if for instance it embodies assumptions about other processes and systems or it relies upon an abstract representation that is not faithful to the final implementation (see above) (2 marks). There are also general problems associated with the biases of white box testing that can be raised in this context (1 mark).

Other solutions might go on to consider the task of verifying the correctness of the code generator and similar boot strapping issues (2 marks for mentioning it, more on a pro rata scale for an in depth analysis).

2.

a) The US Army recently announced that it is likely to support a combination of the Linux operating system and Intel processors for its Future Combat System (FCS) program. FCS is intended to support the next-generation of manned and robotic air and ground systems connected by a fast, secure communications network. There are four layers identified in the high-level architecture: applications, services, transport and standards and it is estimated that the eventual system will integrate more than 35 millions lines of computer code.

Explain the potential benefits of using the Linux-Intel combination for the development of potentially safety-critical applications.

[4 marks]

[seen problem]
In the lectures, we have covered some of the problems that the US military have faced in sourcing processor technology for long to medium term programmes (1 mark). We have looked at preferred part suppliers and the use of these relationships to guarantee quality and supply (1 mark). The choice of mass market and COTS technologies have also been mentioned with economies of scale and help in driving out bugs/design errors through the benefits of large scale FRACAS (failure reporting systems) (2 marks). Equally we have mentioned the problems of commercial confidentiality when using mass market processor technologies. These issues can be contrasted with the choice of Linux which is largely open source (1 mark) but at the same time also has some of the benefits of a COTS/mass market application; there is a strong support infrastructure that rivals commercial systems even though it may not be commercial in the usual sense of the term (2 marks). The safety issues include the ability to review key elements of the OS code that is critical when considering heterogeneous integration across the range of platforms in the FCS programme (1 mark).

b) One of the leading figures in the development of the Future Combat System (FCS) project, General Charles Cartwright, recently confessed that they were unsure about "how you put ground troops and robots together at the same time". In particular, it can be difficult to guarantee safety and liveness properties. For example, a safety property might require that an FCS vehicle never fires on friendly troops. A liveness property might require that information from friendly forces eventually arrives at the intended FCS vehicle.

Describe some of the technical difficulties in demonstrating safety and liveness properties for complex, safety-critical systems such as the FCS applications.

[6 marks]

[unseen problem]
Safety properties involve proofs to show that a theorem *always* holds or conversely that some property is never the case. Liveness properties show that a theorem *eventually* holds (1 mark). Traditionally, these properties can be established across software designs using techniques such as model checking (1 mark) or theorem proving (1 mark) but also including term rewriting etc. In model checking, theorems can be asserted using languages such as temporal logic, finite state machines can then be constructed to determine whether safety or liveness properties hold for a particular system. The state explosion problem creates obvious problems for the termination of some model checkers – showing that a property holds across every possible state can require careful construction of the state machine to avoid exhaustive search across many millions of states (1 mark). Further problems relate to the difficulty of identifying appropriate theorems that can then be proven or disproven using these technologies (1 mark). This is a particular issue for the FCS project because it is unclear what safety and liveness properties one might want to express across a battlefield system that integrates human and robotic/autonomous elements (2 marks). Safety properties cannot simply be asserted over the software running on a single processor (1 mark). We must consider the real-time interaction between thousands of independent or semi-autonomous agents (1 mark). These real-time issues raise further technical challenges in model checking where, for example, the base time granularities may not be the same across all agents and hence it can be difficult to construct traditional interval temporal models with which to assert safety and liveness properties (1 mark).

c) A key component in the Future Combat System (FCS) software is the System-of-Systems Common Operating Environment (SOSCOE). This architecture uses commercial off-the-shelf hardware to produce:

> "…a nonproprietary, standards-based component architecture for real-time, near-real-time, and non-real-time applications. SOSCOE also contains administrative applications that provide capabilities including login service, startup, logoff, erase, memory zeroize, alert/emergency restart and monitoring/control. The SOSCOE framework allows for integration of critical interoperability services that translate Army, Joint, and coalition formats to native, internal FCS message formats using a common format translation service. Because all interoperability services use these common translation services, new external formats will have minimal impact on the FCS software baseline. The FCS software is supported by application-specific interoperability services that act as proxy agents for each Joint and Army system. Battle Command (BC) can access these interoperability services through application program interfaces that provide isolation between the domain applications, thereby facilitating ease of software modifications and upgrades.
>
> (Acknowledgement: http://www.army.mil/fcs, last accessed 11th January 2007)

Identify the features of the System-of-Systems Common Operating Environment that have an implication for the safety of the Future Combat System (FCS). Explain any techniques that you would use to address the safety concerns that arise from your analysis.

[10 marks]

[seen problem/unseen problem]

There are many different solutions to this question. During the course, we have practiced taking documentation from 'real world' projects to analyze and identify the implications for the development of safety-critical systems and this is another example. For instance, they might focus on monitoring and control within the SOSCOE administrative functions – this is ambiguous so solutions could talk about the monitoring used either for redundancy and fault correction or in the sense of intrusion detection – either way there is a link to safety via dependability and answers that draw on both interpretations would be very acceptable (2 marks). The memory zeroise functionality is similar and in some of the lectures we have spoken about this wider interaction between safety and security, which may often be incompatible or at least at odds with each other (2 marks). The development of internal FCS message formats provides a further focus for potential solutions. We have covered some basic issues in the development of safety-critical protocols including the use or error correction and monitoring (2 marks). Similarly, we have spoken a little about the use of proxy agents and interaction across different communications architectures, for example mentioning the problems of network monitoring and of benign and pathological failure modes (2 marks). A key issue here is to establish the isolation from faults that may be introduced when new nodes are added to a network or existing nodes develop a fault (1 mark). The introduction of a new application should not introduce significant additional verification and validation requirements on existing systems (1 mark). While this seems relatively straightforward using the interfaces and proxies envisaged in the SOSCOE white paper, it is less easy to determine how these properties might be carried through at the application level and this motivates General Cartwright's comments cited in part b) of this question (2 marks).

3.

a) During 2007, the European Space Agency (ESA) hopes to launch the Jules Verne spacecraft; this is the first of a proposed series of Autonomous Transfer Vehicles (ATVs). These ATVs are extremely large, approximately the size of a double-decker bus and will initially dock with the International Space Station. Because the autonomous design limits human intervention, there is redundancy in both the hardware and software. For example, the main Fault Tolerant Computer (FTC) that navigates the ATV mission is composed of three identical processing chains all executing the same code. In addition, a completely independent Monitoring and Safing Unit (MSU), incorporating two processing chains, monitors the performance of the FTC. If it detects an anomaly, it will first isolate the FTC and then execute a collision avoidance maneuver.

Explain why the MSU hardware and software must be independent from the FTC and why the MSU has been described as 'a satellite within a satellite'.

[5 marks]

[seen problem / unseen problem]
We have looked at several different hardware and software architectures from space missions, such as the Shuttle's General Purpose Computing systems. However, we have not discussed the new architectures for the ATV and associated missions. The challenge is in applying what we have discussed for other systems to this novel domain. The identical software in the FTC creates common vulnerabilities (1 mark). Identical software is likely to fail in identical ways because bugs and design errors will be replicated (1 mark). Hence by insuring independence in terms of software design and implementation there is the hope that the MSU will not fail in the same manner (1 mark). However, experience has shown that requirements flaws often propagate between diverse areas of a project and N-version programming may not always deliver the promised resilience (1 mark). The hardware independence is trivially important because otherwise we are likely to have common mode failures when the loss of a single component can take out the FTC and MSU (1 mark). Hence we have a 'satellite within a satellite' – a system that should be sufficient to identify an erroneous condition and take over from the FTC when necessary – this phrase simply summarizes the use of redundancy to guard against system failure (2 marks).

b) The Autonomous Transfer Vehicle is of considerable strategic importance for the European Space Agency because there is a long term plan to use it in a modular approach to human space flight. This would involve the development of a Crew Transport Vehicle (CTV) which would add an accommodation module onto the existing ATV design. The software architecture would remain the same with the addition of an independent processor monitoring the accommodation capsule. However in order to support human spaceflight, the MSU emergency procedure would have to support more than the existing avoidance maneuvers.

Identify the new functions the MSU would have to support before trusting the lives of a crew to this architecture and identify any problems that you would envisage during the introduction of this new functionality. (Hint: you should make clear any assumptions about the MSU functionality in the ATV or CTV applications).

[5 marks]

[seen problem/ unseen problem]
Part a) of this question describes the current role for the MSU within the ATV. If it detects a potential problem with the Fault Tolerant Computer (FTC) the MSU takes control and executes a collision avoidance maneuver. This functionality is essential to avoid damaging the International Space Station or other close targets. It is also sufficient for the ATV because there is no essential requirement to bring the ATV back to earth in the case of a failure (1 mark). The vehicle payload can remain in orbit to either be retrieved during subsequent missions or to burn up on re-entry (1 mark). By restricting the MSU to collision avoidance, ESA and its partners can strictly limit the amount of code associated with the highest assurance levels in the ATV architecture (1 mark). However, this is not sufficient for the CTV because once a collision avoidance maneuver has been executed there remains the problem of how to bring the crew back to earth (1 mark). We cannot simply leave them in orbit for an indefinite period while a rescue mission is organized

(1 mark). Hence the MSU functionality must be significantly enlarged (1 mark). Providing a de-orbit and return capability vastly increases the processing requirements of the MSU and will probably imply changes in memory requirements, hardware processor power etc for the CTV (2 marks). In addition, this increased functionality will significantly increase the amount of code and the complexity of that code at the highest assurance levels within a CTV as opposed to a AVT mission (1 mark). This will massively increase the costs of verification and validation etc (1 mark).

c) Two different companies are involved in the validation of the existing MSU software. EADS is responsible for primary development under contract from the European Space Agency (ESA), while Rovsing of Denmark provide independent verification and validation of the code.

Imagine you are an employee of Rovsing, write a brief technical report for the project managers in the ESA explaining how you might use a combination of white and black box testing to increase confidence in the MSU software. (Hint: your answer should consider the limitations of independent testing in the context of such a complex mission)

[10 marks]

[unseen problem]
There are many different approaches to this question but the technical reports should be based around the properties of white and black box testing, which have been presented in the course. White box testing involves some knowledge of the internal architecture used in the software development. Black box testing assumes that the verification authority has no detailed knowledge of the code (1 mark). Clearly in this case where we have an independent verification and validation company, it is possible to conduct black box testing that is often difficult or hard to sustain within the same organization that did the development (1 mark). There are often limited numbers of competent engineers within many companies and hence the same individuals who are qualified to do the testing are often also called upon to support development and this can compromise in-house black box tests (2 marks). Conversely, black box testing is well suited for component based programs because it encourages engineers to develop clean interfaces between modules so that each element can be tested in isolation and without knowledge of implementation (2 marks). Hence there can be degrees of knowledge disclosed about the underlying architecture where verification bodies may understand the interfaces and component structure but not the implementation details (sometimes called gray box testing) (2 marks).

However, as the question suggests there can be significant problems in conducting black box tests for such complex systems. Simply by studying the requirements at the level of detail required to conduct the tests can involve a sustained and in-depth cooperation with the development teams during which it is difficult or impossible to avoid the exchange of architectural and implementation information (2 marks). Similarly, the interactions and real time characteristics of the MSU within the ATV make it difficult for any team to conduct adequate tests without understanding at least a little of the underlying software architectures (2 marks). Hence it is likely that there will be combinations of the two approaches. One recognized technique is to use an alpha group from the external body to conduct black box tests while the beta group within the contractor develops a suite of diagnostic white box tests to determine the reasons why any black box test fails. The beta team may or may not include members of the development group (2 marks).

4. The legal systems in Scotland and in England and Wales provide for the common law offence of 'manslaughter' using the *mens rea* or 'controlling mind' principle. The prosecution must identify a responsible individual in order to establish guilt. In particular, the legal systems do not allow not allow for aggregation where the actions of a number of people over a period of time cumulatively provide the necessary evidence of a 'guilty mind' even when there is no individual who might exhibit this degree of culpability.

   Briefly explain the influence that these provisions have upon the operation and management of safety-critical technologies and state the reasons why you feel that *mens rea* should either be retained or removed.

[20 marks]

[Essay]

This is an open-ended question, the class have been told to expect an essay and I will provide them with some broad hints to look at recent papers on Corporate Manslaughter legislation and the provisions of the Health and Safety at Work Act, 1974. I am looking for some acknowledgement that most large-scale organisations that operate in complex, safety-critical industries devolve decisions through teams of co-workers so that it is difficult if not impossible to identify a single 'controlling mind' (1 mark). It is for this reason that attempts to prosecute organisations for Corporate Manslaughter often fail (1 mark). The only exceptions tend to be small to medium sized businesses and, in particular, sole traders where it is not possible to devolve responsibility through intermediate levels of management (1 mark). I have provided several examples of unsuccessful prosecutions. These include the manslaughter acquittals of two directors and P&O European Ferries following the 1987 loss of the Herald of Free Enterprise. Similarly, manslaughter charges were thrown out against Barrow Borough Council following the deaths of seven members of the public from legionnaire's disease in 2002. The judge ruled that there was no case to answer. The jury failed to reach a verdict on seven counts of manslaughter against an individual design services manager. Partly as a result, the UK Crown Prosecution Service (2005) decided there was no prospect of convicting any individual or company for manslaughter by gross negligence following the 2002, Potters Bar rail crash (6 marks – I would not expect this level of detail for full marks but some examples and case studies should be briefly outlined).

I have also cited successful cases; the director of an English waste paper recycling business was given a 12 month jail sentence following the death of an employee in December 2003. The accused pleaded guilty to manslaughter and other health and safety charges. His company was also fined £30,000 with costs of £55,000. The employee climbed into a paper-shredding machine to clear blockages. The machine contained a series of hammers that revolved at high speed. The subsequent investigation into the accident found that there was no means of securely isolating the machine while blockages were being cleared and the control system was contaminated with dust. In this case, the individual director was found guilty of manslaughter as the 'directing mind'. Both he and his company pleaded guilty to offences under section 2(1) and 37(1) of the Health and Safety at Work etc Act 1974 (HSWA). Section 2(1) of HSWA states "It shall be the duty of every employer to ensure, so far as is reasonably practicable, the health, safety and welfare at work of all his employees." Section 37(1) states "Where an offence under any of the relevant statutory provisions committed by a body corporate is proved to have been committed with the consent or connivance of, or to have been attributable to any neglect on the part of, any director, manager, secretary or other similar officer of the body corporate or a person who was purporting to act in any capacity, he as well as the body corporate or a person who was purporting to act in any such capacity, he as well as the body corporate shall be guilty of that offence and shall be liable to be proceeded against and punished accordingly." (6 marks – again, I would not expect this level of detail but I would expect some recognition that the HSAW can be used when manslaughter charges cannot).

The closing sections might discuss the controversy over the Home Office proposals for Corporate Manslaughter legislation as well as the counter proposals from the Scots panel of experts. Alternately reference may be made to the C-45 Westray Bill in Canada or the Australian Capital Territories legislation (2 marks for each brief description up to 6 in total). These measures reveal the tensions that arise when attempting to remove *mens rea*. Firstly, the need to establish guilt is intended as a safeguard in most legal systems that would protect minors and individuals with mental disabilities who might lack the necessary understanding of the consequences of their actions (1 mark). Secondly, creating the opportunity for collective guilt might dissuade well qualified individuals from working the field of safety-critical systems. This would have the paradoxical effect of making systems less safe because few would want the risks of criminal prosecution with significantly less protection than exists at the moment (2 marks). Thirdly, in the event of an accident the threat of criminal prosecution would make companies less willing to disclose information about their management and decision making processes and hence less insight may be obtained from the investigatory process (1 mark). And so on…

Balanced against these concerns are strong arguments in favour of corporate manslaughter legislation without *mens rea.* At the moment, the burden of proof is too high for effective prosecution. Hence financial sanctions tend to be applied to companies – this can be little or no disincentive for large corporations where direct personal liability of senior executives might do more to focus minds on safety issues (2 marks). The threat of individual liability might be more persuasive than is presented in the previous paragraph – a small number of prosecutions could be effective in creating examples and then the mass of potential violations could be dealt with more sympathetically providing the company and management cooperated with an investigation (1 mark as I think this is a weaker argument and not equitable). Public opinion often demands corporate manslaughter legislation in the aftermath of major accidents and to continue to stall when (for instance) action has been promised by several recent administrations is to deny a manifesoe promise. In addition, there are some industries that seem to be particularly slow in enacting health and safety improvements (1 mark). And so on…

It is possible to argue both for and against the mens rea principle but the students will know that my own personal view is that it should be retained.