A0.1

Lent Term 2003

http://www.comp.scran.ac.uk/~setzer/index.htm

Dr. Anton Setzer

02/index.htm

http://www.comp.scran.ac.uk/~setzer/lectures/lectures/critsys/

CS.411 Critical Systems:

A0.2

Critical Systems: CS.411 Lent Term 2003, Sec. A0

(c) Anton Setzer 2003 (except for pictures)

A0.3

Definition: A critical system is a system of a critical system failing.

(c) Anton Setzer 2003 (except for pictures)

A0.4

- Failure may result in the failure of the business using that system.
  - Business-critical system.
- Failure may result in the failure of some goal-directed activity.
  - Mission-critical systems.
- Failure may cause injury or death to human beings.
  - Safety-critical systems.

(c) Anton Setzer 2003 (except for pictures)

A0.1 Introduction, Overview

A0.2 Literature

A0.3 Plan

A0.4 Administrative Issues

(b) Two aspects of critical systems.

(a) A case study of a safety-critical system failing.

A0.1

Three Kinds of Critical Systems.

(c) Anton Setzer 2003 (except for pictures)

(a) A Case Study of a Critical System Failing

Definition:

A critical system is a computer, electronic or electromechanical system the failure of which may have serious consequences, such as:

- Injuries or death of human beings.
- Substantial environmental damage.
- Substantial financial losses.
- The failure of which may have serious consequences, such as:

  - Complete, electronic or electromechanical system.
Examples of Critical Systems

- Medical Devices
  - Aerospace
  - Military aviation
  - Manned space travel
- Chemical Industry
  - Nuclear Power Stations
  - Chemical Industry
- Transport
  - Traffic control
  - Railway control system
  - Road traffic control (e.g., traffic lights)
- Defense
- Areas where secrecy is required
- Online shopping cart
- Customer account system in a bank
- Other military equipment

Mission-Critical

- Navigation system of a space probe
- Area where secrecy is required
- Civil aviation
- Railroad

Business Critical

- Customer account system in a bank
- Online shopping cart
- Secret service
- Defense
- Areas where secrecy is required

Examples of Critical Systems (Cont.)

- Astra Train accident (January 5, 2000)

Report from November 6, 2000

http://odin.dep.no/jd/norsk/publ/rapporter/aasta/
Figur 3.1: Rørosbanens beliggenhet i Norge. Rørosbanen markert med rødt
Kilde: Jernbaneverket

Figur 3.5: Rørosbanen Hamar-Støren med stasjoner, regiongrenser og grense mellom
de områder togledersentralene overvåker
Kartgrunnlag: Jernbaneverket

Figure 2-3. Rudstad station track layout

Figur 3.1: Rørosbanen beliggende i Norge. Rørosbanen markert med rødt
Kilde: Jernbaneverket

Figur 3.5: Rørosbanen Hamar-Støren med stasjoner, regiongrenser og grense mellom
den områder togledersentralene overvåker
Kartgrunnlag: Jernbaneverket
Train 2302 is 21 minutes behind schedule.

When reaching Rena, delay is reduced to 8 minutes, in order to cross 2369 at Rudstad.

Train 2369 leaves after a brief stop Rudstad 13:06:17.

Train 2369 leaves after a brief stop Rena 13:06:15.

Train 2302 is 21 minutes behind schedule.

Train 2302 leaves after a brief stop Rena 13:06:15, in order to cross 2369 at Rudstad.

Leaves Rena after a stop with green exit signal 13:06:15.

When reaching Rena, delay is reduced to 8 minutes.

Figur 3.12: De to togsettene etter at brannen var slokket
Foto: Politiet/Kripos

Figur 3.13: Motorvogn, lokomotiv, vogn nr. 3 og nr. 2 etter at brannen var slokket
Foto: Politiet/Kripos

Traffic controller saw the train, because of use of mobile telephones.

Train 2369 leaves after a brief stop Rudstad 13:06:17.

According to timetable crossing of trains at Rudstad.

Railway with one track only. Therefore crossing of trains only at stations.

Sequence of events:

Trains collide 13:12:35, 19 persons are killed.

The correct number hadn’t been passed on to him.

Rail traffic controller sees alarm approx. 13:12.

Alarm signal to the rail traffic controller (no audible signal).

Local train shouldn’t have had green signal.


Local train shouldn’t have had green signal.


Local train shouldn’t have had green signal.
Investigations

Notechnicalfaultsofthesignalsfound.

Traindriverwasnotblindedbysun.

Fourincidents
ofwrongsignalswithsimilarsignalingsystemsreported:

- Exitgreensignal.
- Distriansignalgreen,mainsignalsignaled,trainpassesovermainsignal,and
- Trafficcontrollertoldhimnotinthecorner,signaled.
- Mainlightsignaledandtrainstopped.

18April2000:Trainhasgreensignal.
Traindriverseesredsignaland
pullover.

SINTEF(FoundationforScientificandIndustrialexpertise)foundnosignificanterrortotheaccident.

"Criticalcondition:
- Plusback.
- Distriansignalgreen,mainsignalsignaled,trainpassesovermainsignaland
- Trafficcontrollertellsignaled,signaled.
- Exitgreensignalandtrainstopped.

Fourincidentsofwrongsignalswithsimilarsignalingsystemsreported.

Traindriverwasnotblindedbysun.

Notechnicalfaultsfromthesignalsfound.

Investigations

Conclusions

Itispossiblethatthetraindriverofthelocaltrainwasdrivingagainstaredsignal.
Thefactthathewasstopping,andthatatthesametimethatothertrainssafesignaltwoandahalfminutesaheadoftime,isthinkable.

Itiscriticalthattheoriginalsignalwasthedrivingsignal.

Investigations

Severalsafety-criticaldeficienciesinthesoftwarefound(someknownbefore!)

Thesoftwarewascompletelyreplaced.

Evenifthisparticularaccidentwasnotduetoasoftwareerror,apparently,

Investigations

Criticalsystems,CS
411,Lentterm2003,Sec.A0
A0-15

SINTEF(FoundationforScientificandIndustrialResearchattheNorwegianInstituteofTechnology)foundnomistakeleadingdirectlytotheaccident.

ConclusionofSINTEF:
Noindicationofabnormalsignalstatus.

(Mistakeoftraindriver(diedintheaccident).

(HumanError).

AssessmentofreportbyRailcert

Criticisms:
- SINTEFwasonlylookingforsinglecausefaults,notformultiple.

Cases:
- SINTEF: foursignalingerrorsforthesamecausequestionedbymultiple.(HumanError)

Criticalcondition:
- Exitgreensignal.
- Distriansignalgreen,mainsignalsignaled,trainpassesovermainsignal,and
- Trafficcontrollertoldsignaled,signaled.
- Exitgreensignalandtrainstopped.

Fourincidentsofwrongsignalswithsimilarsignalingsystemsreported.

Traindriverwasnotblindedbysun.

Notechnicalfaultsfromthesignalsfound.

Investigations

Conclusions

Itispossiblethatthetraindriverofthelocaltrainwasdrivingagainstaredsignal.
Thefactthathewasstoppingandleftalmostatthesametimeastheothertrain,makesithinkable.

Itiscriticalthattheoriginalsignalwasthedrivingsignal.
In the protocol of an extremely simple installation, which was established in 1975, and exists in this form in 40 installations in Sweden, a safety-critical error was found when verifying it with a theorem prover formally 1997.

Lotsother errors in the Swedish rail system were found during formal verification. We consider a three-level model (Leveson, pp. 48–51).

Causal Factors.
We consider a three-level model (Leveson, pp. 48–51).

Level 1: Chain of events.
- Described above.

Level 2: Conditions, which allowed the events on level 1 to occur.
- Software error.
- Inadequate red signals for a short period of time.
- The local train left early.

Level 3: Conditions and Constraints.
- Those local trains do not stop at the moment of the red light.
- The local train does not stop at the moment of the red light, which stops trains.
- Due to the fact that the train driver was relying on his watch (which was not an ATC (automatic train control) installation), which stops trains.
- The local train does not stop at the moment of the red light, which stops trains.

We consider a three-level model (Leveson, pp. 48–51).

Root causes.
- Problems found in a level 3 analysis from the root causes.
- Successful of social and economic policies and conditions.
- Technical and physical conditions.
- First level.
- Second level.
- Third level.

Root causes.
Dichtung a red light.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.

Root causes.
- The local train does not stop at the moment of the red light.
The trace controller didn't see the control light.

Control panel was badly designed.

A visual warningsignal is not enough, in case the system detects a possible collision of trains.

Therail controller couldn't warn the driver, since he didn't know the mobile telephonenumber.

To rely on a mobile telephony network in a safety critical system is extremely careless.

Mobilephones often fail.

The connectionsmight be overcrowded.

Connection to mobiles might not work in certain areas of the railway network.

Theprocedure for passing on the mobile phone numbers was badly handled.

- Need for verified design of such software.
- Very difficult to write correct protocols for distributed algorithms.
- Control of railway signals is a safety-critical system and should be designed with high level of integrity.

The rail controller couldn't warn the driver, since he didn't know the mobile telephonenumber.

- A visual warning signal is not enough, in case the system detects a possible collision of trains.
- Control panel was badly designed.
- The traffic controller didn't see the control light.

The safety of the train engines was not very good.
Lessons to be learned (cont.)

- The railway controller was overworked.
- Overconfidence in ICT.
  - Otherwise one wouldn’t have used such badly designed software.
  - Otherwise one wouldn’t have simply relied on them (at least a special agreement with the mobile phone companies should have been set up).
- Flaws in management practices.
  - A mechanism should have been established to thoroughly investigate.
  - No protocol for dealing with mobile phone numbers.
  - No record for dealing with mobile phone numbers.
  - Training of operators.
  - Procedures the operators have to follow.
- System aspects.
  - Safety critical systems are very complex.
  - Software, which includes parallelism.
  - Hardware.
    - Might fail (light bulb of a signal might burn through), relays age.
  - Adverse conditions (low temperatures, rain).
    - Have to operate under adverse conditions.
  - Human-computer interaction.
    - Protocols the operators have to follow.
    - Ameliorating events can prevent the accident or reduce its impact.
    - Events that may propagate or ameliorate the accident.
- Intermediate events.
  - A sequence of events had to happen in order for the accident to take place.

Elements of an accident or reduce its impact.

- and additional safeguards, which prevent a triggering event from
  - If possible by using several independent safeguards.
- Preventing triggering events
  - When different safety critical systems are used.
- Ameliorating events can prevent the accident or reduce its impact.
- Events that may propagate or ameliorate the accident.

Lessons to be learned (Cont.)

- Cultural habits.
  - Train the operators.
  - A mechanism should have been established to thoroughly investigate.
- System aspects.
  - Safety critical systems are very complex.
  - Software, which includes parallelism.
  - Hardware.
    - Might fail (light bulb of a signal might burn through), relays age.
  - Adverse conditions (low temperatures, rain).
    - Have to operate under adverse conditions.
  - Human-computer interaction.
    - Protocols the operators have to follow.
    - Ameliorating events can prevent the accident or reduce its impact.
    - Events that may propagate or ameliorate the accident.
- Intermediate events.
  - A sequence of events had to happen in order for the accident to take place.
Usually at the end of an investigation conclusion "human error".

The architecture of the software was investigated but no detailed search.

Most failures of safety-critical systems were caused by multiple failures.

For a bug was done.

The architecture of the software was investigated but no detailed search.

Lessons to be learned (cont.)
Two Streams in this Module (cont.)

Stream B (interleaved with Stream A):

- Closer look at one prototype example of a tool which allows to write 100% correct software.
  - Programming with dependent types, based on Martin-Löf type theory.
  - Use of the theorem prover Agda.
  - Experimentation system.
  - Still area of research, few successful industrial applications (using the theorem prover Coq).
- Goal is that students have been in contact with one proof assistant.
  - Templates in C++ or soon in Java (??) is one approximation.
  - Interesting: use of a theorem prover, which is used like a programming language.
- Stream B (interleaved with Stream A):
  - 80% Exam.
  - 4 small assignments. Each counts 5% (plan, might be changed).
  - Two lectures per week.
  - Web page contains overhead slides from the lectures.

Assessment:

- 80% Exam.
- 20% Coursework:
  - Mainly associated with Stream B.
  - Four two-weeks later.
  - Five questions concerning Stream B.
  - One question mixture of Stream A and B.
  - One question concerning Stream A.
  - Due two weeks later.
  - Handed out, answered, every second week.

Course material will be continuously updated.

Page dimensions: 595.0x842.0

Address:
Dr. A. Setzer
Dept. of Computer Science
University of Wales Swansea
Singleton Park
Swansea
SA2 8PP
UK
Tel. (01792) 513368
Fax. (01792) 295651
Email a.g.setzer@swansea.ac.uk

Assessment:
80% Exam.

One question mixture of Stream A and B.
One question concerning Stream A.
One question concerning Stream B.
- Mainly associated with Stream B.
- Due two weeks later.
- Handed out, answered, every second week.

Course material will be continuously updated.

- Templates in C++ or soon in Java (??) is one approximation.
- Interesting: use of a theorem prover, which is used like a programming language.
- Still area of research, few successful industrial applications (using the theorem prover Coq).
- Goal is that students have been in contact with one proof assistant.

Timetable:
- Two lectures per week.
  - Monday, 13:00, Robert Recorde Room.
  - Thursday, 12:00, Robert Recorde Room.
- Webpage contains overhead slides from the lectures.
- Course material will be continually updated.

(© Anton Setzer 2003 (except for pictures))
Learning outcome:

- Familiarity with issues surrounding safety-critical systems.
- Understanding of techniques for specifying and verifying high-integrity software.
- Experience with one proof assistant.

(e) Literature

In general, the module is self-contained.

(e) Literature

In the following list of books, which might be of interest, if you later have

(e) Literature

Probably not all topics covered (last year only A2 was reached).

B1. Introduction.
B2. The logical framework.
B3. Data types.
B4. Interactive programs in dependent type theory.
B5. Case studies.

B Stream B

Plan Stream A

B Stream B

Plan Stream A

(e) Literature

In general, the module is self-contained.

(e) Literature

In the following list of books, which might be of interest, if you later have

(e) Literature

Probably not all topics covered (last year only A2 was reached).

B1. Introduction.
B2. The logical framework.
B3. Data types.
B4. Interactive programs in dependent type theory.
B5. Case studies.

B Stream B

Plan Stream A

B Stream B

Plan Stream A
Books Relevant for Stream A

Main course book:

Supplementary books on software engineering aspects:

Supplementary books on software configuration management:

Critical Systems, CS 411, Lent Term 2003, Sec. A0

A0-39

Some books on general formal methods:
Jonathan Jacky: The way of Z. Practical programming with formal methods.


John Barnes: High integrity Ada. The SPARK approach.


A0-40

Software.

A report on a lot of (100?) accidents and incidents of critical errors in software.


Critical Systems, CS 411, Lent Term 2003, Sec. A0

A0-41

Literature Relevant for Stream B

Supplementary books on software engineering aspects:

Main course book: