Introduction to Laboratory Class 1

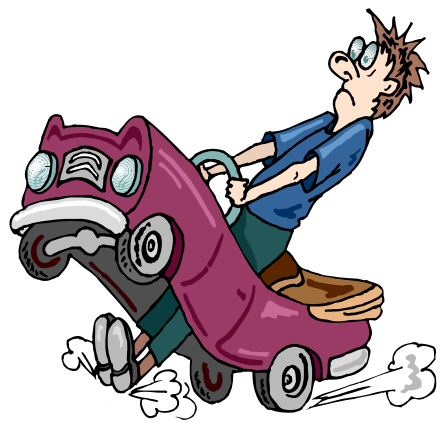
Dependability Modeling

Fault-Tolerant Computer Systems

Exercise 4, 2012

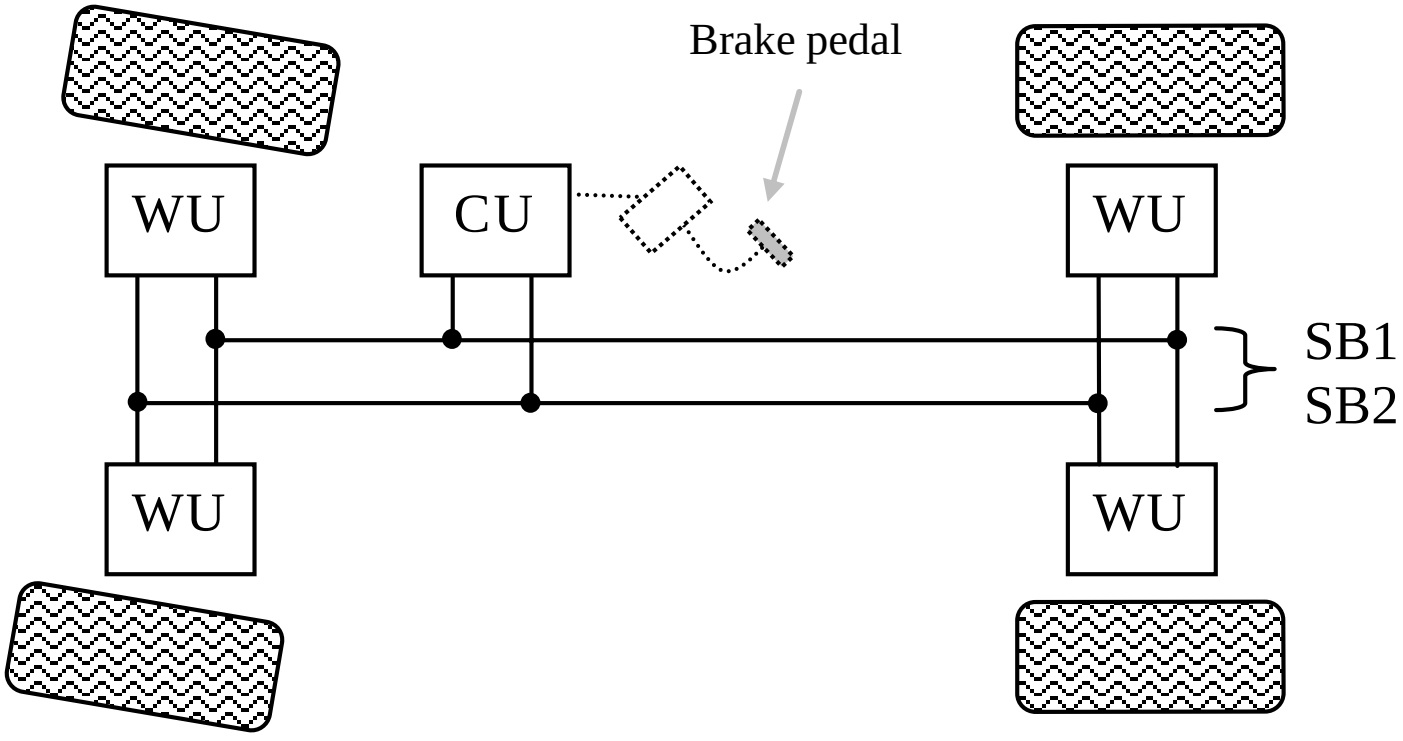
Laboratory Class 1 – Objective

Compare the MTTF and the reliability of two fault-tolerant architectures for a brake-by-wire system using SHARPE.



Outline

Brake-by-Wire System



Candidate Architectures

Distributed architecture

- Computer modules located at each wheel (WUs) execute a control algorithm locally to calculate the brake force to apply.

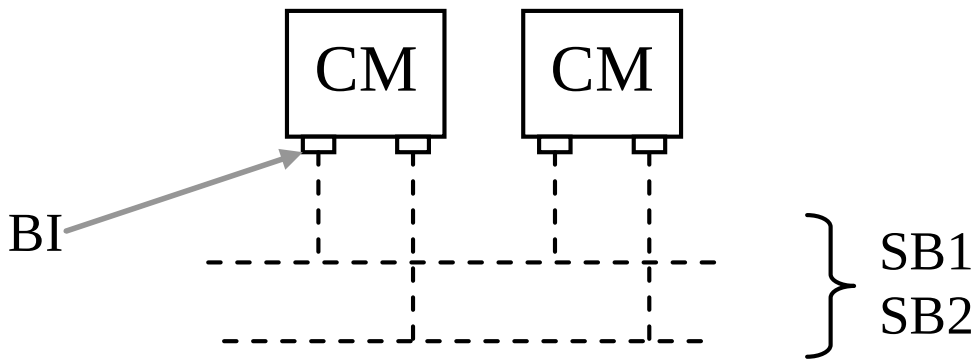
Centralized architecture

- A central computer module (CU) executes a control algorithm to determine the brake force to apply to each wheel.
- The calculated actuator commands are sent to computer modules located at each wheel (WUs).

Central Unit (CU) – Distributed Architecture

Duplex system

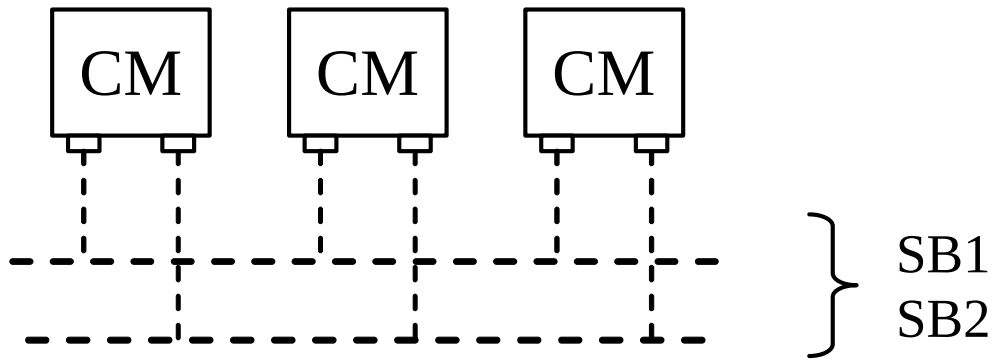
- Two fail-silent computers operating in active redundancy.
- Error detection coverage is 99%



Central Unit (CU) – Centralized Architecture

Triplex/Duplex system

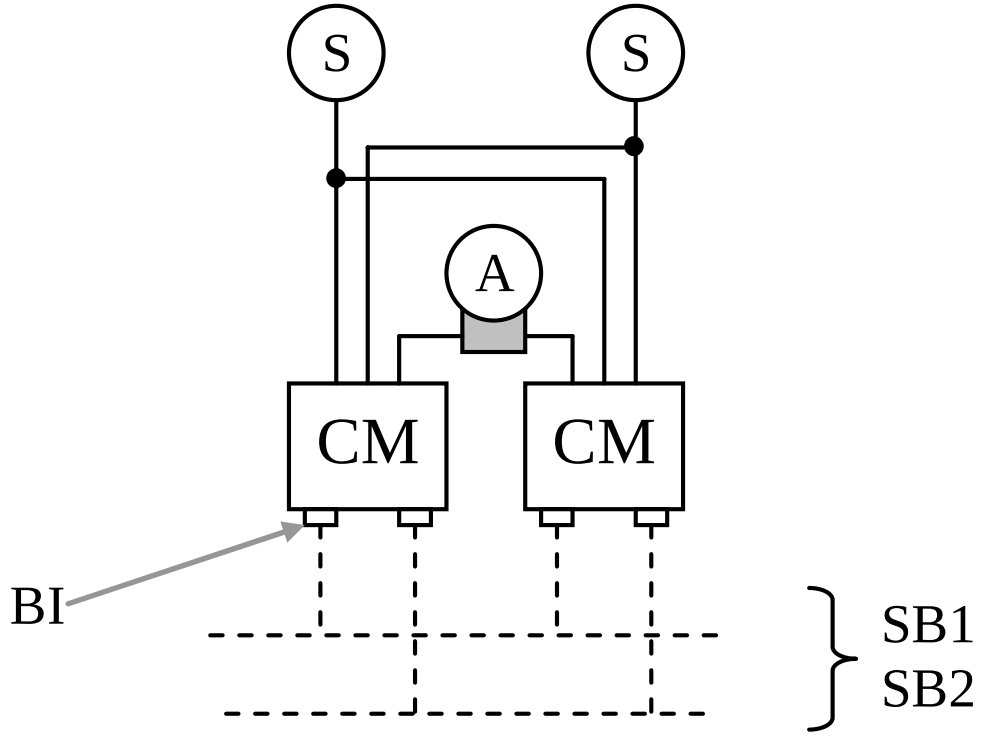
- Three fail-silent computers operating in active redundancy.
- Coverage is 100% as long as three CMs are operational.
- Reconfigured to a duplex system after the first CM fails.



Wheel Unit (WU)

Each WU consists of

- Two fail-silent computer modules
- Two sensors and one actuator
- Four bus interfaces



Laboratory Class Problem

Problems

1. Compare the reliability after 2 years of operation, and the MTTF of the two architectures.
2. Compare the operation time for a reliability of 0.92.

Use SHARPE to analyze the two architectures for two levels of functionality:

- Full functionality – Four wheels operational.
- Degraded functionality – At least three wheels operational.

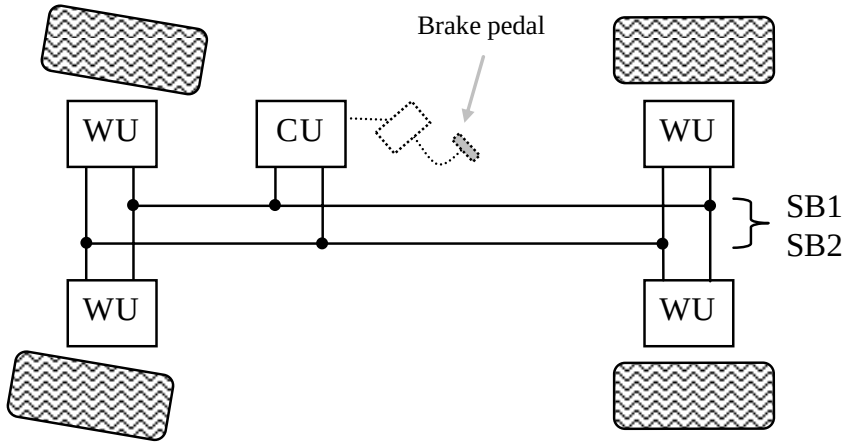
SHARPE

Symbolic Hierarchical Automated Reliability and Performance Evaluator

Provides a specification language and methods to solve reliability models such as:

- Reliability block diagrams
- Fault trees
- Markov chains

SHARPE

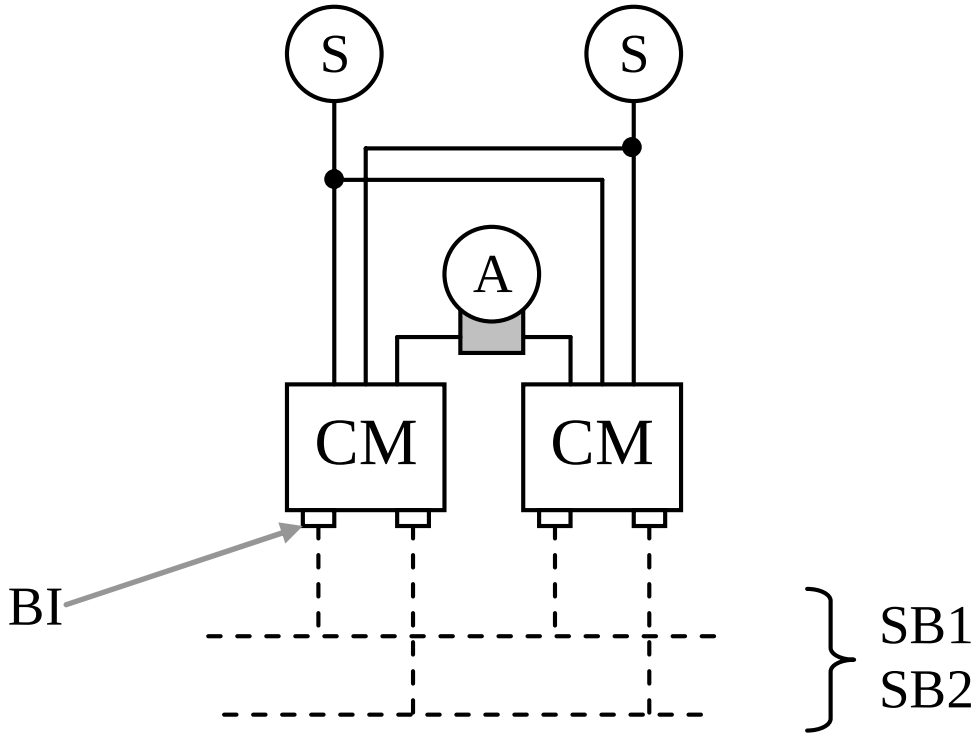


The following models are needed for the lab:

- Reliability block diagram of a wheel unit.
- Fault tree of the wheel unit subsystem.
- Markov models of the central unit for the two architectures.
- Fault tree of the entire system.

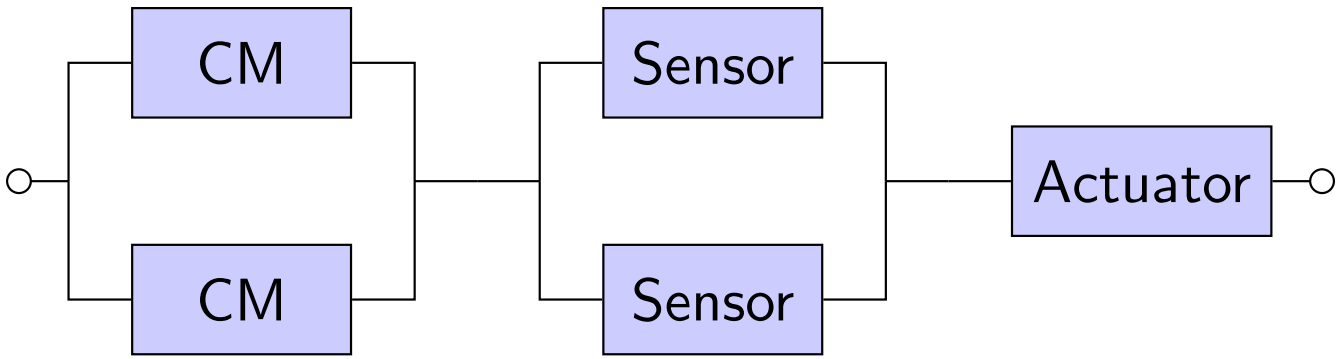
SHARPE – Reliability Block Diagrams

Reliability block diagram of a wheel unit?



SHARPE – Reliability Block Diagrams

Wheel unit:



SHARPE – Reliability Block Diagrams

```
1  bind
2     lambdaWU_CM  20e-6
3     lambdaWU_S  1e-6
4     lambdaWU_A  7e-7
5  end
```

SHARPE – Reliability Block Diagrams

```
1  bind
2     lambdaWU_CM  20e-6
3     lambdaWU_S  1e-6
4     lambdaWU_A  7e-7
5  end
6
7  * RBD of a wheel unit
8  block WU_distributed
```

SHARPE – Reliability Block Diagrams

```
 1  bind
 2     lambdaWU_CM  20e-6
 3     lambdaWU_S  1e-6
 4     lambdaWU_A  7e-7
 5  end
 6
 7  * RBD of a wheel unit
 8  block WU_distributed
 9     comp CM exp(lambdaWU_CM)
10     comp Sensor exp(lambdaWU_S)
11     comp Actuator exp(lambdaWU_A)
```

SHARPE – Reliability Block Diagrams

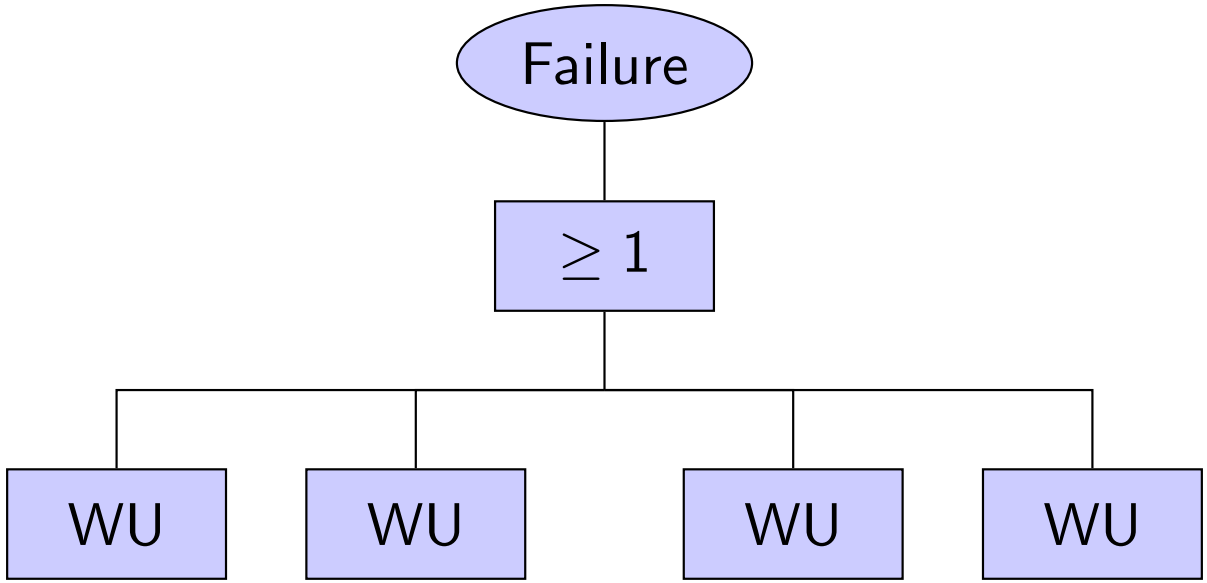
```
1  bind
2     lambdaWU_CM  20e-6
3     lambdaWU_S  1e-6
4     lambdaWU_A  7e-7
5  end
6
7  * RBD of a wheel unit
8  block WU_distributed
9     comp CM exp(lambdaWU_CM)
10    comp Sensor exp(lambdaWU_S)
11    comp Actuator exp(lambdaWU_A)
12
13    parallel twoCM CM CM
14    parallel twoSensors Sensor Sensor
15    series top twoCM twoSensors Actuator
16 end
```

SHARPE – Fault Trees

Wheel unit subsystem – Full functionality:

SHARPE – Fault Trees

Wheel unit subsystem – Full functionality:



SHARPE – Fault Trees

```
1  ftree WUfull_distributed
```

SHARPE – Fault Trees

```
1  ftree WUfull_distributed
2     basic WU cdf(WU_distributed)
```

SHARPE – Fault Trees

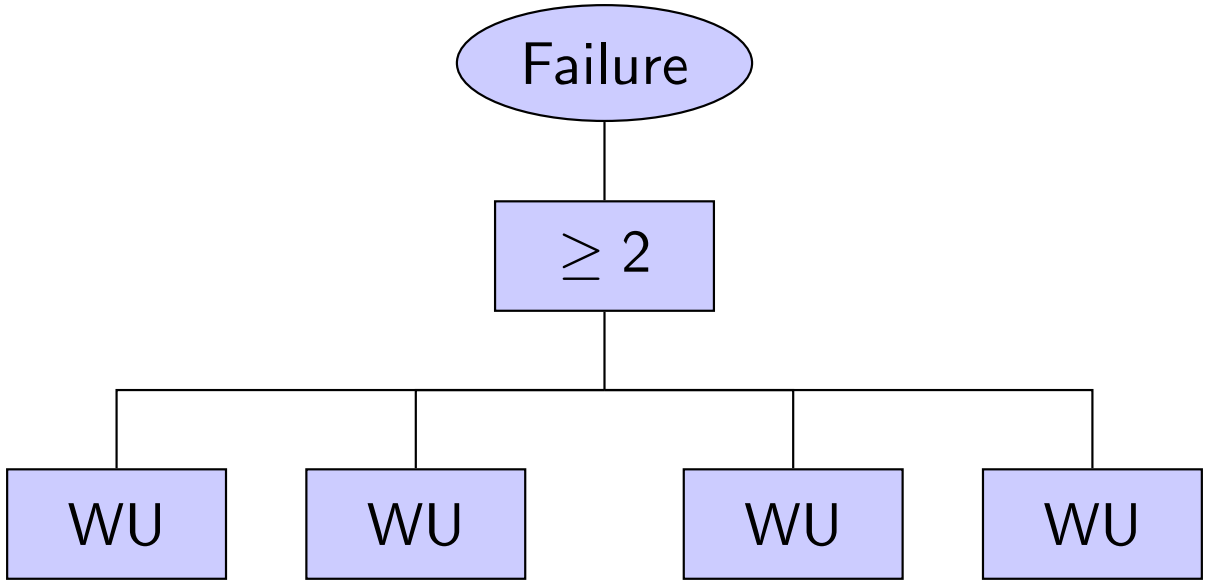
```
1  ftree WUfull_distributed
2     basic WU cdf(WU_distributed)
3     or top WU WU WU WU
4  end
```

SHARPE – Fault Trees

Wheel unit subsystem – Degraded functionality:

SHARPE – Fault Trees

Wheel unit subsystem – Degraded functionality:



SHARPE – Fault Trees

```
1  ftree WUdegraded_distributed
```

SHARPE – Fault Trees

```
1  ftree WUdegraded_distributed
2     basic WU cdf(WU_distributed)
```

SHARPE – Fault Trees

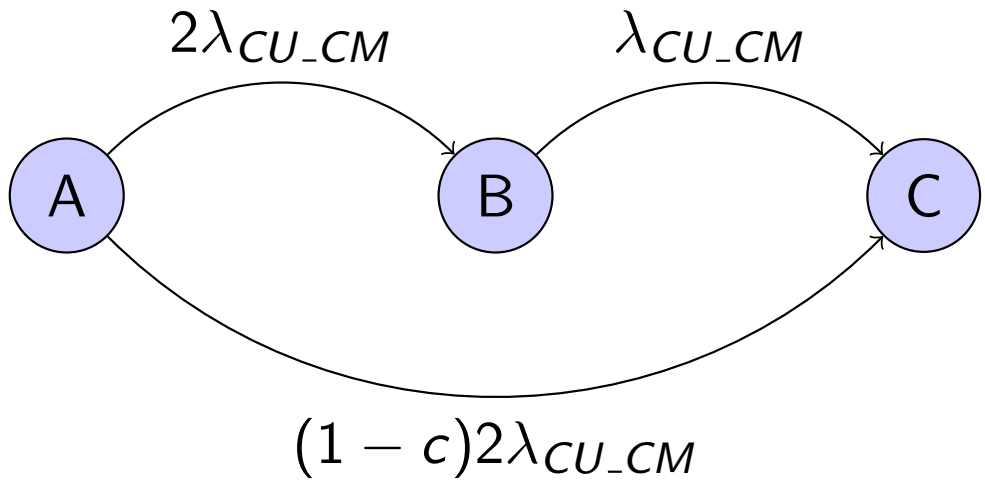
```
1  ftree WUdegraded_distributed
2     basic WU cdf(WU_distributed)
3     kofn top 2,4,WU
4  end
```

SHARPE – Markov Chains

Central Unit – Duplex:

SHARPE – Markov Chains

Central Unit – Duplex:



2 working 1 working Failure

SHARPE – Markov Chains

```
1  bind
2     lambdaCU_CM  8e-6
3     c  0.99
4  end
```

SHARPE – Markov Chains

```
1  bind
2     lambdaCU_CM  8e-6
3     c  0.99
4  end
5
6  markov  CU_duplex
```

SHARPE – Markov Chains

```
1  bind
2     lambdaCU_CM  8e-6
3     c  0.99
4  end
5
6  markov  CU_duplex
7     *  Transitions
8     A  B  2*lambdaCU_CM*c
```

SHARPE – Markov Chains

```
1  bind
2     lambdaCU_CM  8e-6
3     c  0.99
4  end
5
6  markov  CU_duplex
7     *  Transitions
8     A  B  2*lambdaCU_CM*c
9     B  C  lamdaCU_CM
```

SHARPE – Markov Chains

```
1  bind
2     lambdaCU_CM  8e-6
3     c  0.99
4  end
5
6  markov  CU_duplex
7     *  Transitions
8     A  B  2*lambdaCU_CM*c
9     B  C  lamdaCU_CM
10    A  C  2*(1-c)*lambdaCU_CM
11 end
```

SHARPE – Markov Chains

```
1  bind
2     lambdaCU_CM  8e-6
3     c  0.99
4  end
5
6  markov  CU_duplex
7     *  Transitions
8     A  B  2*lambdaCU_CM*c
9     B  C  lamdaCU_CM
10    A  C  2*(1-c)*lambdaCU_CM
11 end
12    *  Initial  probabilities
13    A  1.0
14    B  0.0
15    C  0.0
16 end
```

SHARPE – Calculate the Reliability

Reliability after 2 years for the wheel unit:

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
expr 1-value(17520; WU_distributed)
```

SHARPE – Calculate the Reliability

Reliability after 2 years for the wheel unit:

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
expr 1-value(17520; WU_distributed)
```

- Invoke SHARPE at the command prompt:

```
>sharpe.exe bbw_distributed.sharpe
1-value(17520; WU_distributed):    9.0123e-001
```

- $R(t = 2 \text{ years}) \approx 0.90$

SHARPE – Calculate the MTTF

MTTF for the entire system:

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
expr mean(system)/24/365
```

SHARPE – Calculate the MTTF

MTTF for the entire system:

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
expr mean(system)/24/365
```

- Invoke SHARPE at the command prompt:

```
>sharpe.exe bbw_distributed.sharpe
mean(BBWfulldup)/24/365:    3.0315e+000
```

- $MTTF = 3.0315$ years.

SHARPE – Plot the Reliability

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
* Start time, end time and increment time
eval (system) 0 8760 730
```

SHARPE – Plot the Reliability

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
* Start time, end time and increment time
eval (system) 0 8760 730
```

- Invoke SHARPE:

```
>sharpe.exe bbw_distributed.sharpe | eval_filter
  | complement > distributed.dat
```

SHARPE – Plot the Reliability

- In text file containing SHARPE models (e.g., bbw_distributed.sharpe):

```
* Start time, end time and increment time
eval (system) 0 8760 730
```

- Invoke SHARPE:

```
>sharpe.exe bbw_distributed.sharpe | eval_filter
  | complement > distributed.dat
```

- Start gnuplot

```
>wgnuplot.exe
gnuplot> plot "distributed.dat" with lines
```

Lab report

- Results shall be documented in a short technical report.
- The report must fulfill the requirements given in the lab pm.

Outline of the report

1. Introduction – Purpose of the report, background, and problem statement.
2. Overview of candidate architectures.
3. Description of models – Textual descriptions of each model, fault tree and Markov model.
4. Results – Report the results without any interpretation or discussion.
5. Discussion – pros and cons of the two designs, limitations, which design is the best?
6. Conclusion – Your recommendations based on the results.

Report requirements

- *Make sure the report fulfills all requirements in the lab-pm!*

- The report shall follow the outline given in the lab-pm.
- Copying text from other authors is **not allowed**.
  - However, you are allowed to copy figures from the lab-pm.
- Figures should be numbered and referred to in the text.
- The report shall contain references
  - For formatting, see reference list in paper by Kopetz and Bauer [**?**].
  - It is not allowed to use the lab-pm as a reference.

SHARPE Demonstration

Wheel Unit

```
1  bind
2     lambdaWU_CM  20e-6
3     lambdaWU_S       1e-6
4     lambdaWU_A       7e-7
5  end
6
7  block WU_distributed
8     comp CM exp(lambdaWU_CM)
9     comp sensor exp(lambdaS)
10    comp actuator exp(lambdaA)
11    parallel CMs CM CM
12    parallel sensors sensor sensor
13    series top CMs sensors actuator
14 end
```

Central Unit

```
1  bind
2     lambdaCU_CM        8e-6
3     CU_c        0.99
4  end
5
6  markov  CU_distributed
7     A  B  2*lambdaCU_CM*CU_c
8     B  C  lambdaCU_CM
9     A  C  (1-CU_c)*2*lambdaCU_CM
10 end
11    A  1.0
12 end
```

Printing the Results

```
1  * Reliability after 2 years
2  echo R(t = 2 years) for the CU
3  expr 1-value(17520; CU_distributed)
4
5  * MTTF
6  echo MTTF for the CU
7  expr mean(CU_distributed)/24/365
8
9  * Print R(t), 0 <= t <= 35040
10 eval (CU_distributed) 0 35040 730
```

Plotting with Gnuplot

```
gnuplot> cd 'h:\courses\ftcs\2008\...'
gnuplot> plot 'r_cu.dat' with lines
gnuplot> set terminal postscript color
Terminal type set to 'postscript'
Options are 'landscape noenhanced defaultplex \
   leveldefault color colortext \
   dashed dashlength 1.0 linewidth 1.0 butt \
   palfuncparam 2000,0.003 \
   "Helvetica" 14 '
gnuplot> set output 'r_cu.eps'
gnuplot> replot
```