SAM’04 Design Contest
Access Control System

Christian Webel
University of Kaiserslautern
webel@informatik.uni-kl.de
1. Requirement Analysis

Requirement Analysis (1)

Req. 1: „The door can be manually opened, so the lock controls access and the motor provides power assistance. As the motor is an auxiliary mechanism, the ACS does not rely on it working.“

Req. 2: „When the door is closed the access code can be entered at the console keypad. The door is allowed to be open (...) from the point the correct code is entered.“
2. System Design

System Architecture (1)

ACS and Environment

- console
- ACS
- door

- display
- keypad
- tone
- button
- solenoid
- oSwitch
- cSwitch
- motor

Christian Webel, University of Kaiserslautern

SAM’04 Design Contest
System Architecture (1)

ACS and Environment

- button
- display
- keypad
- toneGen
- solenoid
- oSwitch
- microSw
- cSwitch
- motor
- acs
System Architecture (2)
System Behaviour (1)

MSC S1_openDoor

keypad

keyPressed

(key)

keyPressed

(key)

door closed, no input

code checking...
code ok

acs

solenoid

motor

openSwitch

closeSwitch

KeyPressed

(key)

powerOff

openDoor

swClosed

on

swOpen

off
2. System Design

System Behaviour (2)
2. System Design

System Behaviour (2)

acs

- displayCtrl
- clock
- mainCtrl
- doorCtrl
- button

 ACS

- display
- keypad
- tone

- open(code)

- solenoid
- oSwitch
- cSwitch
- motor

Christian Webel, University of Kaiserslautern 9/28 3.6.04
2. System Design

System Behaviour (2)

- acs
- display
- keypad
- tone
- displayCtrl
- consoleCtrl
- mainCtrl
- clock
- doorCtrl
- button
- solenoid
- oSwitch
- cSwitch
- motor
- keypad
- tone
- display
- acs

Christian Webel, University of Kaiserslautern
10/28
3.6.04

SAM’04 Design Contest
2. System Design

System Behaviour (2)

acs

- display
- keypad
- tone

mainCtrl

- ok
- openFor(dur)

button

- doorCtrl

- displayCtrl
- clock

- consoleCtrl
- toneCtrl

- solenoid
- oSwitch
- cSwitch
- motor

Christian Webel, University of Kaiserslautern 11/28

SAM’04 Design Contest

3.6.04
2. System Design

System Behaviour (2)
2. System Design

System Behaviour (2)

![System Behaviour Diagram]

Christian Webel, University of Kaiserslautern

SAM’04 Design Contest
2. System Design

System Behaviour (2)
2. System Design

System Behaviour (2)

ACS

display

consoleCtrl

toneCtrl

mainCtrl

doorCtrl

button

display

keypad

tone

solenoid

oSwitch

cSwitch

motor

openDoor

Christian Webel, University of Kaiserslautern 16/28 3.6.04
System Design

System Behaviour (2)

acs

display

displayCtrl

consoleCtrl

toneCtrl

mainCtrl

doorCtrl

button

solenoid

doSwitch

cSwitch

motor

tone

keypad

Christian Webel, University of Kaiserslautern

17/28
3.6.04
System Design

System Behaviour (2)

acs

display

button

display

clock

doorCtrl

calendCtrl

consoleCtrl

toneCtrl

display

keypad

tone

motor

solenoid

cSwitch

oSwitch

powerOff

cSwitch
System Behaviour (2)

- **acs**
  - display
  - keypad
  - tone

- **displayCtrl**
- **clock**
- **mainCtrl**
- **doorCtrl**

- **button**
- **solenoid**
- **oSwitch**
- **cSwitch**
- **motor**

- **keyPressed (key)**
- **on/off**
- **swClosed**
- **swOpen**
- **openDoor/powerOff**
- **on**
- **open(code)**
- **openFor(dur)**
- **on/off**

Christian Webel, University of Kaiserslautern
19/28
3.6.04
Reuse (1): Concepts

• Design Pattern:
  
generic solution for recurring problems

• Design Component:
  
ready-to-use component which encapsulates a well-defined functionality

• Framework:
  
skeleton of a system that has to be adapted
Reuse (2): SDL-Design Pattern *SynchronousInquiry*

Req. 2: „(...) The door is allowed to be open (...) from the point the correct code is entered.“
3. Reuse

Reuse (2): SDL-Design Pattern *SynchronousInquiry*

**Req. 2:** "(...) The door is allowed to be open (...) from the point the correct code is entered."”

MSC S3b_checkCode

- keypad
  - keyPressed
    - (key)
    - keyPressed
      - (key)
    - keyPressed
  - keyPressed
    - open
      - (code)
    - keyPressed
      - (key)
- keyPressed
  - wait4response
  - check(code) = true
    - ok
  - code checking ok
    - opening door

process mainCtrl

- closedSimple
  - open(code)
    - via
      - ANY_GATE
    - CALL check(code)
      - true
        - ok
          - via
            - door
          - ok
            - via
              - secure
            - ok
              - open
Reuse (3): SDL-Design Pattern *SynchronousInquiry*

**process consoleCtrl**

```
ready

keyPressed(key)
  VIA ANY OATE

(length)

4

open

wait4Open

OK
  VIA ANY OATE

risk
  VIA ANY OATE

button
  VIA ANY OATE

keyPressed(Up)
  VIA ANY OATE

repeat
  VIA ANY OATE

displayCodeOk
  VIA dis

displayCodeNck
  VIA dis

displayButton
  VIA dis

warning
  VIA ton

open

ready
```
3. Reuse

Reuse (4): Design-Component *Clock*

**Req 8:** „The console display shows (...) the time, which is always visible ...“
Demo of the System and the GUI (1)
Demo of the System and the GUI (2)
Summary

• systematical development
• ALL requirements met (including the optional ones)
• designed using SDL+ and SAFIRE
• reuse (SDL-Design Pattern and Design Components)
• GUI
• Testsuite
• about 1600 LoC (GUI additional 700 LoC)
Summary

• systematical development
• ALL requirements met (including the optional ones)
• designed using SDL+ and SAFIRE
• reuse (SDL-Design Pattern and Design Components)
• GUI
• Testsuite
• about 1600 LoC (GUI additional 700 LoC)

THANK YOU FOR YOUR ATTENTION.