CSC 7003 : Basics of Software Engineering

J Paul Gibson, D311

paul.gibson@telecom-sudparis.eu

http://www-public.telecom-sudparis.eu/~gibson/Teaching/CSC7003/

Lift Requirements Sample Solution

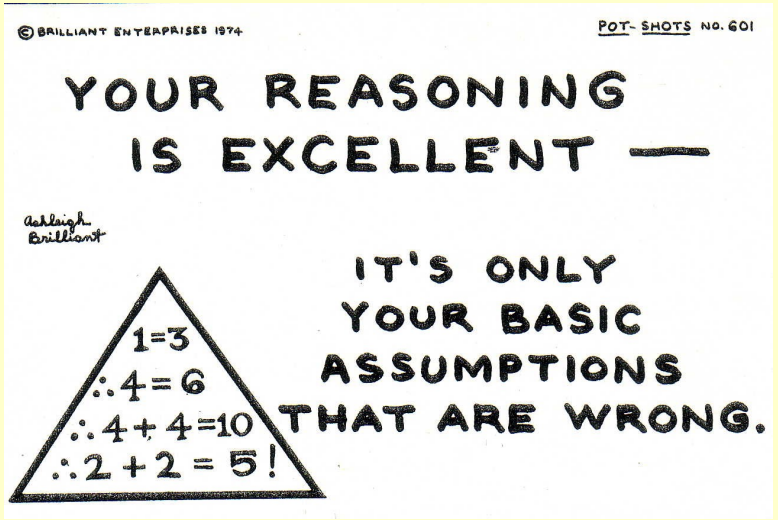
/~gibson/Teaching/CSC7003/L3-Requirements-SampleSolution.pdf

Elements of a requirements document

Assumptions
Legal Obligations
Standards
System Parameters
System Interface
Operational /Behavioural Semantics
Safety Issues
Liveness/Fairness/Performance Issues – QoS
Options and Configuration
Validation Tests – Use Cases

For contractual requirements – cost/time/maintenance/support

Requirements For A Generic Lift System



Assumptions

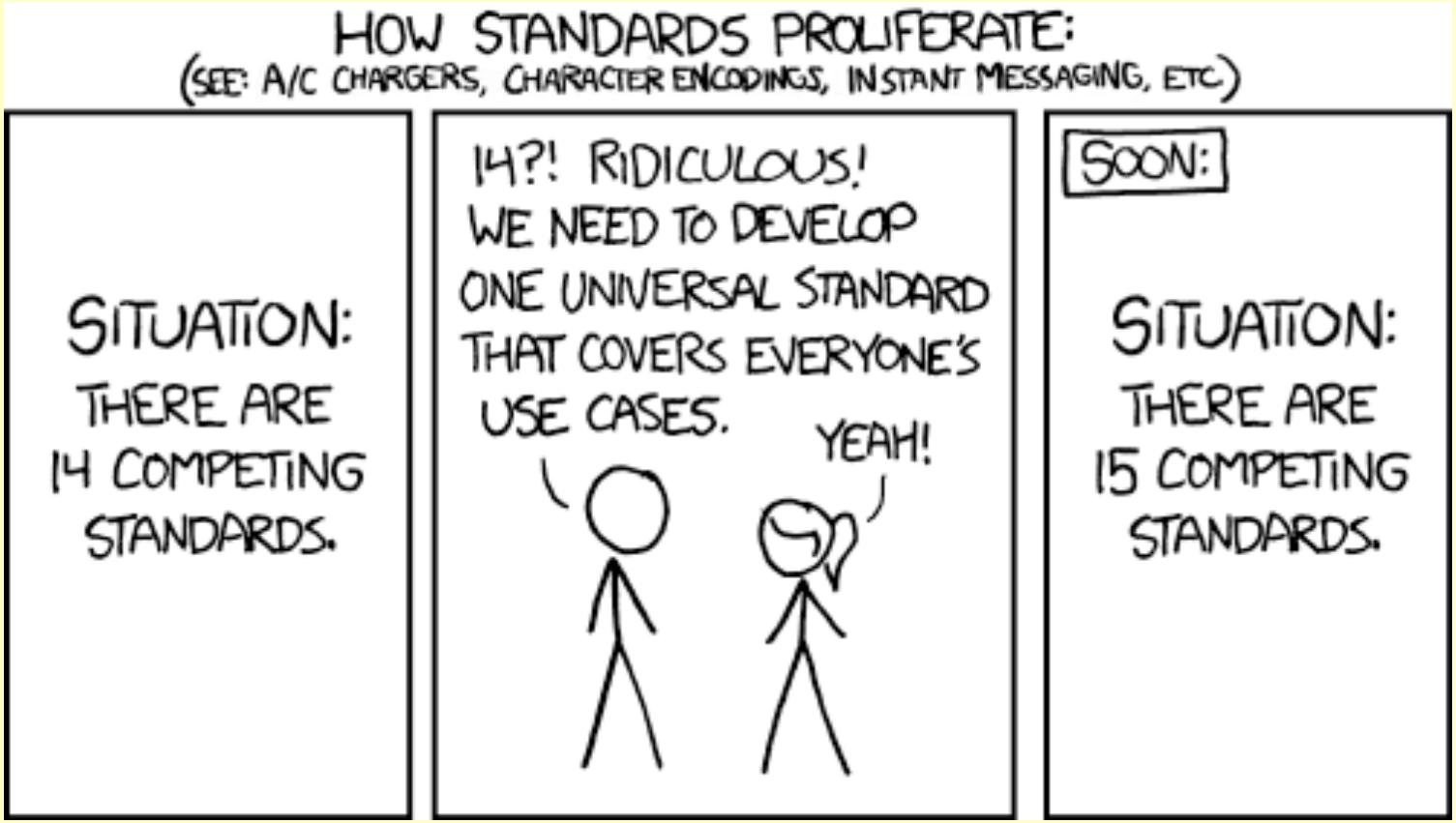
The elevator is to be used by humans

The elevator provides a safe physical environment for the users:

- light
- heat,
- air,
- no unnecessary risk of injury,
- etc…

Requirements For A Generic Lift System

Legal Obligations & Standards



Requirements For A Generic Lift System

System Parameters

- *Bottom* = lowest floor
- *Top* = highest floor
  - *NumberOfFloors* = 1+Top-Bottom (a derived parameter)
- *NumberLifts* = number of lifts operating between *Bottom* and *Top* and available at same physical location at each floor

Requirements For A Generic Lift System

System Interface

Users can request the lift at a floor by specifying direction they wish to travel (towards Bottom or towards Top)

Users can specify the floor they wish to go to inside the lift (All choices between and including Bottom to Top)

Users can block the doors at the floor at which the lift is resting

A lift can be put in and out of an emergency state

Requirements For A Generic Lift System

Operational – Behavioural Semantics : Lift State Changes

A lift can move up (towards Top) or down (towards Bottom)

A lift can rest/stop at a floor (not moving)

A lift can open/close inner doors of the lift (except when blocked)

A lift can open/close outer doors of a floor (except when blocked)

Requirements For A Generic Lift System

Safety Rules

Doors can be open only when the lift is resting at the floor.

A lift can move only if all doors are closed

When in emergency state the lift must immediately stop at nearest floor, open doors and not move

Requirements For A Generic Lift System

Liveness Property: *In general operation, every user request will eventually be serviced (if it is not prohibited by emergencies and/or blocked doors)*

Requirements For A Generic Lift System

Quality of Service Rules – refine the liveness property

For every user, a lift can pass a user at a floor without stopping to pick them up at most one time.

For every user, a lift can change direction at most two times between the time that a user has requested the lift (at a floor) and the time that the lift stops at the floor to collect the user.

When there are user requests (inside the lift or at a floor) then the lift is obliged to move to service the requests (except in the emergency state)

When there are no user requests, the lift saves energy by not moving.

Requirements For A Generic Lift System

Performance Requirements

We need to consider physical requirements with respect to time, and time-dependent properties like speed, acceleration, etc…

Depending on the system being modelled, we may have to consider real-time issues: relative/absolute time, global clocks?, synchronous or asynchronous communication, granularity/error, etc…

QUESTION: can we abstract away from real-time issues in our requirements model/specification?

NOTE: Performance is also related to safety: "the lift should go as fast as it can without breaking the safety constraints (for speed and acceleration)". "The lift should never stop too abruptly"

We will return to these when we work on the design phase of the problem

Requirements For A Generic Lift System

Configuration Options

Client can specify :

- No doors
- No emergency service
- Weight limit detector
- Different types of user feedback (on floor and in lift)
- Movement protocols (plug and play controllers):
  - Energy efficiency
  - Priority (eg for floor with restaurant at meal times)
- Specific user input devices (eg buttons, touch screens, etc..)
- Security and access rules (eg for locking certain floors)

The lift is a good case study for understanding the problems in building a software product line (SPL)

Requirements For A Generic Lift System

Validation – use cases

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (known in Unified Modelling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.

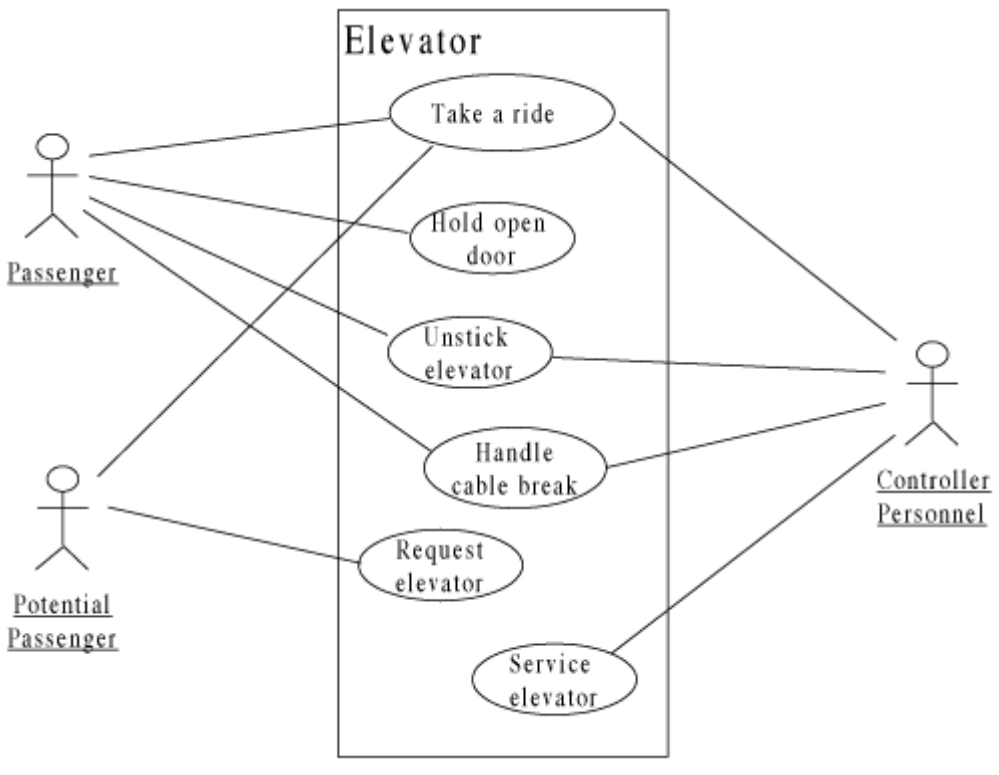
http://en.wikipedia.org/wiki/Use_case

"To my knowledge, no other software engineering language construct as significant as use cases has been adopted so quickly and so widely among practitioners. I believe this is because use cases play a role in so many different aspects of software engineering."

Ivor Jacobson

Requirements For A Generic Lift System

Validation – use cases



http://www.ece.ufrgs.br/~cpereira/temporeal_pos/www/chap2-3.html

Requirements For A Generic Lift System

Validation – use cases

An elevator is a fairly simple device, but even a use case as simple as "Take a ride" can produce many scenarios:

- *A suitable elevator is already on the floor*
- *An elevator is available but must travel to the floor*
- *A suitable elevator exists, but has a pending request*
  - *pending request must be handled first*
  - *pending request should be handled after picking up passenger*
- *No suitable elevator exists, request must pend, waiting for next available elevator*
  - *Passenger engages Stop on the Stop-Run switch as elevator is on the way to pick up the passenger*
  - *Picked up passenger engages Stop en route to selected floor*
  - *Passenger selects floor not in the direction originally selected (request must be serviced last)*

http://www.ece.ufrgs.br/~cpereira/temporeal_pos/www/chap2-3.html

Requirements For A Generic Lift System

Validation – use cases

Further reading

Jacobson, Ivar. "Use cases–Yesterday, today, and tomorrow." *Software and Systems Modeling* 3 (2004): 210-220.

Cockburn, Alistair. "Structuring Use Cases with Goals." (1997).

Cockburn, Alistair. *Writing effective use cases*. Vol. 1. Reading: Addison-Wesley, 2001.

Requirements For A Generic Lift System

Further Reading – control algorithms need to be tested/ validated (through simulation of users)

Barto, A. G., and R. H. Crites. "Improving elevator performance using reinforcement learning." *Advances in neural information processing systems* 8 (1996): 1017-1023.

Pepyne, David L., and Christos G. Cassandras. "Optimal dispatching control for elevator systems during uppeak traffic." *Control Systems Technology, IEEE Transactions on* 5.6 (1997): 629-643.

Cortés, Pablo, Juan Larrañeta, and Luis Onieva. "Genetic algorithm for controllers in elevator groups: analysis and simulation during lunchpeak traffic." *Applied Soft Computing* 4.2 (2004): 159-174.

Requirements For A Generic Lift System

How to move from requirements to design?

We consider – as a first step – the core functionality for moving the lift in order to service requests

We need to structure the design as a bridge between the requirements and the implementation

QUESTION: What do you know about software design?

TO DO: Produce a design for the elevator controller targeted at an implementation language of your choice.

Lets Try Out Some Design With Our Lift/Elevator Problem