Addressing Degraded Service Outcomes and Exceptional Modes of Operation in Behavioural Models

Sadaf Mustafiz, Jörg Kienzle

School of Computer Science
McGill University, Montreal, Canada

{sadaf, joerg} @cs.mcgill.ca

Andrey Berlizev

Laboratory of Advanced Software Systems
University of Luxembourg, Luxembourg

andrey.berlizev@uni.lu
Motivation

• Dependability not usually addressed in current mainstream software development methods
• Lack of methods for elicitation and specification of degraded service outcomes and modes of operation
• Not addressed in current modelling formalisms
• Necessary to identify, specify, and analyze dependability concerns during the early stages of software development
• Need to precisely define system behaviour in exceptional situations
• Need to satisfy users and maintain system safety and reliability
Overview

• Background
• Degraded service outcomes
• Modes of operation
• Requirements Engineering and DREP
• Modelling degraded outcomes
• Modelling modes
• Related work
• Conclusion
Dependability

- **Dependability** is that property of a computer system such that trust can justifiably be placed on the service it delivers\(^1\).
  - Reliability - *aptitude to provide continuity of service*
  - Safety - *lack of catastrophic failures*

---

Degraded Service Outcomes

- System providing a set of services
- Exceptional situation does not allow the system to complete the task at hand
- Discuss with stakeholders
- Handle current situation
- Provide partial or degraded service
- Provision of degraded outcome better than complete service failure
  - success, failure, degraded success
Exceptional Modes of Operation

- Provision of some services not possible due to exceptional situation
- Do not offer services that cannot be provided
- Mode determines the set of services currently offered
- Switch into a different mode
Requirements Engineering

• Requirements Development
  • discovery and elicitation
  • definition and specification
  • analysis of the requirements
  • system validation against the requirements

• Dependability-Focused Requirements Engineering Process (DREP)
Model-based DREP

Use Case: ...
Dep. Exp.: 99.999%

Standard Use Cases + Dependability Expectations

Use Case: ...
Primary Actor: ...
Secondary Actors: ...
Dep Exp.: 99.999%
Main:
1. ... Rel.: 99.83%
2. ... Rel.: 99.91%
   Safety: L2
3. ... Rel.: 99.94%
Extensions:
1a. Exception {...}
1b. Exception {...}
3a. Exception {...}

Exceptional Use Cases

Activity Diagrams

DA-Charts

Markov Chains

Exceptional Use Case Diagram

Exceptions

Modes

+ Exception Table
+ Mode Table
Dependability in Use Cases

• Use Cases capture interactions between the system and the environment to achieve user goals

• Error detection - detection of exceptional situations by means of secondary actors (e.g., sensors) and timeouts
  • situation that threatens the successful completion of the user goal (reliability)
  • situation puts user in danger (safety)

• System recovery - describing interactions with the environment required to
  • continue to deliver the current service (reliability)
  • offer a degraded service (reliability)
  • take actions that prevent a catastrophe (safety)
Elevator Control System Case Study
Elevator Arrival Exceptional Use Case (1)

**Use Case:** ElevatorArrival

**Intention:** System wants to move the elevator to the User's destination floor.

**Main Success Scenario:**

1. System asks Motor to start moving towards the destination floor.
2. Movement Sensor informs System that cabin is moving.
3. Approaching Floor Sensor informs System that the cabin is approaching destination floor.
4. System requests Motor to stop.
5. Floor Sensor informs System that elevator is stopped.
6. System requests Door to open.
7. Door Sensor informs System that door is open.

Use case ends in **<< success >>** ReachedDestination.
Elevator Arrival Exceptional Use Case (2)

**Extensions:**

2a. Exception {MotorStartFailure}
   2a.1 Use case continues at step 1.
      2a.1.a Retried 3 times.
         Use case ends in **<< failure >>** MotorFailure.

3a. Exception{MissedFloor}
   3a.1 System chooses to approach a neighboring floor. Use case continues in **<< degraded >>** DifferentFloor at step 1.

5a. Exception{MotorFailure}
   5a.1 Use case ends in **<< failure >>** MotorFailure.

7a. Exception{DoorStuckClosed}
   7a.1 Use case continues at step 6.
      7a.1a Retried 3 times.
         Use case ends in **<< failure >>** DoorStuckClosed.
Outcomes in the EA Use Case

• Success
  • Reached Destination

• Degraded success
  • Different Floor

• Failure
  • Motor Failure
  • Door Stuck Closed
Activity Diagrams

• Models workflow behaviour
• Emphasizes the sequence and conditions
• Models activities
  • Actions
  • Sending or receiving messages
  • Control flow
  • Object flow
• Hierarchical structuring
Outcomes in Activity Diagrams (1)
Outcomes in Activity Diagrams (1)
Outcomes in Activity Diagrams (2)
Statecharts

- Modelling of reactive system behaviour
- Used for documentation, analysis, and simulation
- DA-Charts
  - probabilistic extension of statecharts
  - model-driven assessment for use cases
  - tool support to compute the probability of ending in a safe state or of completing a goal
Outcomes in DA-Charts

Sequencing according to use case, goalSuccess/goalFailure states
Fault-free - no probabilities

System State

Actors can fail with certain probability!

Outcome State

Safety State
Outcomes in DA-Charts

Sequencing according to use case, goalSuccess/goalFailure states
Fault-free - no probabilities

System State

Safety State

Actors can fail with certain probability!
Exceptional Modes

• Normal mode
  • services available and correctly functioning

• Degraded mode
  • offers limited services
  • degraded QoS provided

• Emergency mode
  • normal services suspended
  • emergency services offered

• Restricted mode
  • subset of normal services offered
  • emergency services available
Modelling Modes

- **Normal Modes**
  - **Standard Mode**
    - at 7am
    - at 3pm
    - do/TakeElevator
    - do/ReserveElevator
    - do/UserEmergency
  - **Rush Hour Mode**
    - at 10am
    - at 6pm
    - do/TakeElevator
    - do/UserEmergency

- **Emergency Mode**
  - do/ReturnToGroundFloor
  - <<exception>>
  - EmergencyOverride

- Resume Normal Operation
## Mode Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Description</th>
<th>Expected Reliability</th>
<th>Expected Safety</th>
<th>Other Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Task-based DREP

Elicitation and Discovery

Definition and Specification

Dependability Analysis

Dependability-Based Refinements

Requirements Summary
Related Work

• CORA
  • proposed by Berlizev and Guelfi
  • analysis method for embedded system modelling
  • activity model, domain model, operation model
  • degraded service outcomes addressed in activity models

• Work on degraded modes of operation proposed by Srivasta et al, Shea and Johnson, Lygeros et al.
Conclusion

• Introduced concepts of degraded service outcomes and exceptional modes of operation
• Have shown how to model the concepts in different formalisms
  • use cases, activity diagrams, statecharts
• Integration in requirements engineering
  • Illustrated by incorporating concepts in DREP
  • Elicitation and specification of outcomes
    • exceptional use cases to elicit
    • exceptional activity diagrams to specify
    • DA-Charts to analyze
  • Elicitation and specification of modes
• Leads to a complete requirements specification document
References (1)


References (2)


Thank you!

??? Questions ???

Contact: sadaf@cs.mcgill.ca