Design Test Process in Component-Based Software Engineering: An Analysis of Requirements Scalability

Mariem Haoues, Asma Sellami and Hanêne Ben-Abdallah

FSEG, University of Sfax, Tunisia
ISIMS, University of Sfax, Tunisia
King Abdulaziz University, KSA
Outline

- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective
Problematic

- Building software that will satisfy Functional User Requirements (FUR) within the quality, time, and budget constraint is a challenge
- The complexity of software & the instability of FUR may cause
  - High development costs
  - Risk of software failures
- Researchers agree that Component Based Software Engineering (CBSE)
  - Cost reduction
  - Faster development
  - Software extension more easier (FUR Scalability)
  - Higher cost of testing phase and error detection
Objectives

- Improve the CBD model to decrease the effort at testing phase
- Propose a design test process: component testing, integration testing, and system testing
  - COSMIC Functional Size Measurement
  - Traceability matrix
- The proposed design test is illustrated using Communication Diagram (UML-CM) and Use Case Diagram (UML-UC)
COSMIC-ISO 19761

- COSMIC Functional Size Measurement –V 3.0.1

- Allow the quantification of any type of software (business, real-time, embedded, …) from user’s point of view
- Independent of any quality or technical criteria
- Free on the web: http://www.cosmicon.com/
CBD Model

Outline:
- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective

- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective
Software is decomposed into components (C1, C2, C3, and C4)
- A component encapsulates one or more functionalities

Outline:
- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective
A New CBD Model

Outline:
- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective

Diagram:
- FUR
  - Requirements Definition
  - Storage Components
- Design
  - Components Definition
  - Components Selection
  - Components created/selected
- Implementation
  - Validation test
- Deployment
  - Test
    - Component testing
    - System testing
    - Integration testing
Design Testing Process

- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective
Mapping COSMIC on UML-CM

Outline:
- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective

![Diagram of Mapping COSMIC on UML-CM](image-url)
Measuring the UML-CM

- The functional size of an UML-CM depends on the number of messages exchanged between:
  - Actors and objects
  - Objects and objects

\[
FSM \ (UML - CM ) = FSM \ (event ) + \sum_{i=1}^{n} FSM \ (m) + \sum_{j=1}^{m} FSM \ (cond)
\]  

- FSM (UML-CM): functional size of the UML-CM
- FSM (event): FSM of the event (1 CFP)
- FSM (m): the functional size of a message (1 CFP)
- \( n \): the number of messages
- FSM (cond): the functional size of a condition (1 CFP)
- \( m \): the number of conditions
# Traceability Matrix

<table>
<thead>
<tr>
<th>UML-UC elements</th>
<th>UML-CM elements</th>
<th>Name: Number</th>
<th>Actor</th>
<th>Message</th>
<th>Event</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Seq</td>
<td>Cond</td>
<td>Res</td>
</tr>
<tr>
<td>Name: Number</td>
<td></td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actor</td>
<td></td>
<td>Primary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Secondary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td></td>
<td>Pre-cond</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Post-cond</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td>Include: use case</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extend: use case</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic flow</td>
<td></td>
<td>Number</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td></td>
<td>Pre-condition</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td></td>
<td>Actor</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative flow</td>
<td></td>
<td>Number</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td></td>
<td>Pre-condition</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td></td>
<td>Actor</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td></td>
<td>Number</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative flow</td>
<td></td>
<td>Actor</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td></td>
<td>Number</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative flow</td>
<td></td>
<td>Actor</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td></td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-functional requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The UML-UC diagram includes four UC
- View alarms
- View monitoring data
- Generate alarm
- Generate monitoring data

Every UC is detailed by a textual description

Every UC is represented by a UML-CM diagram

Use case <View Alarms>
Actors: <Monitoring Operator>
Pre-condition: <The monitoring operator is logged in>
NS /* Nominal scenario */
Begin
<1> <Monitoring Operator> <The monitoring operator requests to view the outstanding alarms>
<2> <System> <The system displays the outstanding alarms. For each alarm, the system displays the name of the alarm, alarm description, location of alarm, and severity of alarm (high, medium, low)>
End
AS /* Alternative scenario */
Begin <Emergency situation, begin at 2>
<2.1> <System> <System displays emergency warning message to operator>
End

Component Testing

FSM of UML-CM “View Alarms”

\[
FSM_{(UML-CM)} = FSM_{(view)} + \sum_{i=1}^{n} FSM_{(FP)} + \sum_{j=1}^{m} FSM_{(cond)}
\]

\[
1 \leq FSM_{CM_{(View Alarms)}} \leq FSM_{MaxUC_{(View Alarms)}}
\]

\[
1 \leq 4 \text{ CFP} \leq 5 \text{ CFP}
\]

- The absence of intra-component errors is verified
Integration Testing

FSM

- We suppose that “View Alarms” is the new component to be integrated
- Verify the possibility of integrated “View Alarms” component
  - Verify the FSM for UML-UC (FSM_{base} ("View Alarms")) and (FSM_{CM} ("View Alarms"), FSM_{CM} ("View monitoring data"), FSM_{CM} ("Generate alarm"), and FSM_{CM} ("Generate monitoring data"))
  - Check the compatibility using traceability matrix

<table>
<thead>
<tr>
<th>Component</th>
<th>FSM results</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;View Alarms&quot;</td>
<td>1 CFP ≤ 4 CFP ≤ 5 CFP</td>
</tr>
<tr>
<td>&quot;View Monitoring Data&quot;</td>
<td>1 CFP ≤ 4 CFP ≤ 5 CFP</td>
</tr>
<tr>
<td>&quot;Generate Alarm&quot;</td>
<td>1 CFP ≤ 4 CFP ≤ 5 CFP</td>
</tr>
<tr>
<td>&quot;Generate Monitoring Status&quot;</td>
<td>1 CFP ≤ 4 CFP ≤ 5 CFP</td>
</tr>
</tbody>
</table>
# Integration Testing

Traceability Matrix (1st level)

**Outline:**
- Introduction
- CBD Model
- Design Testing Process
  - Illustrative Example
- Conclusion & Perspective

<table>
<thead>
<tr>
<th>UML-UC</th>
<th>UMC-CM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: View Alarms</td>
<td>&quot;View Monitoring data&quot;</td>
</tr>
<tr>
<td>Pre-cond: The monitoring operator is logged in.</td>
<td></td>
</tr>
<tr>
<td>Actor: Monitoring Operator</td>
<td></td>
</tr>
<tr>
<td>Action: Description</td>
<td>The monitoring operator requests to view the outstanding alarms.</td>
</tr>
<tr>
<td></td>
<td>The system displays the outstanding alarms.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UML-UC</th>
<th>UMC-CM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: View Alarms</td>
<td>&quot;Generate Alarms&quot;</td>
</tr>
</tbody>
</table>

---

**Integration Testing**

**Traceability Matrix (1st level)**
### Integration Testing

#### Traceability Matrix (detailed level)

<table>
<thead>
<tr>
<th>UML-CM &quot;View Alarms&quot;</th>
<th>Name: Number &quot;View Alarms&quot;: 1</th>
<th>Operator Request</th>
<th>Alarm Request</th>
<th>Alarm</th>
<th>Display Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: Number</td>
<td>Monitoring Operator</td>
<td>Operator</td>
<td>S1</td>
<td>S.1.2</td>
<td>S.2.1 Display Info</td>
</tr>
<tr>
<td>Actor</td>
<td></td>
<td>Request</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Pre-condition

**Actor:** Monitoring Operator

**Description:** The monitoring operator is logged in.

#### Basic flow

**Action**

**Number 1**

**Pre-condition**

**Actor:** Monitoring Operator

**Description:** The Monitoring Operator requests to view the outstanding alarms.

#### Alternative flow

**Action**

**Number 2.1**

**Pre-condition**

**Actor:** Monitoring Operator

**Description:** The system displays the outstanding alarms. For each alarm, the system displays the name of the alarm, alarm description, location of alarm, and severity of alarm (high, medium, low).

**Alternative flow**

**Action**

**Number 2.1**

**Pre-condition**

**Actor:** Monitoring Operator

**Description:** Emergency situation. System displays emergency warning message to operator.
System Testing

- System testing is used to verify that the system satisfy all FUR
- Check the correspondance between FSM (UML-CM) and FSM (UML-UC)

<table>
<thead>
<tr>
<th>Use Case</th>
<th>FSM using UML-UC</th>
<th>FSM using UML-CM</th>
<th>Formula (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Alarms</td>
<td>$1 \leq FSM_{base} \leq 5$</td>
<td>4 CFP</td>
<td>$1 \leq 4 \leq 5$</td>
</tr>
<tr>
<td>View Monitoring Data</td>
<td>$1 \leq FSM_{base} \leq 5$</td>
<td>4 CFP</td>
<td>$1 \leq 4 \leq 5$</td>
</tr>
<tr>
<td>Generate Alarm</td>
<td>$1 \leq FSM_{base} \leq 8$</td>
<td>4 CFP</td>
<td>$1 \leq 4 \leq 8$</td>
</tr>
<tr>
<td>Generate Monitoring Status</td>
<td>$1 \leq FSM_{base} \leq 6$</td>
<td>4 CFP</td>
<td>$1 \leq 4 \leq 6$</td>
</tr>
</tbody>
</table>
Conclusion & Perspective

- Introduction
- CBD Model
- Design Testing Process
- Illustrative Example
- Conclusion & Perspective
**Conclusion**

- Propose a design test process in CBSE
  - Component testing for intra-component errors detection using FSM
  - Integration testing for inter-component errors detection using FSM and traceability matrix
  - System testing for system defects detection using FSM

- A measurement interval between UML-UC and UML-CM is proposed
  - Identify modeling errors
  - Ensure the compatibility between these diagrams in terms of CFP

- Illustration through the “Emergency Monitoring System” case study
Perspective

- Further works
  - Apply the design test process on some real-word case studies to ensure its efficiency
  - Analysis of the FUR impact change in terms of functional size
  - Implement the proposed design test process
Thank you!

Mariem Haoues
Asma Sellami
&
Hanène Ben-Abdallah

e-mail:
mariem_haoues@yahoo.fr
asma.sellami@isimsf.rnu.tn
hbenabdallah@kau.edu.sa
Questions?