Semantic-Based Interaction Detection in Aspect-Oriented Scenarios

Gunter Mussbacher\textsuperscript{1}, Jon Whittle\textsuperscript{2}, Daniel Amyot\textsuperscript{1}

\textsuperscript{1}SITE, University of Ottawa, Canada
\textsuperscript{2}Dept. of Computing, Lancaster University, UK

September 4, 2009
Background: Aspect-oriented Modeling

• Aspects address the problem of one concern **crosscutting** other concerns in a system or model
• Aspects can encapsulate concerns even if they are crosscutting

Without Aspects

Concern A  Concern B  Concern C

- Scattering
- Tangling

With Aspects

Concern A

Concern B

Concern C

Aspect1

Aspect2

Aspect3

... 3 Crosscutting Concerns (Aspect1, Aspect2, Aspect3)

(Each aspect contains a **composition rule** illustrated by the arrows that defines where to add the aspect)

**Abstraction**

**With Aspects**

**Modular Reasoning**

**Compositional Reasoning**

**Aspectual Properties**
Background: AO Requirements Engineering

• Improved support for separation of crosscutting functional and non-functional properties during requirements engineering

• Establish critical trade-offs even before the architecture is derived

• Improved understanding of the problem and ability to reason about it

• Today’s focus is on scenario models
Motivation (1)

• Aspect Interaction Problem
  • Multiple aspects may be applicable at a given point in the base model

• In the best case, aspects may simply be ordered
  • E.g., an aspect may assume certain modeling elements in the base
    are introduced by another aspect

• In the worst case, there may be deep semantic conflicts
  • E.g., inherent trade-offs between two non-functional aspects such as
    security and performance
    • Security mechanisms must be enforced → performance impact
    • Performance aspect may cache results → security implications
Motivation (2)

- Our approach to address semantic interactions
  - Lightweight *semantic annotations* of aspect models
  - Model the semantic impact of aspects on each other in a *goal model* called an *influence model*

- Identify and trade-off semantic aspect interactions with the *influence model*
- Reason about stakeholder needs and aspect interactions with the help of built-in qualitative or quantitative evaluation mechanisms applied to the *influence model*
Table of Contents

- Example: Electronic Voting Machine
  - Reporting Use Case (Base Model)
  - Aspects
  - Composed Model

- Goal-oriented Requirement Language (GRL)
- Goal Model for Electronic Voting Machine

- Summary of our Approach

- Case Study

- Conclusion and Future Work
Electronic Voting Machine: Reporting Use Case

- Poll Official
- :Voting Machine
- : Backend Server

- presentOptions
- selectReport
- saveResults
Electronic Voting Machine: Authentication Aspect

Poll Official

:Voting Machine

displayLogin

enterPIN

insertSmartCard

authenticate

<<confidential>>

: Authentication Server

presentOptions

alt

[fail]

[success]

any

takeSmartCard

ejectSmartCard
Electronic Voting Machine: Remote Service Aspect
Electronic Voting Machine: Caching Aspect

```
<<cache>>

: |Machine

|checkCache

|alt

|found

|not found

|message

|saveToCache

<<remote>>

: |Remote Server

Intro
Base Model
Aspects
Composed Model
GRL
Goal Model
Evaluation
Summary
Case Study
Concl./F. Work

Semantic-Based Interaction Detection in Aspect-Oriented Scenarios. RE 2009. © 2009 Gunther Mussbacher, Jon Whittle, Daniel Amyot
Electronic Voting Machine: Composed Model

- **Poll Official**:
  - insertSmartCard
  - displayLogin
  - enterPIN

- **:Voting Machine**:
  - authenticate

- **:Backend Server**

- **:Authentication Server**

- **Remote Service**

**Reporting Use Case**
- Authentication
- Remote Service
- Caching

---

Semantic-Based Interaction Detection in Aspect-Oriented Scenarios. RE 2009. © 2009 Gunter Mussbacher, Jon Whittle, Daniel Amyot
Goal-oriented Requirement Language (GRL)

• GRL is integrated with Use Case Maps (UCM), a scenario notation, in the User Requirements Notation (URN)
  • URN is the first and currently only standard which explicitly addresses goals in addition to scenarios in a graphical way in one unified language (International Telecommunication Union, ITU-T Z.150 series)

• GRL is based on i* (concepts / syntax) and the NFR Framework (evaluation mechanism)
  • Ideally suited to capture qualitative relationships (as required by the influence model)
  • Reason about stakeholder needs and aspect interactions with the help of qualitative or quantitative evaluation mechanisms that are applied to the influence model
Electronic Voting Machine: Goal Model

Goal (intermediate node for combining semantic markers)

Softgoal (for NFR addressed by aspect)

Contribution (for impact of semantic marker on its own aspect’s NFR)

Remote Service

Consistency

Confidentiality

Remote Server

Local Server

Authentication

Caching

Encryption

Decomposition

Task (for semantic markers)

Performance

Correlation (for impact of semantic marker on another aspect’s NFR)

GRL Contribution Types:

- Make
- Some Positive
- Help
- Hurt
- Some Negative
- Break
- Unknown
Initial Satisfaction Level (100 for semantic marker in use; indicated by *)

Remote Service

Confidentiality

Consistency

Authentication

Caching

Encryption

Initial Satisfaction Level (0 for semantic marker not in use; default value)

GRL Satisfaction Levels:

- Denied
- Weakly Denied
- None
- Weakly Satisfied
- Satisfied
- Unknown
- Conflict
Electronic Voting Machine: Evaluated Goal Model (2)

GRL Satisfaction Levels:
- Denied
- Weakly Denied
- None
- Weakly Satisfied
- Satisfied
- Unknown
- Conflict
**Summary**

**Step 1: Base and aspect models**

- **GRL Goal Model**
  - High-Level Goal 1
  - High-Level Goal n
  - Strategies
  - Semantic Markers

**Step 2: Composed scenario model**

- **GRL Goal Model**
  - Evaluation
  - +75
  - 0
  - -100
  - 100

**Step 3: Instantiation of influence model**

- **MATA**
  - Semantic Markers (SM)
  - Composition
  - <<SM.A>>O1 <<SM.B>>O2

**Step 4: Evaluation of influence model**

- **MATA**
  - Semantic Markers (SM)
  - Evaluation
  - +75
  - 0
  - -100
  - 100

**Map semantic markers (SM) to initial satisfaction levels of strategies**

**GRL ... Goal-oriented Requirement Language**

**MATA ... Modeling Aspects Using a Transformation Approach**

---

**Introduction**

- Base Model
- Aspects
- Composed Model
- GRL
- Goal Model
- Evaluation
- Summary
- Case Study
- Conclusion/Work

---

**Semantic-Based Interaction Detection in Aspect-Oriented Scenarios**

**RE 2009**

© 2009 Gunter Mussbacher, Jon Whittle, Daniel Amyot
Case Study – Methodology

• Can our technique detect semantic interactions in practice that cannot be found by applying syntactic interaction detection?

• Industrial Case Study: Software Defined Radio Application
  • 40 page document, 11 primary use cases + a number of auxiliary use cases, 8 UML sequence diagrams (SDs) – all not aspect-oriented

1. Refactor original sequence diagrams to modularize crosscutting concerns
2. Develop a set of semantic markers for each aspect domain and annotate the aspects with these markers
3. Develop an influence model for these markers and related non-functional requirements
4. Apply our techniques to detect syntactic interactions
5. Apply our techniques to detect semantic interactions
Case Study – Results

• Base model with 5 SDs
• 4 aspect domains with 14 MATA SDs
• 11 semantic markers

<table>
<thead>
<tr>
<th>Id</th>
<th>Domain</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>Audit Trail</td>
<td>General Logging</td>
</tr>
<tr>
<td>1B</td>
<td>Audit Trail</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td>2A</td>
<td>Security</td>
<td>Tamper Proof</td>
</tr>
<tr>
<td>2B</td>
<td>Security</td>
<td>Authorization</td>
</tr>
<tr>
<td>2C</td>
<td>Security</td>
<td>Authentication of Downloads</td>
</tr>
<tr>
<td>2D</td>
<td>Security</td>
<td>Decryption</td>
</tr>
<tr>
<td>2E</td>
<td>Security</td>
<td>Filtering</td>
</tr>
<tr>
<td>2F</td>
<td>Security</td>
<td>Authentication - General</td>
</tr>
<tr>
<td>2G</td>
<td>Security</td>
<td>Authentication of Keys</td>
</tr>
<tr>
<td>2H</td>
<td>Security</td>
<td>Encryption</td>
</tr>
<tr>
<td>3</td>
<td>Cache</td>
<td>Download</td>
</tr>
<tr>
<td>4A</td>
<td>Fault Tolerance</td>
<td>Reporting – Fault</td>
</tr>
<tr>
<td>4B</td>
<td>Fault Tolerance</td>
<td>Reporting – Failure</td>
</tr>
<tr>
<td>4C</td>
<td>Fault Tolerance</td>
<td>Retry</td>
</tr>
</tbody>
</table>

• Syntactic interactions
  • Benign / non-benign

• Semantic interactions
  • Type I: Performance conflict
  • Type II: Storage / security conflict
  • Type III: Functionality / security conflict

• Disjoint
Case Study – Lessons Learned

• Results show that our techniques do indeed discover interactions not discoverable by our earlier efforts on syntactic interactions

• Usefulness beyond interactions
  • Application during iterative modeling
  • False positives
  • Granularity of markers
  • Return on investment
  • Scalability and complexity
Conclusion and Future Work

• Presented an approach for semantically detecting interactions between aspect models based on lightweight semantic annotations

• Tool support
  • MATA tool for UML SD (jUCMNav for AoUCM), jUCMNav for GRL
  • Not fully automated at this point

• The case study presented here constitutes only a first step in a longer term planned validation effort
  • Further empirical studies are needed to compare the benefits versus the additional effort required
  • Reusable, generic aspects and incrementally defined influence model?

• Use existing, domain-specific, standardized profiles for lightweight semantic annotations
  • Proved too complicated or did not cover the required domains