Semantic-Based Interaction Detection in Aspect-Oriented Scenarios

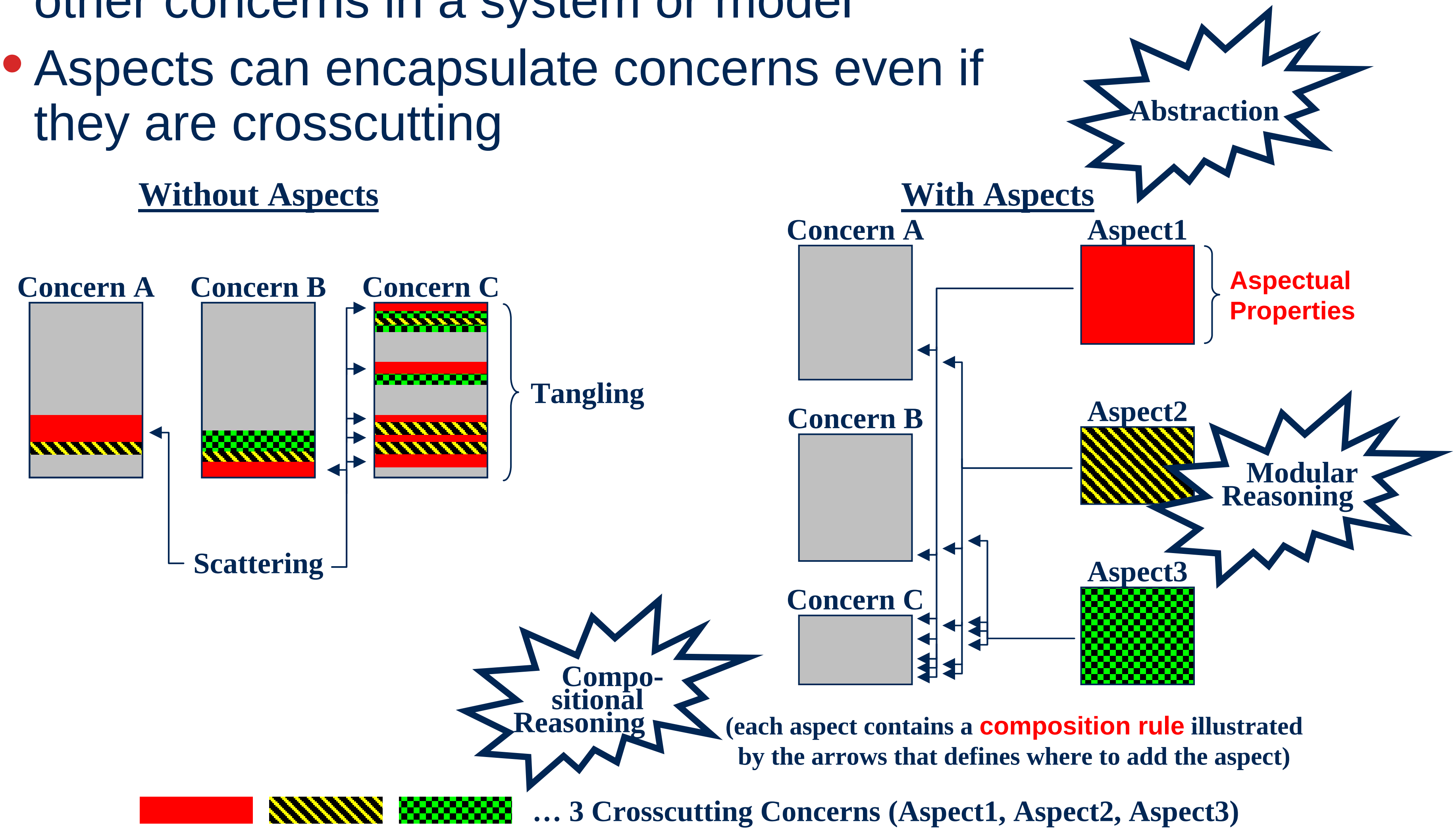
Gunter Mussbacher[1], Jon Whittle[2], Daniel Amyot[1]

[1]SITE, University of Ottawa, Canada
[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



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)

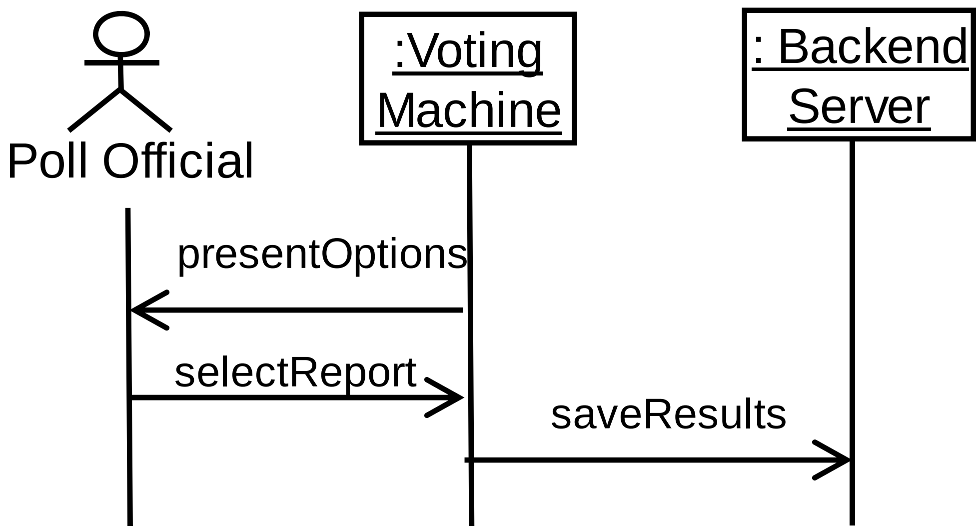
- Aspect Interaction Problem
  - Multiple aspects may be applicable at a given point in the base model
- In the best case, aspects may simply be ordered
  - E.g., an aspect may assume certain modeling elements in the base are introduced by another aspect
- In the worst case, there may be deep semantic 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)

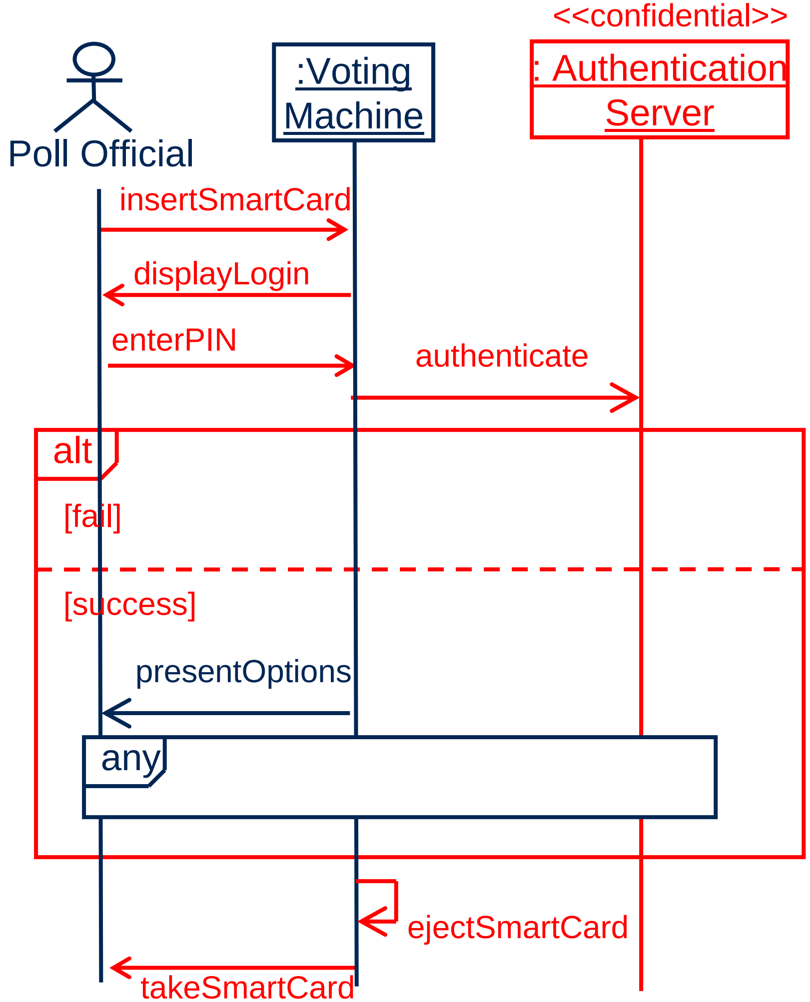
- Our approach to address semantic interactions
  - Lightweight semantic annotations of aspect models
  - Model the semantic impact of aspects on each other in a goal model called an influence model
  - Identify and trade-off semantic aspect interactions with the influence model
  - Reason about stakeholder needs and aspect interactions with the help of built-in qualitative or quantitative evaluation mechanisms applied to the influence model

Table of Contents

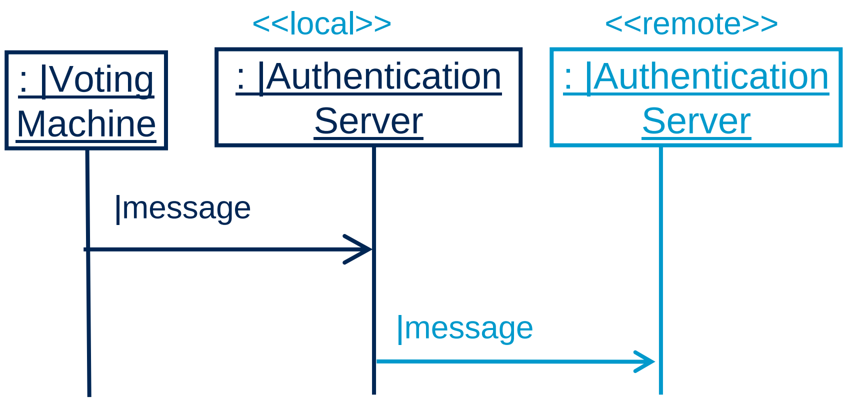
Electronic Voting Machine: Reporting Use Case



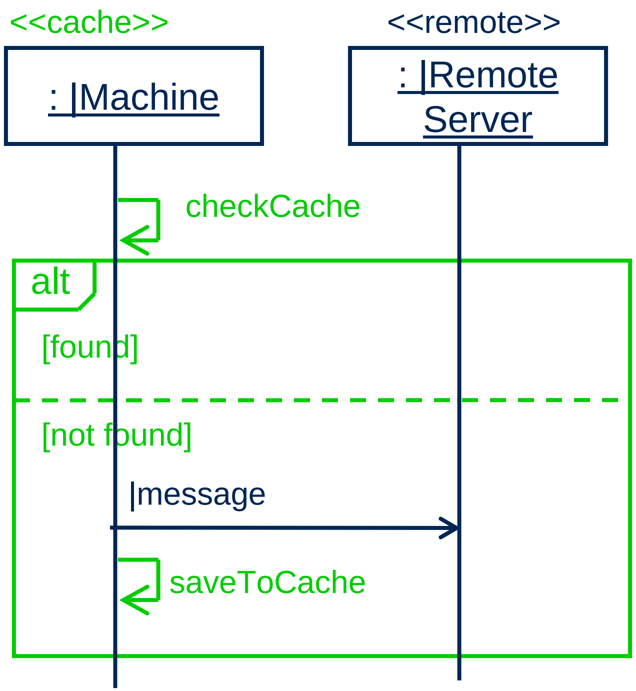
Electronic Voting Machine: Authentication Aspect



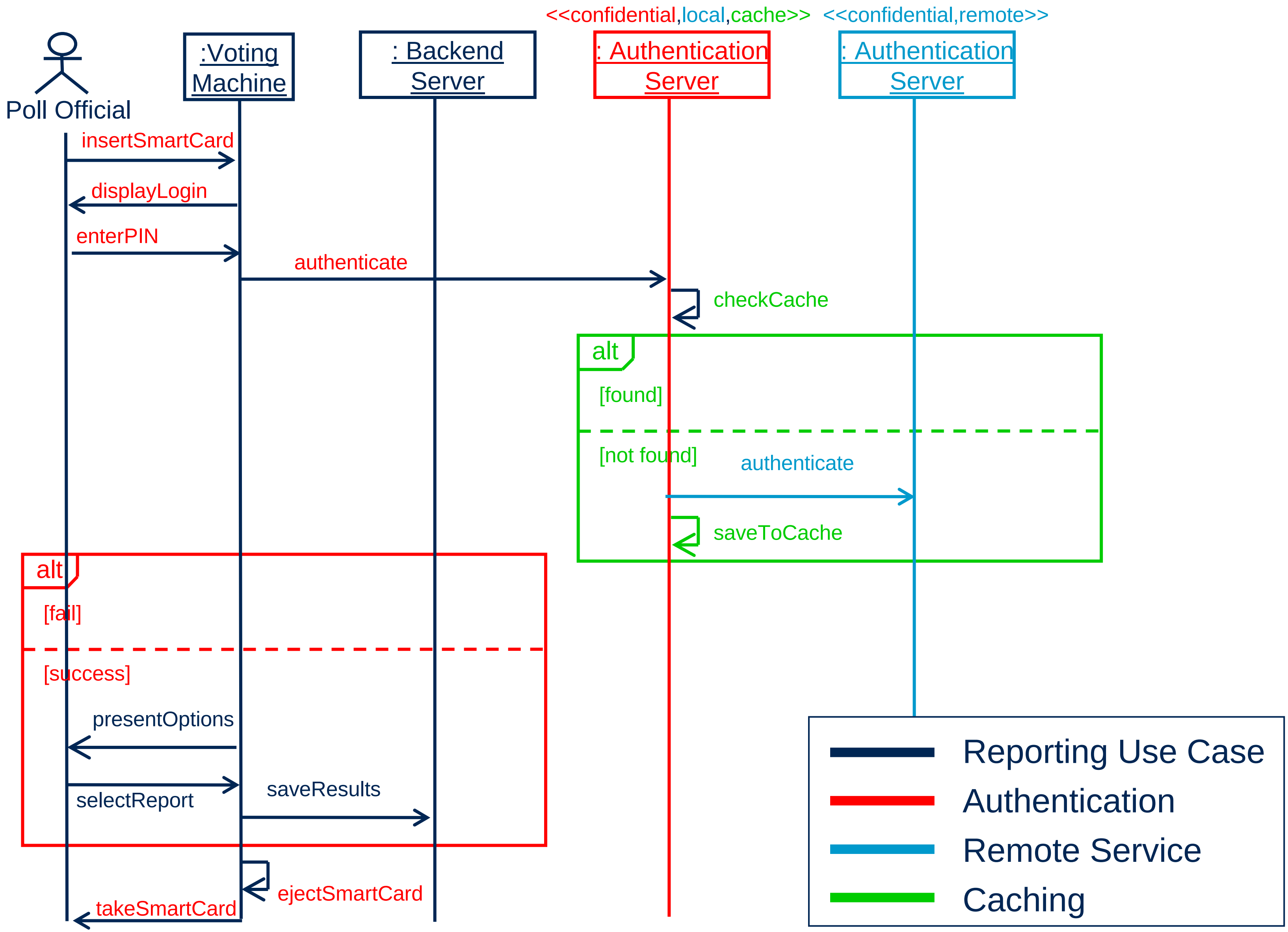
Electronic Voting Machine: Remote Service Aspect



Electronic Voting Machine: Caching Aspect



Electronic Voting Machine: Composed Model

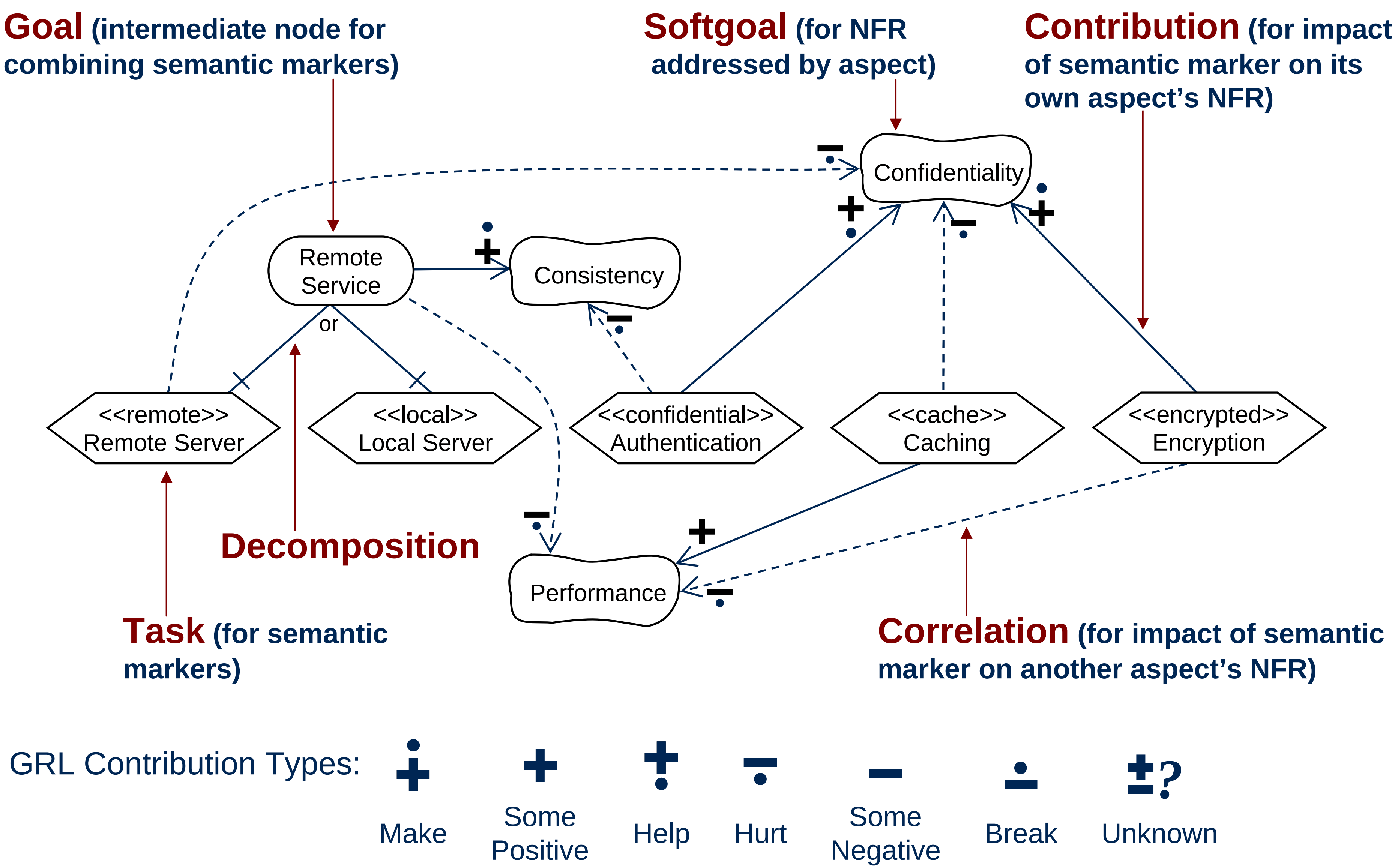


Goal-oriented Requirement Language (GRL)

- 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

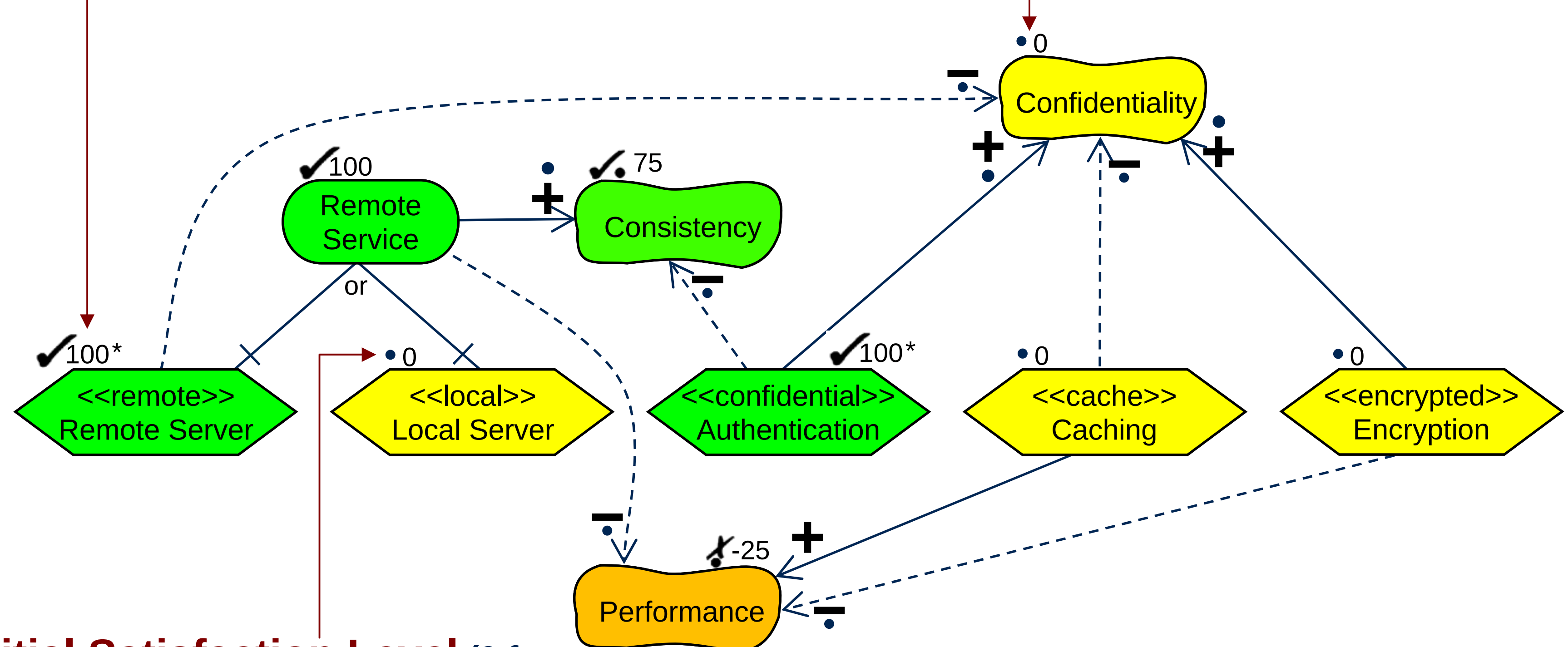
Electronic Voting Machine: Goal Model



Electronic Voting Machine: Evaluated Goal Model (1)

Initial Satisfaction Level (100 for semantic marker in use; indicated by *)

Propagated Satisfaction Level (for each aspect's NFR)

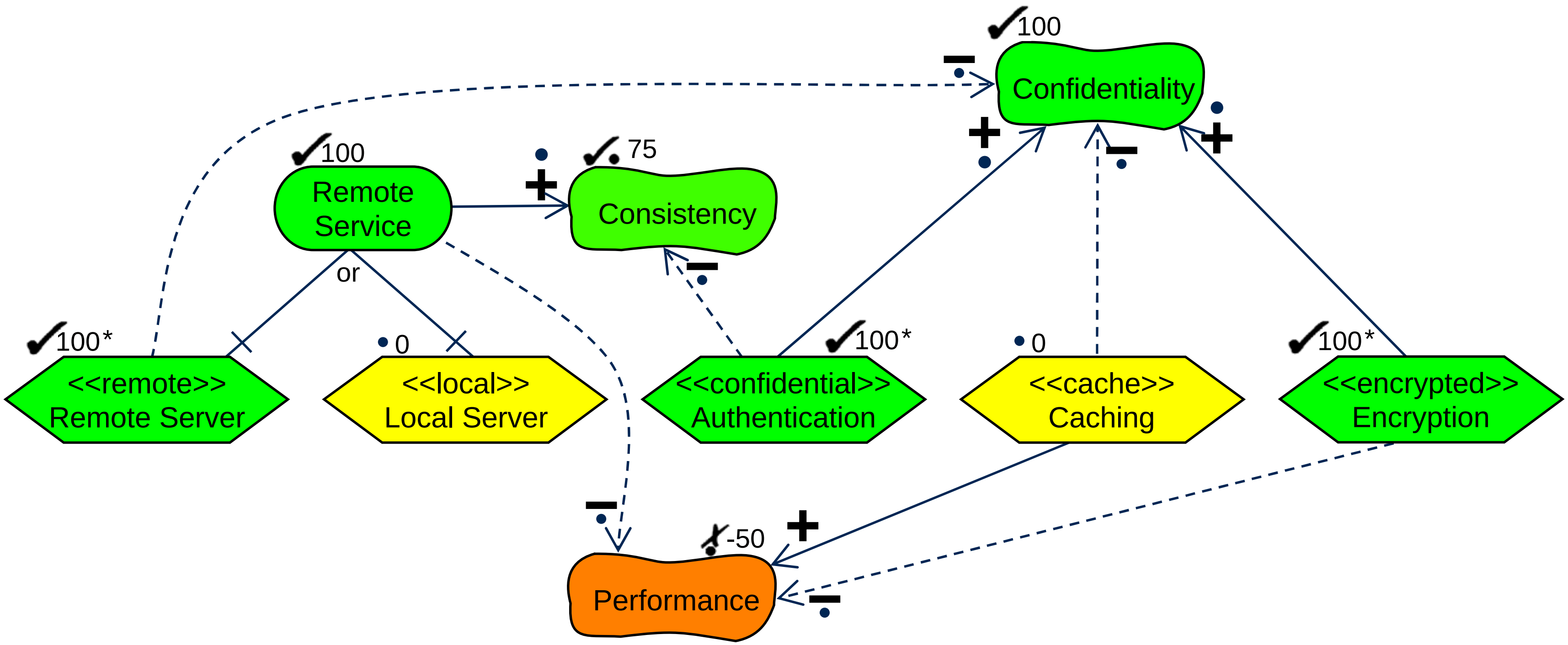


Initial Satisfaction Level (0 for semantic marker not in use; default value)

GRL Satisfaction Levels:

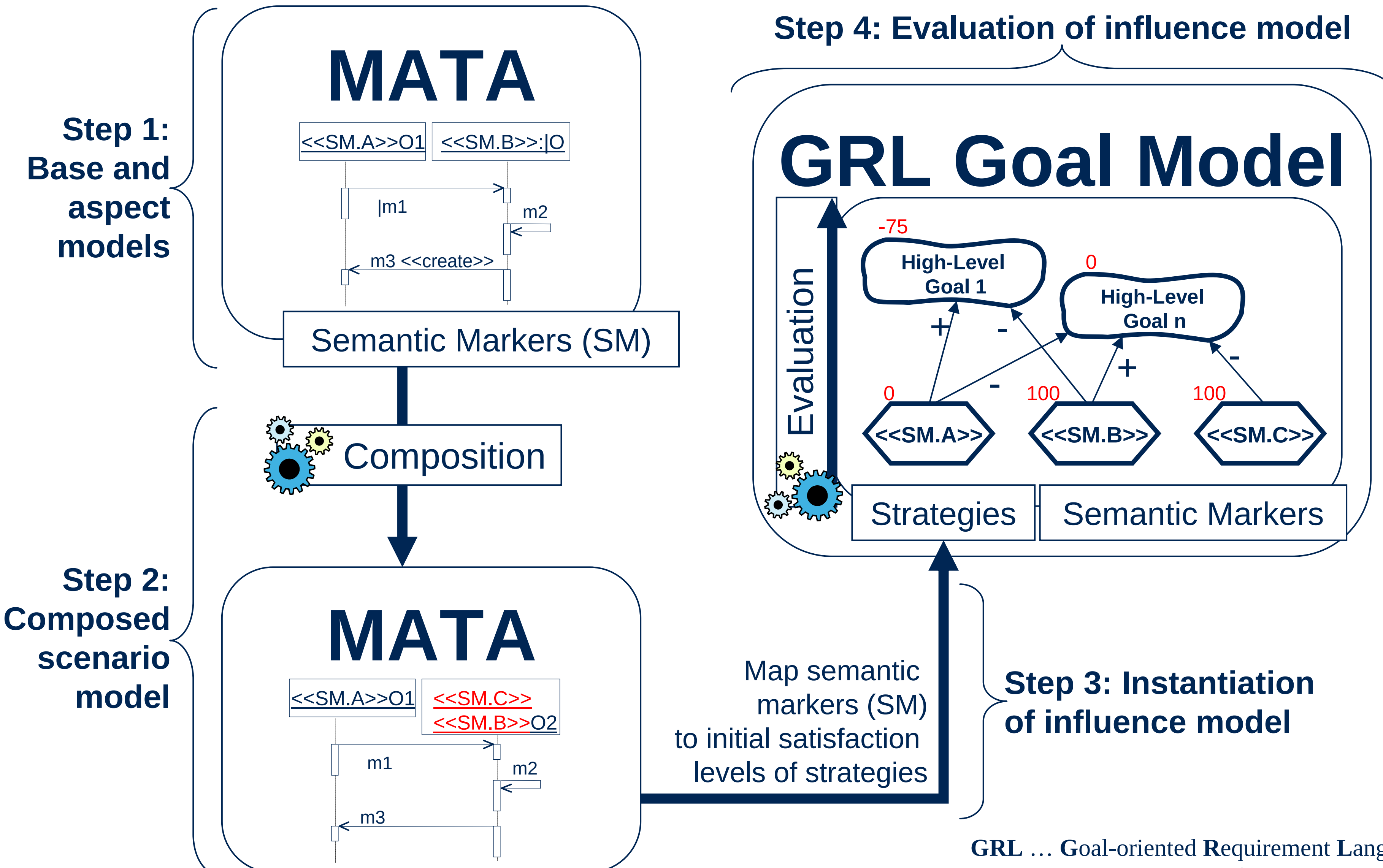
Denied | Weakly Denied | None | Weakly Satisfied | Satisfied | Unknown | Conflict

Electronic Voting Machine: Evaluated Goal Model (2)



GRL Satisfaction Levels: Denied | Weakly Denied | None | Weakly Satisfied | Satisfied | Unknown | Conflict

Summary



GRL … **G**oal-oriented **R**equirement **L**anguage
MATA … **M**odeling **A**spects Using a **T**ransformation **A**pproach

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

❶ Refactor original sequence diagrams to modularize crosscutting concerns

❷ Develop a set of semantic markers for each aspect domain and annotate the aspects with these markers

❸ Develop an influence model for these markers and related non-functional requirements

❹ Apply our techniques to detect syntactic interactions

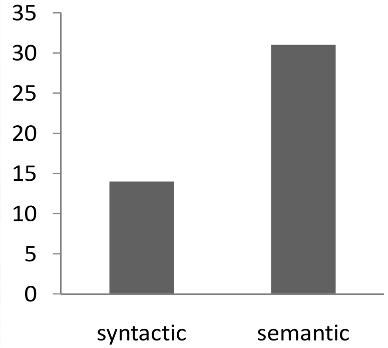
❺ Apply our techniques to detect semantic interactions

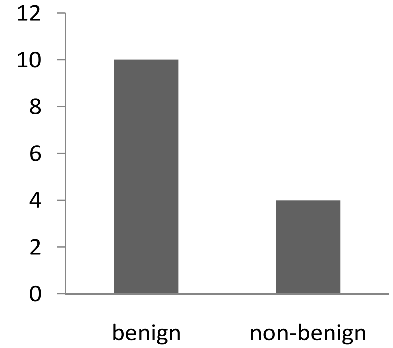
Case Study – Results

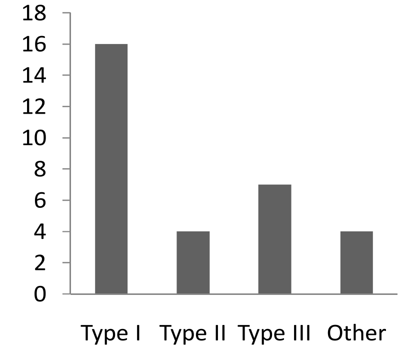
- Base model with 5 SDs
- 4 aspect domains with 14 MATA SDs
- 11 semantic markers

id	Domain	Aspect
1A	Audit Trail	General Logging
1B	Audit Trail	Non-repudiation
2A	Security	Tamper Proof
2B	Security	Authorization
2C	Security	Authentication of Downloads
2D	Security	Decryption
2E	Security	Filtering
2F	Security	Authentication - General
2G	Security	Authentication of Keys
2H	Security	Encryption
3	Cache	Download
4A	Fault Tolerance	Reporting – Fault
4B	Fault Tolerance	Reporting –Failure
4C	Fault Tolerance	Retry

- 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

