



DO MORE
www.rise.com.br

Towards an Approach for Service-Oriented Product Line Architectures

Flávio Mota Medeiros

Eduardo Santana de Almeida

Silvio Romero de Lemos Meira

Outline

- Introduction
- Variability Levels and Mechanisms
- Case Study
- Reuse Across Product Lines
- Conclusions and Future Work

Introduction

- Service-Oriented Architecture (SOA) [1]
 - » Develop distributed applications
 - » Self-contained, reusable and loosely coupled

- SOA Highlights [2]
 - » Align IT and business goals
 - » Increase reusability
 - » Flexibility to change (business agility)

General Scenario

- Systematic planned reuse
 - » SOA reuse is not planned and systematic as in SPL [3]
 - » Ad-hoc reuse strategy
- Artifacts reuse
 - » Services are the reusable entities [4]
 - » Diagrams and business process models
- High customization
 - » Through variability [5]

Research Goal

- Develop SPL with SOA
 - » Explore commonality and variability
 - » Domain-specific service reuse
 - » Artifacts with variability
- Raise reuse level
 - » Increase productivity
 - » Decrease development costs and time
- Service-oriented applications
 - » Systematic planned reuse
 - » High customization
 - » Customers or market segments needs

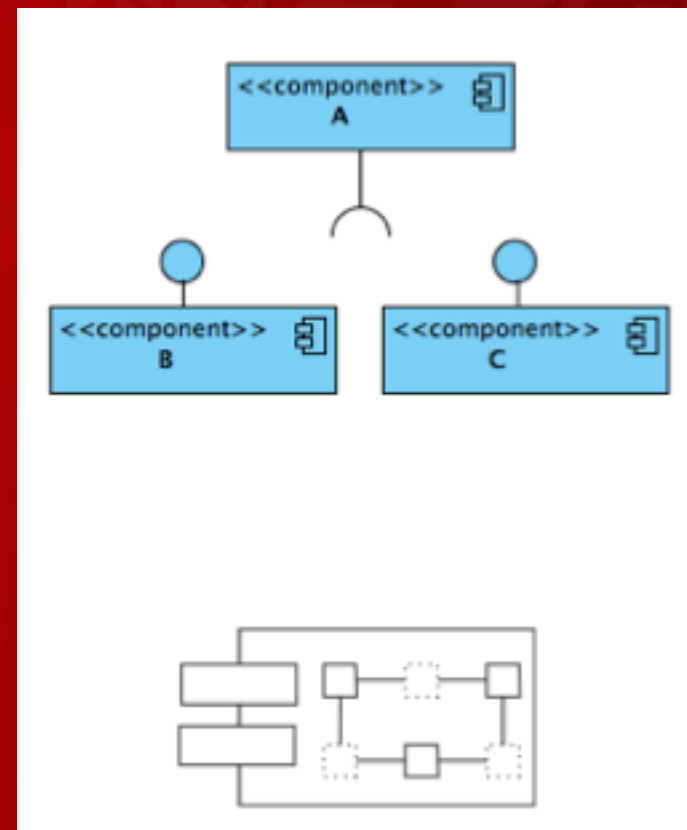
Outline

- Introduction
- **Variability Levels and Mechanisms**
- Case Study
- Reuse Across Product Lines
- Conclusions and Future Work

Variability Levels [6] [7]

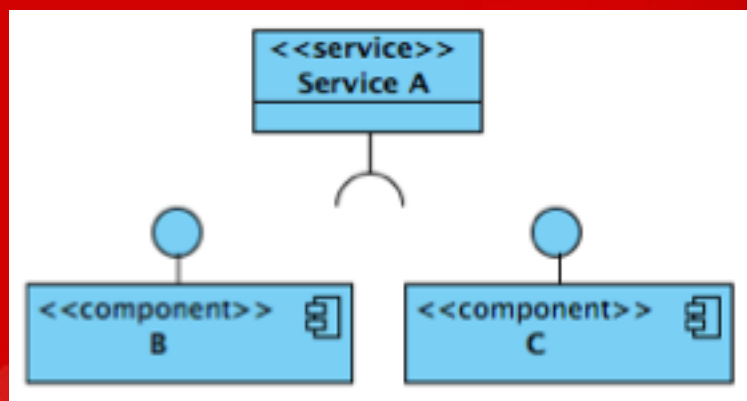
- Configuration variability
 - » Select services or components
 - » Architecture variability
 - » High granularity
 - Different classes

- Customization variability
 - » Introduce internal variability
 - » Customize component and services
 - » Low granularity
 - Class attributes or methods



Variability Mechanisms

- Configuration Variability [8] [9]
 - » Dependency injection
 - » Parameters
 - » Configuration files
 - » Aspect orientation



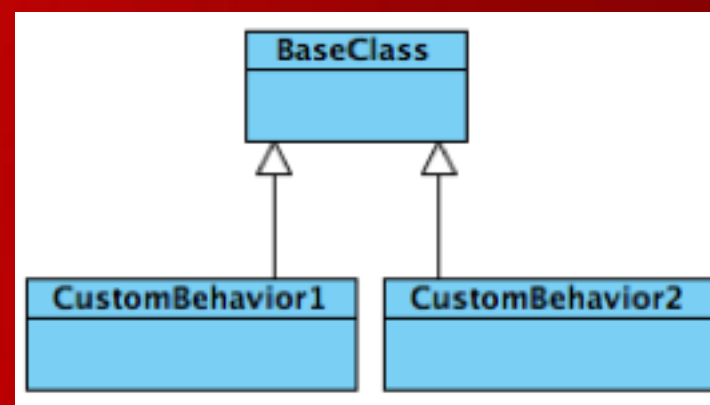
```

If ( condition ) {
    //Binding Component B
    //Call service B
} else {
    //Binding Component C
    //Call service C
}

```

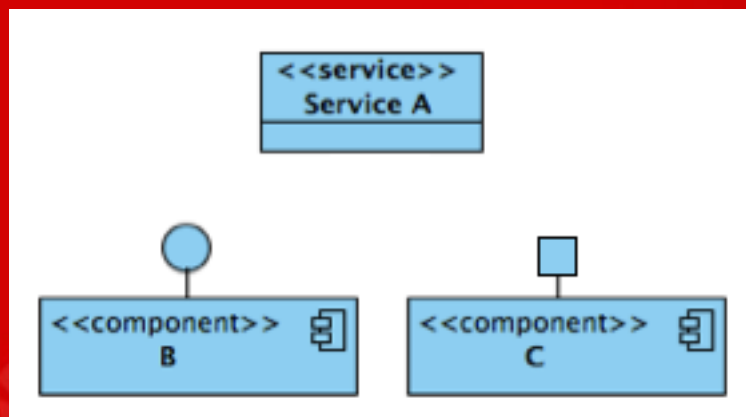
Variability Mechanisms

- Customization variability [10]
 - » Design patterns
 - » Configuration files
 - » Parameters
 - » Aspect orientation



Variability Mechanisms

- Configuration and customization variability together
 - » Select component B or C
 - » Component B and C with different interfaces
 - » Service A exposes different interfaces
 - » Aspect orientation



```

public aspect ComponentB {
    public void ServiceA.operationB(){
        // Implementation
    }
}
  
```

```

public aspect ComponentC {
    public void ServiceA.operationC(){
        // Implementation
    }
}
  
```

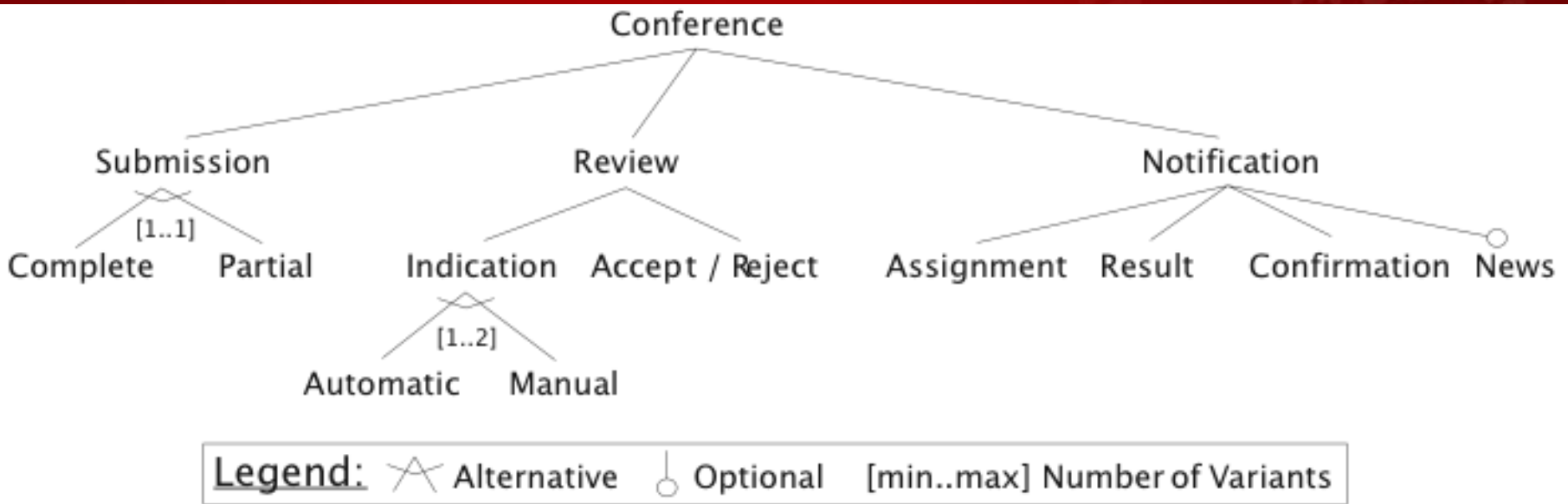
Outline

- Introduction
- Variability Levels and Mechanisms
- **Case Study**
- Reuse Across Product Lines
- Conclusions and Future Work

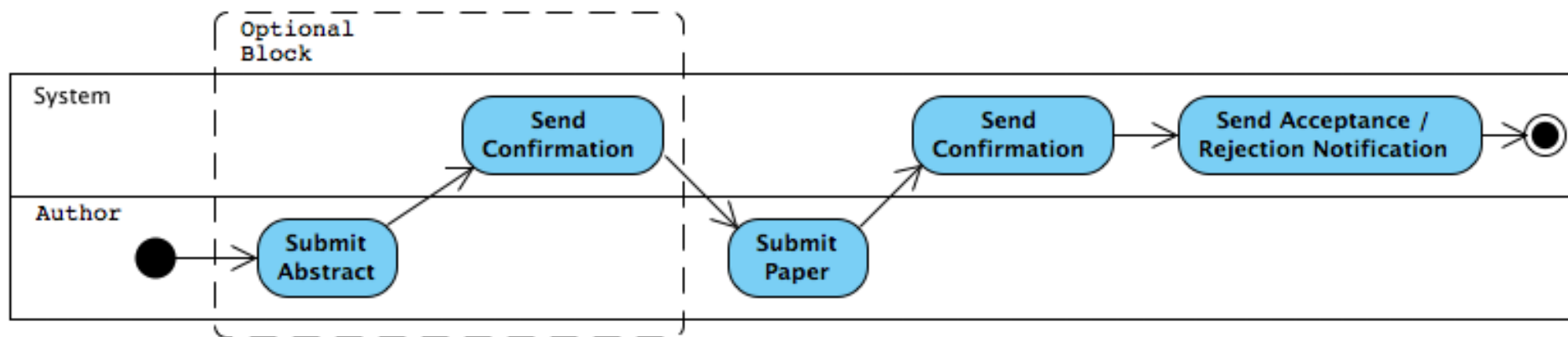
Case Study

- Conference management
 - » Service-oriented product line
 - » Explore commonality and variability
 - Cyber chair, easy chair, IS Technology journal, etc
 - » Customized according to the requirements
- SPL project
 - » Full product line version
 - » Reduced scope in the context of SOAPL

Scope

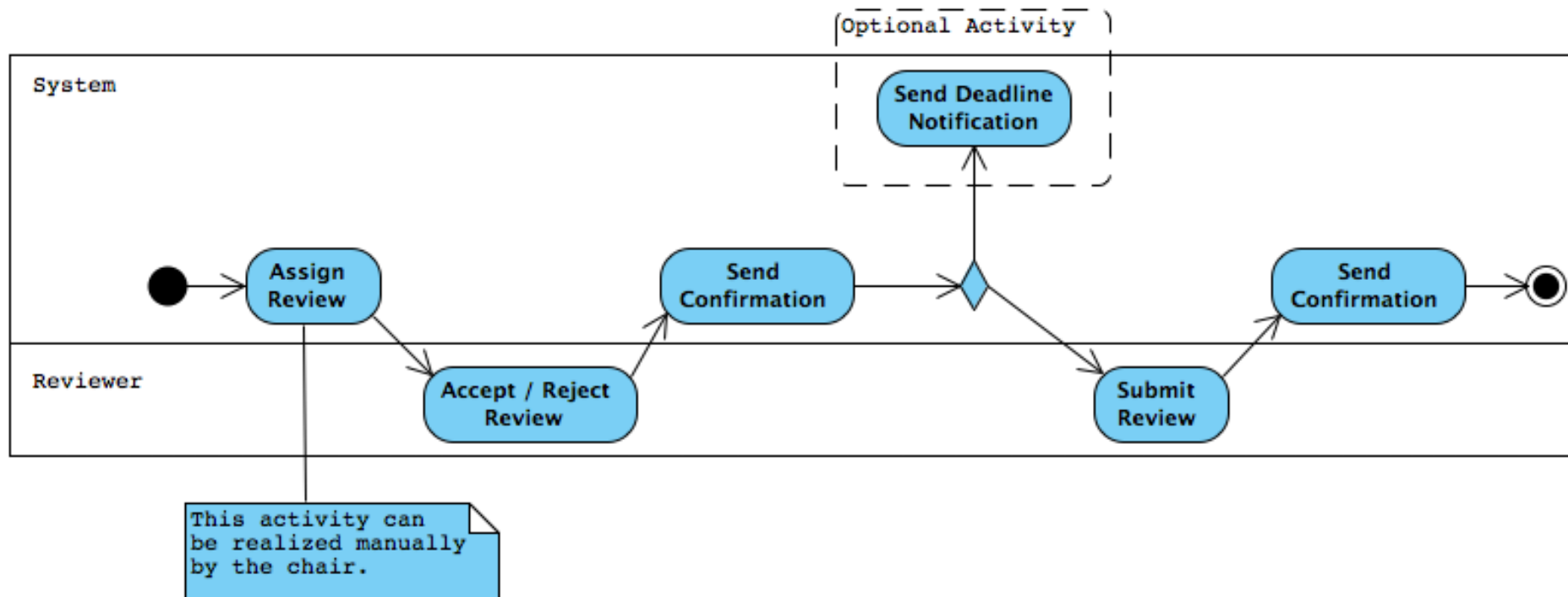


Submission Business Process



- Partial or complete submission
 - » Alternative feature

Review Business Process



- » Event news notification (optional)
- » Manual or automatic assignments (OR)

Where we're going

- Generate scenarios for users (submitter, reviewer, admin)
- Identify possible services
- Identify variations
- Enhance feature model
- Layer by services and workflows

Services

- Registration (author, reviewer, email confirmation/reregister)
- Scheduling (submission cut off, review cut off, reminders)
- Bidding/Selection/Assignment
- Submissions (abstracts, papers, reviews)
 - » Uploads (browse)
 - » Withdraw submission
 - » Conflicts
- Notification (paper, review summary, conflict, Reviews complete)
- Format (administrator)
 - » Paper (keywords, file, size, cycles (review, revise, final))
 - » Review form
 - » Review (criteria, levels, numbers of reviewers)
 - » Knowledge areas

Services - 2

- Review summary
 - » Accept/reject criteria
 - » Conflict resolution criteria
- Storage/retrieval (papers, reviews, names)
 - » tracking

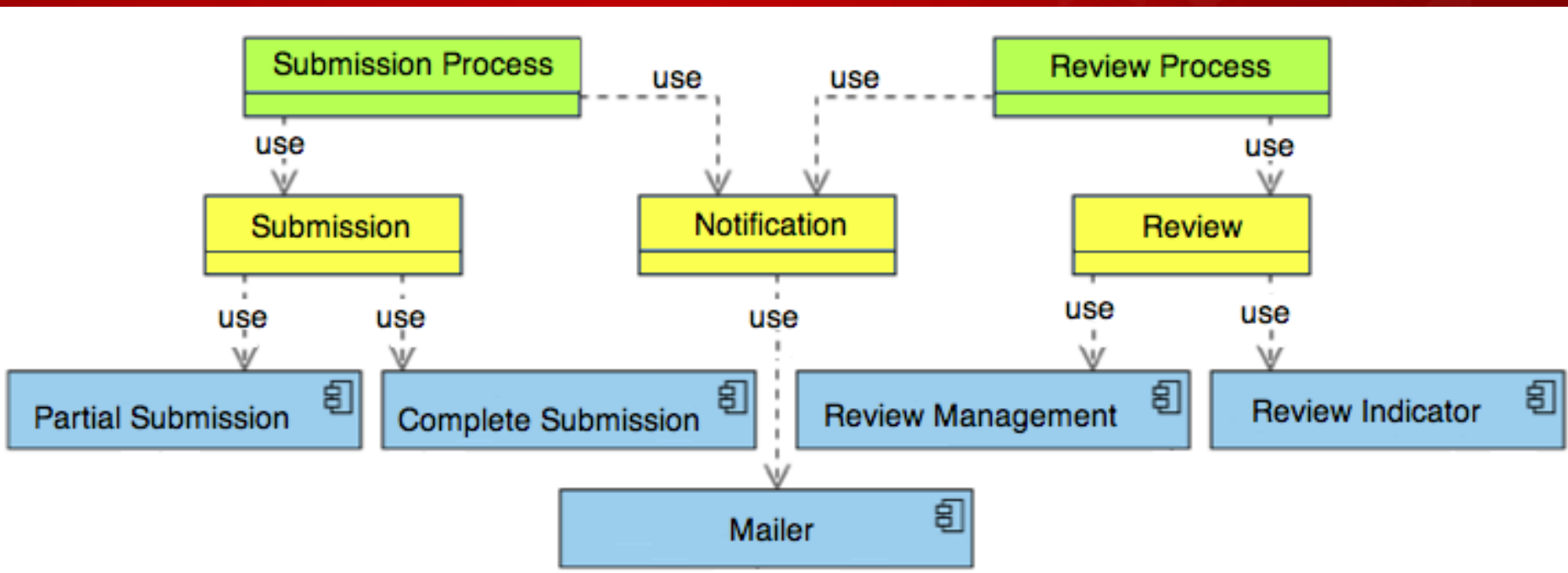
Submission

- Submit (alt: abstracts, papers, reviews, response by author to comments)
 - *Medical tests*
- Meta data
 - » Conference identification
 - » Author anonymity (option)
 - » Conflicts
 - *Patient, insurance,*
- Uploads (opt: browse)
 - » Word → .pdf
 - *.pdf, images, audio, video*
 - » Submission window (cut and paste into window)
 - » Protocols (ftp, web,
 - *Secure protocols, auditing*
 - » Format verifier, pages, words in abstract, file size, combine words/graphics in word count
 - » Format preview
 - » Via email
- Withdraw submission
- For review, as revised or final submission

Workflow -

- So far
 - » Author identifies paper (*Clinic identifies*)
 - » Submits paper (*Clinic submits*)
 - » System stores
- Next
 - » Provide summaries to reviewers → bidding service
 - » Place in queue for reviewer → start clock on review submissions
 - *Place medical record in queue for reviewer*
 - » Allow for withdraws/updates

Architectural Elements



- Orchestration Layer
- Service Layer
- Component Layer

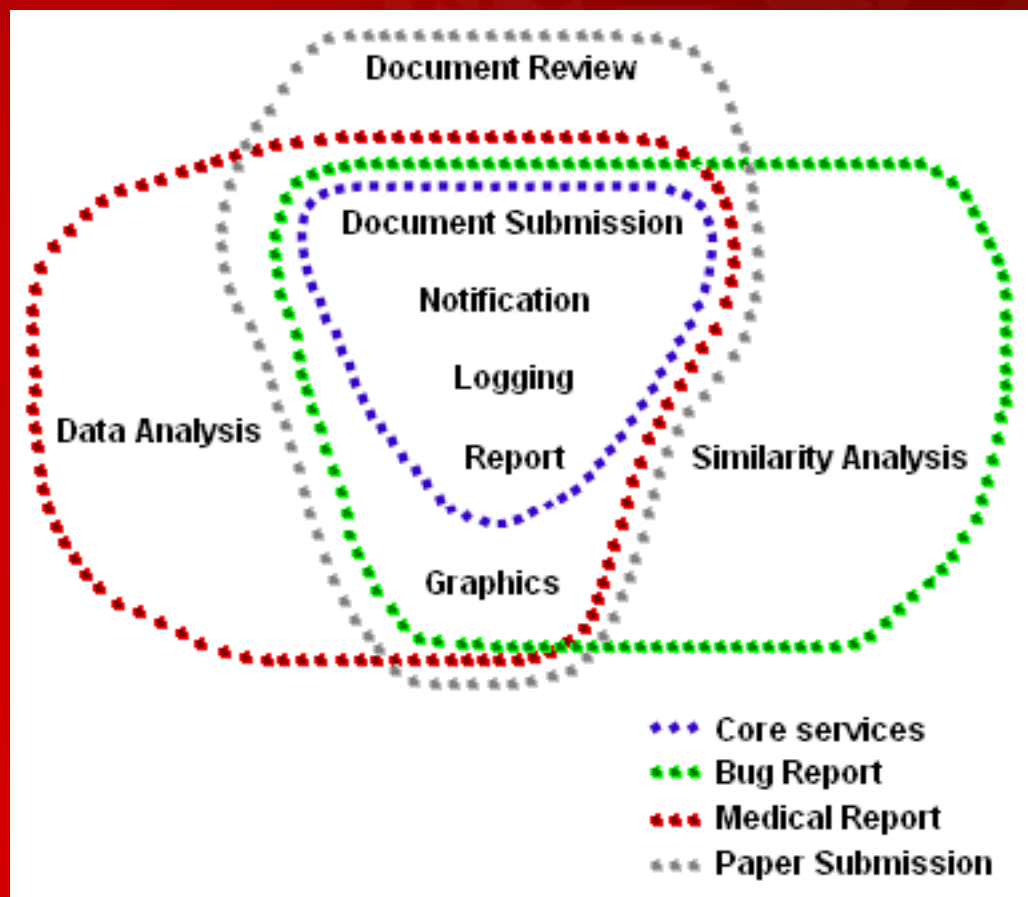
Outline

- Introduction
- Variability Levels and Mechanisms
- Case Study
- **Reuse Across Product Lines**
- Conclusions and Future Work

Services for Different SPLs

- Domains

- » Paper submission
- » Medical record review
- » Trouble or Bug report
- » Item order report



Outline

- Introduction
- Variability Levels and Mechanisms
- Case Study
- Reuse Across Product Lines
- Conclusions and Future Work

Conclusions

- SPL principles can be used in service environments
 - » Systematic reuse
 - » High customization (variability)
- Service internal variability
 - » Useful in some cases
 - » Variability granularity is low
- Variability mechanism work in service environment
 - » Design patterns, aspects and configuration files
 - » Dependency injection and parameters

Future Work

- Evolve the case study
 - » Analyze more variability mechanisms
 - » Analyze different binding times

- Extend the SPL to Different domains
 - » Bug report

Thank you!

Any questions or suggestions?

References I

- [1] Thomas Erl. “Service-Oriented Architecture: Concepts, Technology, and Design”, 2005.
- [2] Ali Arsanjani. “Service-Oriented Modeling and Architecture”, 2004.
- [3] Clements and Northrop. “Software Product Lines: Practices and Patterns”, 2001.
- [4] Mikko Raatikainen, Varvana Myllärniemi and Tomi Männistö. “Comparison of Service and Software Product Family Modeling”, 2007.

References II

- [5] Klaus Pohl et al. “Software Product Line Engineering: Foundations, Principles and Techniques”, 2005.
- [6] Eunsuk Ye, Mikyeong Moon, Youngbong Kim and Keunhyuk Yeom. “An Approach to Designing Service-Oriented Product-Line Architecture for Business Process Families”, 2007.
- [7] Nicola Boffoli, Danilo Caivano, Daniella Castelluccia, Fabrizio Maria Maggi and Giuseppe Visaggio. “Business Process Lines to Develop Service-Oriented Architectures through Software Product Lines Paradigm”, 2008.

References III

- [8] Mikael Svahnberg, Jilles van Gorp and Jan Bosch. “A taxonomy of Variability Realization Techniques”, 2005.
- [9] Sergio Segura, David Benavides, Antonio Ruiz-Cortés and Pablo Trinidad. “A Taxonomy of Variability in Web Service Flows”, 2007.
- [10] Cristina Gacek and Michalis Anastasopoulos. “Implementing Product Line Variabilities”, 2001.