Aspects – Part I: Aspect-oriented Programming with AspectJ and AJDT

University of Ottawa

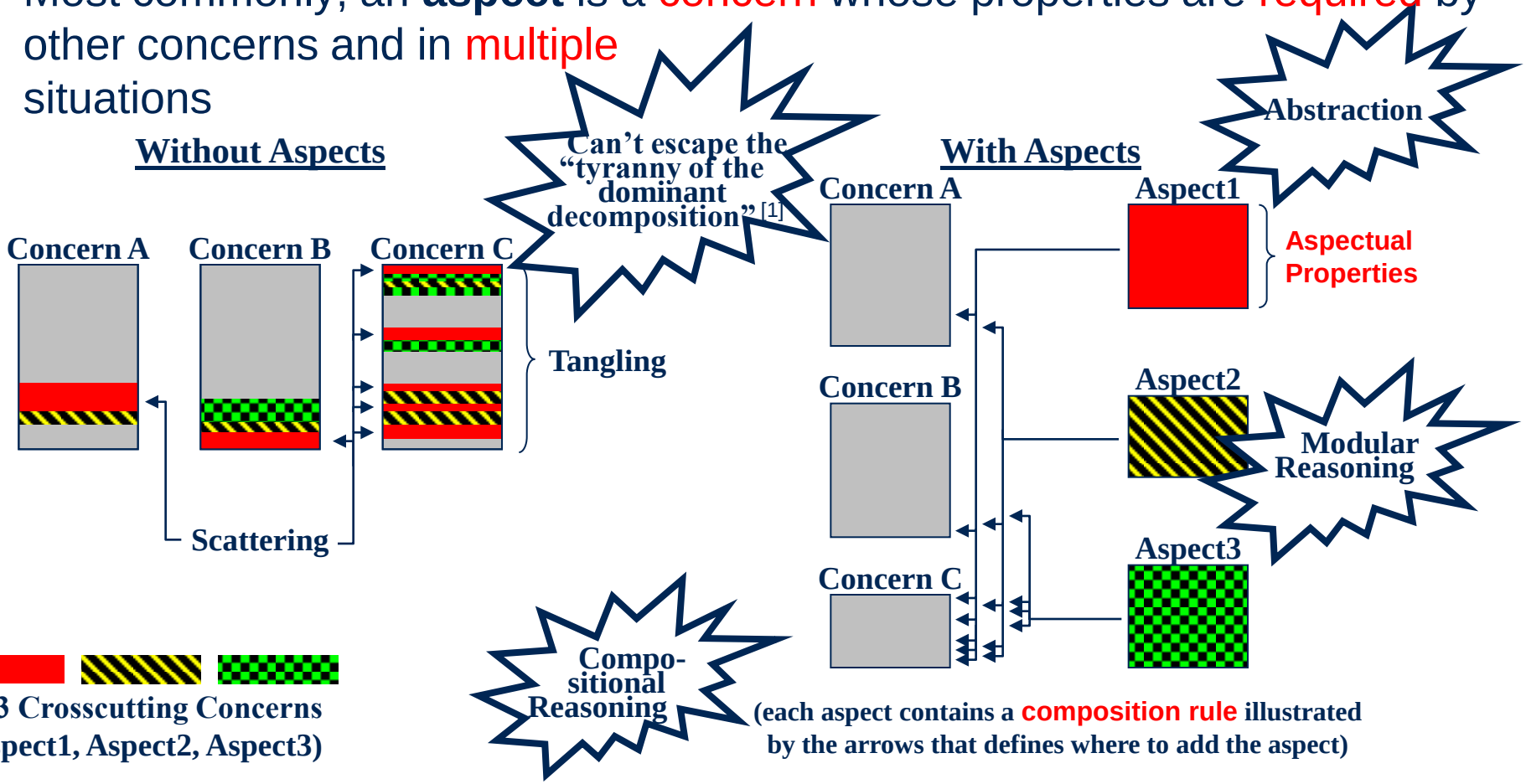
SEG4110

October 2011

Table of Contents

Introduction to Aspects: Crosscutting Concerns

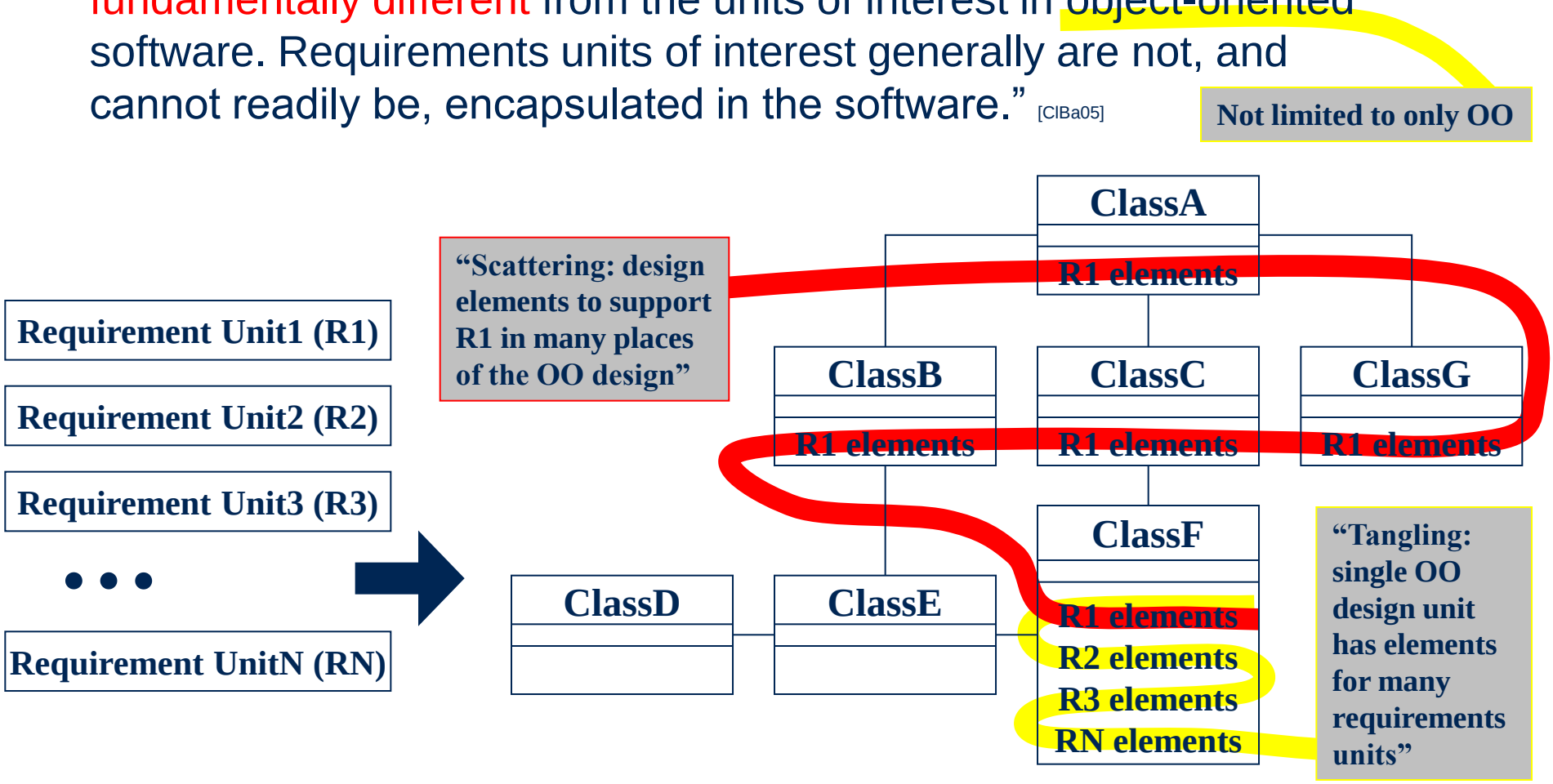
- In general, a **concern** is anything related to a goal, feature, concept, or "kind" of functionality
- Most commonly, an **aspect** is a concern whose properties are required by other concerns and in multiple situations



[1] Tarr, P., Ossher, H., Harrison, W., and Sutton, S.M.: N degrees of separation: Multidimensional separation of concerns. ICSE 99

Introduction to Aspects: The Problem

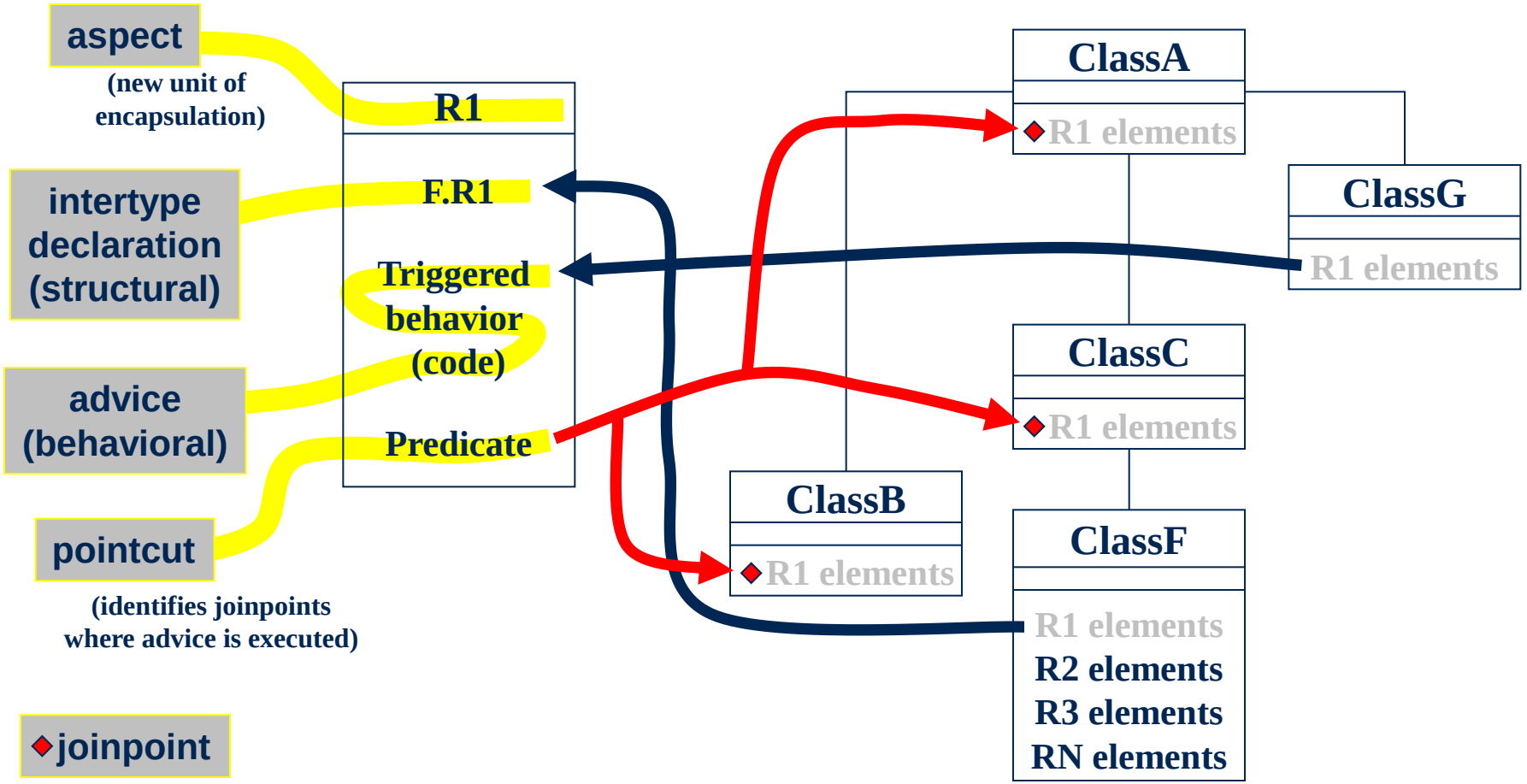
- "Structurally, the units of interest in the requirements domain are fundamentally different from the units of interest in object-oriented software. Requirements units of interest generally are not, and cannot readily be, encapsulated in the software." [ClBa05]



Clarke, S. and Baniassad, E.: *Aspect-Oriented Analysis and Design: The Theme Approach*. Addison-Wesley, 2005, www.thethemeapproach.com

Introduction to Aspects: The Solution with AspectJ

- Aspects address the problem of one concern crosscutting other concerns in the system/model because aspects encapsulate crosscutting concerns



Terminology based on AspectJ: www.eclipse.org/aspectj

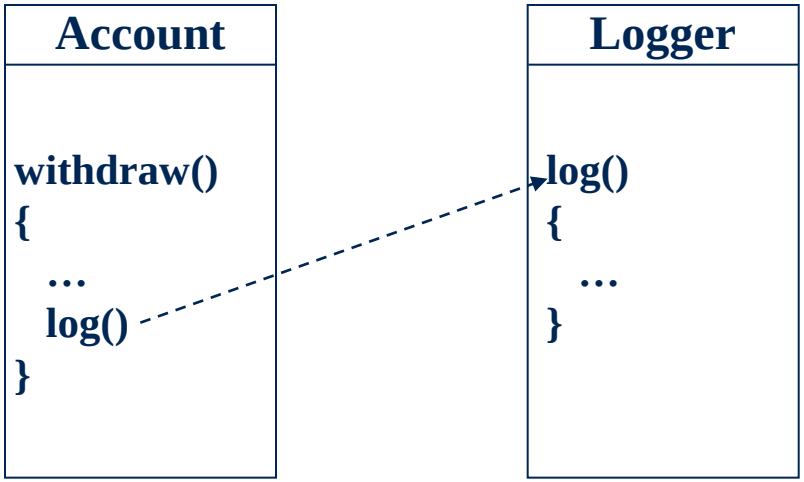
Introduction to Aspects: Examples

- Authorization/Authentication
- Caching
- Concurrency management
- Debugging
- Distribution
- Logging
- Testing
- Transaction management
- …

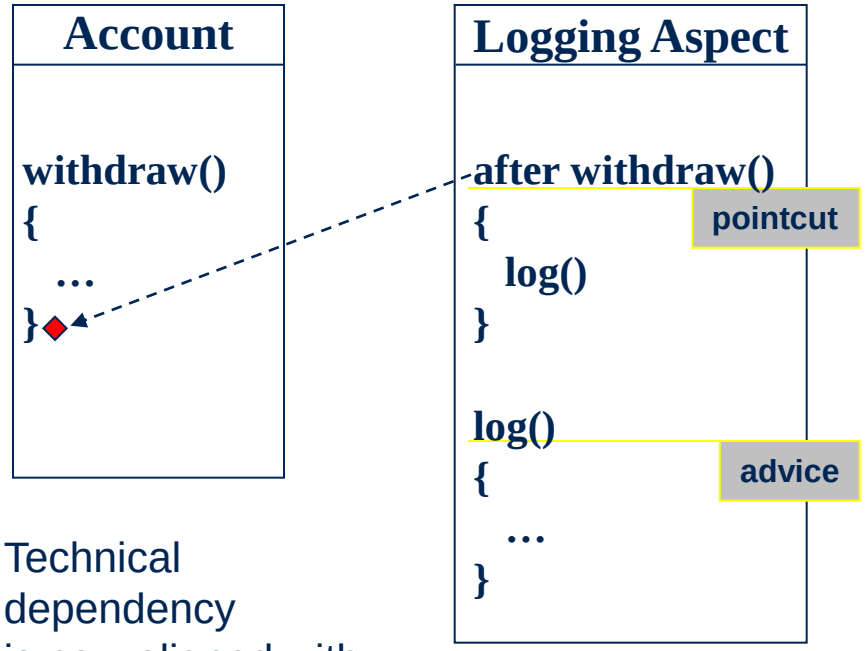
- A feature or use case
- A non-functional requirement (a system quality)
- Alternative architectures
- Alternative communication mechanisms

Introduction to Aspects: Dependencies

- Aspects make it possible to align technical dependencies with logical dependencies to a greater extent than before



Account is dependent on Logger due to technical reasons only. Logically, an account is completely independent from a Logger. On the contrary, the Logger is logically dependent on what needs to be logged (i.e. the Account).



Technical dependency is now aligned with logical dependency.

AJDT: Projects & Basic Views

- Right-click on project, file, … and select *AspectJ Tools*
- Add/remove AspectJ Runtime libraries
  - Create New AspectJ Project
  - Convert Existing Project to AspectJ Project
  - Remove AspectJ Capability from Project
- Aspect file: *.aj
- Perspective
  - Aspect Visualization
- Views
  - Visualiser and Visualiser Menu
  - Cross References

AJDT: Build Configurations

- Save Build Configuration As
- Build Configuration file: *.ajproperties

- src.includes = …

- src.excludes = …

- Apply Build Configuration

AJDT: Overview of Basic Syntax

- Aspect (can be abstract)
  - [*Visibility*] **aspect** *Id* [**extends** *Type*] [**implements** *TypeList*] { *Body* }
- Pointcut (can be abstract)
  - [*Visibility*] **pointcut** *Id*(Args) : *PointcutExpression*;
- Pointcut Expression
  - *PointcutType*(*Visibility ReturnType Package.Class.Method*(*Args*))
  - PointcutType: call, execution (many more…, not all have the same syntax)
  - Wildcard matching: *, Any argument: .., Any subtype: +
  - Any subpackages: add an additional . after the package name
  - && || !
- Advice
  - **before/after**(Args) [**throws** *TypeList*] : *Pointcut* { *Body* }
  - *Type* **around**(Args) [**throws** *TypeList*] : *Pointcut* { *Body* }
  - **proceed**(Args)

AJDT: Overview of Basic Syntax

- Example of pointcut exposing argument and advice using this pointcut
  - **pointcut** pc(*Type Id*) : **call**(* * *Package.Class.Method*(*Type*)) && **args**(*Id*);
  - **before**(*Type Id*) : pc(*Id*) { *Body* }
- Example of pointcut exposing target and advice using this pointcut
  - **pointcut** pc(Type *Id*) : **call**(* * *Package.Class.Method*(..)) && **target**(*Id*);
  - **before**(*Type Id*) : pc(*Id*) { *Body* }
- Intertype declaration (can be abstract)
  - [*Visibility/static*] *Type Type.Id*(*Args*) [**throws** *TypeList*] { *Body* }
- **declare error/warning** : *PointcutExpression* : *String*;
- **thisJoinPoint** … reflective information about the join point

AJDT: Help

- http://www.eclipse.org/aspectj/docs.php
- AspectJ Language
  - Programming Guide
  - Quick Reference
- AspectJ Developer's Notebook