

## Concern Metrics

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

## Measurement and Quality

- Achievement of good design is not trivial
  - Requires different sorts of metrics
- Some concerns deteriorate the system maintainability
  - Conventional metrics cannot easily detect some bad smells related to concerns

## Concern Metrics

- Measurement of concern modularity attributes is required
- Concern metrics capture information about concerns traversing one or more structural modular units
  - Modular units can be classes, methods, attributes, etc.

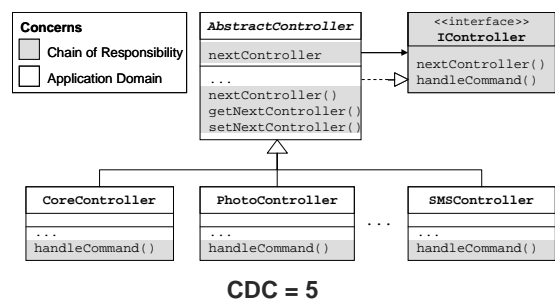
## Example of Concern Metrics

- Concern Diffusion over Components (CDC)
- Concern Diffusion over Operations (CDO)
- Concern Diffusion over Lines of Code (CDLOC)
- Number of Concerns per Component (NCC)
- Lines of Concern Code (LOCC)

## CDC – Concern Scattering

- CDC counts the number of classes and interfaces related to a concern
  - In architecture models, CDC can count the number of architecture components related to a concern
- Less scattered concerns are easier to understand and maintain

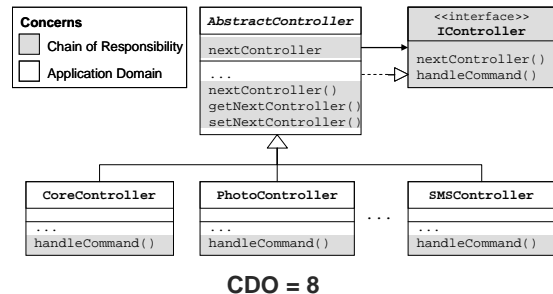
## Example of CDC



## CDO – Methods with Concern

- CDO counts the number of methods and constructors related to a concern
- CDC and CDO quantify concern scattering at different entities

## Example of CDO



## CDLOC – Concern Tangling

- CDLOC counts the number of concern switches through the lines of code
  - A concern switch is a place in the source code where there is a concern change
- Tangled concerns are harder to understand

## Class AbstractController

```

public class AbstractController implements IController {
    private IController nextController;

    ...

    public void nextController(Command command) {
        if (handleCommand(command) == false)
            getNextController().handleCommand(command);
    }

    public IController getNextController() {...}
    public void setNextController(...) {...}
}

```

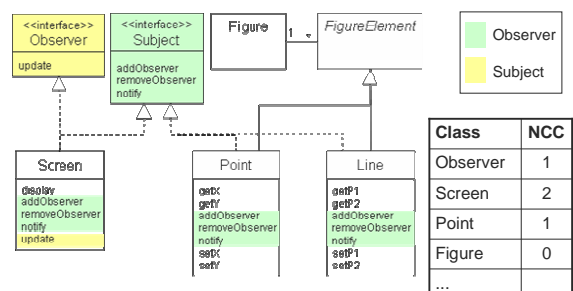
concern switch ↻  
concern switch ↻  
concern switch ↻  
concern switch ↻

**CDLOC = 4**

## NCC – Tangled Class

- NCC counts the number of concerns of interest implemented by a component
- Modules with tangled concerns are harder to reuse

## Example of NCC



## [ LOCC – Concern Size ]

- LOCC counts the number of lines of code which implement a concern
- Large concerns go against modular reasoning
  - We should be able to reason about a small part of the problem each time

## [ Class AbstractController ]

```
public class AbstractController implements IController {  
    private IController nextController;  
    ...  
    public void nextController(Command command) {  
        if (handleCommand(command) == false)  
            getNextController().handleCommand(command);  
    }  
    public IController getNextController() {...}  
    public void setNextController(...) {...}  
}
```

Diagram illustrating the LOCC (Lines of Code Counting) for the `AbstractController` class. The code is annotated with "concern switch" labels and arrows indicating the boundaries of concerns. The code is divided into four distinct concern blocks, each marked with a "concern switch" label and a curved arrow pointing to the start or end of the block.

**LOCC = 8**

## [ Bibliography ]

- CDC, CDO and CDLOC
  - C. Sant'Anna et al. On the Reuse and Maintenance of. Aspect-Oriented Software: an Assessment Framework. SBES 2003.
- NCC
  - E. Figueiredo et al. On the Maintainability of Aspect-Oriented Software: A Concern-Oriented Measurement Framework. CSMR 2008.
- LOCC
  - M. Eaddy et al. Do Crosscutting Concerns Cause Defects? TSE 2008.