Pointcut in AspectJ

Eduardo Figueiredo

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

AspectJ

- The most used and mature aspect-oriented programming language
- It is a simple extension of Java
  - After compilation, it creates .class files as every Java program does
  - AspectJ program can run in a JVM
- AspectJ has good tolling support
  - AspectJ Development Tools (AJDT)
  - AJDT integrates to Eclipse IDE

AspectJ Features

- Joint point
- Pointcut
- Advice
- Inter-type declaration

Joint points

AspectJ Joint Point Model

- It defines how and where classes and aspects are combined
- Examples
  - Calls of methods and constructors
  - Execution of methods and constructors
  - Object instantiation
  - Accesses to fields, etc.

Joint Point Examples

- Calls to the **credit** method in **Account**
  - call(void Account.credit(double))
- Read accesses of the **number** attribute in **Account**
  - get(String Account.number)

Quantification

- We can use “regular expressions” to indicate several joint points (wildcards)
  - Improve expressiveness
- Types of wildcards
  - ***** represents any expression; the number of characters does not matter
  - **+** represents all subclasses of a class
  - **..** represents parameters; the number of parameters does not matter

Examples of Quantification

- All calls to every method of the Client class, but methods must have a single parameter and type of return is void
  - **call(void Client.*(*))**
- All calls to every method in Account beginning with “set” and the number of parameters does not matter
  - **call(* Account.set*(..))**

More Examples

- Calls to a methods beginning with set in the Account class or in its subclasses
  - **call(* Account+.set*(..))**
- Read accesses to every attribute in the Client class, but its type must be String
  - **get(String Client.*)**
- Write accesses to every attribute in every class (entire system)
  - **set(* *.*)**

Pointcuts

Pointcut

- A pointcut is a set of join points
  - It is a mean to identify (name) the join points
- Join points can be composed by using logic operators
  - **&&** (and) intercepts a join point when both conditions are satisfied
  - **||** (or) intercepts a join point when at least one condition is satisfied
  - **!** (not) intercepts all join points that do not match the condition

Examples of Pointcuts

- All calls to the set methods of the Client and Account classes (anonymous)
  - **call(* Client.set*(*)) || call(* Account.set*(*))**

- Equivalent, but naming pointcuts
  - pointcut setClient() : call(* Client.set*(*))
  - pointcut setAccount() : call(* Account.set*(*))
  - pointcut sets() : setClient() || setAccount()

The *Logging* Example

```
public class Account {

  private String number;
  private double balance;

  public void withdraw(double value) {
    if (getBalance() >= value)
      setBalance(getBalance() - value);
    System.out.println("Withdraw done.");
  }

  public void credit(double value) {
    setBalance(getBalance() + value);
    System.out.println("Credit done.");
  }

  ...
}
```

Code in red implements logging and should be separated from the base code

Defining Pointcuts: Keywords

Pointcut and call are keywords

```
pointcut logCredit() :
   call (* Account*.credit(double));

pointcut logWithdraw() :
   call (* Account*.withdraw(double));
```

Defining Pointcuts: Join Points

All calls to the credit method in all classes beginning with Account

```
pointcut logCredit() :
    call (* Account*.credit(double));

pointcut logWithdraw() :
    call (* Account*.withdraw(double));
```

All calls to the withdraw method in all classes beginning with Account

Several Account Classes

```
public class Account {
   ...
}

public class AccountPlus extends Account {
   ...
}

public class AccountReserve extends Account {
   ...
}

public class AccountKids extends Account {
   ...
}
```

All classes beginning with Account

Bibliography

- R. LADDAD. **AspectJ in Action**, 2ª Ed. 2010.
  - Part 1 Understanding AOP and AspectJ