

Impact analysis for AOP

Group 3

Motivation

- Make a change to a system
 - what is the impact of that change?
 - modification of 'base' code vs. aspect
- Assist in semi-automated refactoring
- Can we help a user choose between implementation alternatives?
- Determine effects of weaving an aspect to a woven system

Change Analysis

- What are the potential consequences of the change?
 - categorized consequences
- Which consequences are detectable?
- Which consequences can be analysed to be safe?

Impact analysis

Problem / Consequence	Detectable?	Analysis possible?
change in advice matching <ul style="list-style-type: none">- static- dynamic	<ul style="list-style-type: none">- yes- no	cannot analyse impact of arbitrary code block
multiple matches at one joinpoint	yes	if declare precedence, ok otherwise human intervention
declare precedence	yes	trivial case (no multiple matches), otherwise present places with multi-matches

Impact analysis

Problem / Consequence	Detectable?	Analysis possible?
around advice <ul style="list-style-type: none">- arguments- control flow- return value	Fuzzy Boolean <ul style="list-style-type: none">- Never or Maybe	very hard - human intervention
declare parents: <ul style="list-style-type: none">- instance of- change method/field lookup	yes	trivial cases : <ul style="list-style-type: none">- no instance of tests,- no overridden members
inter-type declarations <ul style="list-style-type: none">- can hide members	yes	no

Impact analysis

Problem / Consequence	Detectable?	Analysis possible?
pointcut used in perXXX aspect	yes	no
declare soft - can create problems for future changes	yes	yes

Making analysis easier

- different types of aspects
 - add invariants that can be statically analysed
 - may reduce scope of checking
 - should be part of language
- involve the user in decisions
 - maybe have tool learn