7 Impact analysis (IA)

• Determines the strategy and impact of change
• Classes identified in concept location are the initial impact set
• Class dependencies are analyzed, and impacted classes are added to the impact set
• Produces estimated impact set
Initial and estimated impact set
Example personal schedules
Example personal schedules

- Jacob
- Henry
- Chuck
- Pat
- James
- Bobby
- Jane
Example personal schedules
Example personal schedules

Jacob
  ↓
  Henry
  ↓
  Chuck

Pat
  ↓
Bobby

James

Jane
Example personal schedules
Class interactions

• Two classes interact if they have something in common
  – One depends on the other
    • There is a contract between them
  – They coordinate
    • They share the same coding, schedule, etc.

• Interactions propagate change
  – In both directions
    • From A to B or from B to A
Class Interaction Graph

- \( G = (X, I) \)
  - \( X \) ... set of classes
  - \( I \) ... set of interactions
- Neighborhood of class A
  \[ N(A) = \{ B \mid (A, B) \in I \} \]
Interactions caused by dependencies

![Diagram showing interactions caused by dependencies]
Interactive IA
Coordination

class C {
    A a; // gets the color code
    B b; // paints the screen
    void foo() {
        b.paint(a.get()); // dataflow between a and b
    }
};
Dependency diagram, Interaction diagram

Dependencies

Interactions
Neighborhood of \textit{Item}
# Status of components (marks)

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>The class was never inspected and is not scheduled for an inspection.</td>
</tr>
<tr>
<td>Changed</td>
<td>The programmers inspected the class and found that it is impacted by the change.</td>
</tr>
<tr>
<td>Unchanged</td>
<td>The programmers inspected the class and found that it is not impacted by the change.</td>
</tr>
<tr>
<td>Next</td>
<td>The class is scheduled for inspection.</td>
</tr>
</tbody>
</table>
A simplified IA process

1. Create interaction diagram and mark all classes as BLANK.
2. Mark the class located during concept location as CHANGED.
3. Mark all BLANK neighbors as NEXT.
4. Are there any classes marked as NEXT?
   - [No]: Mark class as UNCHANGED.
   - [Yes]: Select a class among the classes marked as NEXT.
     What is the new mark for this class?
     - [UNCHANGED] 
     - [CHANGED]

© 2012 Václav Rajlich  
Software Engineering: The Current Practice  Ch. 7
Propagating class: Mailman

• John loaned money to Paul
  – needs the money back
  – His situation changed
• John writes a letter to Paul
  – Mailman takes the letter from John to Paul
  – Paul must take a part-time job
  – a big change that propagated from John to Paul
Interactions

• John interacts with the mailman
• the mailman interact with Paul
• The change originated with John and propagates through the mailman to Paul.
  – mailman is in the middle of the propagation
  – he does not have to change anything
  – he just keeps delivering the letters from one person to another.
## Status of components (marks)

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>The class was never inspected and is not scheduled for an inspection.</td>
</tr>
<tr>
<td>Changed</td>
<td>The class is impacted by the change.</td>
</tr>
<tr>
<td>Propagating</td>
<td>The class is not impacted, a client or supplier of the class is impacted.</td>
</tr>
<tr>
<td>Unchanged</td>
<td>The class is inspected but not impacted by the change.</td>
</tr>
<tr>
<td>Next</td>
<td>The class is scheduled for inspection</td>
</tr>
</tbody>
</table>
Iterative IA

Color codes

<table>
<thead>
<tr>
<th>Unknown</th>
<th>Changed</th>
<th>Next</th>
<th>Unchanged</th>
<th>Propagating</th>
</tr>
</thead>
</table>

© 2012 Václav Rajlich

Software Engineering: The Current Practice  Ch. 7  20
Example: Point of Sale
Change request

- Record cashier sessions

- A cashier session
  - total cash and all sales
  - during the time between the cashier logging in and out.
Example: Point of Sale

```
Store

Inventory

Item

Price

CashierRecord

Cashiers
```
Example: Point of Sale

Cashiers

CashierRecord

Store

Inventory

Item

Price
Example: Point of Sale

Cashiers

CashierRecord

Store

Inventory

Item

Price
Example: Point of Sale

- Cashiers
- CashierRecord
- Store
- Inventory
- Item
- Price
IA including propagating classes

Create interaction diagram and mark all classes as BLANK

Mark the class located during concept location as CHANGED

Mark all BLANK neighbors as NEXT

Are there any classes marked as NEXT?

Select a class marked as NEXT. What is the new mark for this class?

[Yes] [No]

Mark class as UNCHANGED

[UNCHANGED]

Mark class as PROPAGATING

[PROPAGATING]

Mark class as CHANGED

[CHANGED]
Alternatives in software change

• Program displaying a temperature in Fahrenheit
  – change request: display it in Celsius

• Two separate locations deal with temperature
  – sensor data converted to the temperature
  – temperature displayed to the user

• The change can be done in either place
  – impact analysis weights these alternatives
The criteria

• Required effort of the change
• Clarity of the resulting code
• Often, these two criteria contradict each other
  – it is easier to adjust the user interface
  – it is better to have all calculations of the temperature in one place
• Conflict between short-term and long-term goals
Interactive IA

Create interaction diagram and mark all classes as BLANK

Mark the class located during concept location as CHANGED

Mark all BLANK neighbors as NEXT

Are there any classes marked as NEXT?

Mark class as INSPECTED

Mark class as PROPAGATING

Mark class as CHANGED

Select a class marked as NEXT. What is the new mark for this class?

© 2012 Václav Rajlich  Software Engineering: The Current Practice  Ch. 7
Tool support: JRipples

- Automatically identifies and suggests components to be inspected Next

- Keeps track of
  - What was done (Changed, Propagating, Unchanged)
  - What needs to be done (Next)

- Eclipse plug-in
  - Java
  - 15,000 LOC
  - 150 Classes
  - 2,500 Methods

http://jripples.sourceforge.net
### JRipples GUI

![JRipples Hierarchical View](image)

<table>
<thead>
<tr>
<th>Component</th>
<th>Mark</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractOptionPane</td>
<td>Continue</td>
<td>8</td>
</tr>
<tr>
<td>AbbrevsOptionPane</td>
<td>Continue</td>
<td>7</td>
</tr>
<tr>
<td>HeaderMouse</td>
<td>Visited</td>
<td>3</td>
</tr>
<tr>
<td>ActionHandler</td>
<td>Visited</td>
<td>2</td>
</tr>
<tr>
<td>_save</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>_init</td>
<td></td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>Visited</td>
<td></td>
</tr>
<tr>
<td>edit</td>
<td>Visited</td>
<td></td>
</tr>
<tr>
<td>globalAbbrevs</td>
<td>Continue</td>
<td></td>
</tr>
<tr>
<td>AbbrevsModel</td>
<td>Next</td>
<td>2</td>
</tr>
<tr>
<td>AboutDialog</td>
<td>Changed</td>
<td></td>
</tr>
</tbody>
</table>
JRipples marks and Eclipse