

IFT2255 - Génie logiciel

Analyse de l'impact

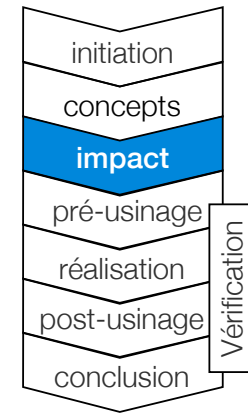
(basé sur le matériel de Václav Rajlich)

Bruno Dufour
dufour@iro.umontreal.ca

Analyse de l'impact

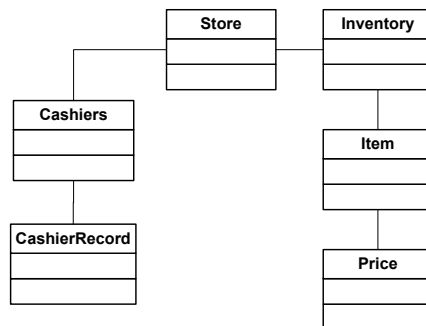
2

- Analyse de l'impact
 - Détermine la stratégie à utiliser et l'impact des changements à apporter
 - Les classes identifiées lors de la localisation des concepts forment l'**ensemble d'impact** initial
 - Les dépendances sont analysées et d'autres classes sont ajoutées à l'ensemble d'impact



Exemple : Point de vente

3



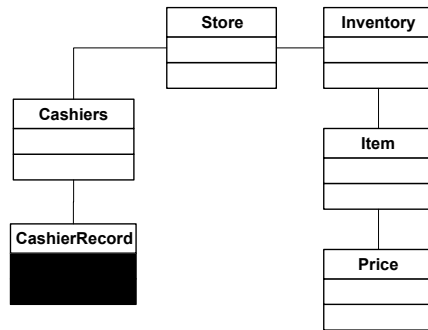
Requête de changement

4

- Enregistrer les sessions d'un caissier
 - Une session débute lorsqu'un caissier se connecte et se termine lorsqu'il se déconnecte
 - La session enregistre toutes les ventes et ainsi que le montant total en argent

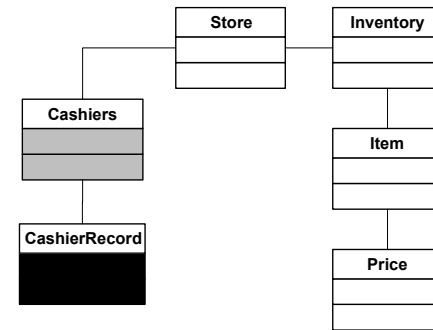
Exemple : Point de vente

5



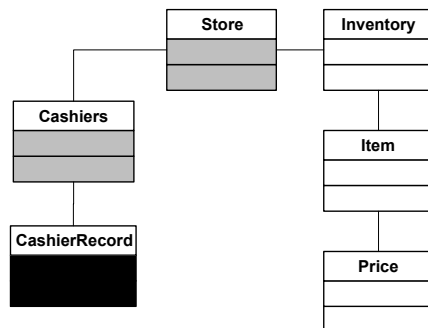
Exemple : Point de vente

6



Exemple : Point de vente

7



Estimer l'impact

8

- L'analyse d'impact permet d'obtenir un ensemble d'impact estimé
 - À partir d'un changement initial et de changements secondaires nécessaires
 - "ripple effect"
- Après que les changements aient été effectués, l'ensemble modifié contient toutes les classes qui ont été changées
 - Ensemble des classes réellement impactées
- Plus l'analyse est précise, plus l'ensemble modifié est près de l'ensemble prédit

Localisation des concepts et AI

9

- La localisation des concepts et l'analyse d'impact utilisent de l'information similaire mais effectuent des tâches différentes
- La localisation des concepts est un processus itératif qui vise à identifier un emplacement dans le code source
 - Le processus s'arrête lorsque l'emplacement est identifié
- L'analyse d'impact construit l'ensemble d'impact de façon itérative et systématique
 - Il est difficile de savoir lorsque l'ensemble complet a été découvert

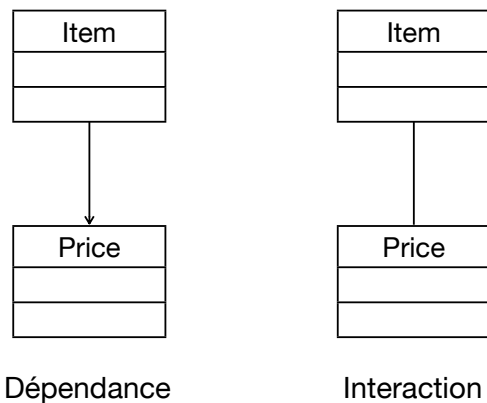
Interactions

10

- Deux classes interagissent si elles ont quelque chose en commun
 - Une classe dépend de l'autre
 - Il existe un contrat entre elles
 - Les classes coordonnent
 - ex: elles partagent le même codage
- Les interactions propagent les changements
 - dans les deux sens : $A \rightarrow B$ et $B \rightarrow A$

Graphe d'interaction

11

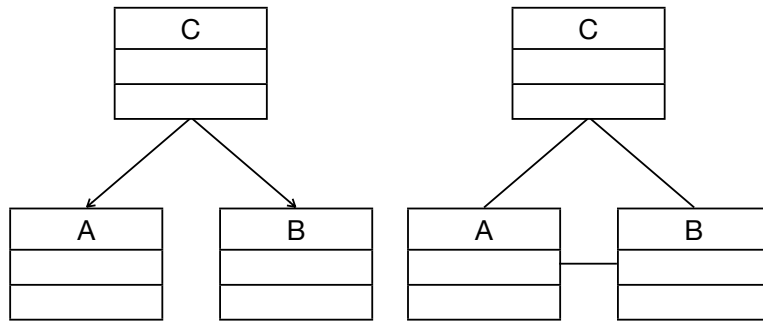


Coordination

12

```
class C {  
    A a; // gets the color code  
    B b; // paints the screen  
  
    void draw() {  
        b.paint(a.get());  
    }  
}
```

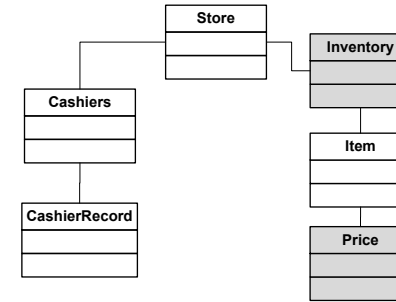
Coordination



Dépendances

Interactions

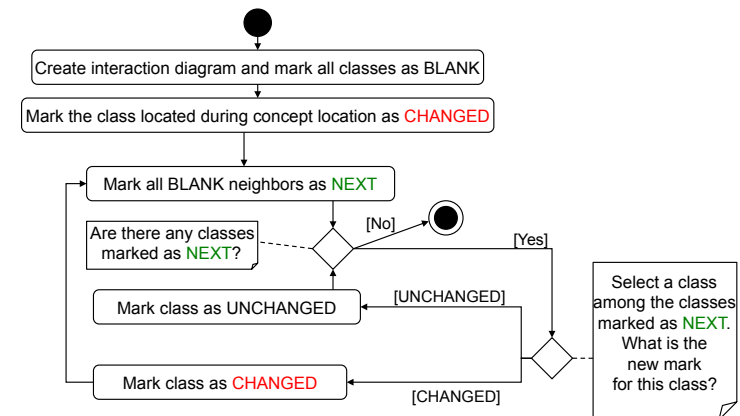
Voisins de Item



Statut des classes

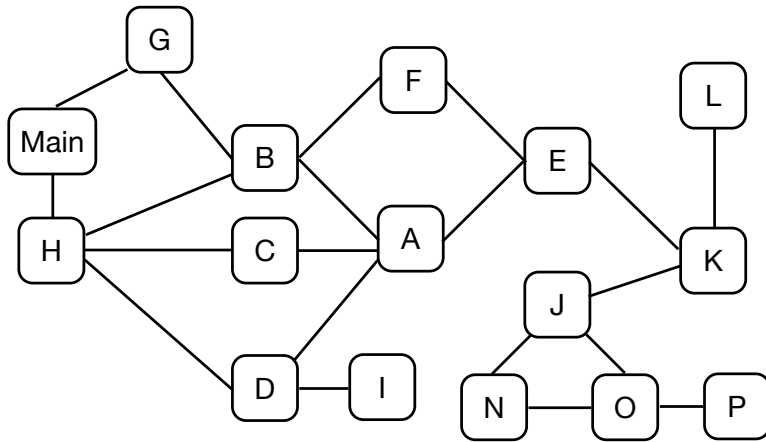
Vide	Classe non-inspectée et qui n'apparaît pas dans la liste de classes à inspecter.
Changée / Impactée	Un programmeur a inspecté la classe et a déterminé qu'elle était impactée par le changement.
Non-impactée	Un programmeur a inspecté la classe et a déterminé qu'elle n'était pas impactée par le changement.
Suivante	La classe apparaît dans la liste de classes à inspecter.

Algorithme simplifié



IA itérative

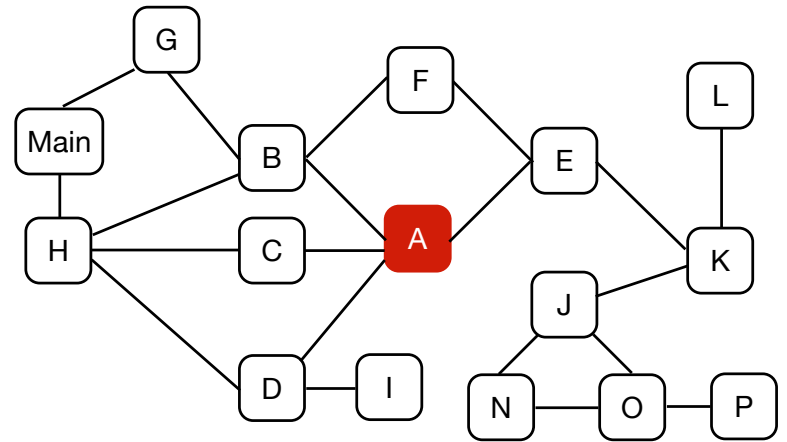
17



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

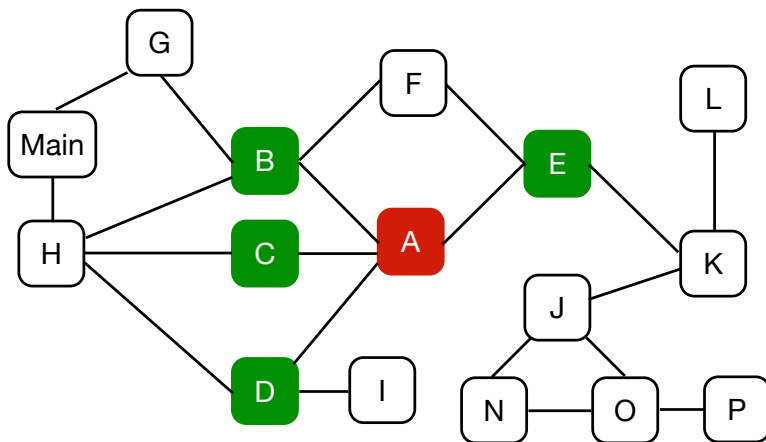
18



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

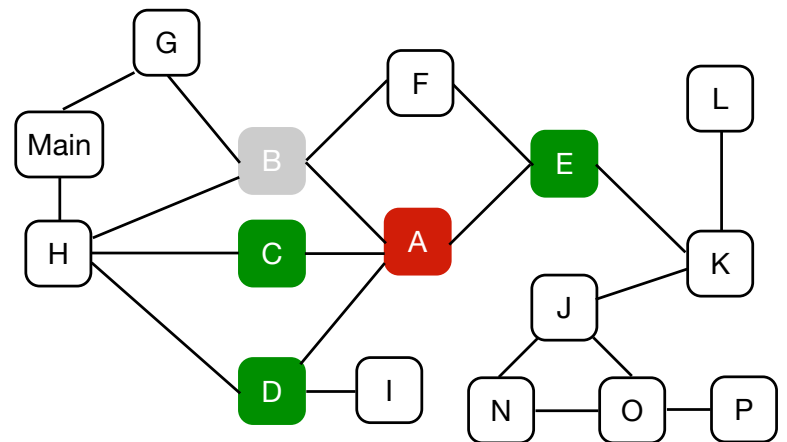
19



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

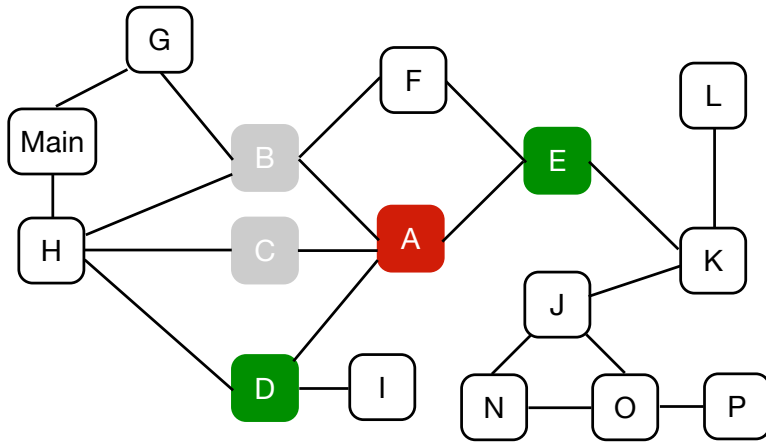
20



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

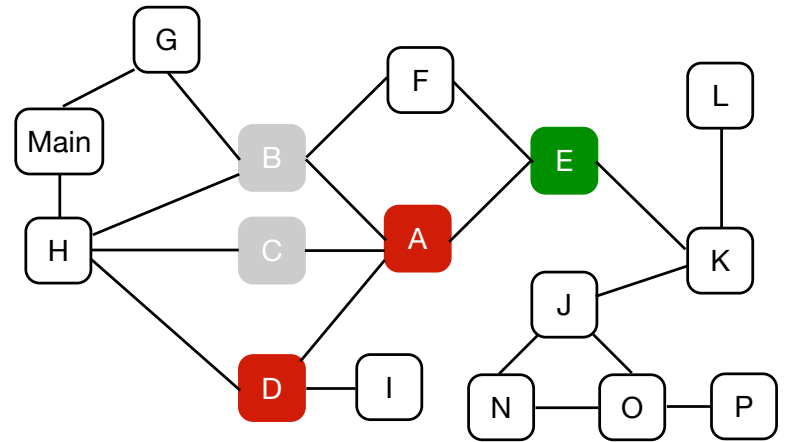
21



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

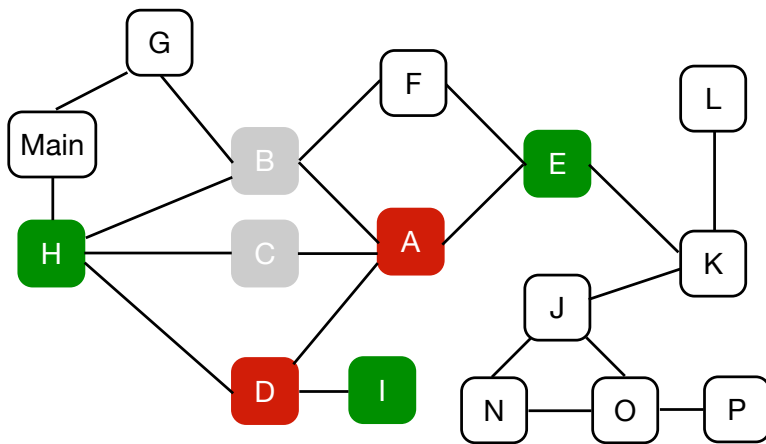
22



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

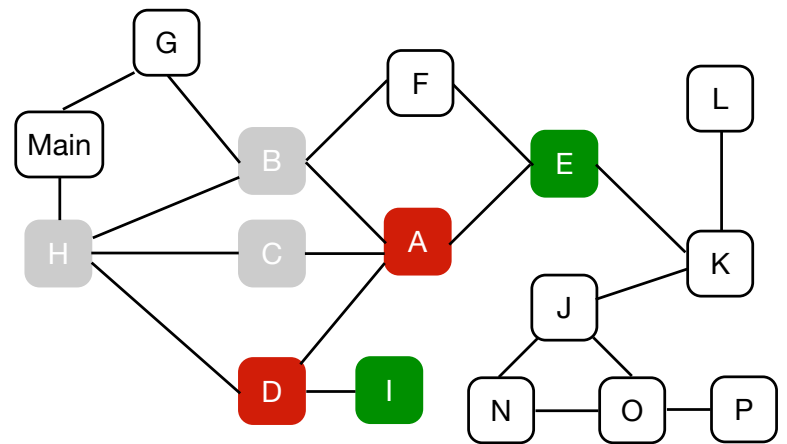
23



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

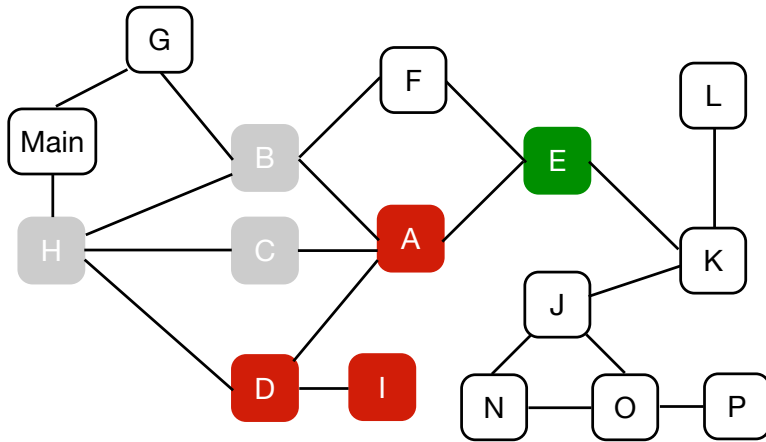
24



○ Inconnu ● Impacté ● À inspecter ○ Non-impacté

IA itérative

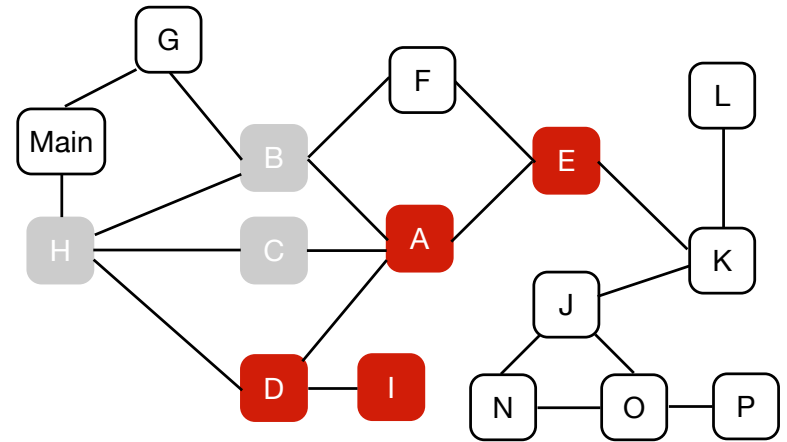
25



Legend: Inconnu (white), Impacté (red), À inspecter (green), Non-impacté (grey)

IA itérative

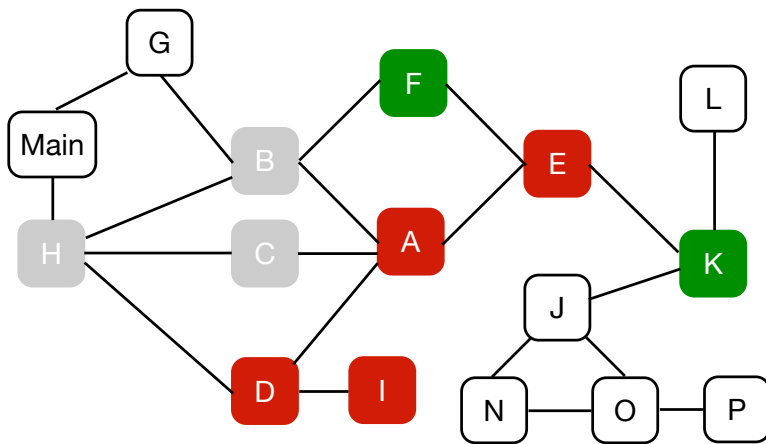
26



Legend: Inconnu (white), Impacté (red), À inspecter (green), Non-impacté (grey)

IA itérative

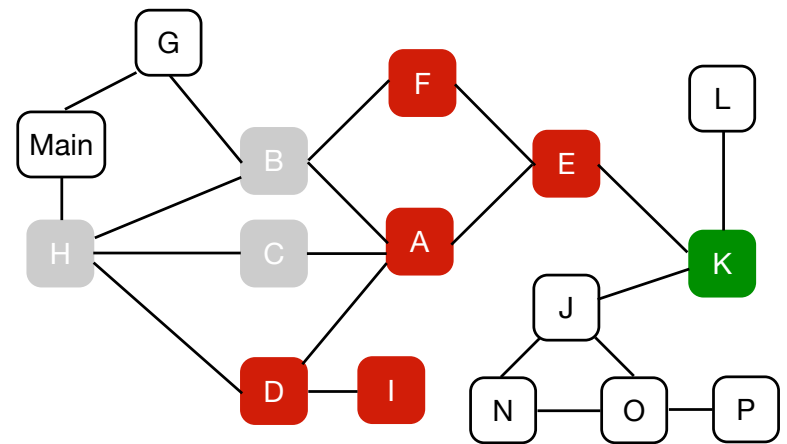
27



Legend: Inconnu (white), Impacté (red), À inspecter (green), Non-impacté (grey)

IA itérative

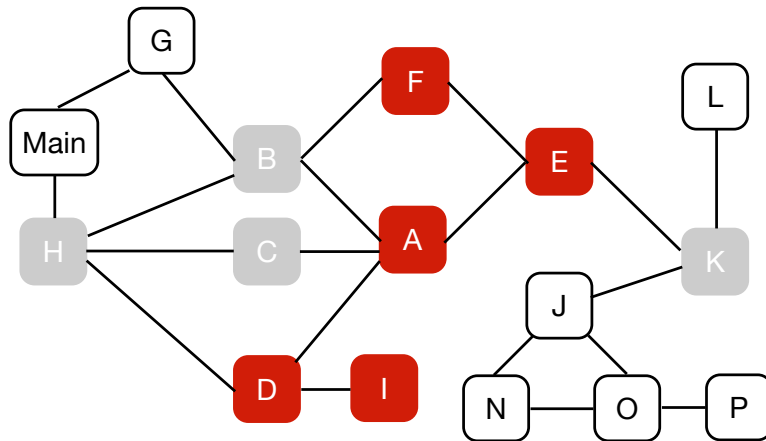
28



Legend: Inconnu (white), Impacté (red), À inspecter (green), Non-impacté (grey)

IA itérative

29



○ Inconnu ● Impacté ● À inspecter ● Non-impacté

Bruno Dufour - Université de Montréal

Classes propageant les changements

30

- Analogie : lettre à la poste
 - Alice a prêté de l'argent à Bob
 - La situation d'Alice change, elle a maintenant besoin de l'argent
 - Alice écrit à Bob pour demander l'argent
 - Le changement de situation d'Alice est propagé à Bob

Bruno Dufour - Université de Montréal

Interactions

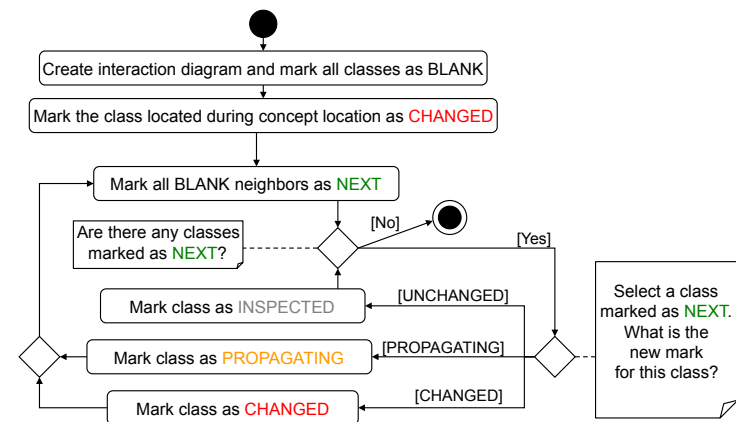
31

- Alice interagit avec le facteur
- Le facteur interagit avec Bob
- Le changement provient d'Alice et se propage à Bob par le facteur
 - Le facteur propage le changement
 - Le facteur ne change pas, il continue à livrer les lettres comme d'habitude

Bruno Dufour - Université de Montréal

Algorithme incluant la propagation

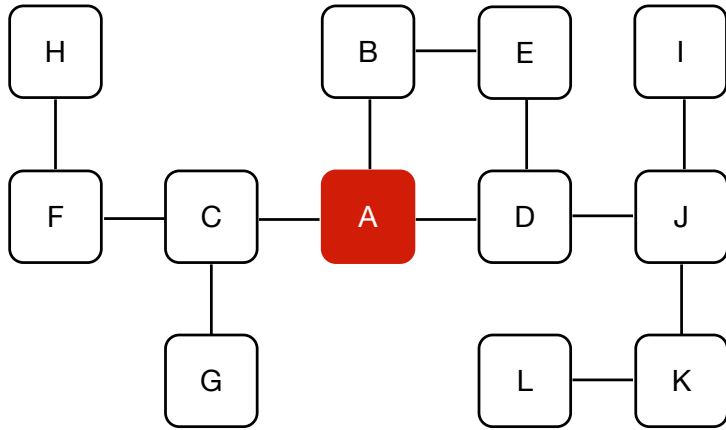
32



Bruno Dufour - Université de Montréal

AI itérative

33

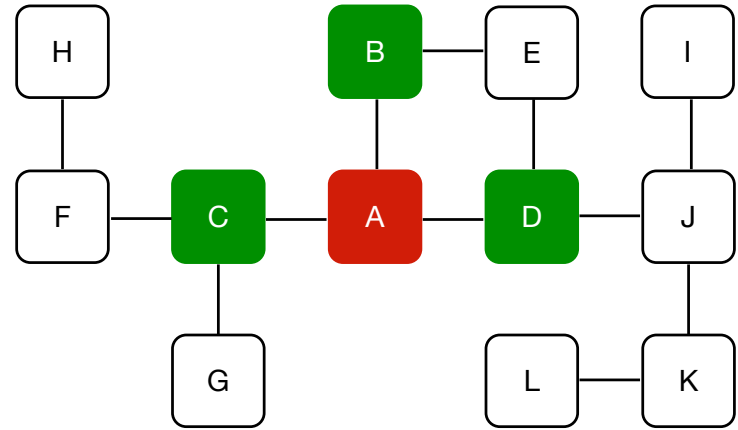


○ Inconnu ● Impacté ● À inspecter ○ Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

34

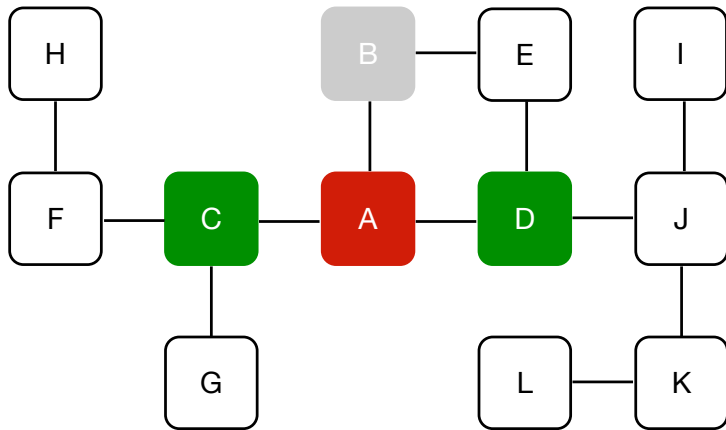


○ Inconnu ● Impacté ● À inspecter ○ Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

35

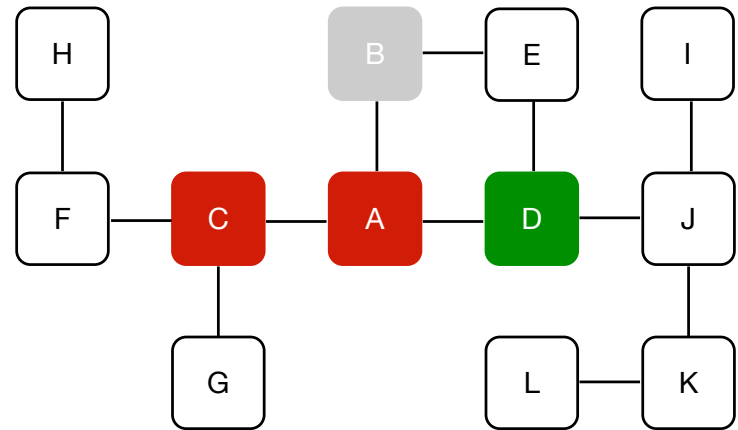


○ Inconnu ● Impacté ● À inspecter ○ Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

36

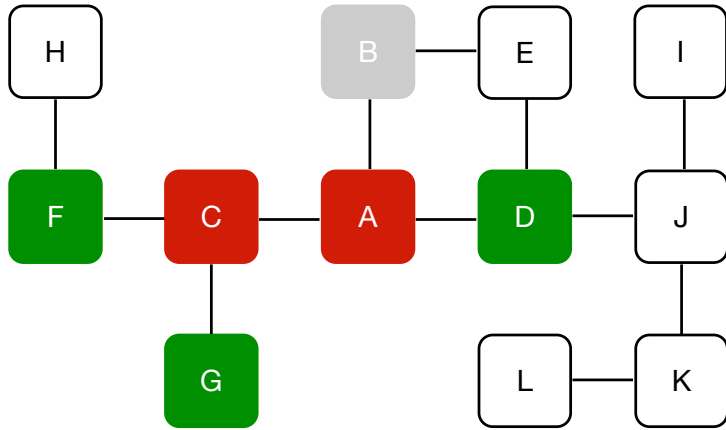


○ Inconnu ● Impacté ● À inspecter ○ Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

37

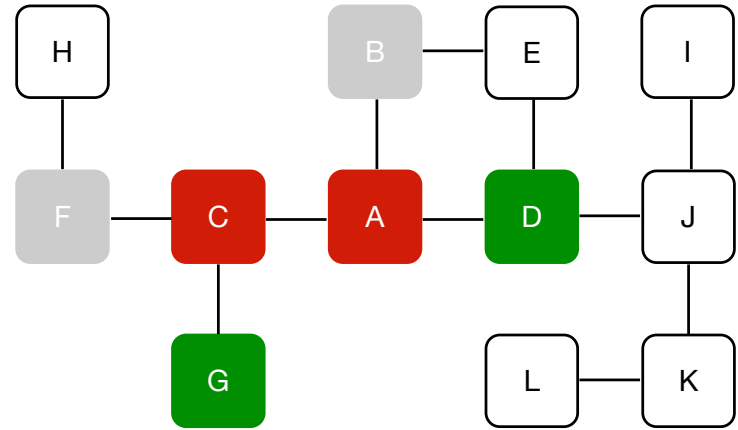


○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

38

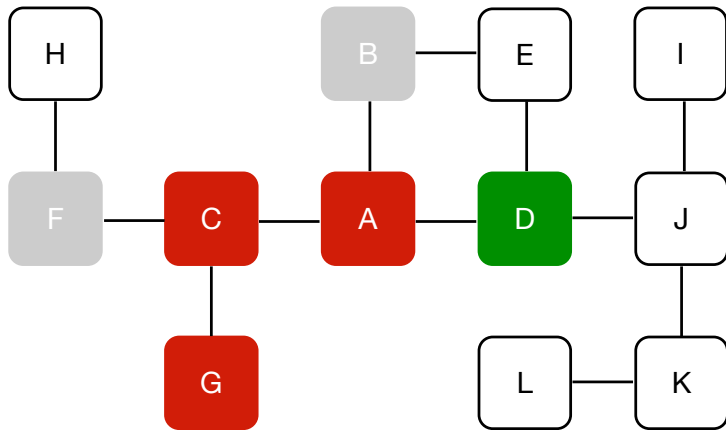


○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

39

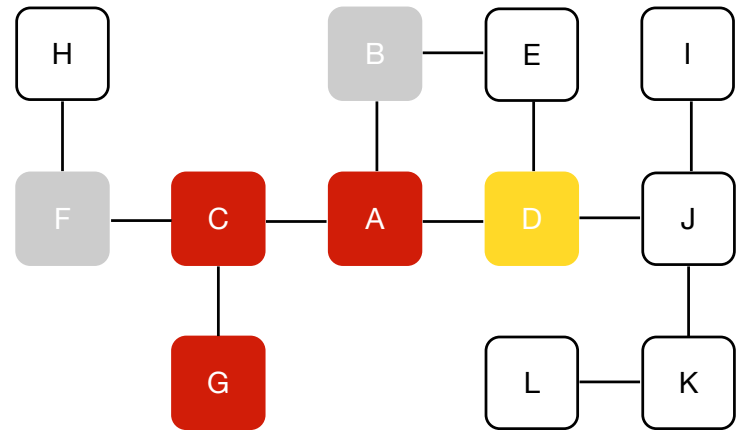


○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

40

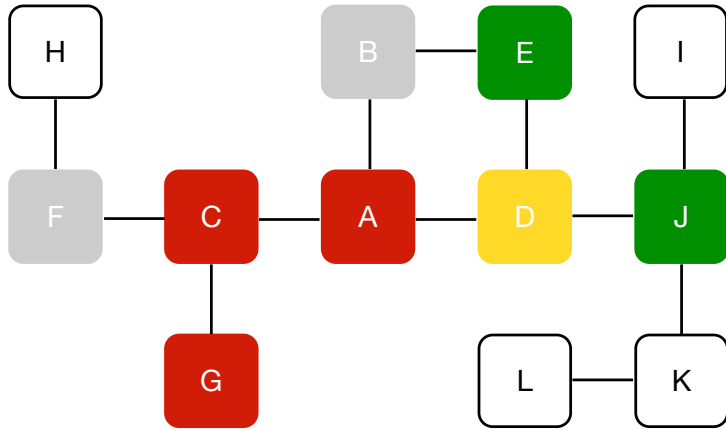


○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

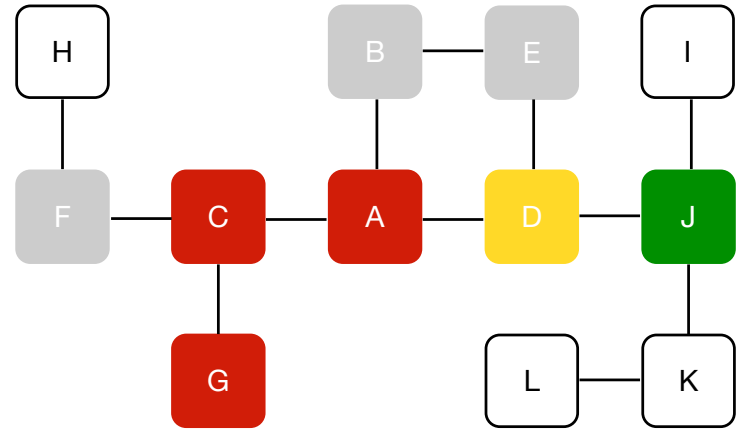
41



○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

AI itérative

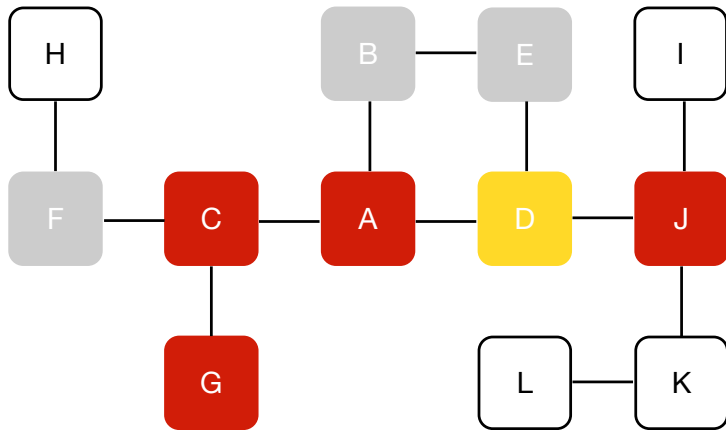
42



○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

AI itérative

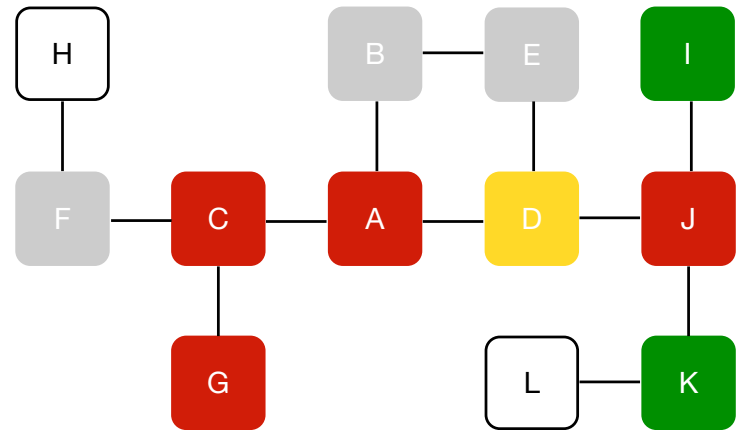
43



○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

AI itérative

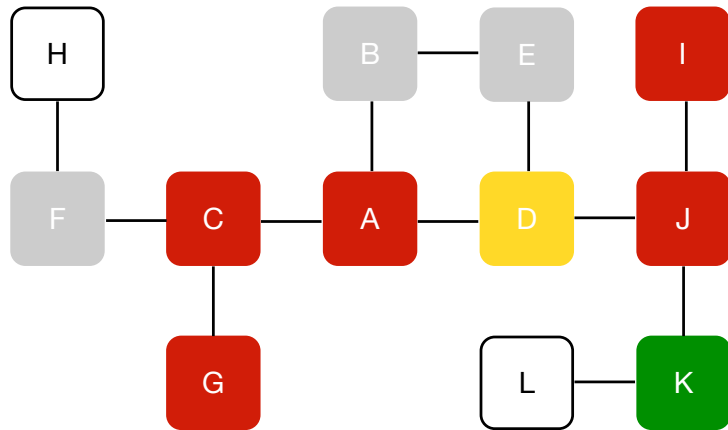
44



○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

AI itérative

45

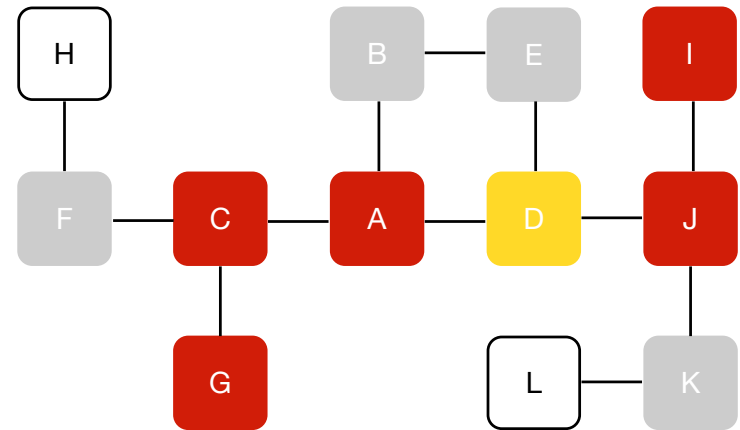


○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

Bruno Dufour - Université de Montréal

AI itérative

46



○ Inconnu ● Impacté ● À inspecter ● Non-impacté ● Propage

Bruno Dufour - Université de Montréal

Alternatives

47

- Une requête de changement peut souvent être implémentée de plusieurs façons
- Ex : un logiciel qui affiche la température en Celcius
 - Requête : afficher la température en Fahrenheit
 - 2 emplacements possibles
 - Conversion à partir du capteur de température
 - Affichage à l'utilisateur
 - Le changement peut être effectué aux deux endroits
 - L'analyse d'impact permet de comparer ces alternatives

Bruno Dufour - Université de Montréal

Critères de comparaison

48

- Effort requis pour le changement
- Qualité du code résultant
- Ces critères sont souvent en contradiction
 - Il peut être plus facile de modifier l'interface
 - Il peut être bénéfique de centraliser le calcul de température

Bruno Dufour - Université de Montréal

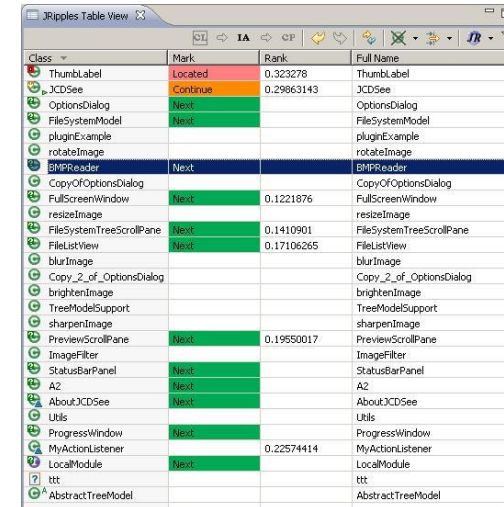
Outils pour l'analyse d'impact

49

- Certains outils supportent le processus d'exploration manuelle
 - ex: JRipples, XRay
- Certains outils peuvent prédire l'impact à partir du code source
 - Se basent sur plusieurs indicateurs (peut inclure l'historique du projet)
- Certains outils peuvent décomposer un changement complexe pour analyser l'impact individuel de chaque changement élémentaire
 - ex: Chianti

JRipples

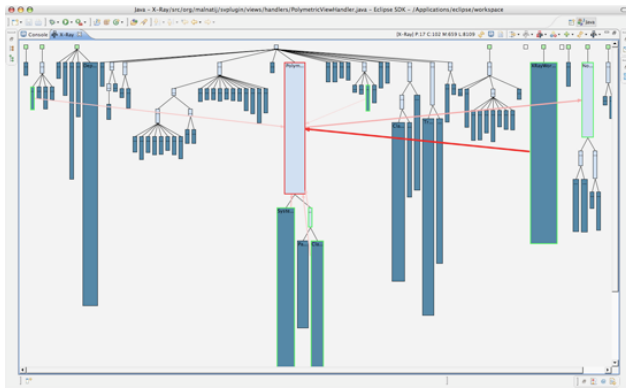
50



Class	Mark	Rank	Full Name
ThumbLabel	Located	0.323278	ThumbLabel
JCDSee	Continue	0.29863143	JCDSee
OptionsDialog	Next		OptionsDialog
FileSystemModel	Next		FileSystemModel
pluginExample	Next		pluginExample
rotateImage	Next		rotateImage
BMPReader	Next		BMPReader
CopyOfOptionsDialog	Next		CopyOfOptionsDialog
FullScreenWindow	Next	0.1221876	FullScreenWindow
resizeImage	Next		resizeImage
FileSystemTreeScrollPane	Next	0.1410901	FileSystemTreeScrollPane
ListView	Next	0.17106265	ListView
blurImage	Next		blurImage
Copy_2_of_OptionsDialog	Next		Copy_2_of_OptionsDialog
brightenImage	Next		brightenImage
TreeModelSupport	Next		TreeModelSupport
sharpenImage	Next		sharpenImage
PreviewScrollPane	Next	0.19550017	PreviewScrollPane
ImageFilter	Next		ImageFilter
StatusBarPanel	Next		StatusBarPanel
A2	Next		A2
AboutJCDSee	Next		AboutJCDSee
Utils	Next		Utils
ProgressWindow	Next		ProgressWindow
MyActionListener	Next	0.22574414	MyActionListener
LocalModule	Next		LocalModule
ttt	Next		ttt
AbstractTreeModel	Next		AbstractTreeModel

XRay

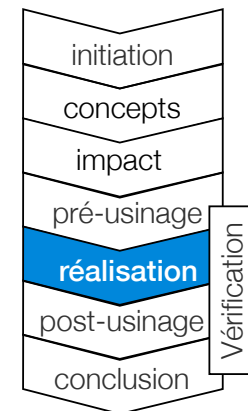
51



Réalisation

52

- Réalisation
 - Ajout de nouveau code qui interagit avec le code existant
 - L'impact de l'ajout ou de modifications se répercute sur les classes voisines
 - Propagation des changements
 - Ondulation (*ripple effect*)



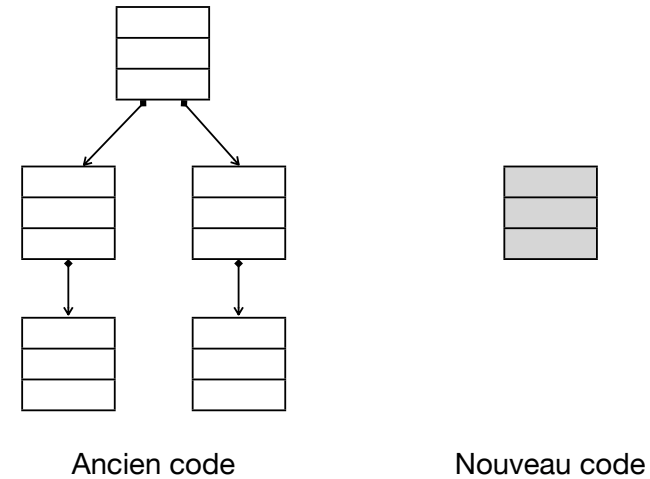
Effectuer les changements

53

- Petits changements
 - Effectués directement dans le code existant
- Changements importants
 - Ajout de nouvelles classes
 - Le changement peut se propager à d'autres éléments du système

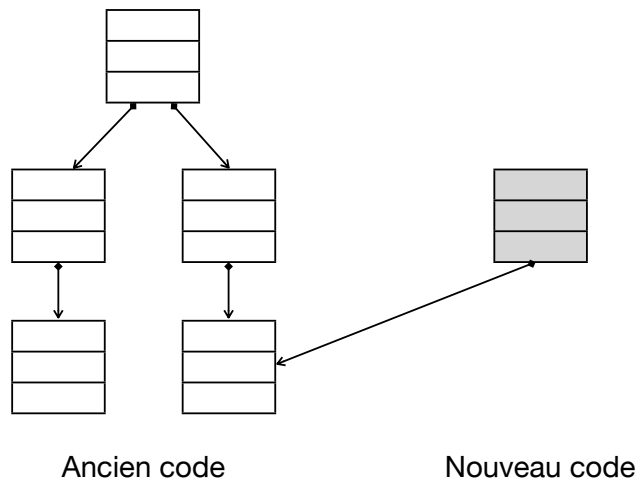
Ajout de fonctionnalité locale

54



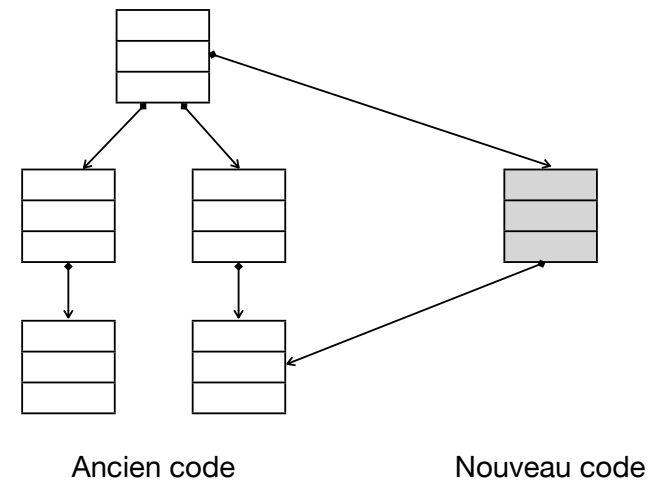
Ajout de fonctionnalité composite

55



Ajout d'un fournisseur

56



- Peut aussi propager le changement
- Toutes les références à la fonctionnalité retirée doivent aussi être retirées
 - Les changements secondaires se propagent à d'autres classes

- L'analyse de l'impact prédit les classes qui seront impactées
- La propagation de changements modifie le code des classes impactées
 - L'analyse d'impact peut surestimer ou sous-estimer les changements à effectuer
 - conséquence de l'invisibilité du logiciel
 - Rend la planification difficile