RhôneAlpes Région

Fiabiliser la réutilisation des patrons par une approche orientée complétude, variabilité et généricité des spécifications

Présentée par : Nicolas ARNAUD

Directrice : Dominique RIEU
Co-Directrice : Agnès FRONT

vendredi 17 octobre 2008

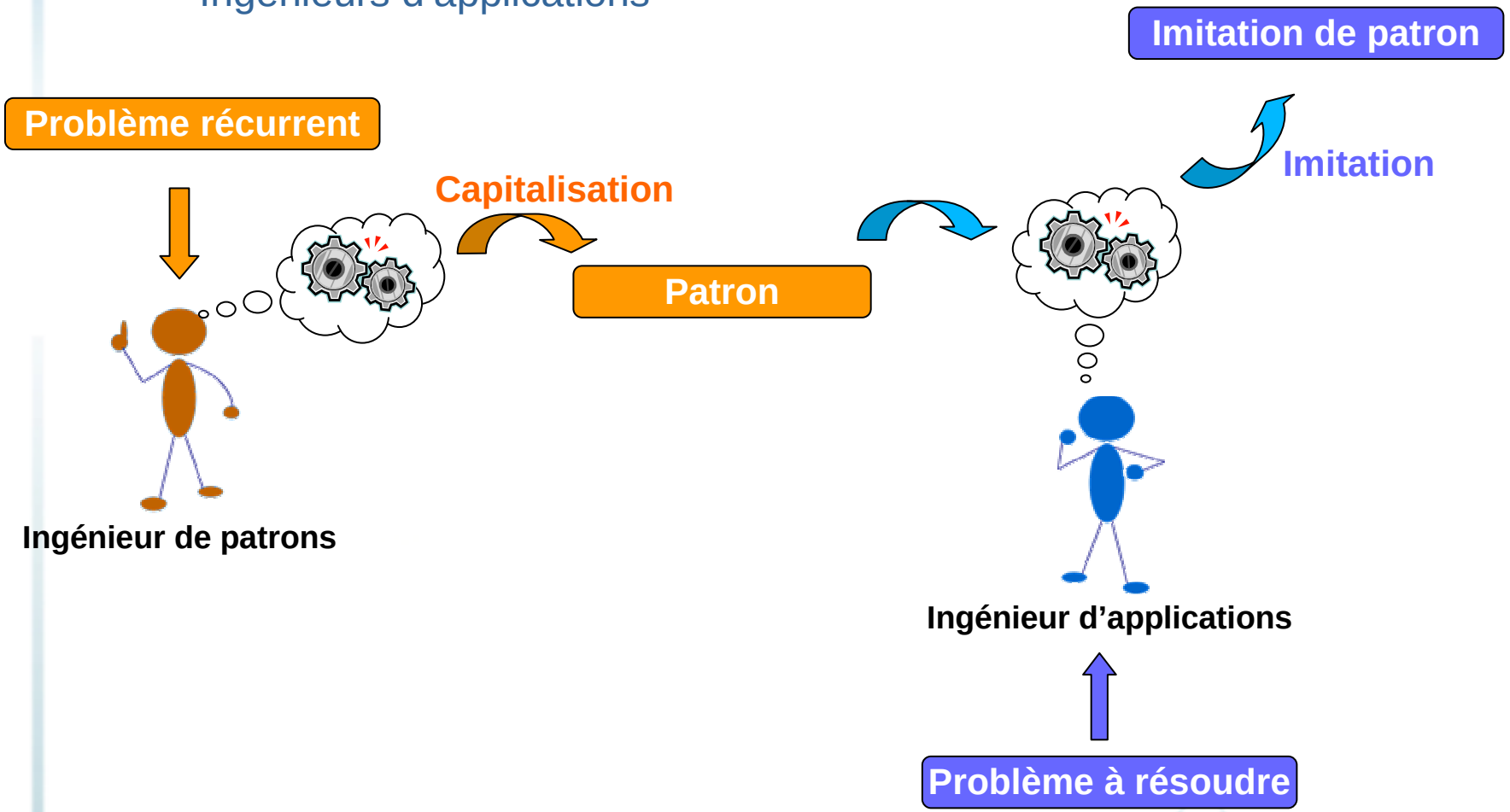
Les patrons

- Patron : forme de capitalisation des savoirs et savoir-faire
  - *« Chaque patron décrit à la fois un **problème** qui se produit très fréquemment dans un **contexte** et l'architecture de la **solution** à ce problème de telle façon que vous puissiez utiliser cette solution des millions de fois sans jamais l'adapter deux fois de la même manière»*
    [Alexander, 1979]

- Dans l'ingénierie logicielle (Analyse & Conception)
  - Principalement orientés objet
  - Un modèle statique (diagramme de classes)
  - Parfois un modèle dynamique (diagramme de séquences)
  - Descriptions textuelles
  - …

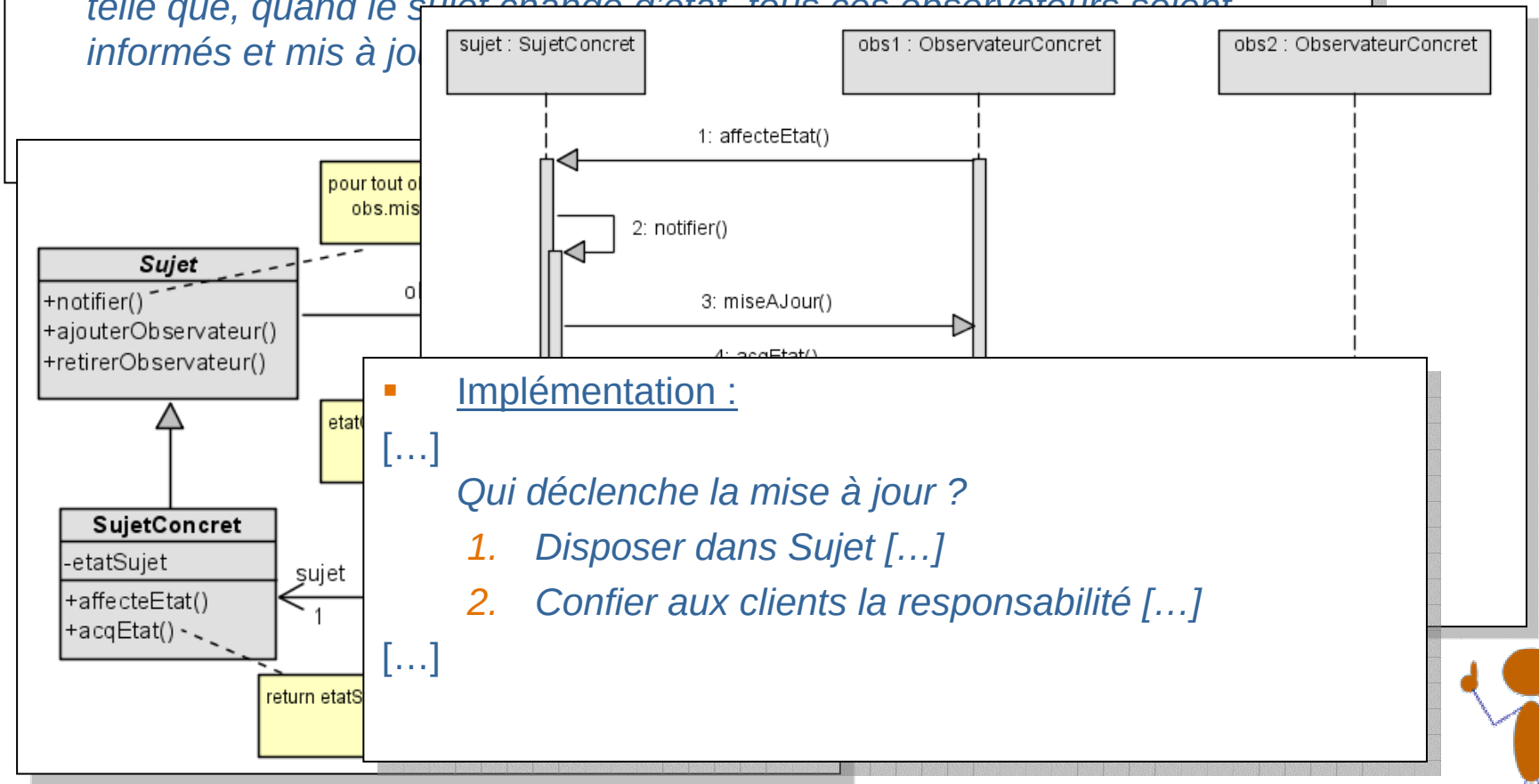
Usage des patrons

- Deux rôles
  - Ingénieurs de patrons
  - Ingénieurs d'applications



Le patron «Observateur»

- Intention : *Définir une dépendance entre les observateurs d'un même sujet telle que, quand le sujet change d'état, tous ses observateurs soient informés et mis à jo*



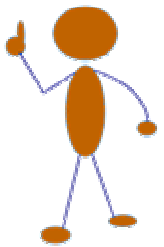
- Implémentation :

[…]

Qui déclenche la mise à jour ?

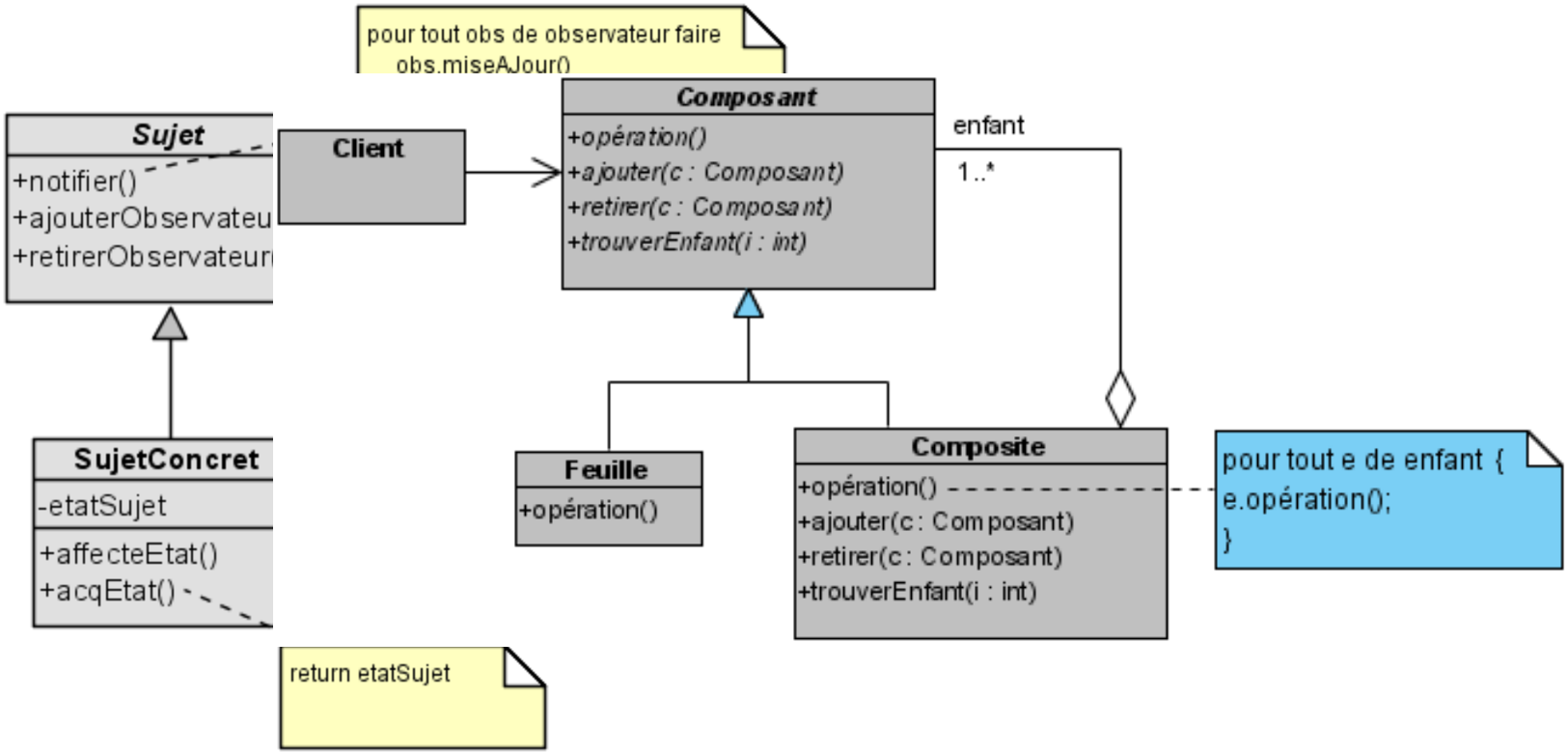
1. *Disposer dans Sujet […]*
2. *Confier aux clients la responsabilité […]*

[…]



E. Gamma
[Gamma et al., 1995]

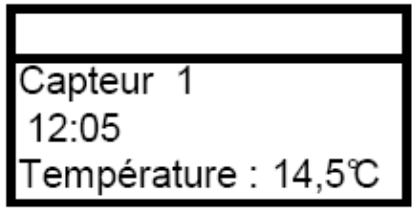
De nombreux patrons et ingénieurs de patrons

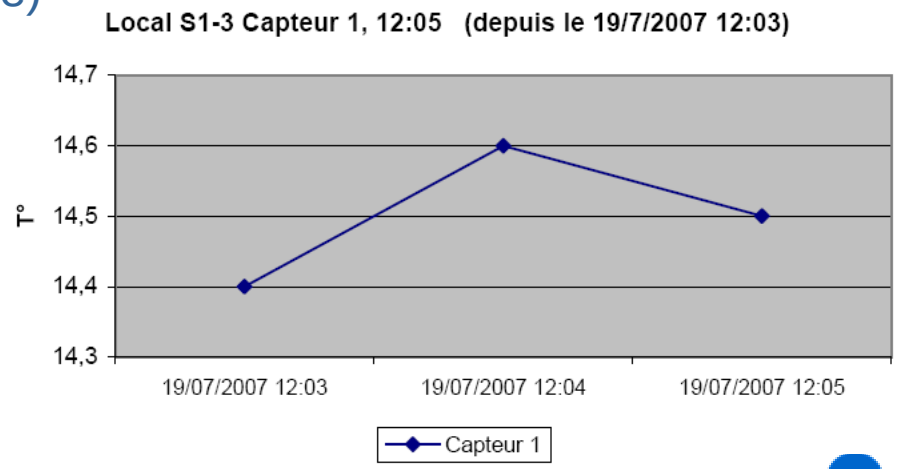


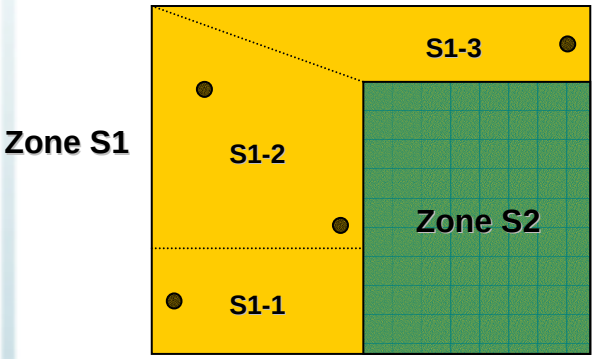
P. Coad
[Coad, 1992]

Des applications et des ingénieurs d'applications

- Système de surveillance de bâtiments industriels
  - Basé sur la mesure de températures ambiantes des locaux
  - Affichage en temps réel des températures
    - Forme Simple (textuelle)
    - Graphique
  - Hiérarchie de locaux







I.A.

Cas d'étude : patrons applicables

- Système de surveillance de bâtiments industriels
  - Basé sur la mesure de températures ambiantes des locaux

Mémoire d'évènements

  - Affichage en temps réel des températures
    - Forme Simple (textuelle)
    - Graphique

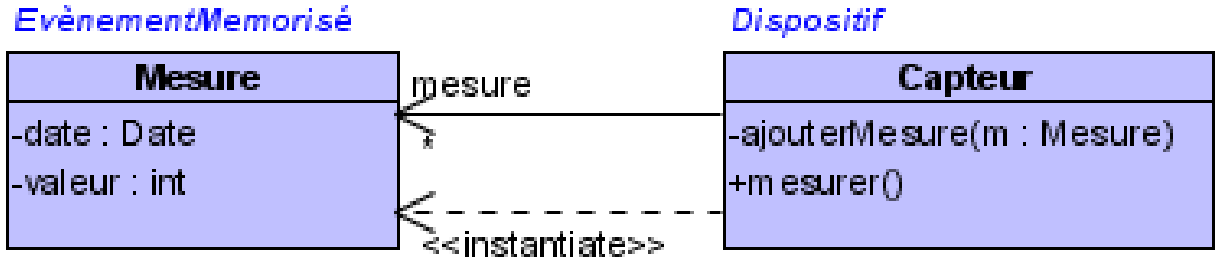
Observateur

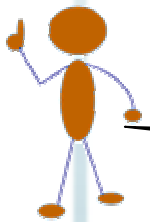
  - Hiérarchie de locaux

Composite

I.A.

Cas d'étude : imitation de « Mémoire d'évènements »





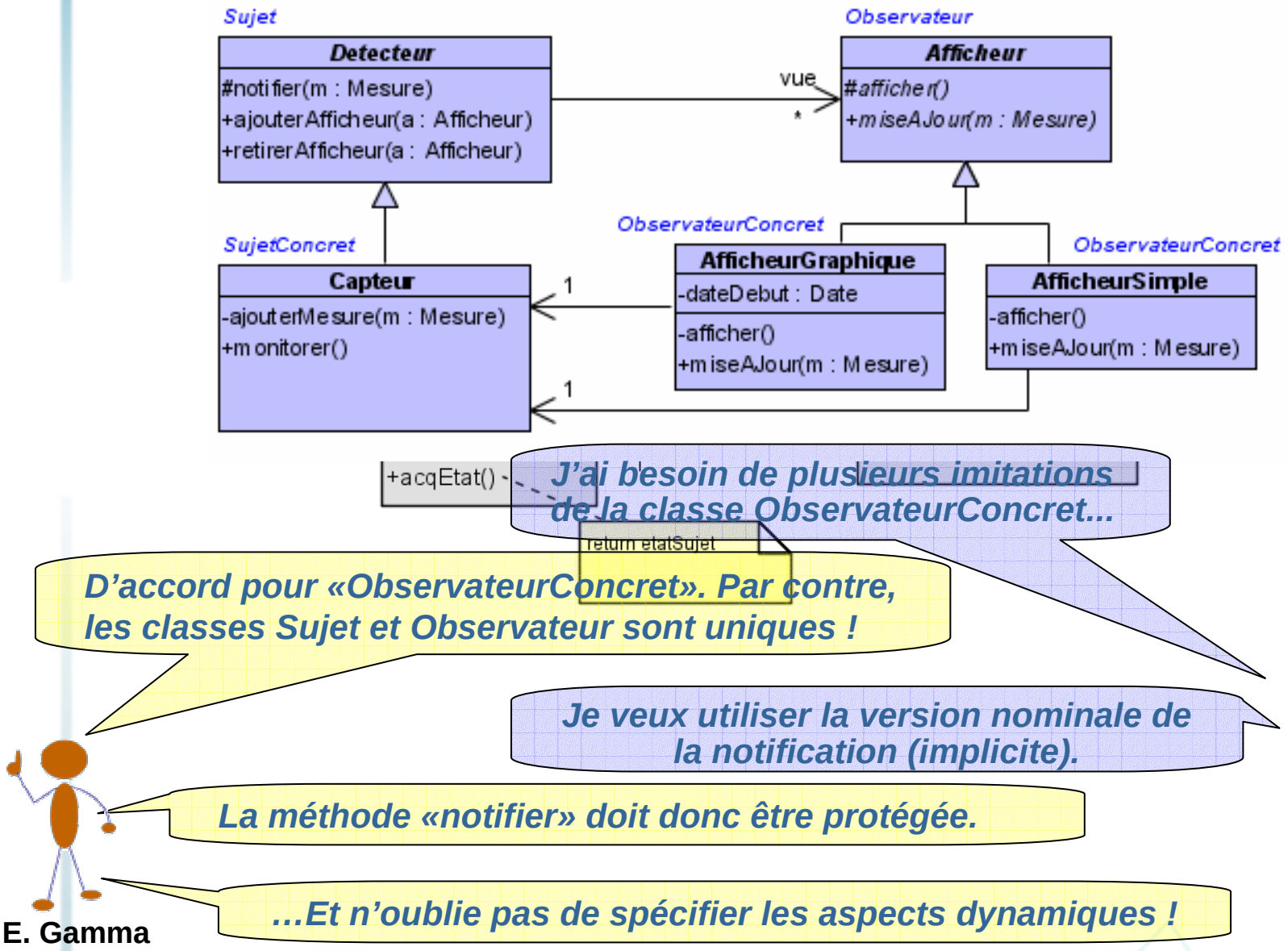
P. Coad

Seulement certains.

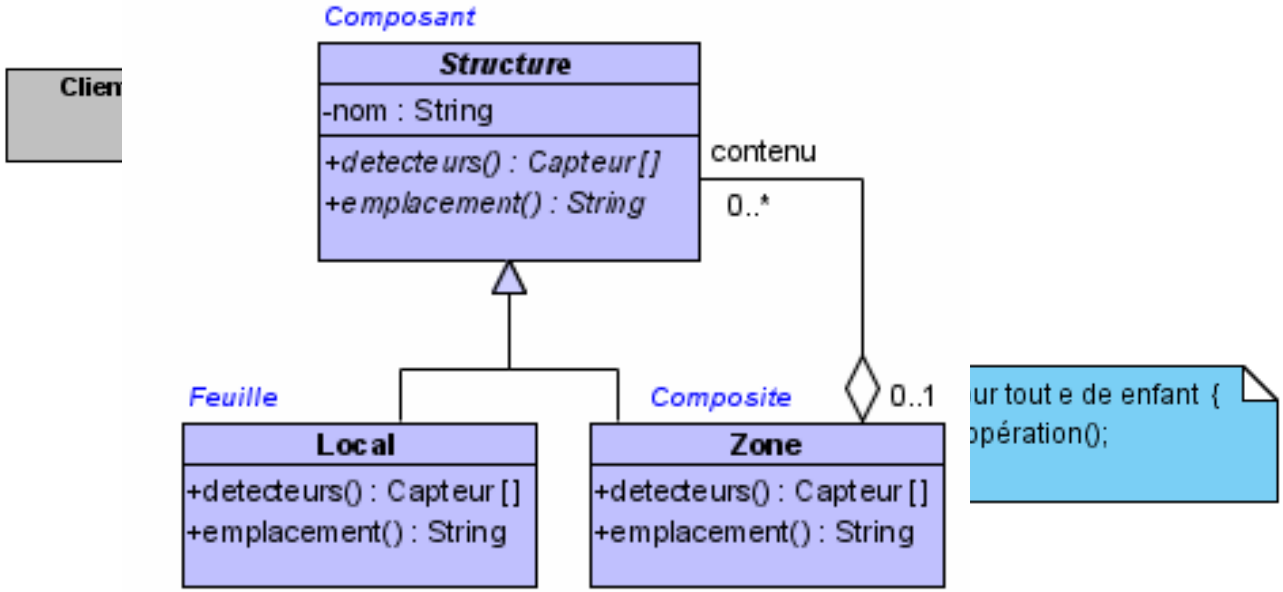
Puis-je supprimer des attributs sans risquer de nuire à la qualité de l'imitation ?

I.A.

Cas d'étude : imitation de « Observateur »



Cas d'étude : imitation de « Composite »



J'ai besoin de plusieurs imitations de la méthode «opération»…

Je n'y vois pas d'inconvénient.

Je n'ai pas besoin des opérations de gestion des enfants…

Elles sont facultatives

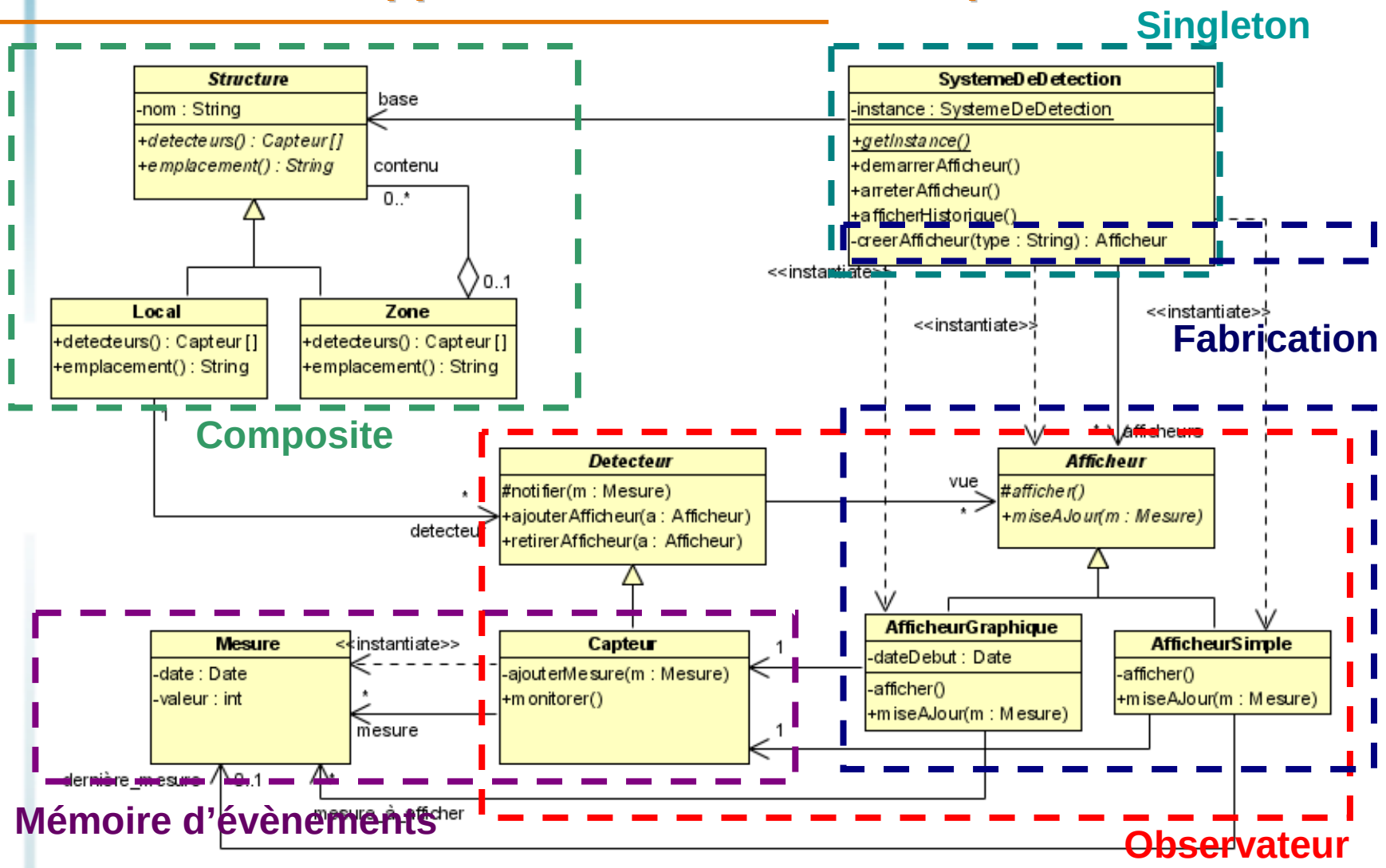
E. Gamma

I.A.

Bilan des problèmes rencontrés lors de l'imitation

- Conséquences des choix de variantes
- Aspect facultatif de certaines fonctionnalités
- Limites d'adaptation de la solution (généricité)
  - Suppression d'un élément
  - Duplication d'un élément
  - …
- Réalisation dynamique
- Composition des imitations

Cas d'étude : applications des différents patrons



Plan

- Cas d’étude
- Problématique et Travaux existants
- Spécifications des solutions
- Processus d’imitation
- Outillage
- Conclusions & Perspectives

Problématique générale

- Objectif : permettre la réalisation d'imitations
  - Correctes (et qui restent correctes)
  - Qui couvrent des vues différentes du système
  - Ajustées fonctionnellement aux besoins (variantes)

- Deux études complémentaires
  - Spécification des solutions (ingénieurs de patrons)
    - Complétude
    - Variabilité
    - Généricité
  - Réutilisation (ingénieurs d'applications)
    - Tirer partie de ces spécifications
    - Traçabilité

Travaux existants : récapitulatif

- Aspects statiqu
- Aspects dynam

- Nombreuses propriétés dans l'existant
- Principe d'
- Nombre d'
- Dimensions
- …

- Variantes d'implémentation
- Variantes fonctionnelles pas ou peu traitées

Quelques Approches	Co			réutilisation
[Maplesden et al., 2001] (DPML)	statique	(dimensions)	✗	✗
[France et al., 2003] (RBML)	statique **dynamique**	(cardinalités et contraintes OCL)	✗	(validation assistée)
[Meijler et al, 1997] (FACE)	statique	(très limitée)	✗	✗
[Albin-Amiot et Guéhéneuc, 2001]	statique **dynamique (code)**	(MM extensible et ajout de contraintes par codage)	✗ (par codage?)	
[Sunye, 1999]	statique **dynamique (code)**	(cardinalités)	(alternatives d'implémentation)	(non décrit)

Variabilité dans l'ingénierie logicielle

- Plusieurs domaines de recherche utilisent la variabilité
  - Lignes de produits. ***[Ziadi et al., 2005], [Clauss, 2001]***
  - Ingénierie des besoins, ***[Bennasri, 2005],***
  - …

- "Un **point de variation** est un endroit du système où des choix doivent être faits afin d'identifier les **variantes** à utiliser."
  [Czarnecki et al., 2000]

- Plusieurs types de variabilité ***[Bachmann et Bass, 2001]***
  - Options,
  - Alternatives,
  - Ensemble d'options
  - Alternatives optionnelles
  - …

Plan

- Cas d'étude
- Problématique et Travaux existants
- Spécifications des solutions
  - Complétude
  - Variabilité
  - Généricité
- Processus d'imitation
- Outillage
- Conclusions & Perspectives

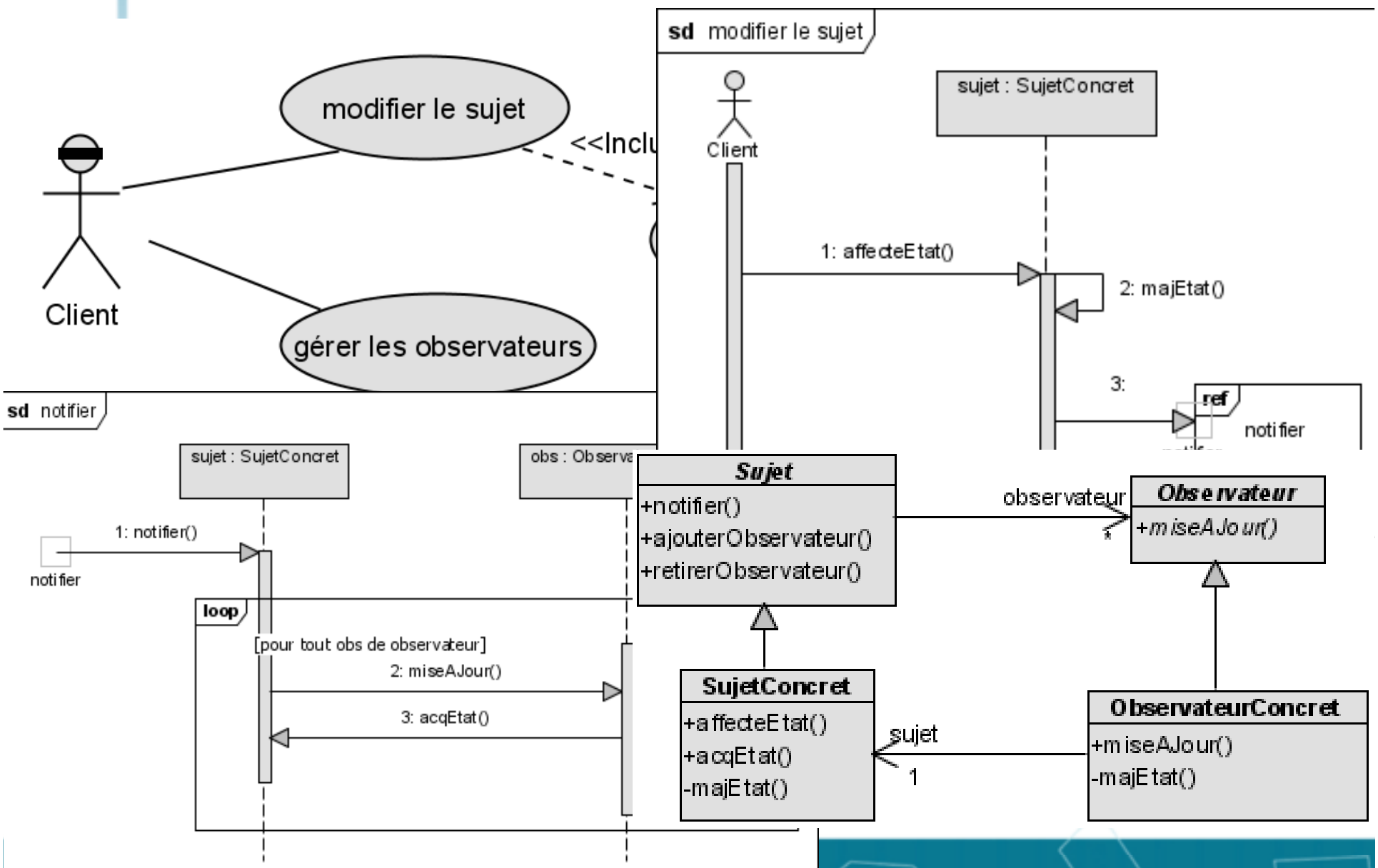
Complétude des solutions

- Un patron → un mini système composé de trois vues

- Vue fonctionnelle
  - Fonctionnalités (cas d'utilisation)
- Vue dynamique
  - Scénarii d'utilisation
  - Déroulement de séquences
  - Algorithmes
- Vue statique
  - Classes
  - Propriétés (opérations, attributs)
  - …

- Spécification à l'aide d'UML
  - Plus aisée pour les ingénieurs de patrons et d'applications
  - Cohérence inter-vues

Patron «Observateur» : mini-système



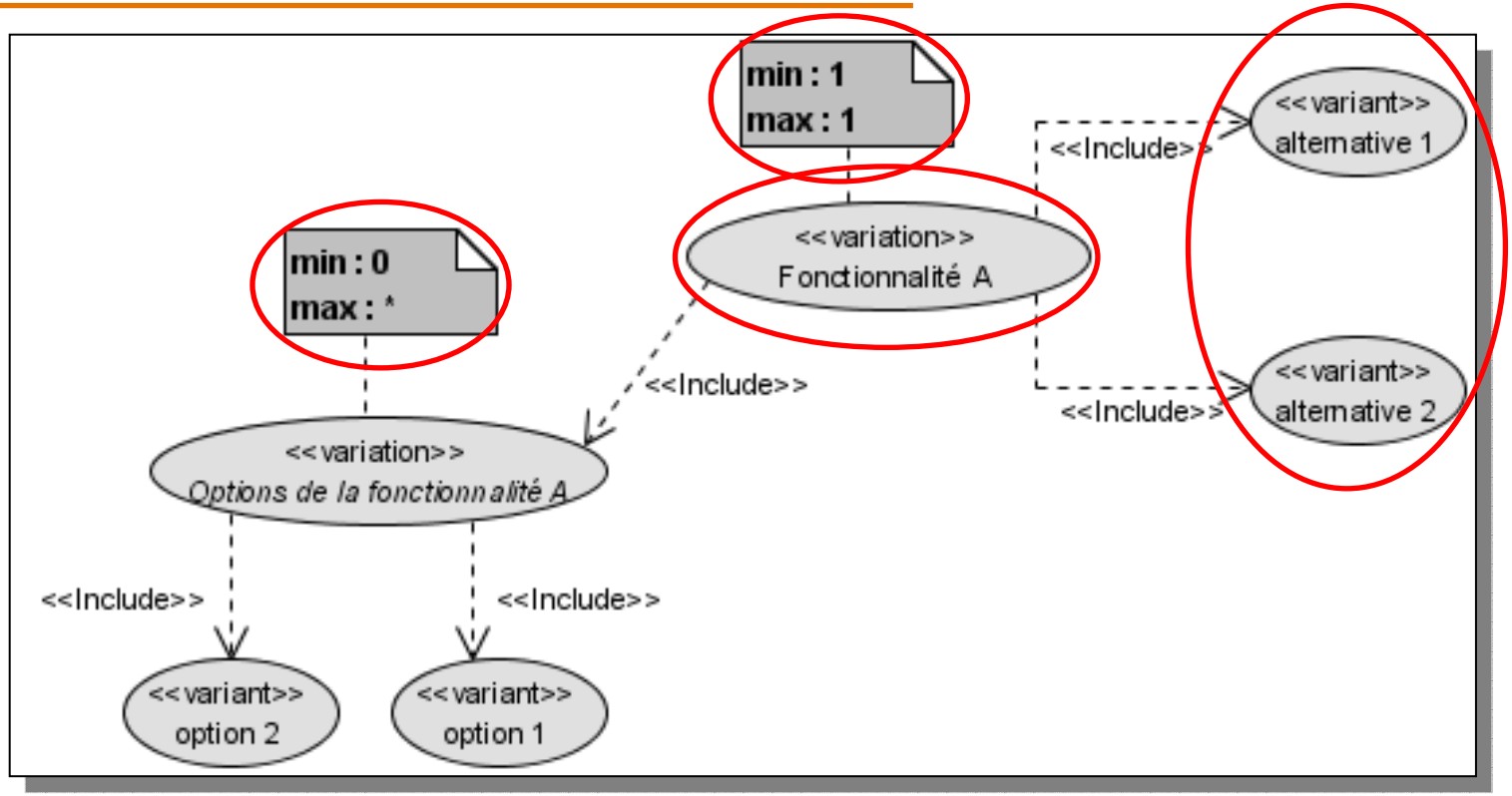
Complétude : bilan

Complétude	▪ vue fonctionnelle ▪ vue dynamique ▪ vue statique
Variabilité	
Généricité	

Variabilité fonctionnelle : concepts

- La variabilité s'exprime sur la vue fonctionnelle du mini-système
  - Pilote la variabilité dans les autres vues
- Deux concepts principaux
  - Point de variation
  - Variante
- Deux types de variabilité
  - Alternatives
  - Options
- Principe de fonctionnalité obligatoire/facultative
- Définition d'un opérateur générique pour la variabilité

Un opérateur générique pour la variabilité fonctionnelle



- Un point de variation (<<variation>>)
- Une ou plusieurs variantes incluse(s) (<<variant>>)
- Des cardinalités min et max associées au PV, définissent le type de variabilité
  - 0..* groupe d'options
  - 1..1 alternative

Fonctionnalités obligatoires/facultatives

- Facultatif = option de premier niveau
  - Racine fonctionnelle
  - La racine est un point de variation admettant des « options » (cardinalité 0..*)
  - Une fonctionnalité facultative est une option de la racine
  - Une fonctionnalité obligatoire est incluse à la racine

Patron «Observateur»

étude de la variabilité

- Informations extractibles de la documentation du patron

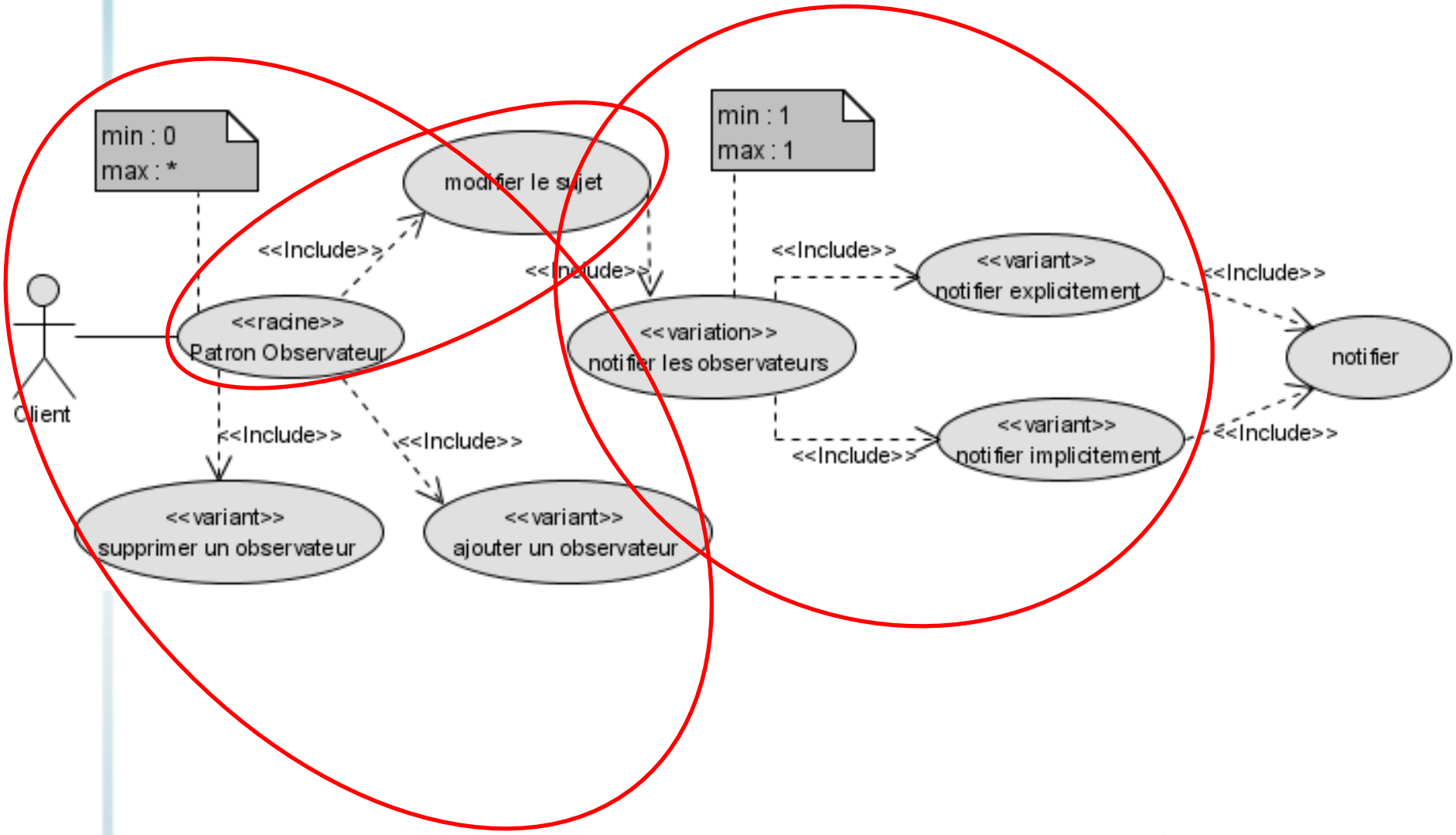
- Fonctionnalité obligatoire : modification du sujet
  - Inclut la notification

- Point de variation : notification
  - Alternative 1 : notification implicite (nominal)
  - Alternative 2 : notification explicite (déclenchée par le client)

- Fonctionnalités facultatives : gestion des observateurs
  - Ajouter un observateur
  - Supprimer un observateur

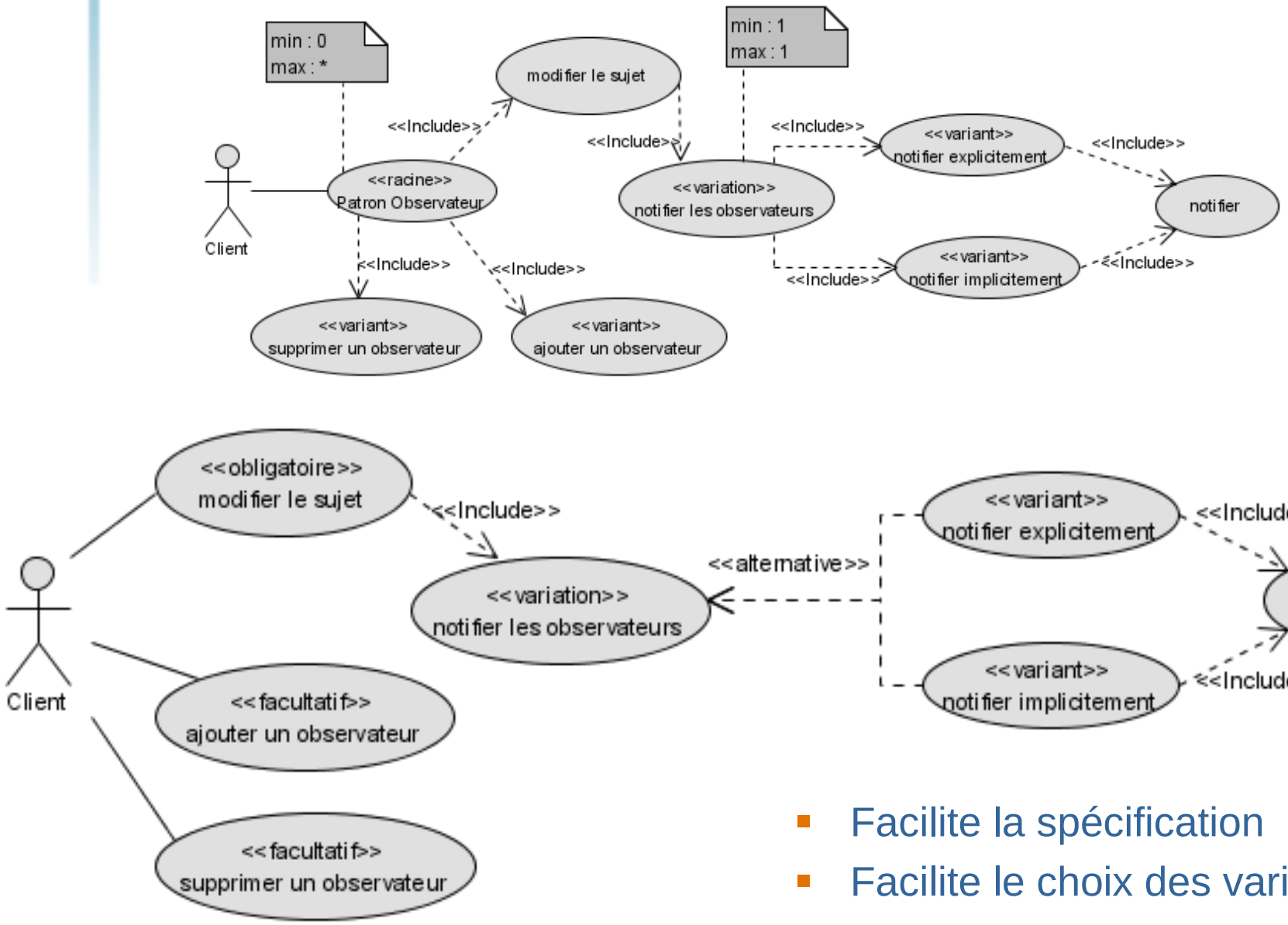
Patron «Observateur»

vue fonctionnelle à variantes avec l'opérateur générique



Patron «Observateur»

vue fonctionnelle à variantes, notation simplifiée



- Facilite la spécification
- Facilite le choix des variantes

Complétude, Variabilité

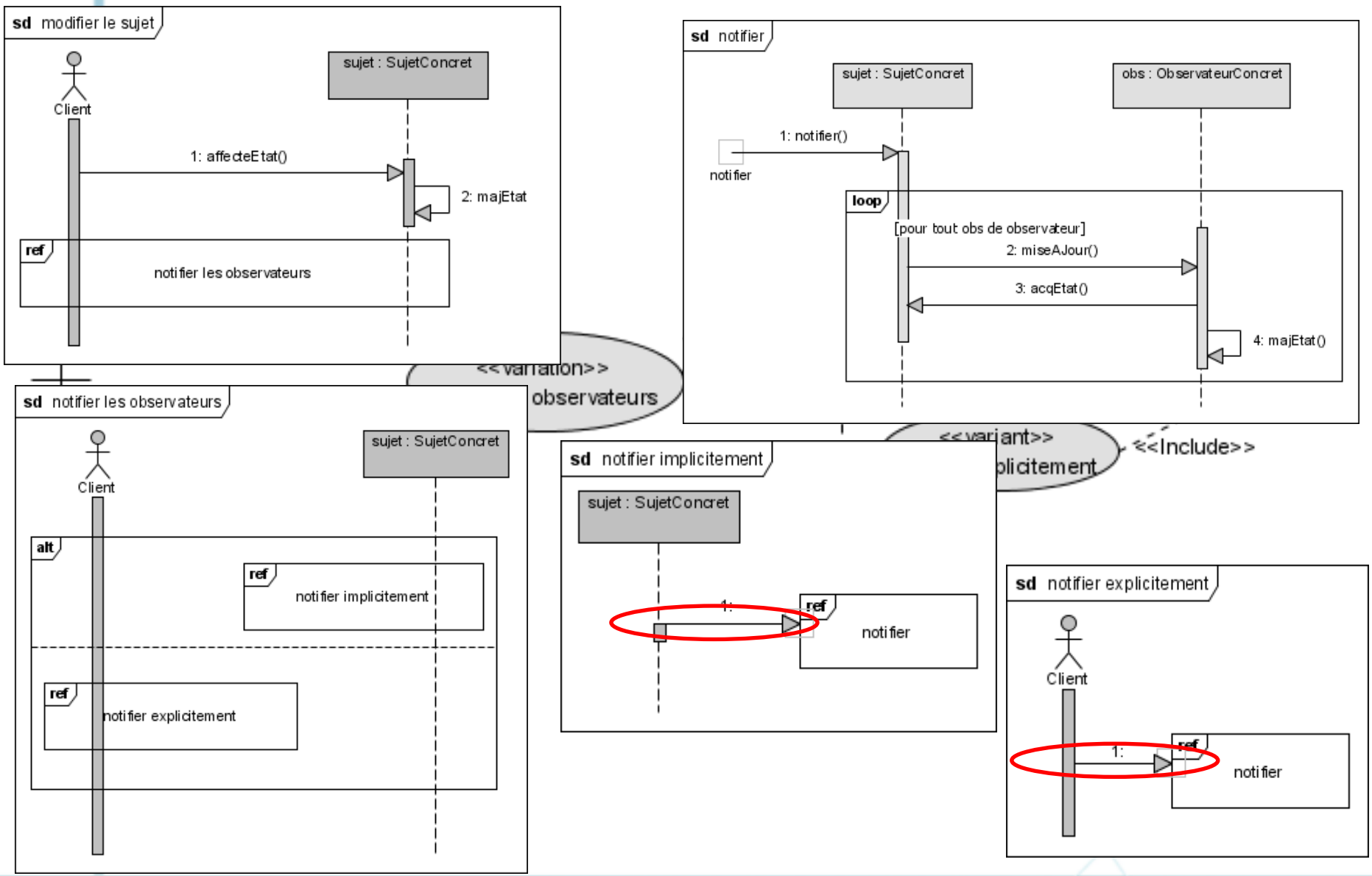
Complétude	▪ vue fonctionnelle ▪ vue dynamique ▪ vue statique
Variabilité	▪ point de variation ▪ variante ▪ alternatives ▪ options ▪ fonctionnalités obligatoire/facultative
Généricité	

- Répercutions sur les deux autres vues ?

Variabilité : impact sur la vue dynamique

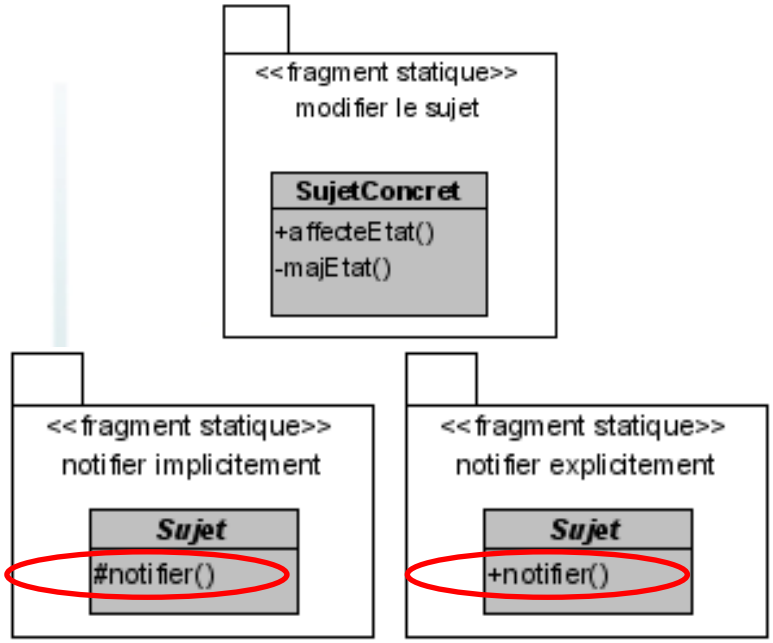
- Un fragment d'interaction par cas d'utilisation
- « include » mis en œuvre par référence d'interaction (UML 2)

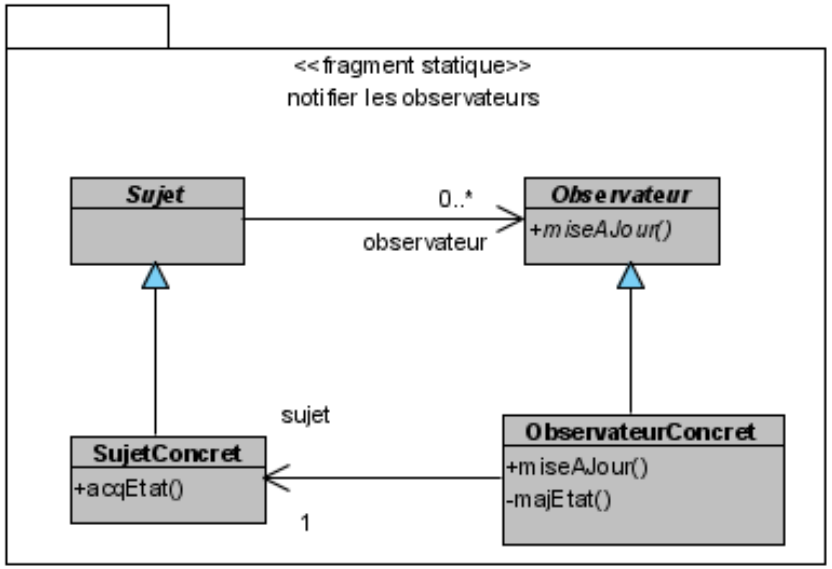
Patron «Observateur» : vue dynamique à variantes



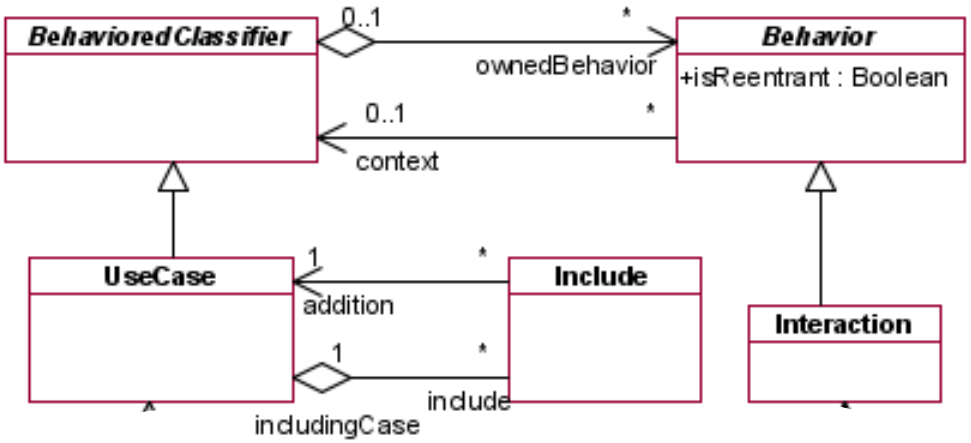
Variabilité : impact sur la vue statique

- La nouvelle vue statique est en grande partie déductible des deux autres vues
- Un fragment statique (package) par cas d'utilisation
  - Uniquement les éléments statiques utilisés dans l'interaction correspondante
  - Généralisation des éléments communs à toutes les variantes d'un même point de variation
- Les fragments seront appariés lors de l'imitation en fonction du choix des variantes…

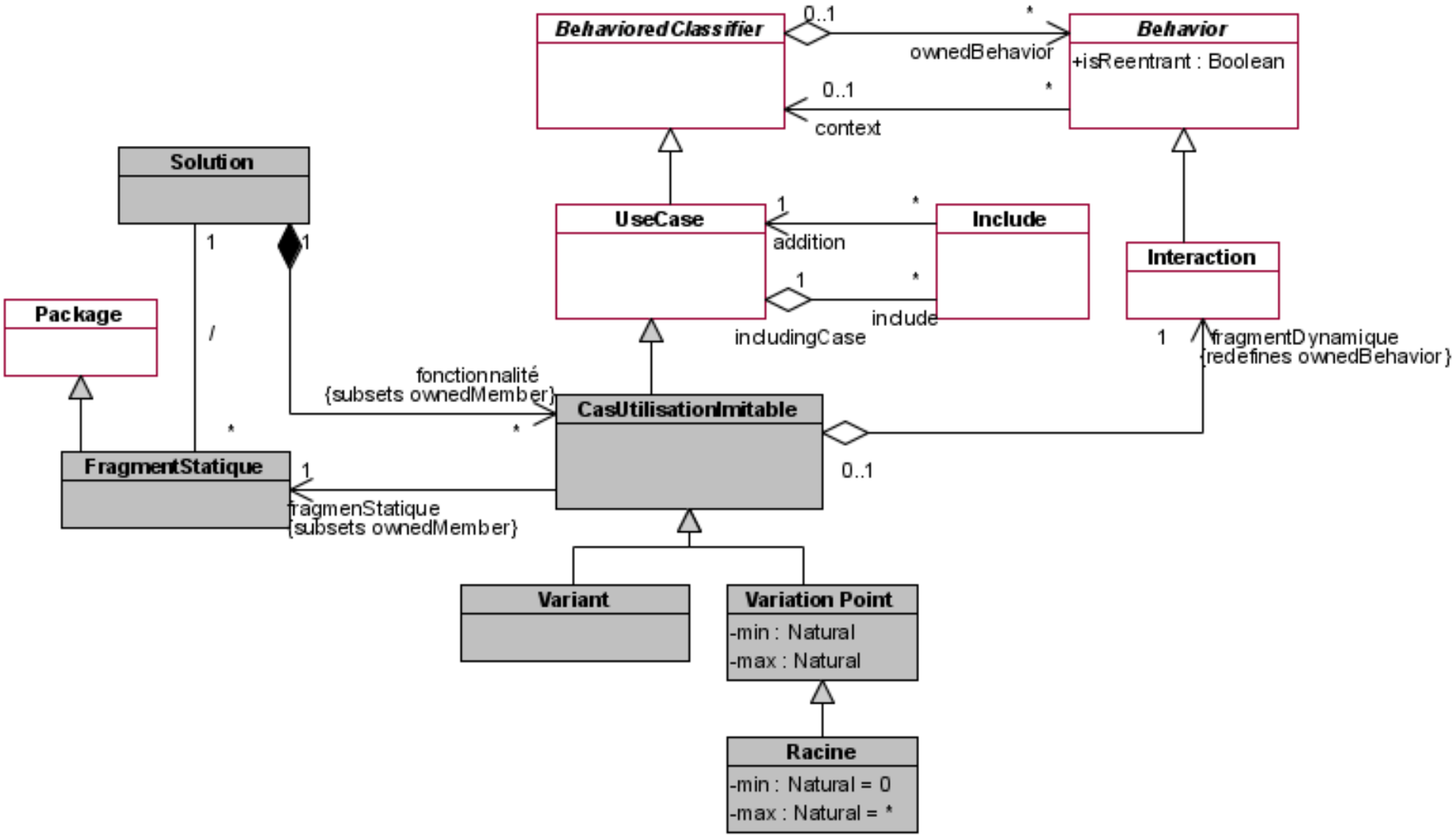




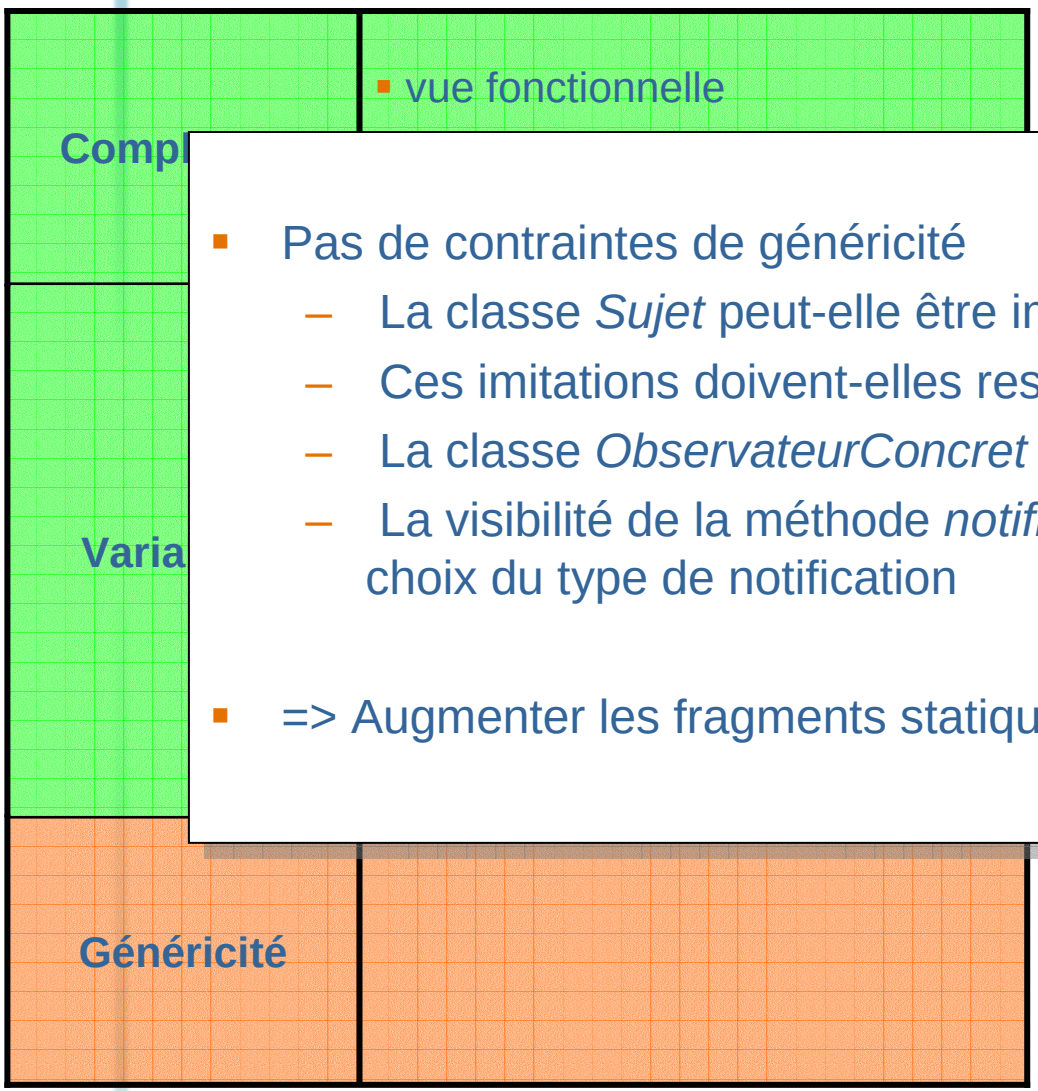
Extension d'UML pour les solutions de patrons (variabilité)



Extension d'UML pour les solutions de patrons (variabilité)



Complétude et Variabilité



- Pas de contraintes de généricité
  - La classe *Sujet* peut-elle être imitée plus d'une fois ?
  - Ces imitations doivent-elles rester abstraites ?
  - La classe *ObservateurConcret* peut-elle être imitée plusieurs fois ?
  - La visibilité de la méthode *notifier* doit rester cohérente avec le choix du type de notification
- => Augmenter les fragments statiques avec des propriétés génériques

Fragments statiques et généricité : principes

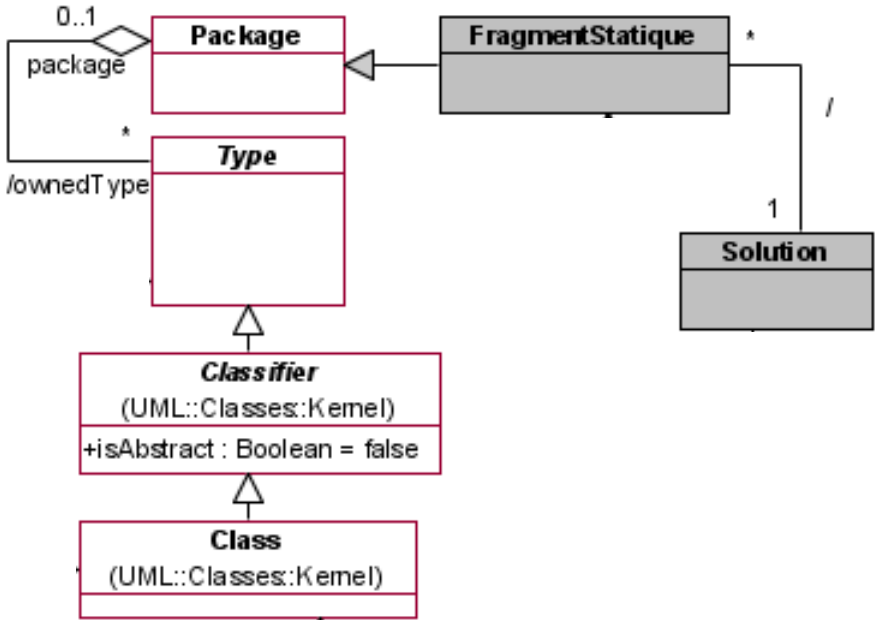
- Propriétés communes à tout type d'élément
  - Nombre d'occurrences d'imitation
    - Classe
    - Méthodes (au sein d'une classe)
    - Attributs (au sein d'une classe)
  - Pérennité de la visibilité

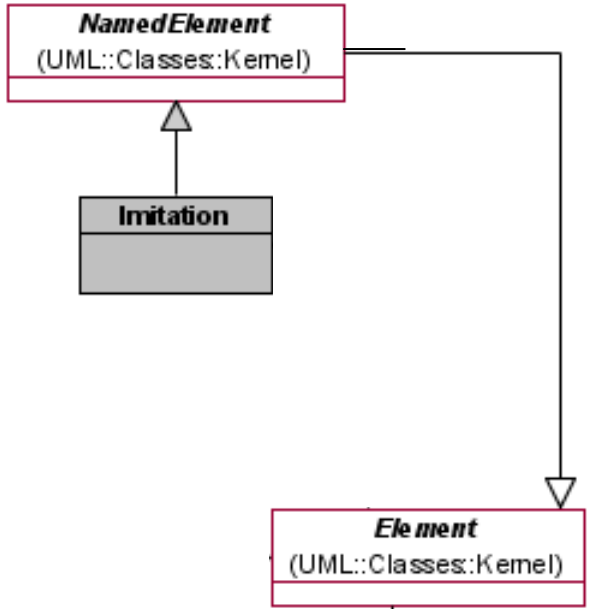
- Propriétés spécifiques à un type d'élément
  - Pérennité de la propriété d'abstraction

- Traçabilité / Vérification
  - Relation forte entre les éléments imités et leur « source »
  - Contraintes OCL exploitant les propriétés

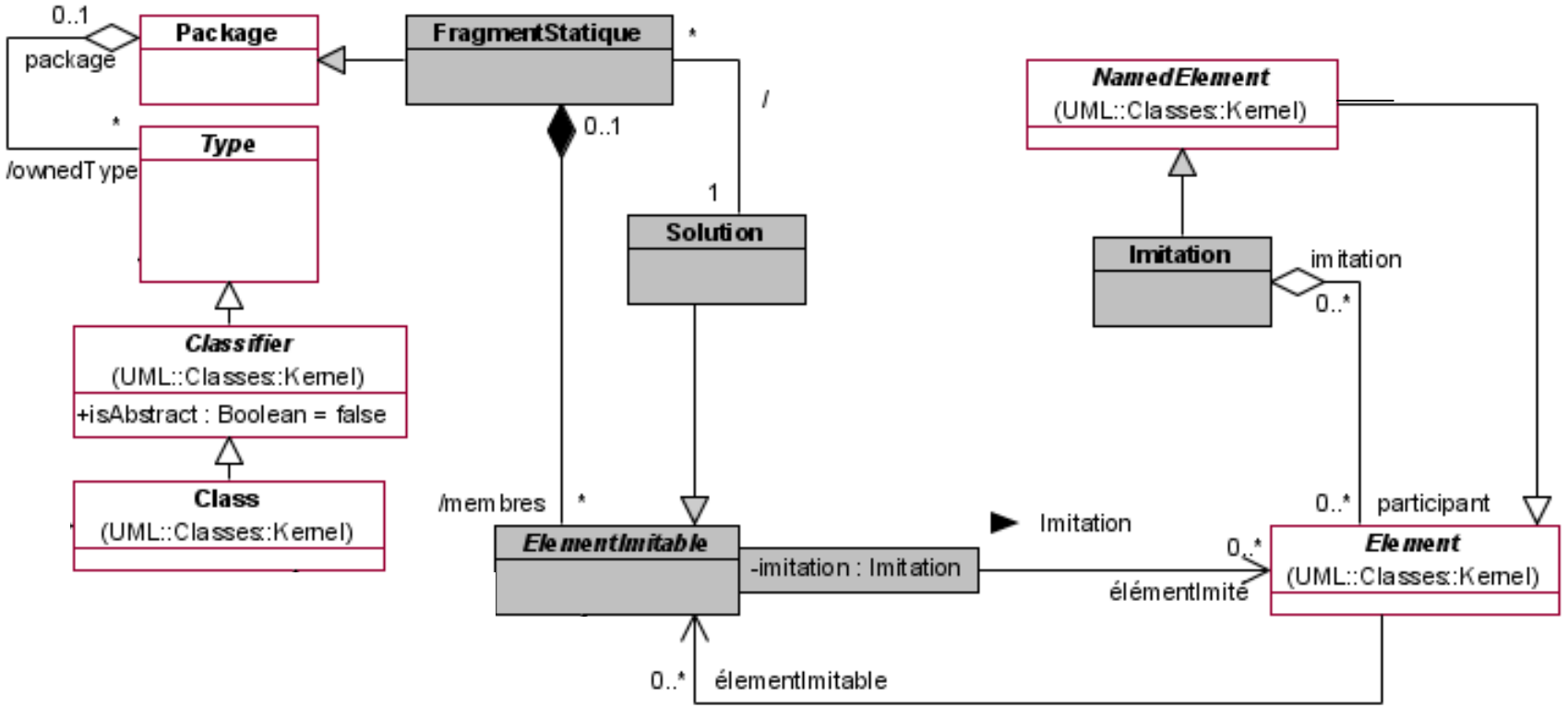
- D'autres propriétés à venir…

Extension d'UML pour les solutions de patrons (généricité)

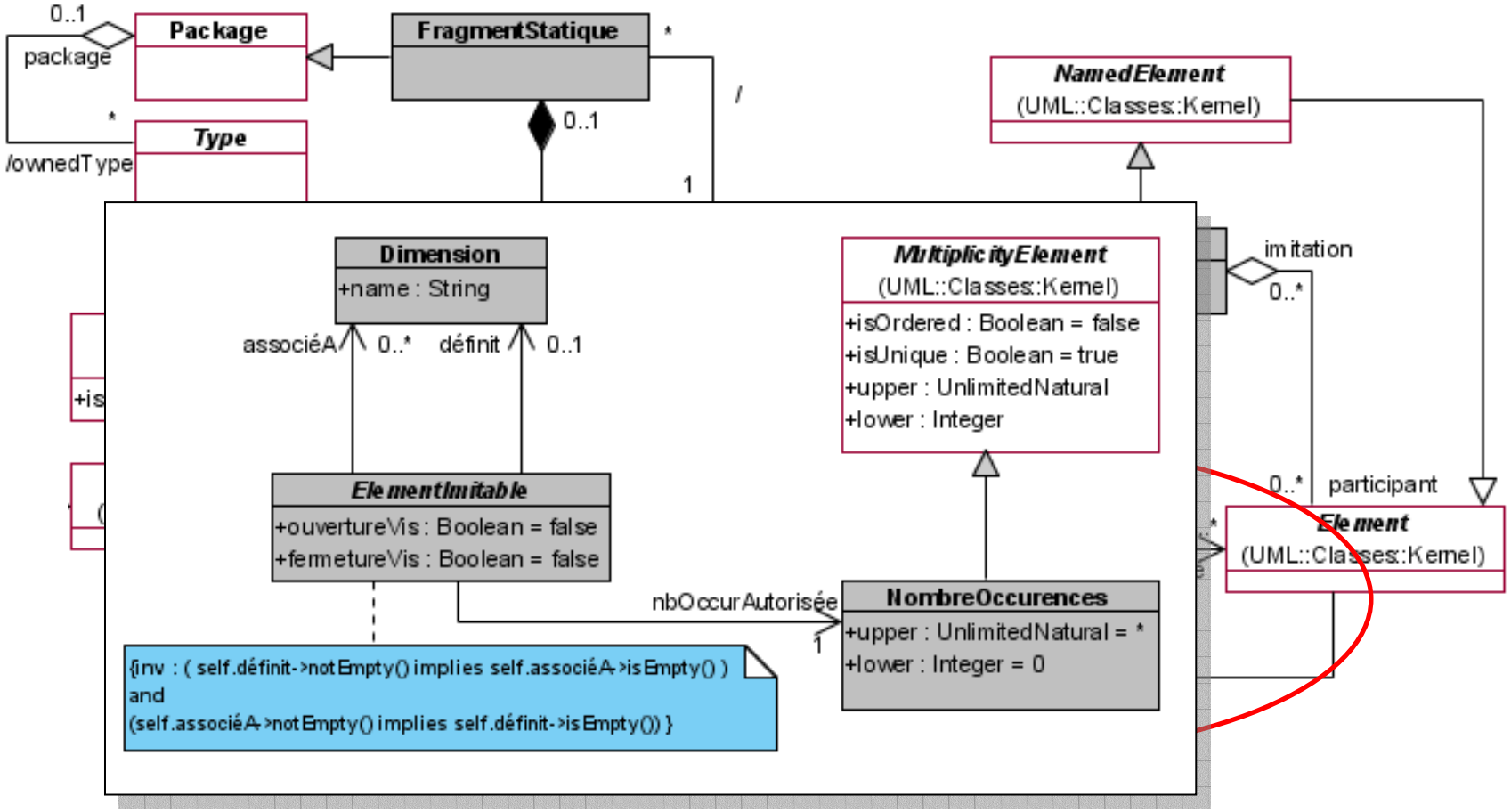




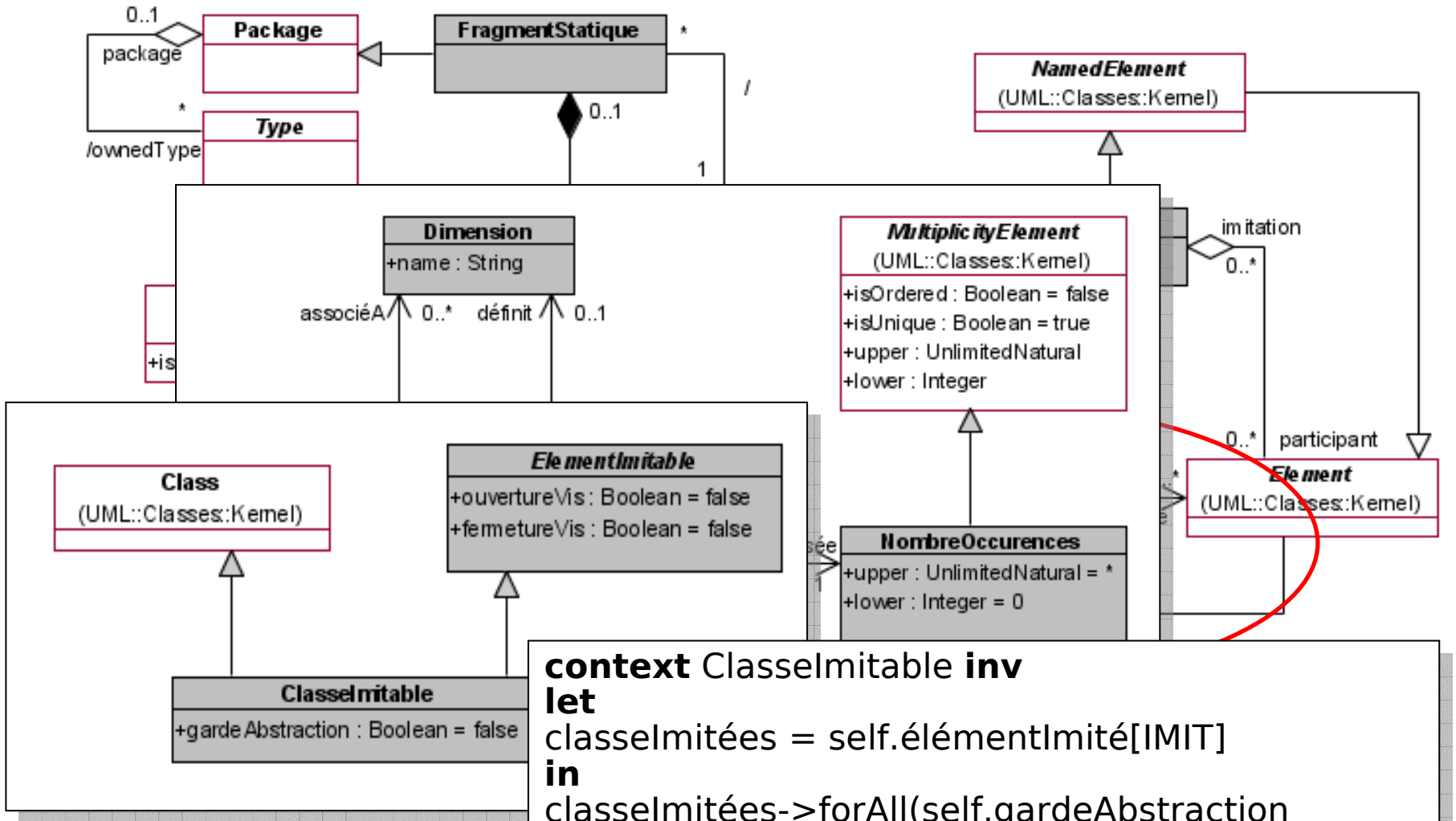
Extension d'UML pour les solutions de patrons (généricité)



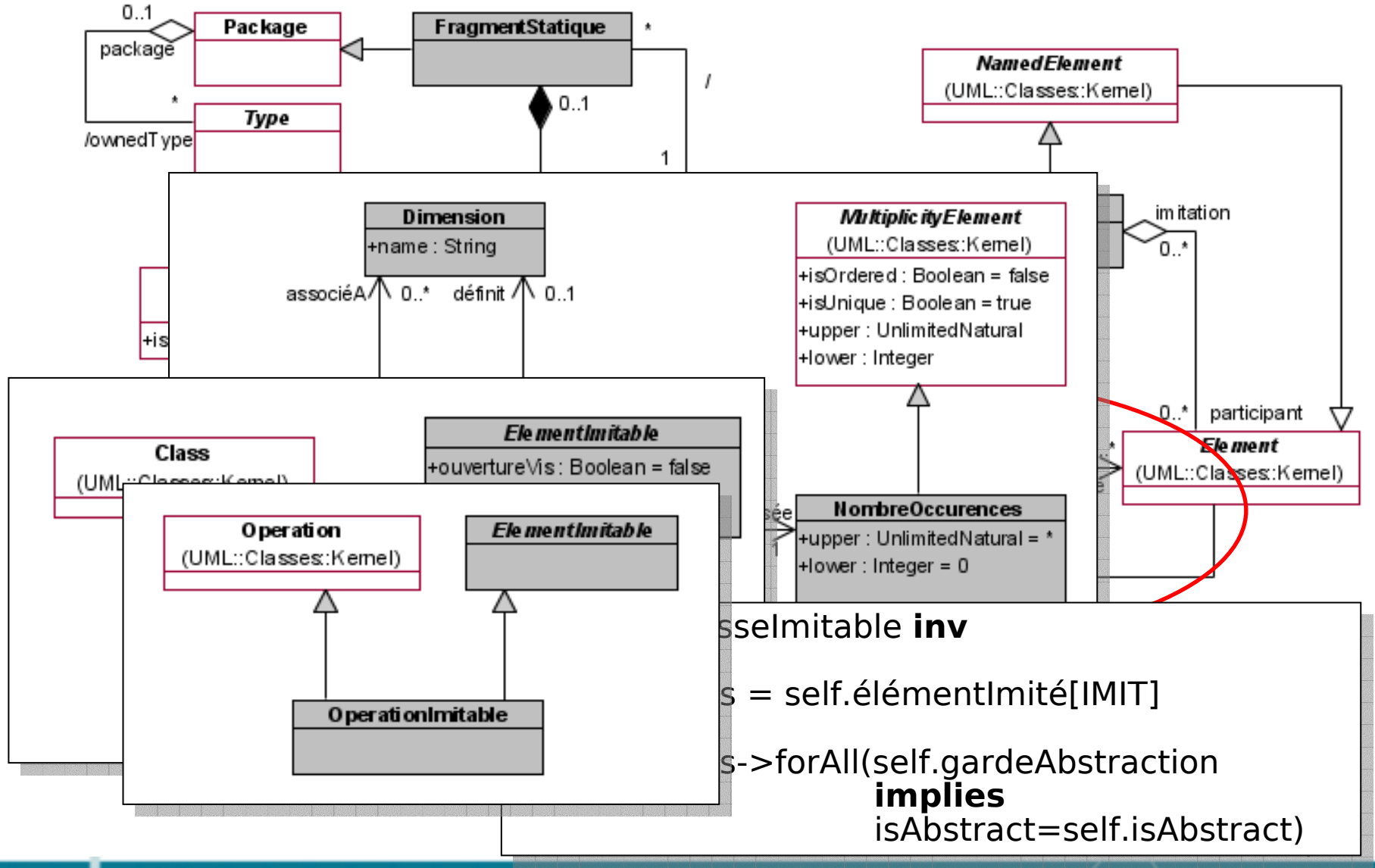
Extension d'UML pour les solutions de patrons (généricité)



Extension d'UML pour les solutions de patrons (généricité)



Extension d'UML pour les solutions de patrons (généricité)



Complétude, Variabilité et Généricité

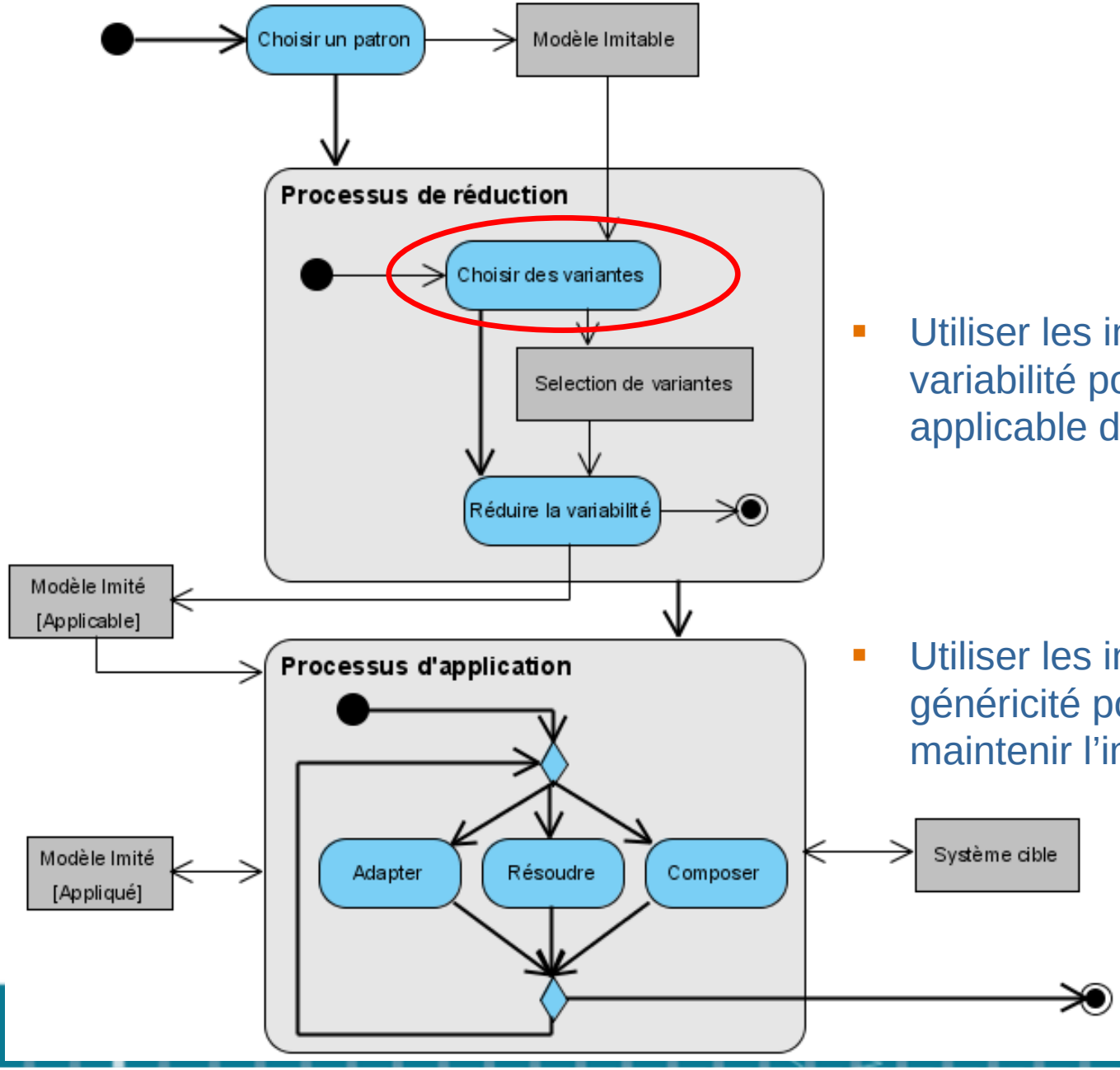
Complétude	▪ vue fonctionnelle ▪ vue dynamique ▪ vue statique
Variabilité	▪ point de variation ▪ variante ▪ alternatives ▪ options ▪ fonctionnalité obli ▪ s'applique sur les
Généricité	▪ éléments statiques ▪ contraintes OCL ▪ évolutif

- Un méta-modèle (extension d'UML)
- Comment produire de telles spécifications ? (ingénieur de patrons)
- Comment les imiter ? (ingénieur d'applications)

Plan

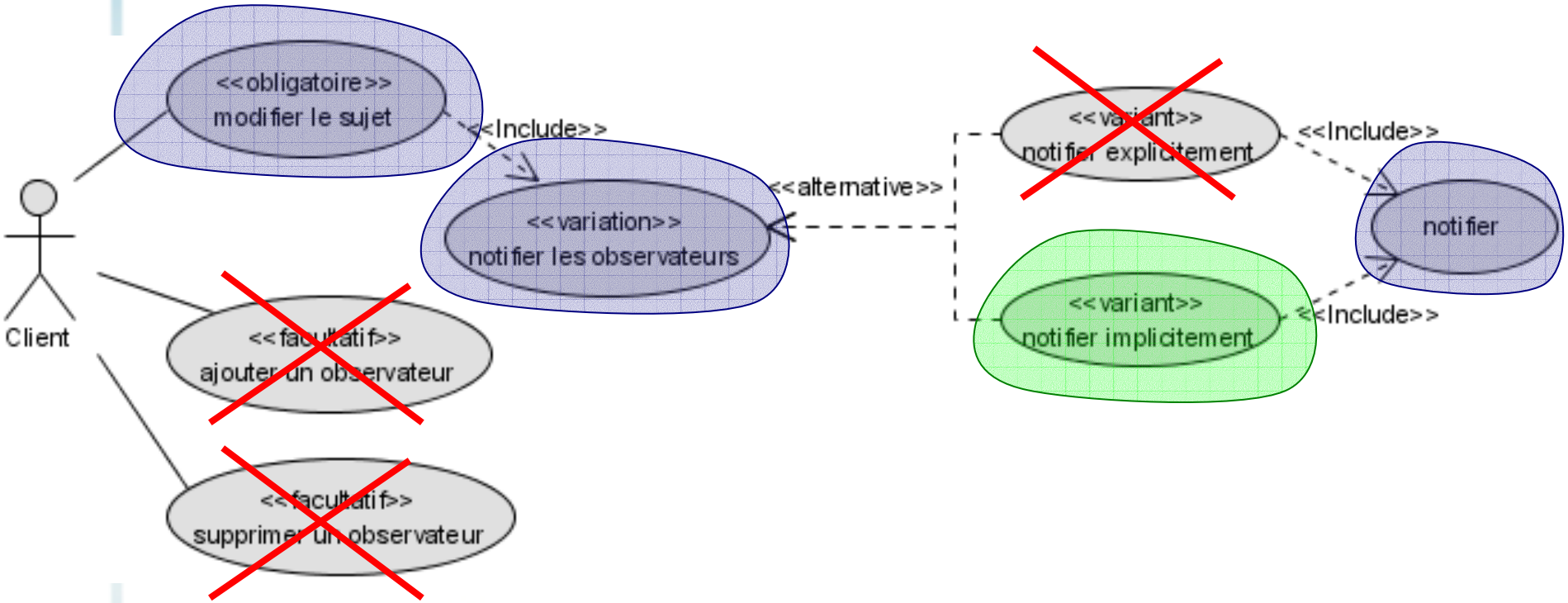
- Cas d'étude
- Problématique et Travaux existants
- Spécifications des solutions
  - Complétude
  - Variabilité
  - Généricité
- Processus d'imitation
- Outillage
- Conclusions & Perspectives

Processus d'imitation



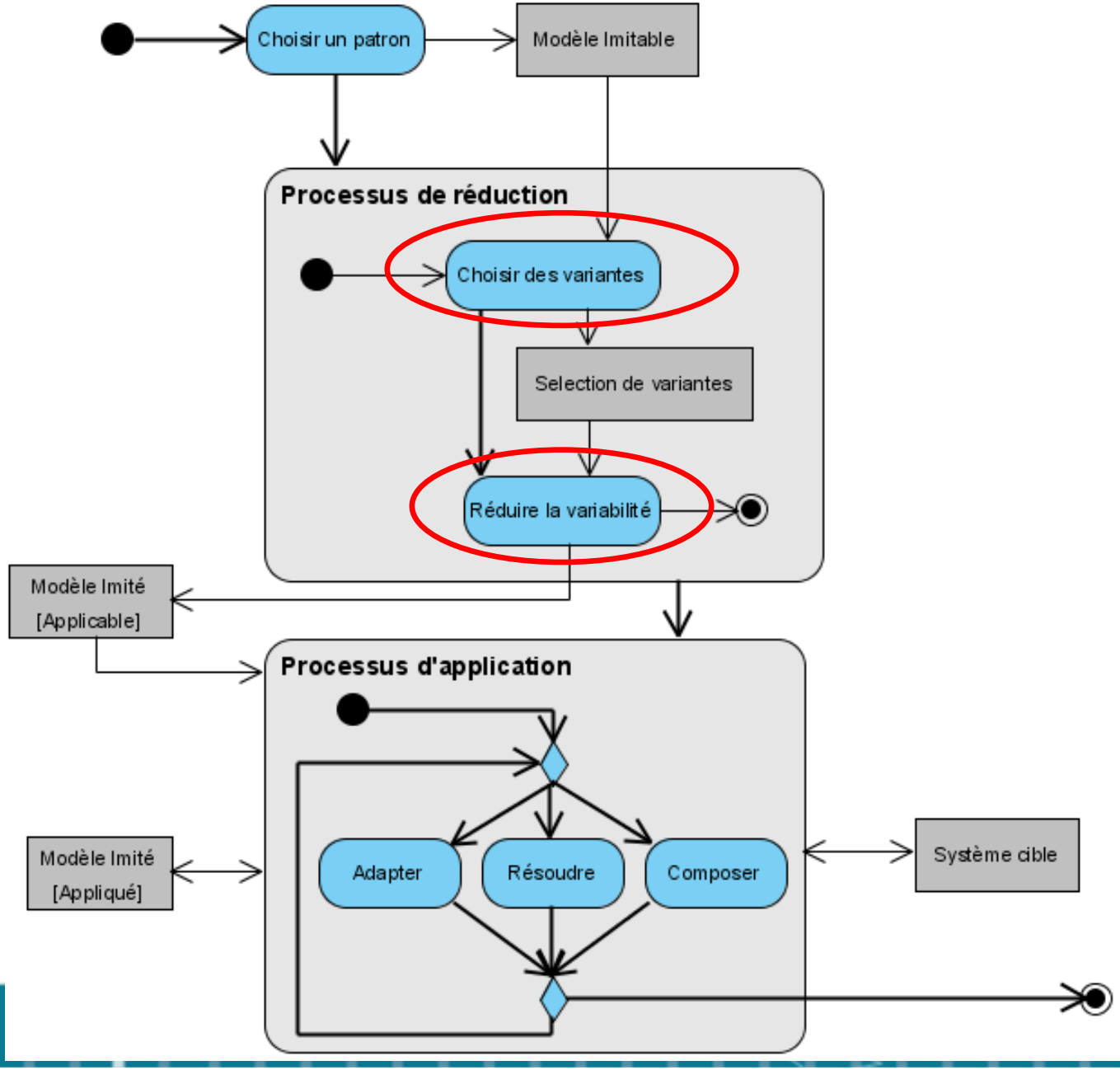
- Utiliser les informations de variabilité pour produire une version applicable de l'imitation
- Utiliser les informations de généricité pour appliquer et maintenir l'imitation

Choix des variantes

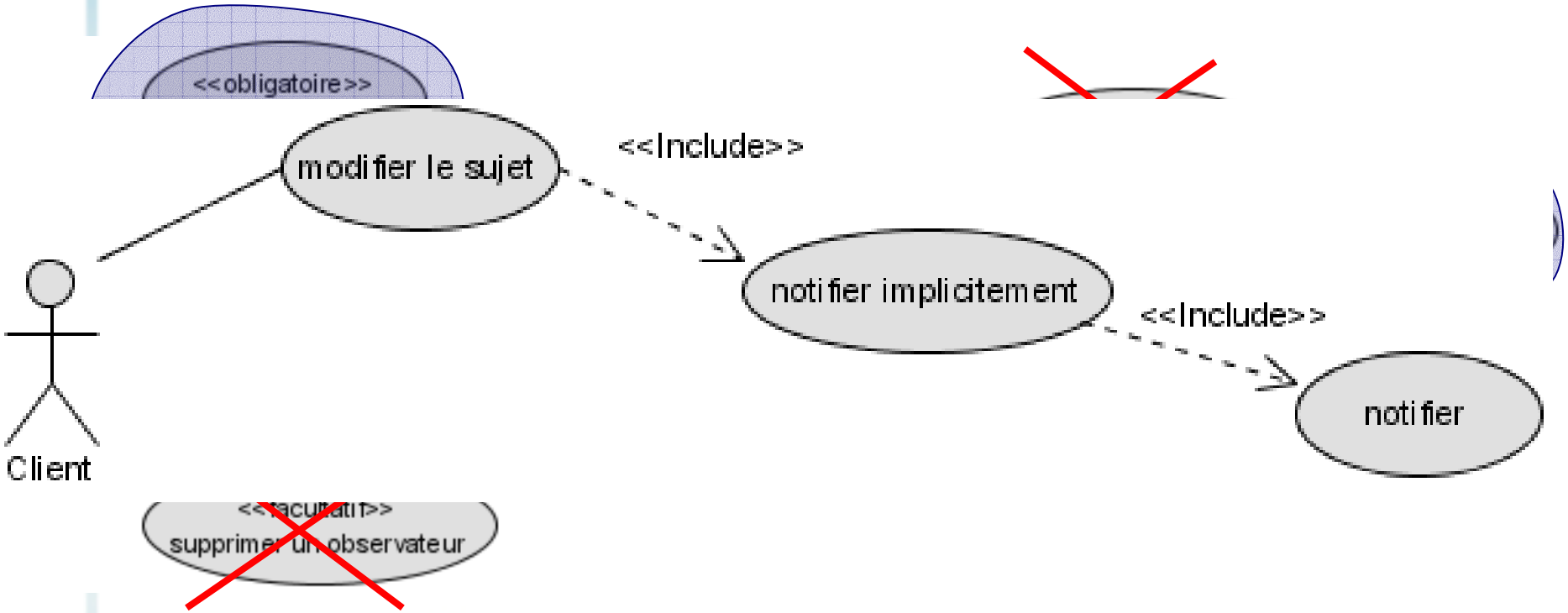


I.A.

Processus d'imitation

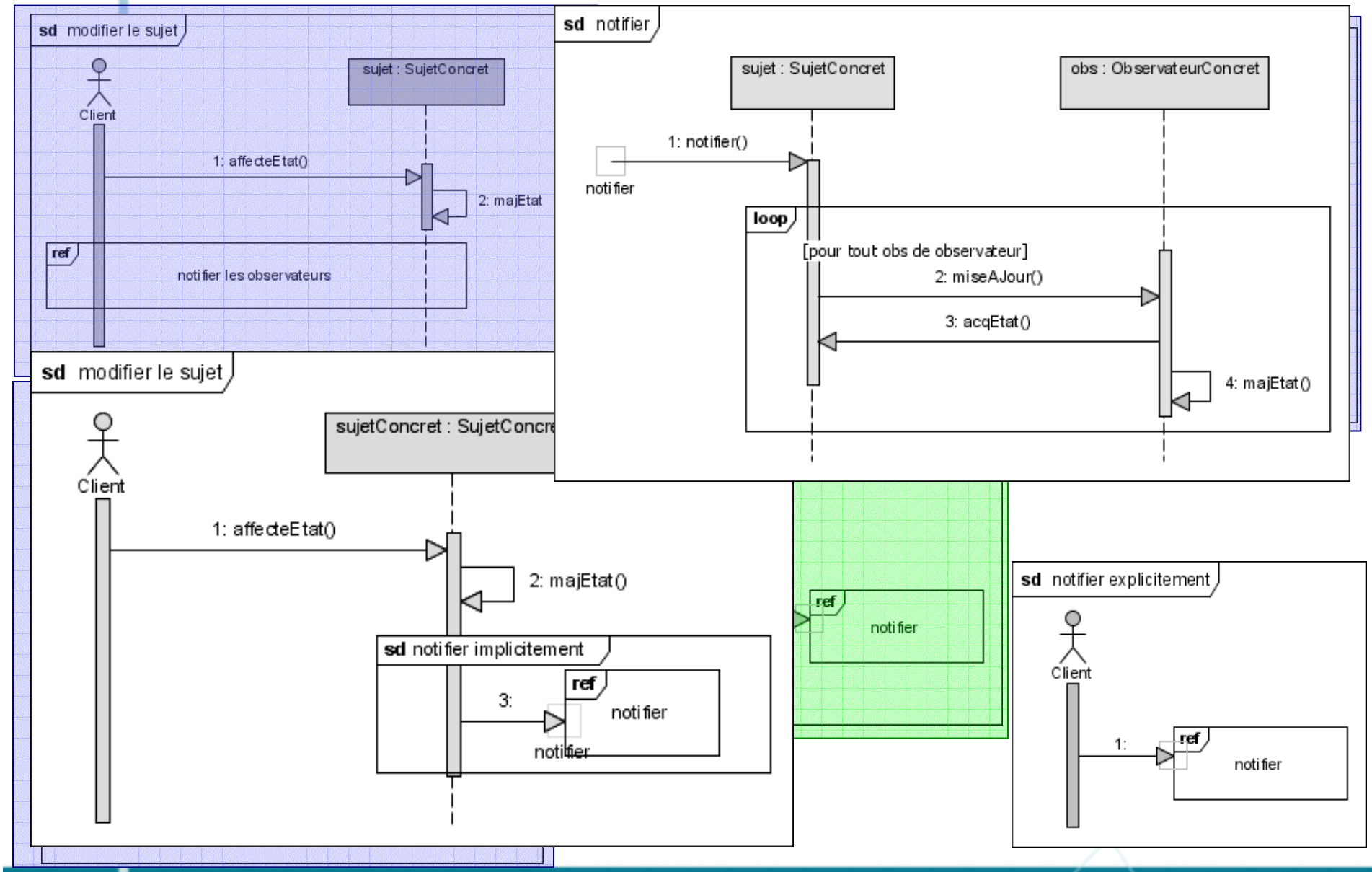


Réduction de la vue fonctionnelle

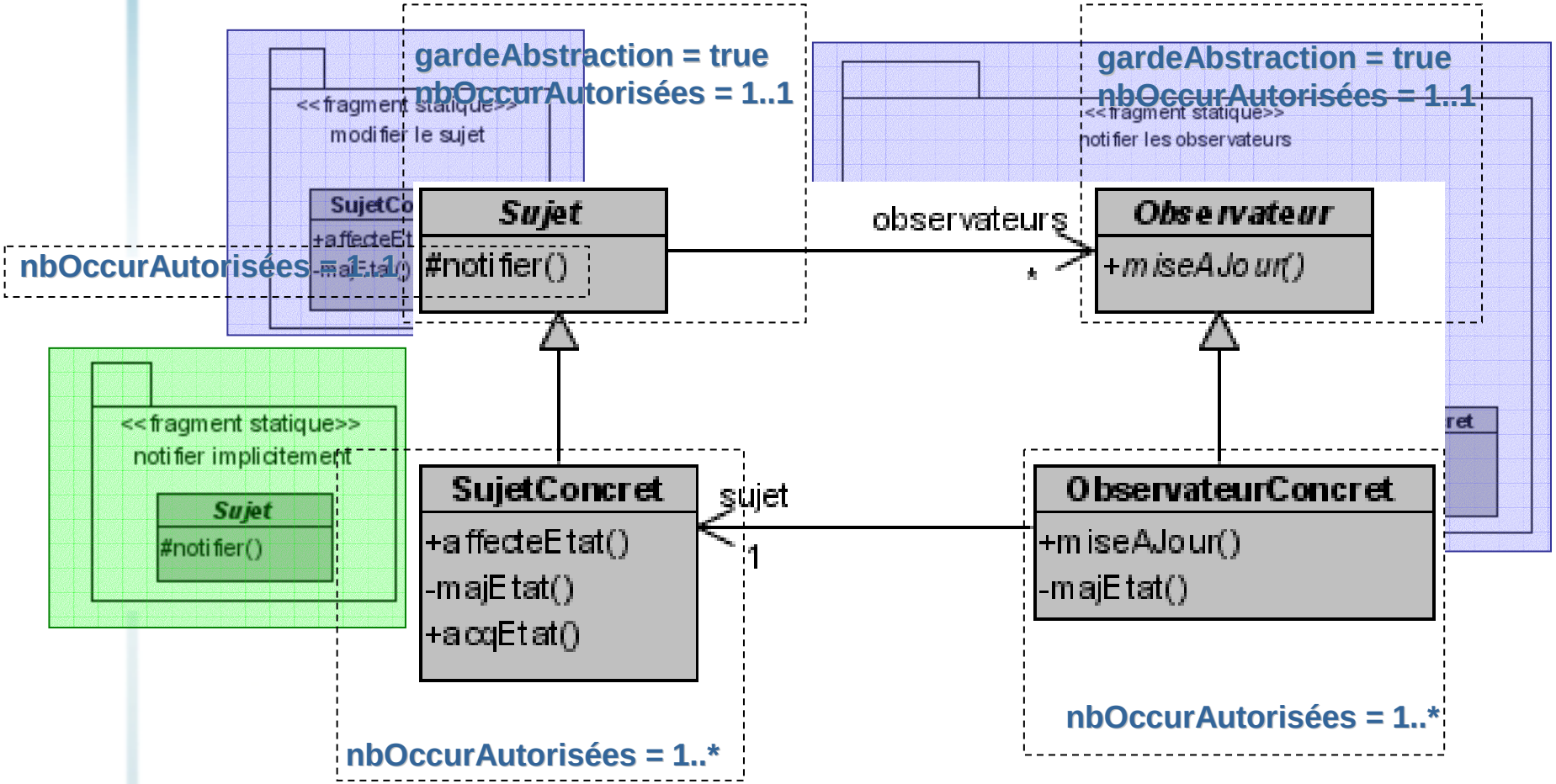


I.A.

Réduction de la vue dynamique

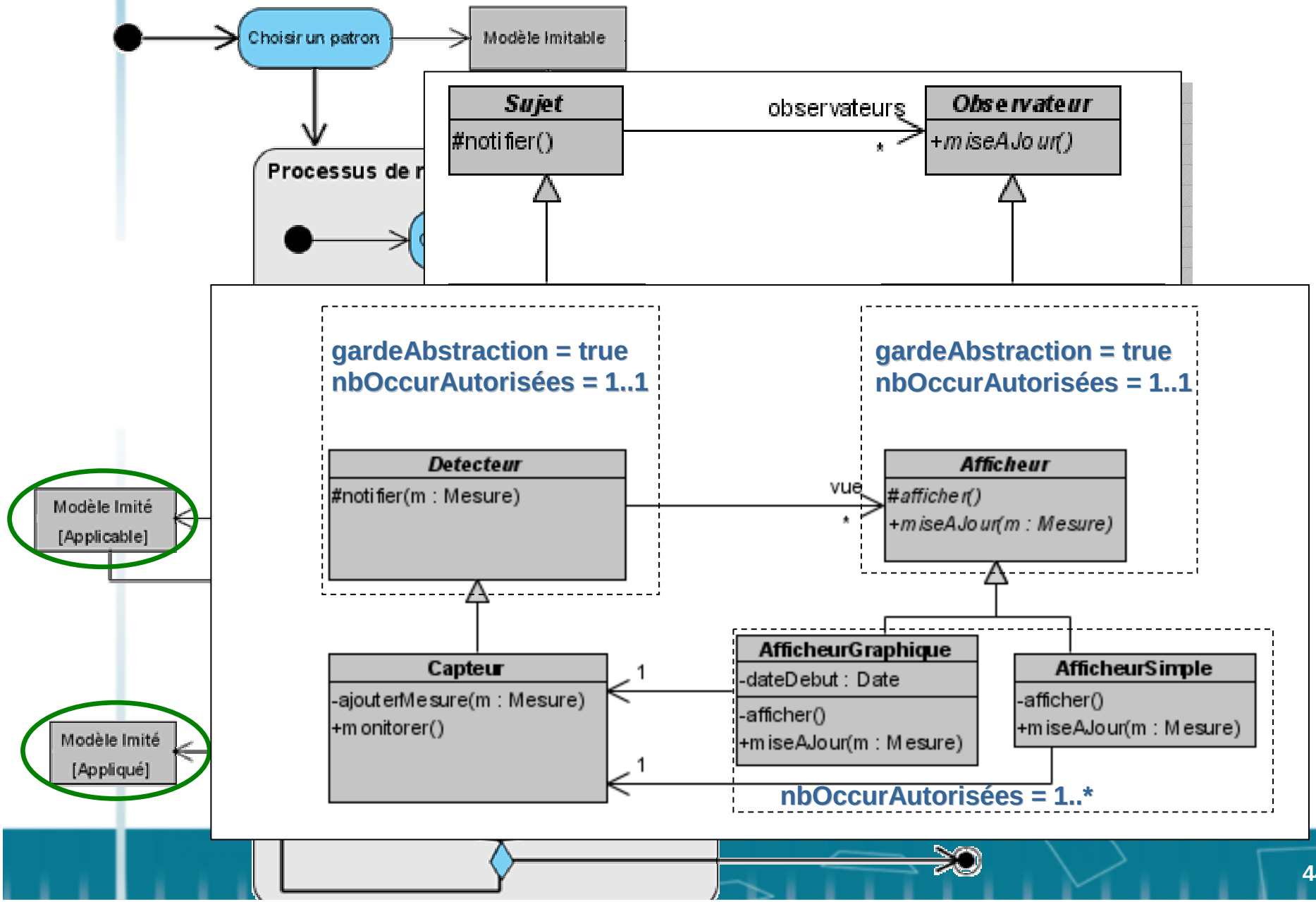


Réduction de la vue statique



- Conservation des propriétés génériques

Processus d'imitation



Plan

- Cas d'étude
- Problématique et Travaux existants
- Spécifications des solutions
  - Complétude
  - Variabilité
  - Généricité
- Processus d'imitation
- Outillage
- Conclusions & Perspectives

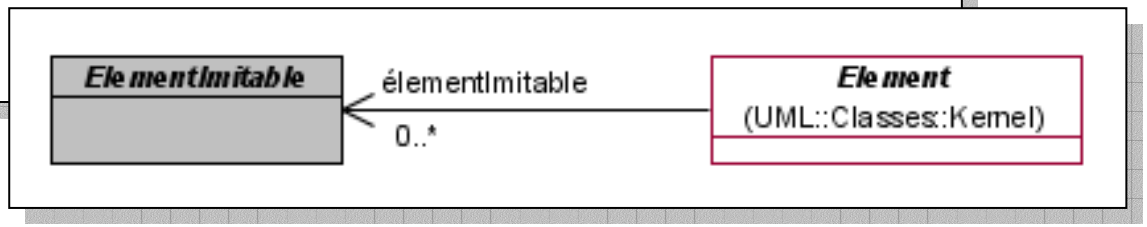
Mise en œuvre avec l'IDM

- Les systèmes cibles sont décrits avec UML

réduction

application

- Mécanismes d'extension d'UML (connus des AGL) n'autorisent pas la modification du méta modèle.
  - Association « Imitation » ?



- Plate-forme Eclipse
  - Eclipse Modelling Framework (EMF)
  - Plugin UML2/EMF
  - Générations EMF pour les 2 méta-modèles *UMLpi* et *UMLpiBind*
  - Transformations de modèles ATL/EMF

Plan

- Cas d'étude
- Problématique et Travaux existants
- Spécifications des solutions
  - Complétude
  - Variabilité
  - Généricité
- Processus d'imitation
- Outillage
- Conclusions & Perspectives

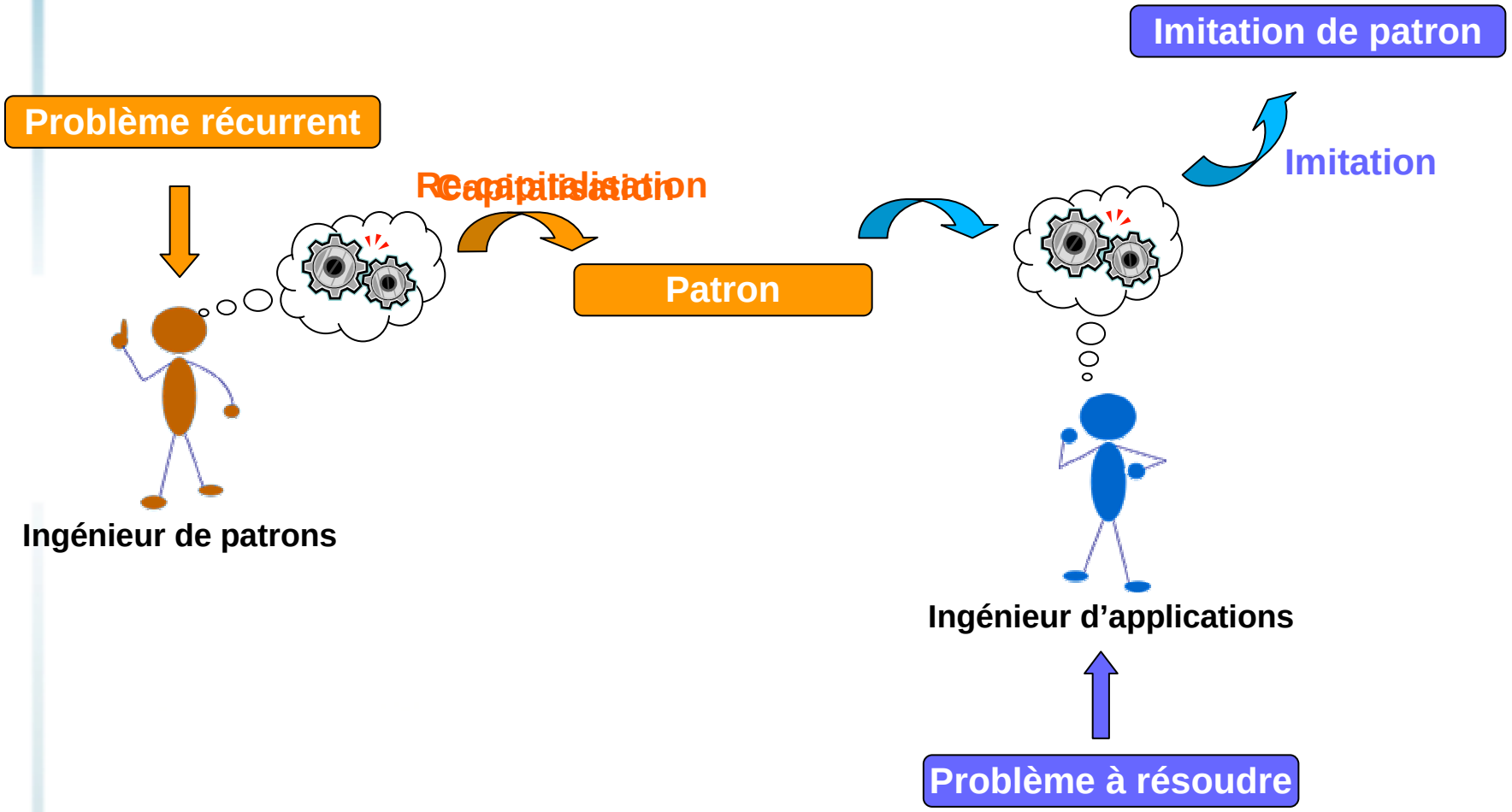
Rappel des propositions

- Nouvelles spécifications des solutions de patrons
  - Trois vues complémentaires
  - Variabilité fonctionnelle qui pilote les autres vues
  - Propriétés génériques sur la vue statique
    - Contraintes pour la validité des imitations
- Processus d'imitation
  - Adéquation fonctionnelle
  - Respect de l' «essence» du patron
- Processus de spécification
- Proposition pour la composition
- Les patrons « originaux » restent des supports didactiques indispensables

Perspectives

- Généricité dans les vues fonctionnelle et dynamique ?
- Méthode pour la création de nouvelles propriétés génériques
- Composition
  - multi-vues
  - avec des spécifications originales
- Application au domaine des composants métiers *[Saidi et al., 2007]*
  - => extensions des principes de variabilité (thèse en cours)
- Impacts sur la notion de patron et sur les usages

Usage des patrons



Merci de votre attention