

Melbeck Nathan, Miesch Nathanaël
Dardenne Gregory, Laury Lucie

SOUTENANCE **OBJECTAID**



INTRODUCTION



- ▼ ☐ objectaid_sae
 - ▼ ☐ controleur
 - ☐ ControleurAffichageGlobal.jav
 - ☐ ControleurButtonMenuAddFle
 - ☐ ControleurButtonNewClass.jav
 - ☐ ControleurButtonProjet.java
 - ☐ ControleurCentreClique.java
 - ☐ ControleurClasseCliquer.java
 - ☐ ControleurClasseGlisseee.java
 - ☐ ControleurFichierGlisse.java
 - ☐ ControleurVueTemporaireClas
 - ☐ HelloApplication.java
 - ▼ ☐ model
 - ☐ Analyseur.java
 - ☐ Classe.java
 - ☐ Fichier.java
 - ☐ Fleche.java
 - ☐ Model.java
 - ▼ ☐ observateur

CONTENT INDEX

RÉCAPITULATIF DES FONCTIONNALITÉS

- Répartition
- itérations

CONCEPTION : VUE GLOBALE

- Diagramme de Classe
- Patrons de Conception

CONCEPTION DE CERTAINES FONCTIONNALITÉS

- Glisser déposer
- Flèches de dépendances
- Enregistrement

RÉCAPITULATIF DES FONCTIONNALITÉS

CONCEPTION : VUE GLOBALE

CONCEPTION DE FONCTIONNALITÉS

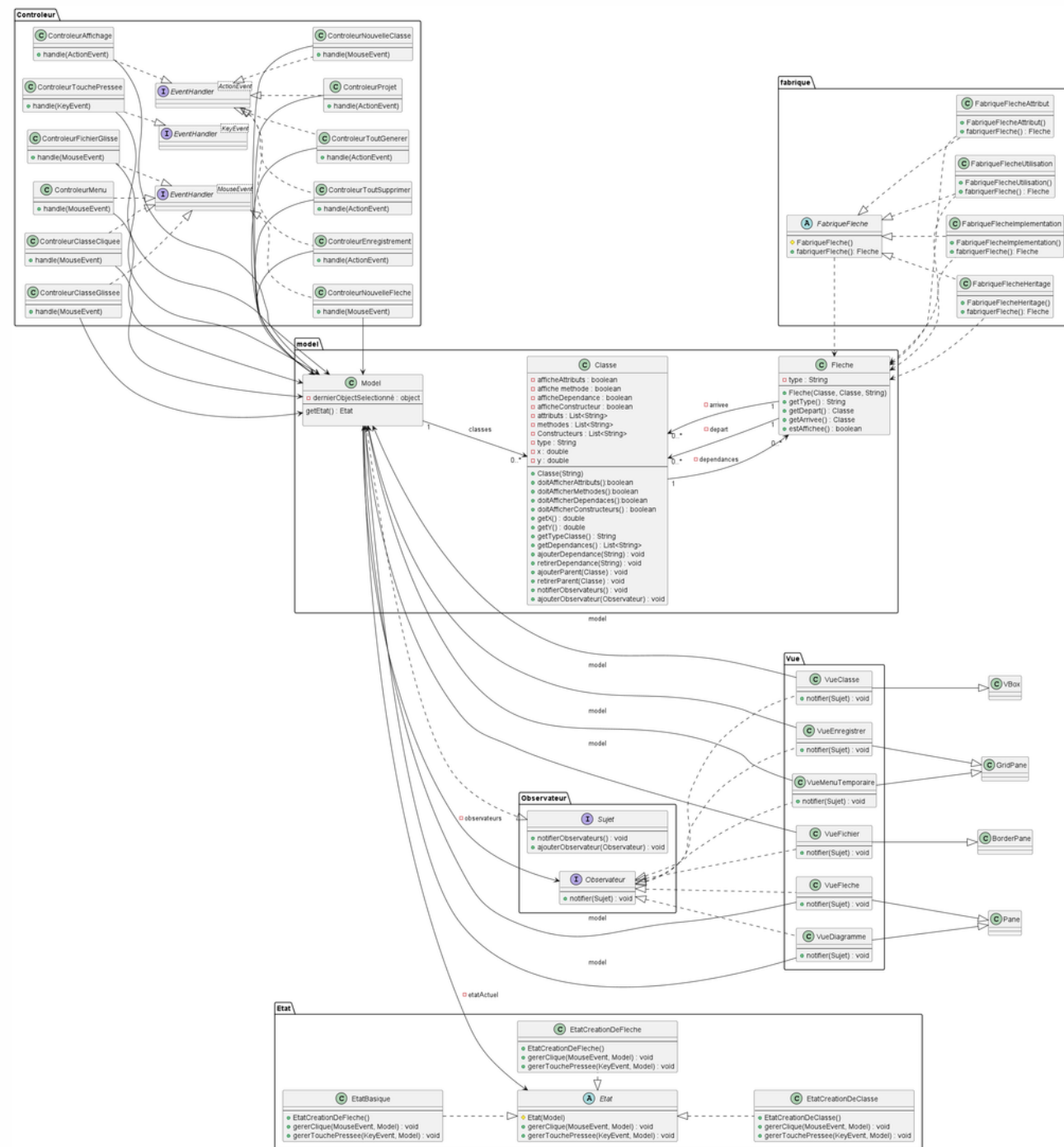
Fonctionnalités	Prévue en Analyse	rajoutées	Faites	acteurs	sprint
afficher l'arborescence du dossier				Nathan	1
générer une classe en glisser-déposer				Nathanaël	1
afficher ses attributs/ constructeurs/ méthodes				Lucie	1
afficher ses dépendances				Nathanaël, Gregory	3,4,5
afficher ses attributs hérités				Nathanaël, Gregory	3,4
déplacer une classe dans le graphe				Lucie	2
déplacer une flèche en déplaçant une classe				Nathanaël	4,5
ajouter des informations à une classe				Nathan, Nathanaël	2
choisir d'afficher certains éléments d'une classe				Gregory, Nathan, Lucie	2
affichage des opérations générales applicable à toutes les classes du graphe				Nathanaël	3
afficher les classes externes utilisées				Gregory	4
générer une diagramme de classe de tout un package				Nathan	3
créer une nouvelle classe				Nathan	3
noter les éléments de cette nouvelle classe				Nathan, Nathanaël	2
créer les dépendances de cette nouvelle classe				Gregory	5
générer le squelette des classes construites ou modifiées dans le diagramme				Gregory	6
enregistrer le document sous le format image				Nathan	4
enregistrer le document sous le format UML				Lucie	4
choisir un titre au document				Nathan, Lucie	4
supprimer une classe existante				Gregory, Nathan	5
tout supprimer du graphe				Nathanaël	6

RÉCAPITULATIF DES FONCTIONNALITÉS

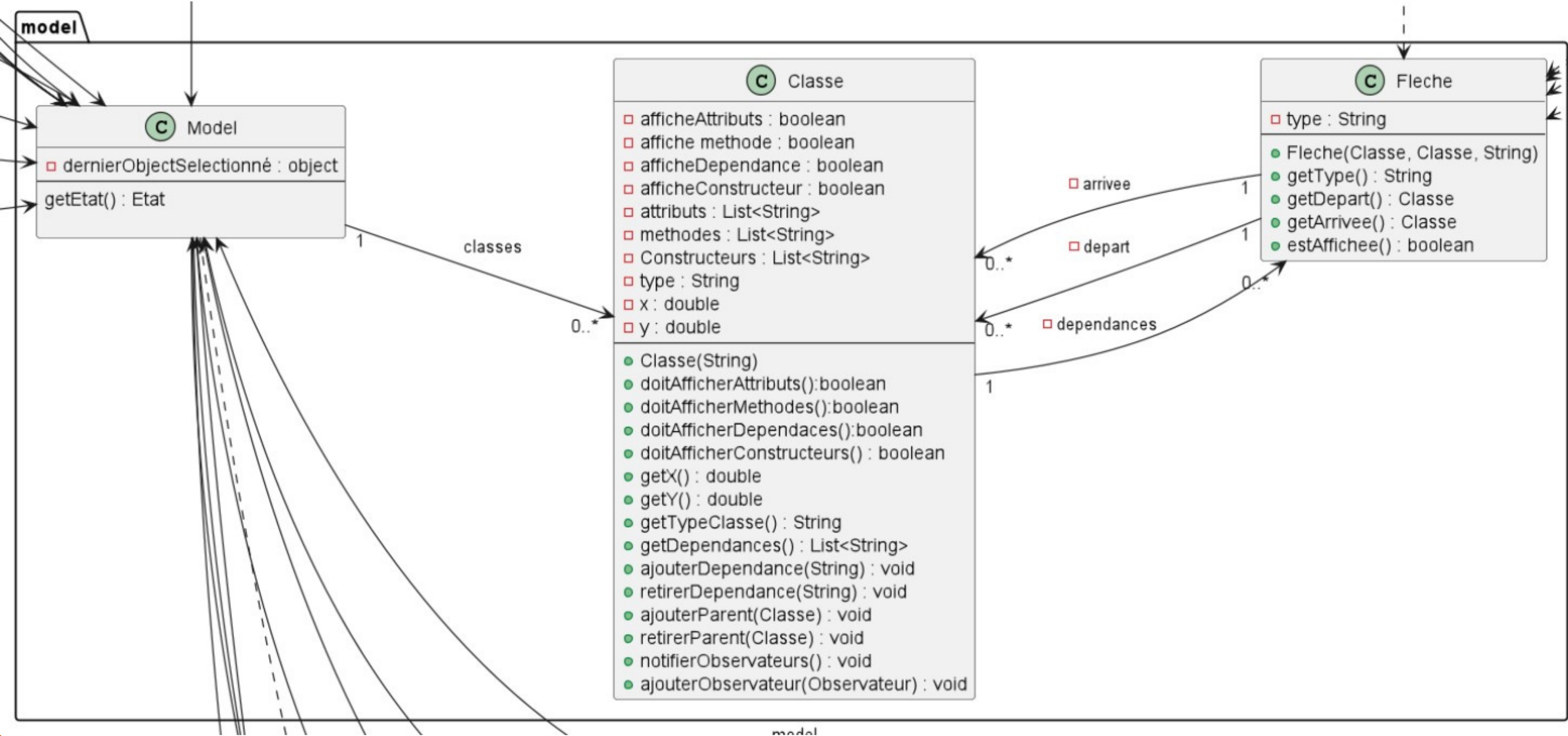
CONCEPTION : VUE GLOBALE

CONCEPTION DE FONCTIONNALITÉS

DIAGRAMME DE CLASSES :



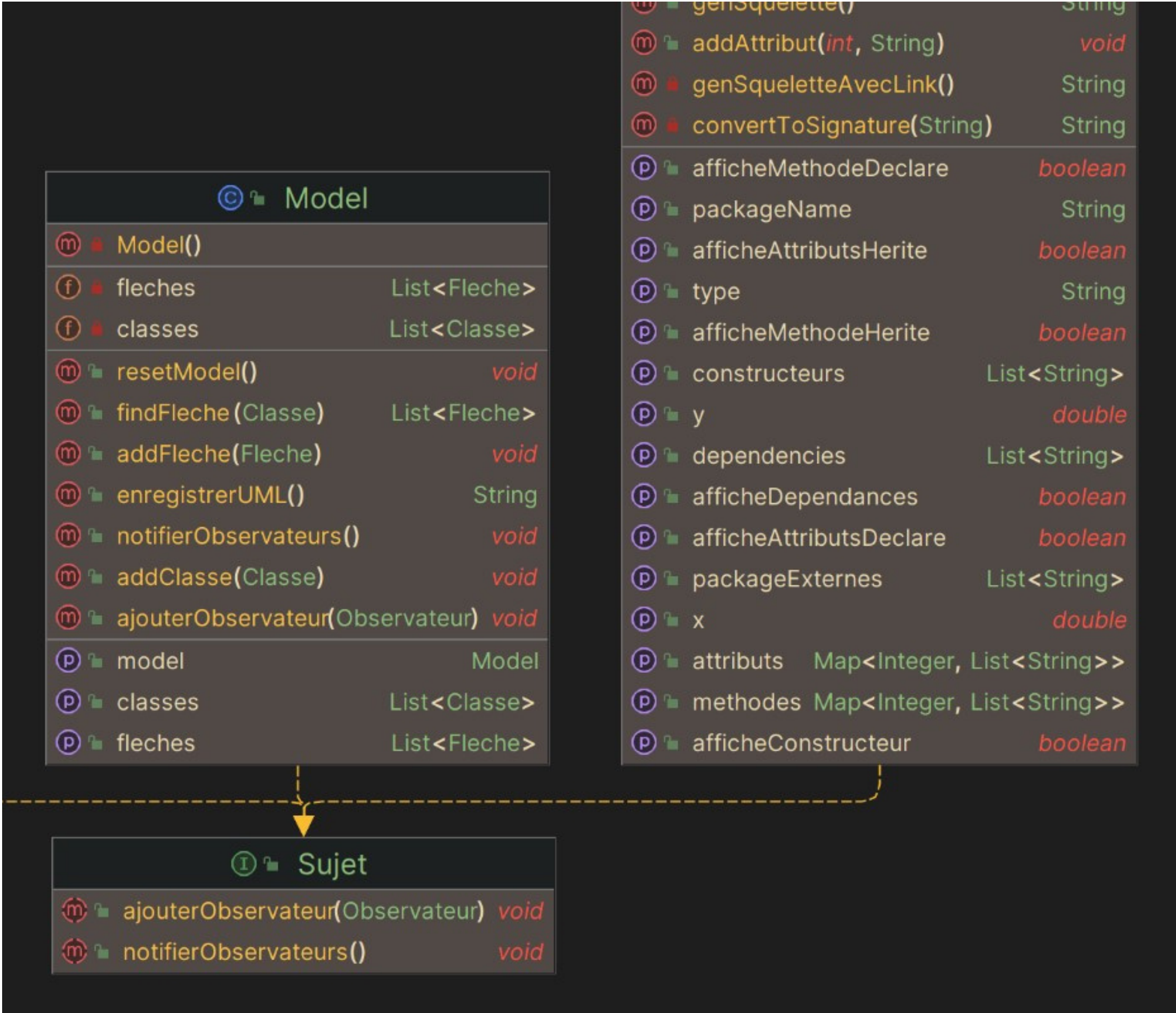
Modele prevu en analyse :



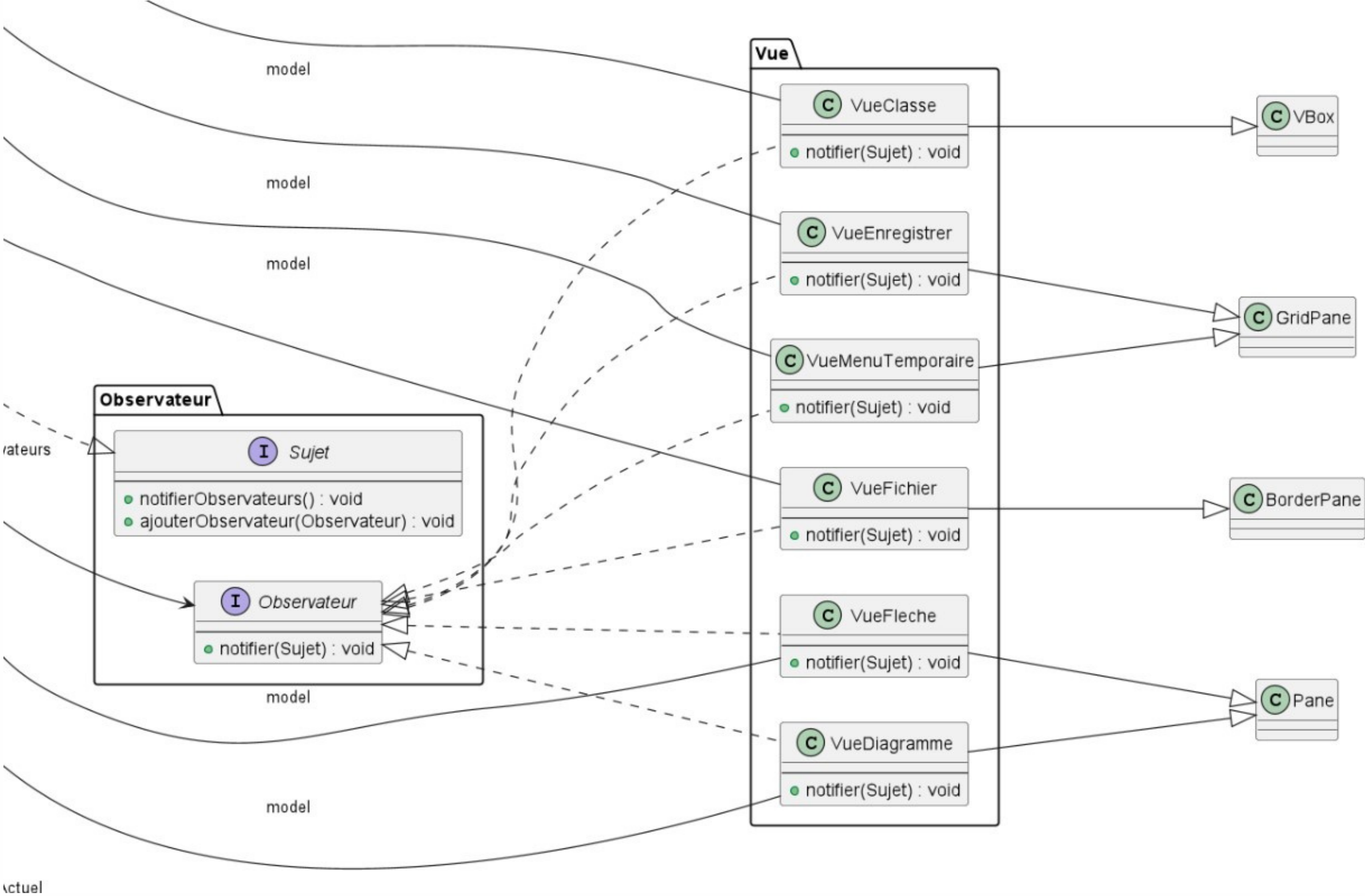
Modèle réalisé:

Classe	Fleche	Analyseur
<div><div>Classe()</div><div><div>f</div><div>type</div><div>String</div></div><div><div>f</div><div>afficheDependances</div><div>boolean</div></div><div><div>f</div><div>y</div><div>double</div></div><div><div>f</div><div>attributs</div><div>Map<Integer, List<String>></div></div><div><div>f</div><div>afficheConstructeur</div><div>boolean</div></div><div><div>f</div><div>afficheAttributsDeclare</div><div>boolean</div></div><div><div>f</div><div>x</div><div>double</div></div><div><div>f</div><div>afficheMethodeDeclare</div><div>boolean</div></div><div><div>f</div><div>packageName</div><div>String</div></div><div><div>f</div><div>dependencies</div><div>List<String></div></div><div><div>f</div><div>constructeurs</div><div>List<String></div></div><div><div>f</div><div>methodes</div><div>Map<Integer, List<String>></div></div><div><div>f</div><div>afficheMethodeHerite</div><div>boolean</div></div><div><div>f</div><div>afficheAttributsHerite</div><div>boolean</div></div><div><div>m</div><div>addMethode(int, String)</div><div>void</div></div><div><div>m</div><div>notifierObservateurs()</div><div>void</div></div><div><div>m</div><div>genSqueletteAttributs()</div><div>String</div></div><div><div>m</div><div>removeAttribut(int, String)</div><div>void</div></div><div><div>m</div><div>addDependencies(String)</div><div>void</div></div><div><div>m</div><div>addConstructeur(String)</div><div>void</div></div><div><div>m</div><div>genSqueletteSignature()</div><div>String</div></div><div><div>m</div><div>toString()</div><div>String</div></div></div>	<div><div>Fleche(Classe, Classe, String, String, VueCentre)</div><div><div>f</div><div>type</div><div>String</div></div><div><div>f</div><div>depart</div><div>Classe</div></div><div><div>f</div><div>fabrique</div><div>FabriqueVueFleche</div></div><div><div>f</div><div>cardinalites</div><div>String</div></div><div><div>f</div><div>nom</div><div>String</div></div><div><div>f</div><div>arrivee</div><div>Classe</div></div><div><div>m</div><div>creerFleches(Classe, VueCentre)</div><div>void</div></div><div><div>m</div><div>actualiserFleches(VueCentre)</div><div>void</div></div><div><div>p</div><div>affiche</div><div>boolean</div></div><div><div>p</div><div>fabrique</div><div>FabriqueVueFleche</div></div><div><div>p</div><div>type</div><div>String</div></div><div><div>p</div><div>nom</div><div>String</div></div><div><div>p</div><div>arrivee</div><div>Classe</div></div><div><div>p</div><div>depart</div><div>Classe</div></div><div><div>p</div><div>cardinalites</div><div>String</div></div></div>	<div><div>Analyseur(String)</div><div><div>m</div><div>genMethods()</div><div>void</div></div><div><div>m</div><div>genAttributs()</div><div>void</div></div><div><div>m</div><div>notClassicType(String)</div><div>boolean</div></div><div><div>m</div><div>genAttributsInClass(int, List<Field>)</div><div>void</div></div><div><div>m</div><div>isContainsMoreThanOneValue(String)</div><div>boolean</div></div><div><div>m</div><div>main(String[])</div><div>void</div></div><div><div>m</div><div>genImplementAndExtendsDependencies()</div><div>void</div></div><div><div>m</div><div>genMethodsInClass(int, List<Method>)</div><div>void</div></div><div><div>m</div><div>genConstructeurs()</div><div>void</div></div><div><div>m</div><div>analyseClasse()</div><div>Classe</div></div><div><div>m</div><div>genClassSignature()</div><div>void</div></div><div><div>m</div><div>genLink(Field)</div><div>boolean</div></div><div><div>m</div><div>getSignature(int)</div><div>String</div></div></div>

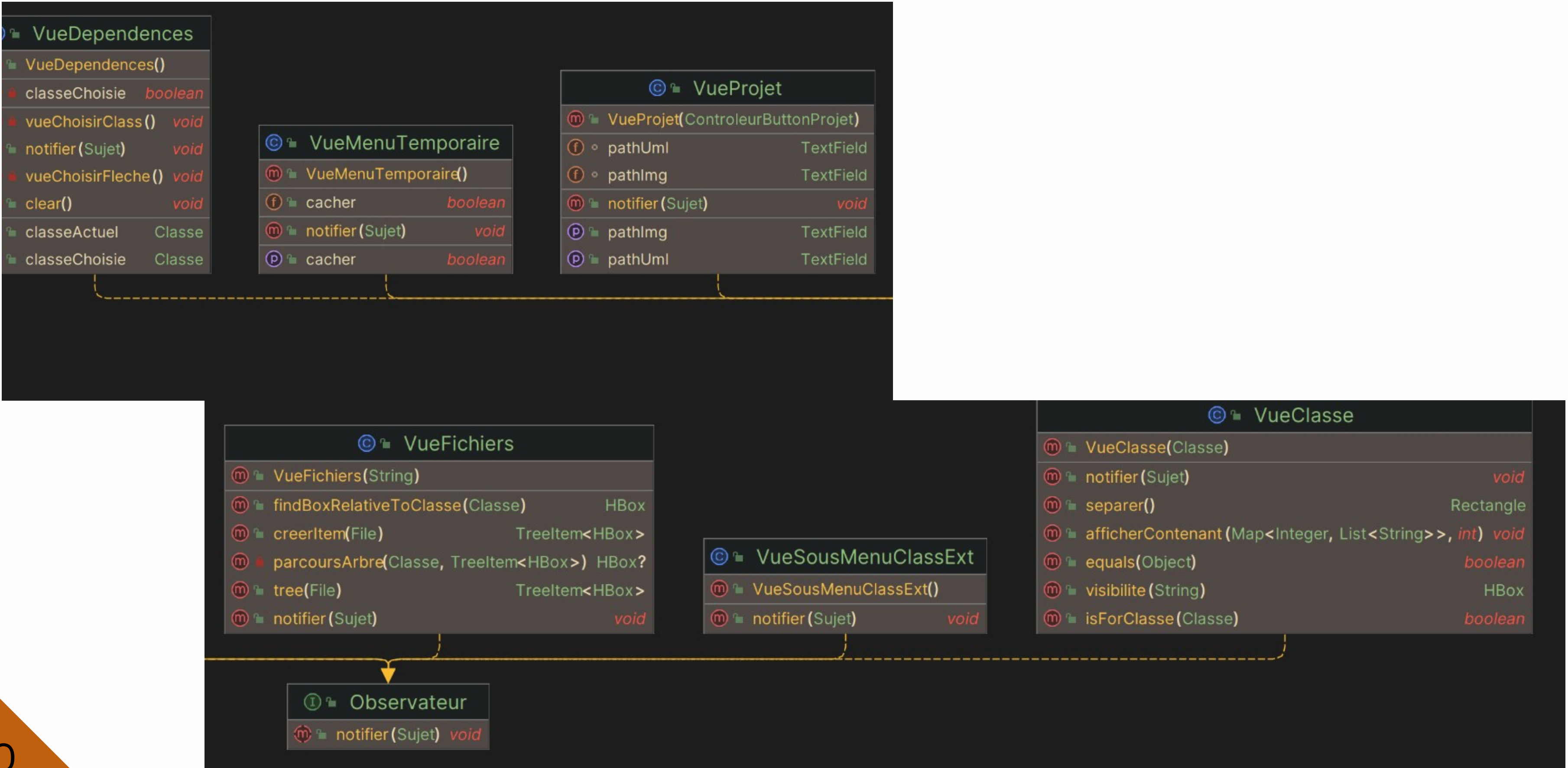
Modèle réalisé:



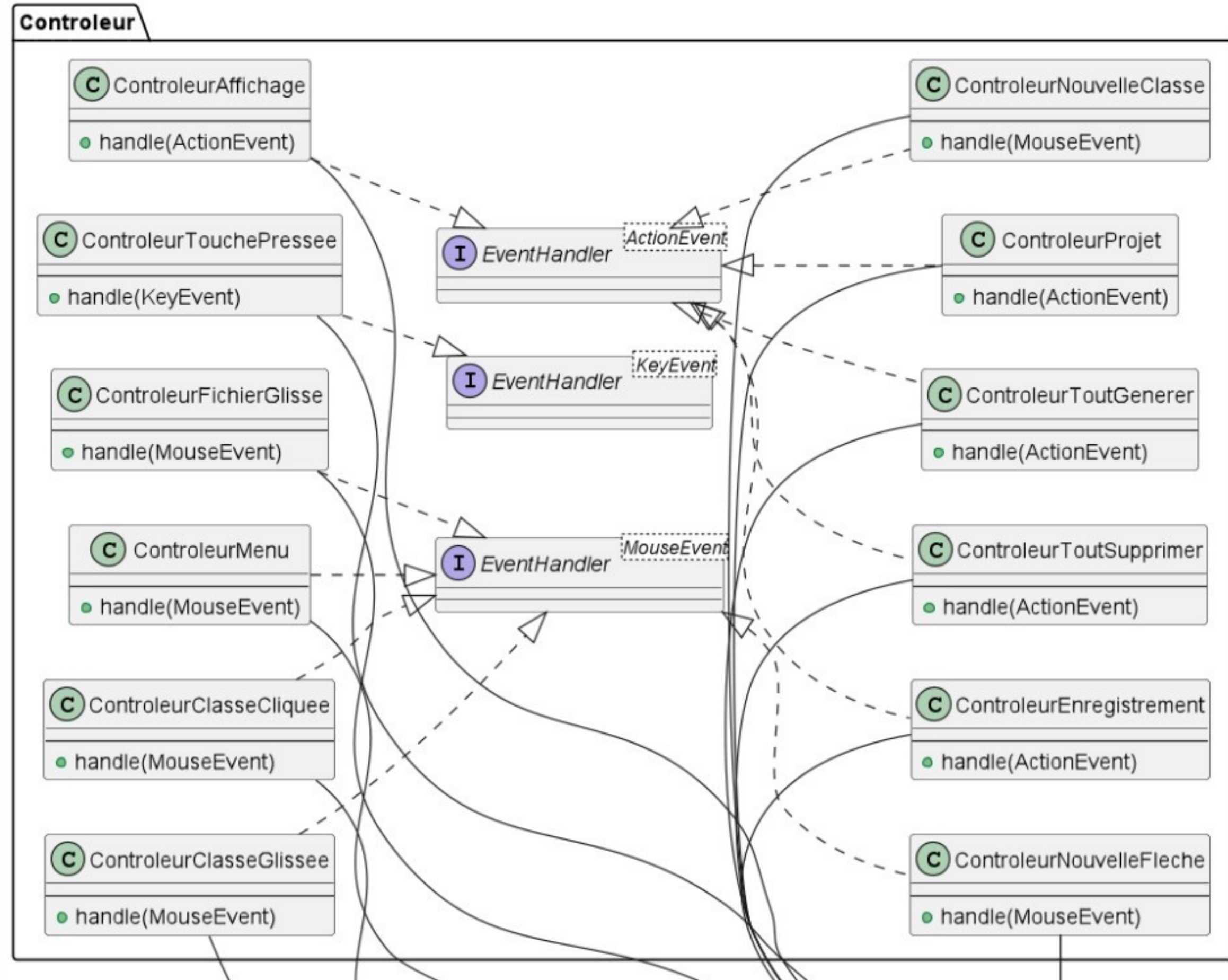
Observateur prévu:



Observateurs réalisés:



Contrôleurs prévus:



Controleurs réalisés:

©  ControleurFichierGlisse

Ⓜ  ControleurFichierGlisse ()

Ⓜ  trouverTIWithPath(String, TreelItem<HBox>) TreelItem<HBox>?

Ⓜ  afficherDossier(String, CheckBox, Model) void

Ⓜ  handle(MouseEvent) void

Ⓜ  afficherUneClasse(String, CheckBox, Model) void

Ⓜ  recupChildrenBranch(TreelItem<HBox>) void

©  ControleurVueTemporaireClasse

Ⓜ  ControleurVueTemporaireClasse(Classe)

Ⓜ  handle(ActionEvent) void

Ⓜ  estMethode(String) boolean

Ⓜ  estAttribut(String) boolean

Ⓜ  affichageCoteDroit(Node) boolean

©  ControleurButtonProjet


Ⓜ  ControleurButtonProjet(VueProjet)


Ⓜ  saveImg(Pane, String) void


Ⓜ  enregUML(String) void


Ⓜ  handle(ActionEvent) void


Ⓜ  openProjetWindow() void


©  ControleurClasseCliquer


Ⓜ  ControleurClasseCliquer(Classe)


Ⓜ  ControleurClasseCliquer(VueMenuTemporaire, Classe)


Ⓜ  handle(MouseEvent) void


©  ControleurButtonNewClass


Ⓜ  ControleurButtonNewClass ()


Ⓜ  handle(ActionEvent) void


Ⓜ  openSecondaryWindow() void


©  HelloApplication


Ⓜ  HelloApplication ()


Ⓜ  main(String[]) void


Ⓜ  start(Stage) void


©  ControleurB


Ⓜ  ControleurButte


Ⓜ  handle(Event)


©  ControleurButtonMenuAddFleche


Ⓜ  ControleurButtonMenuAddFleche ()

Ⓜ  handle(Event) void

©  ControleurCentreClique

Ⓜ  ControleurCentreClique(VueCentre)

Ⓜ  handle(MouseEvent) void

©  ControleurAffichageGlobal

Ⓜ  ControleurAffichageGlobal()

Ⓜ  handle(ActionEvent) void

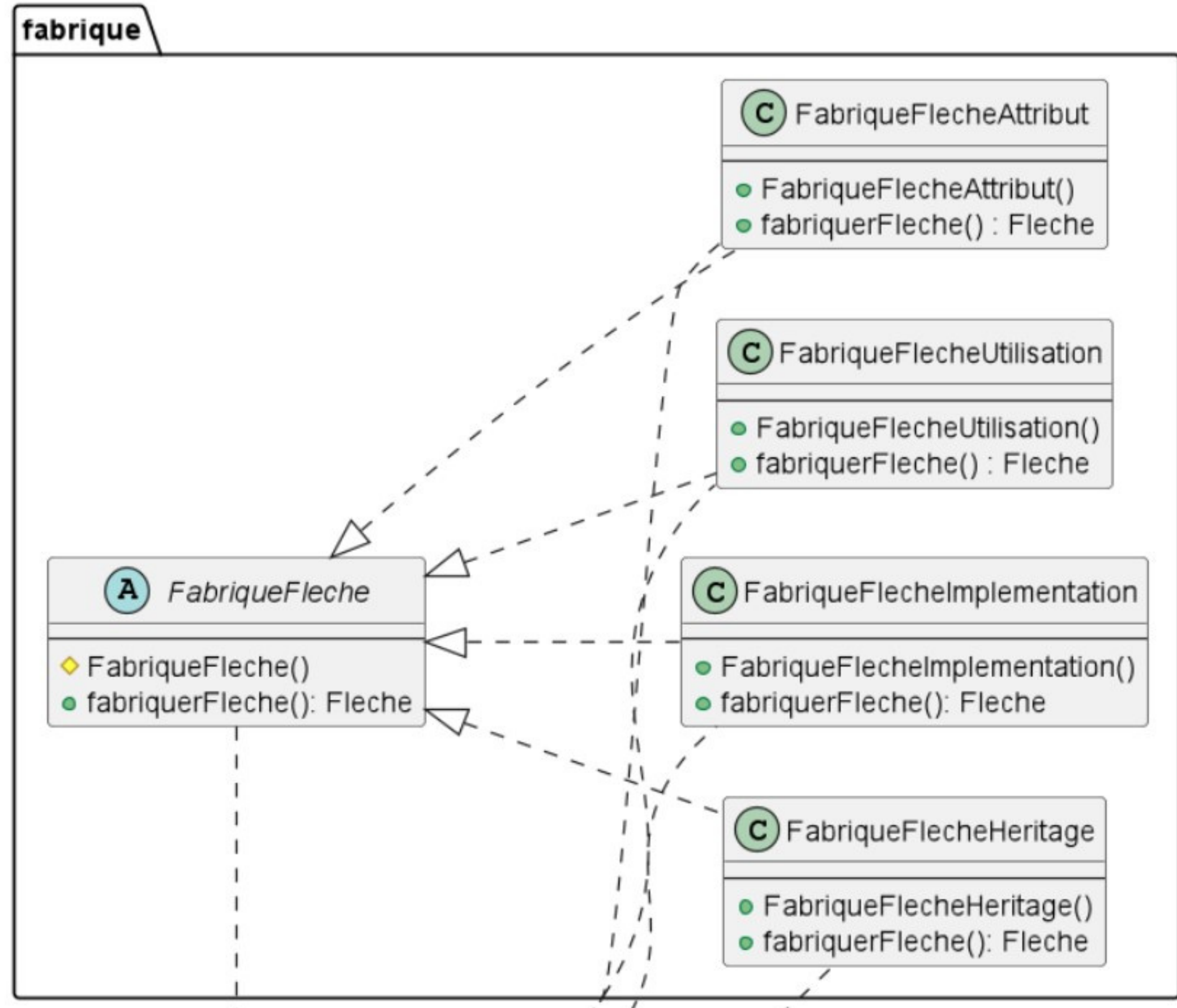
©  ControleurClasseGlissee

Ⓜ  ControleurClasseGlissee(Sujet)

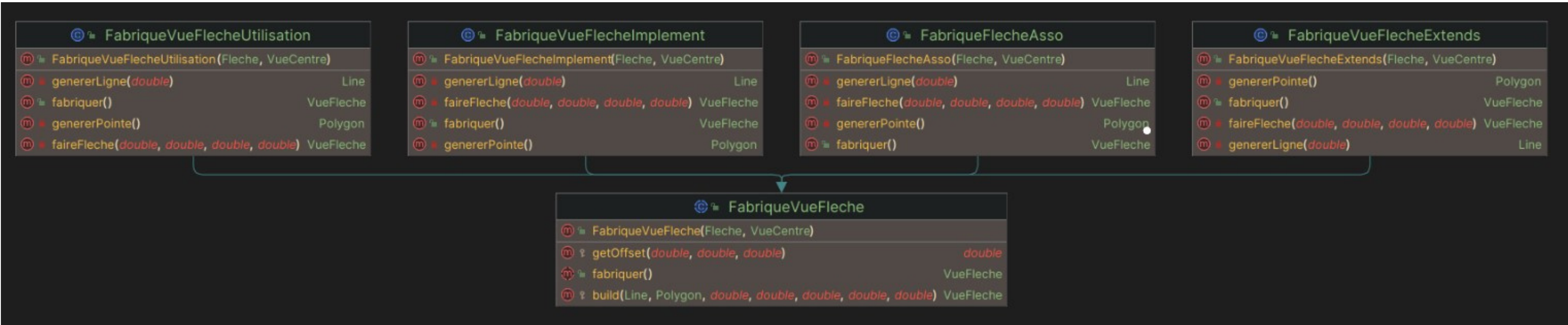
Ⓜ  handle(MouseEvent) void

Ⓟ  classe Classe

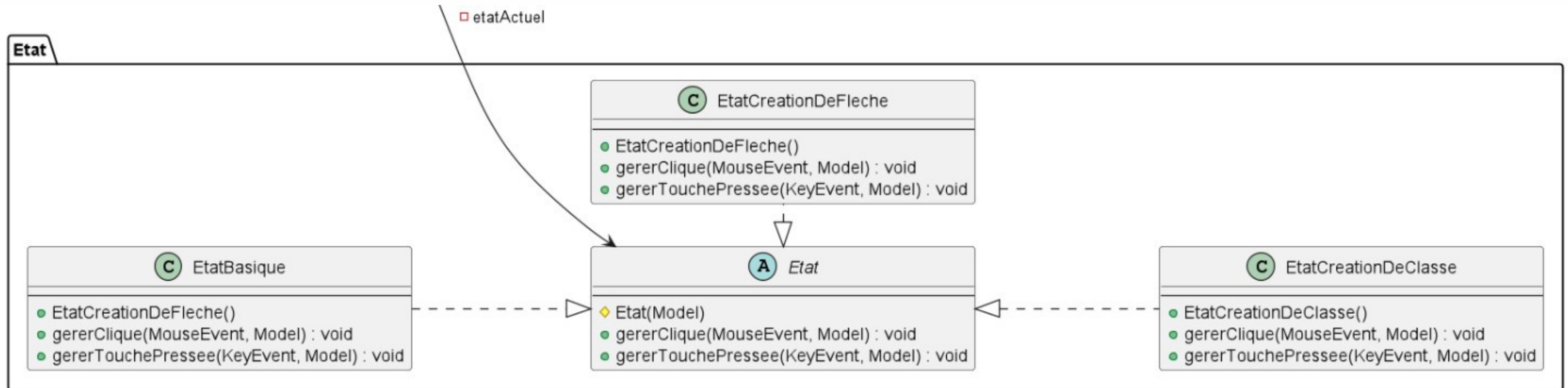
Fabrique prévue:



Fabrique réalisée:



Package état prévu:



Package état réalisé:

PATRONS DE CONCEPTIONS :

Analyse :

- **MVC**
- **Stratégie**
- **Observateur**
- **Fabrique**

final :

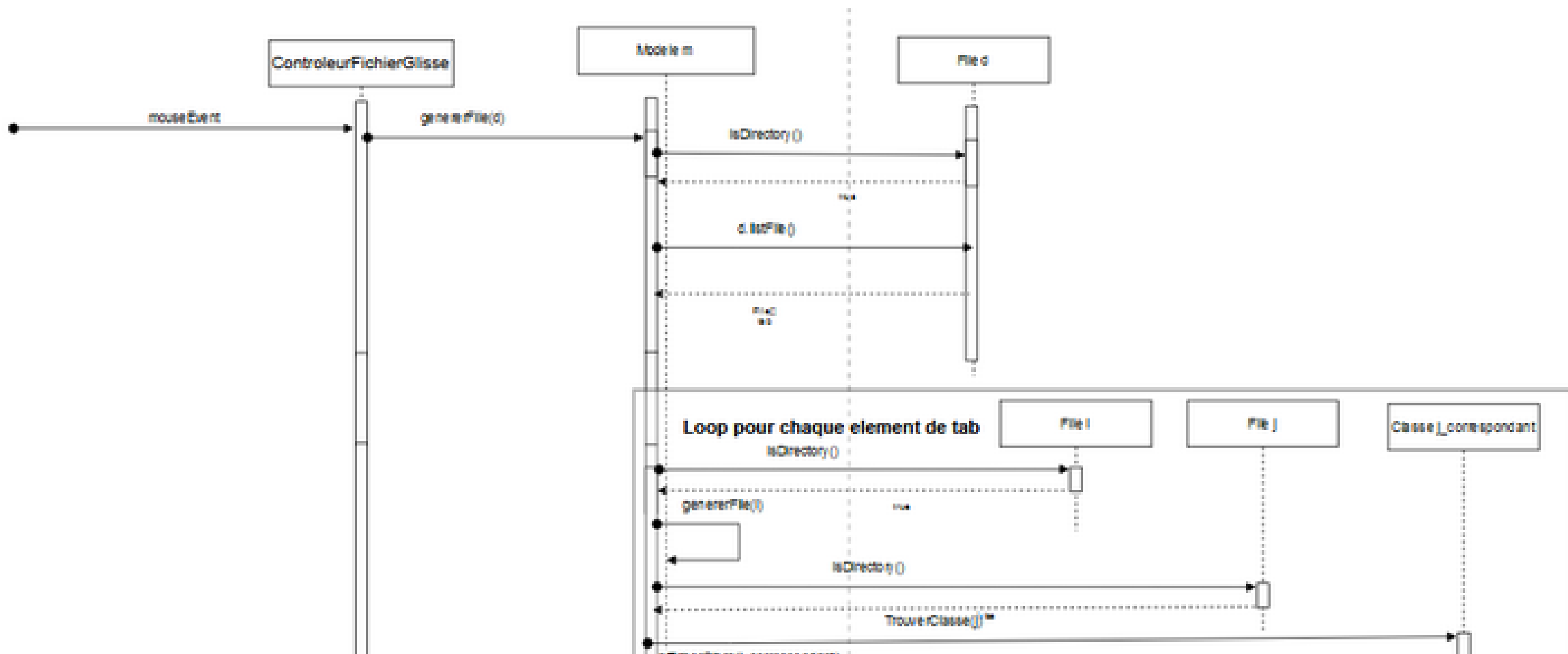
- **MVC**
- **Observateur**
- **Fabrique**
- **Singleton**

Fonctionnalités retenues :

- Glisser déposer des classes.
- Flèches de dépendances
- Enregistrement (format UML ou image)

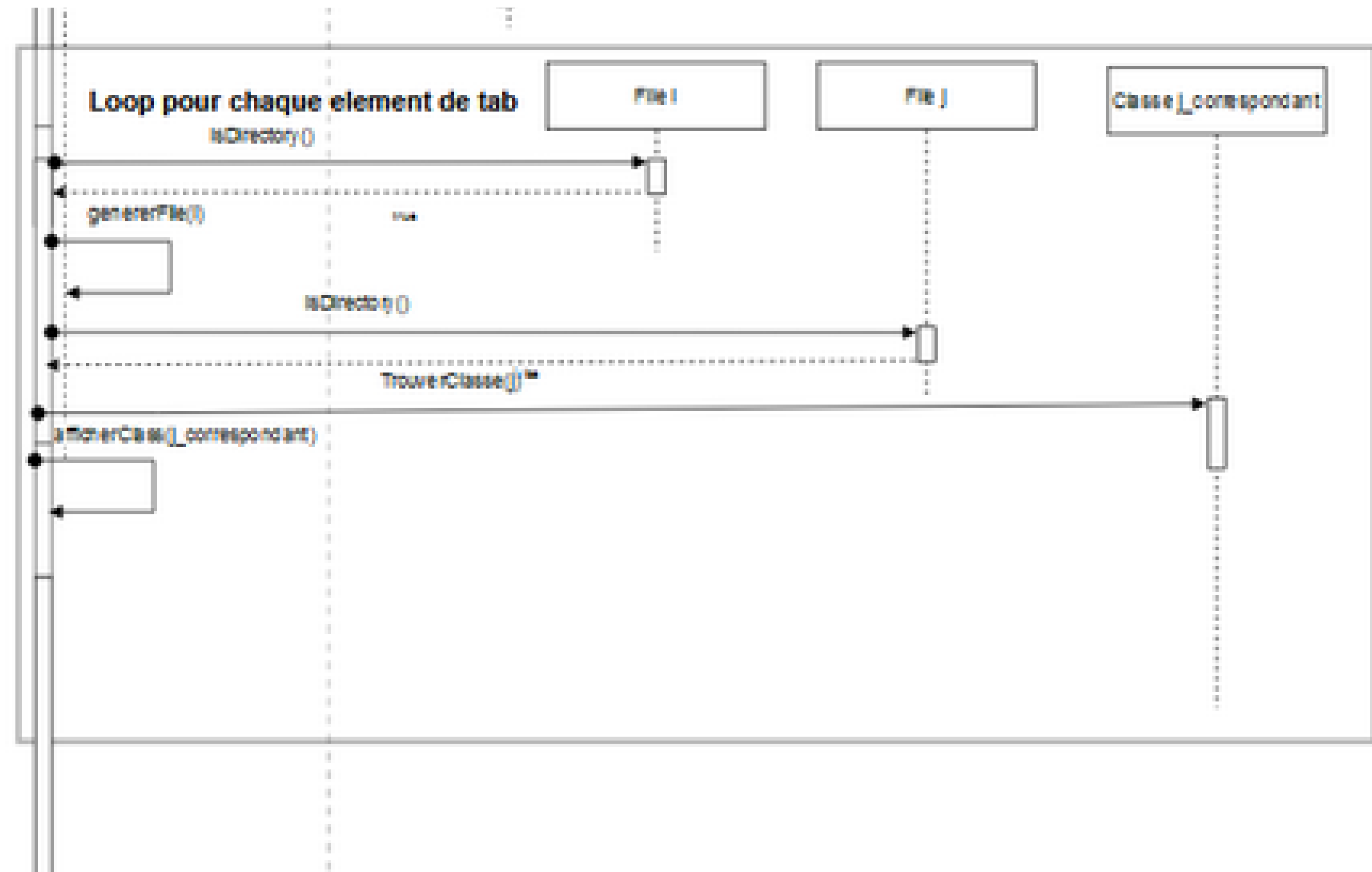
Glisser déposer d'un dossier :

Analyse :



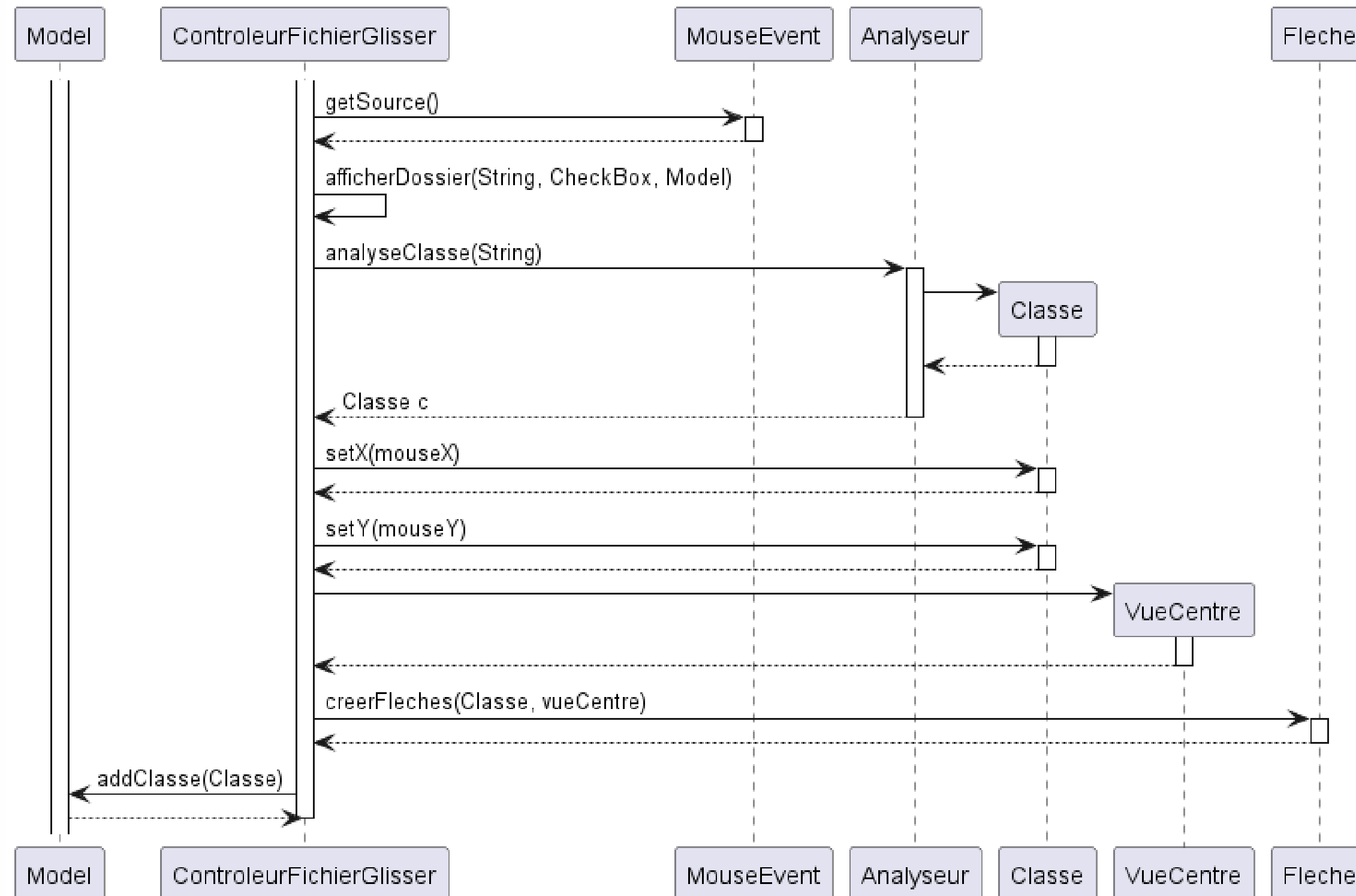
Glisser déposer d'un dossier :

Analyse :



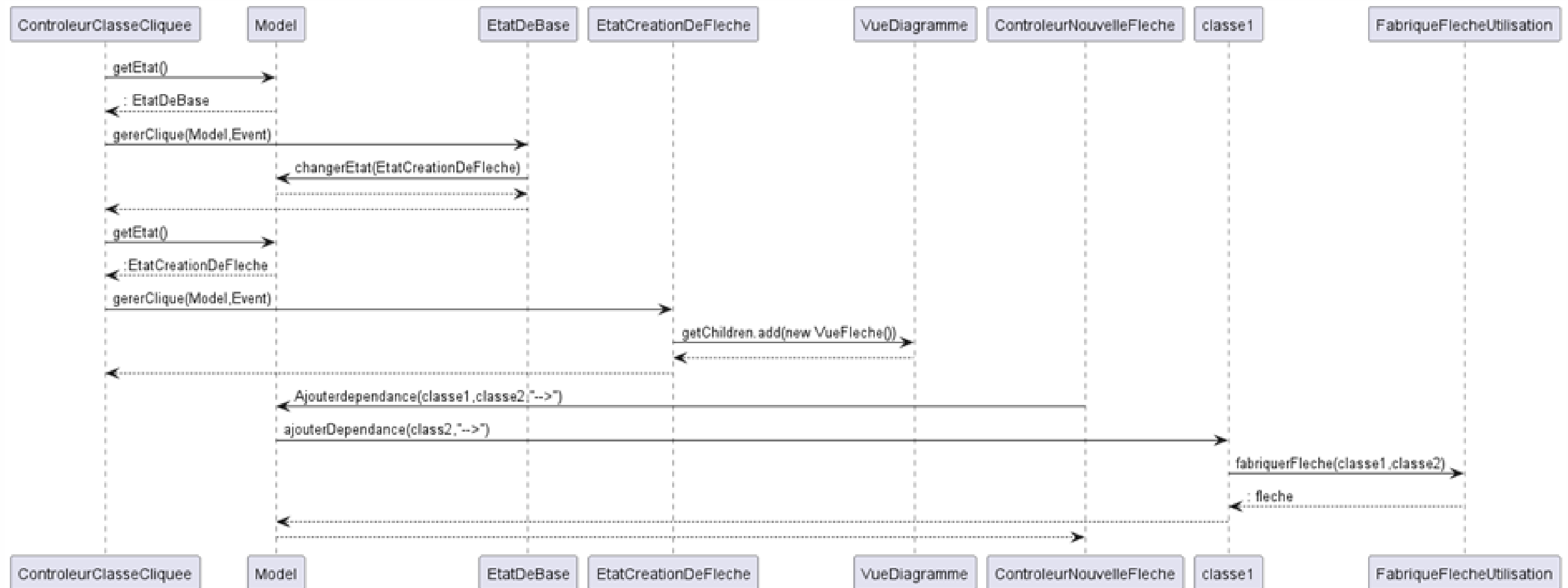
Réalisé :

Glisser déposer d'un dossier :



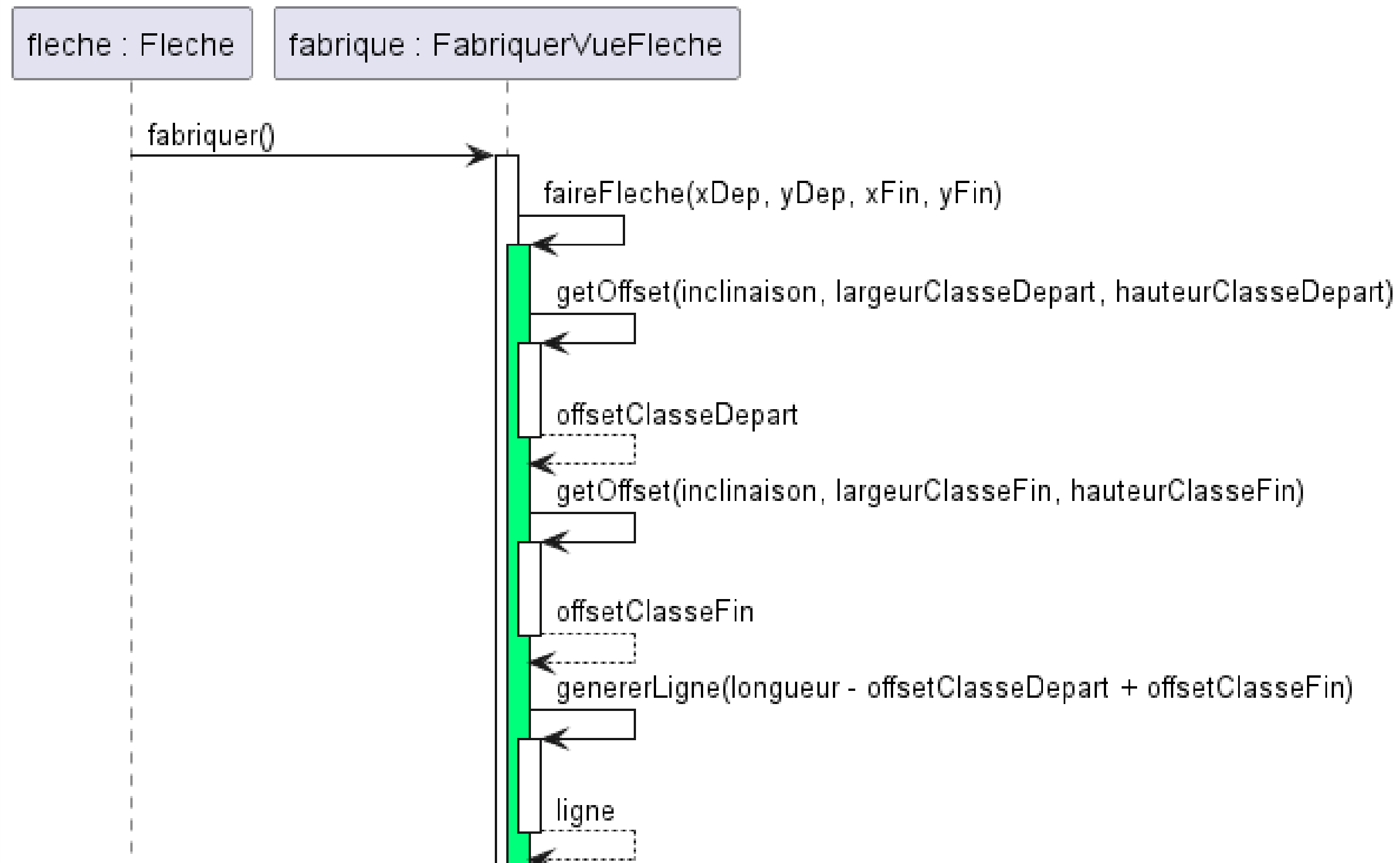
Flèches de dépendances :

Analyse :



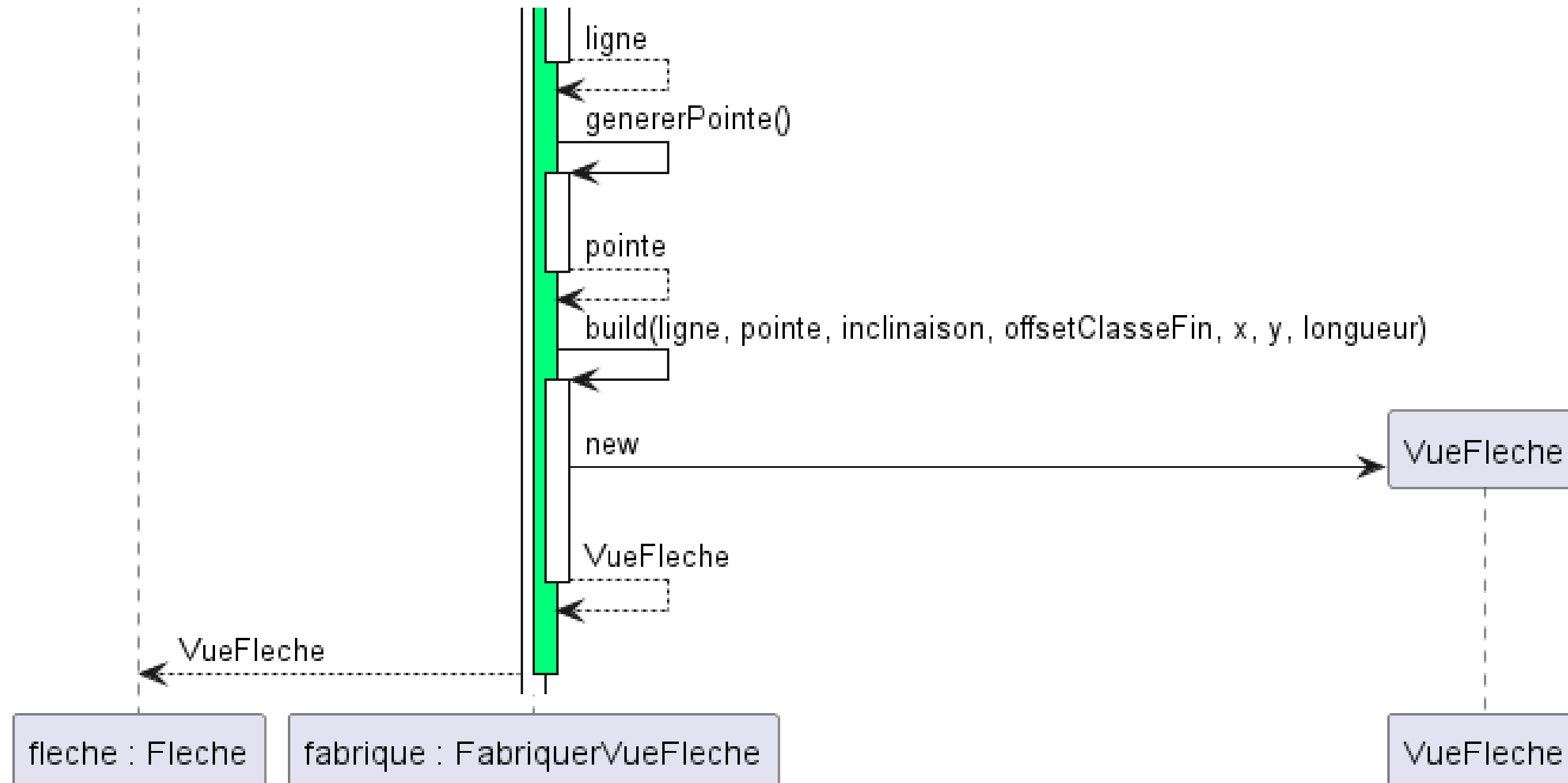
Flèches de dépendances :

Réalisé :



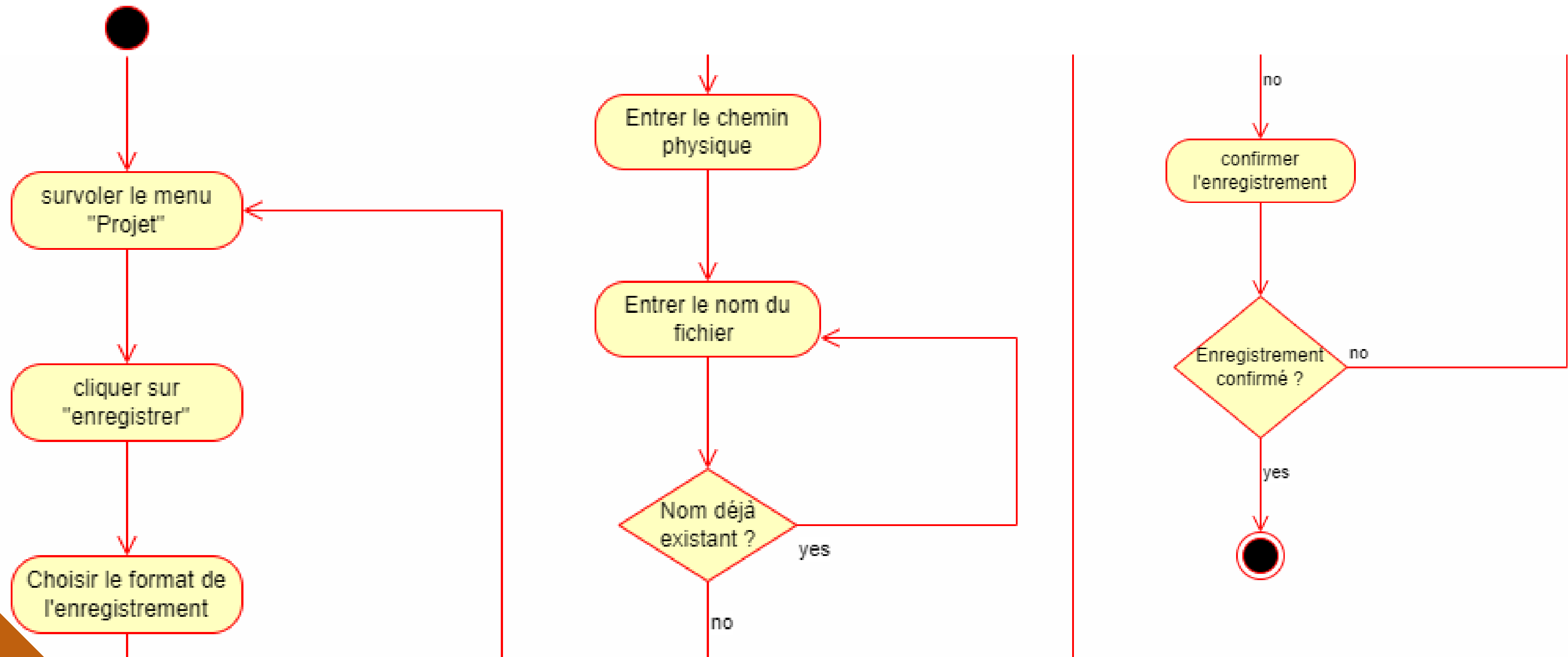
Flèches de dépendances :

Réalisé :



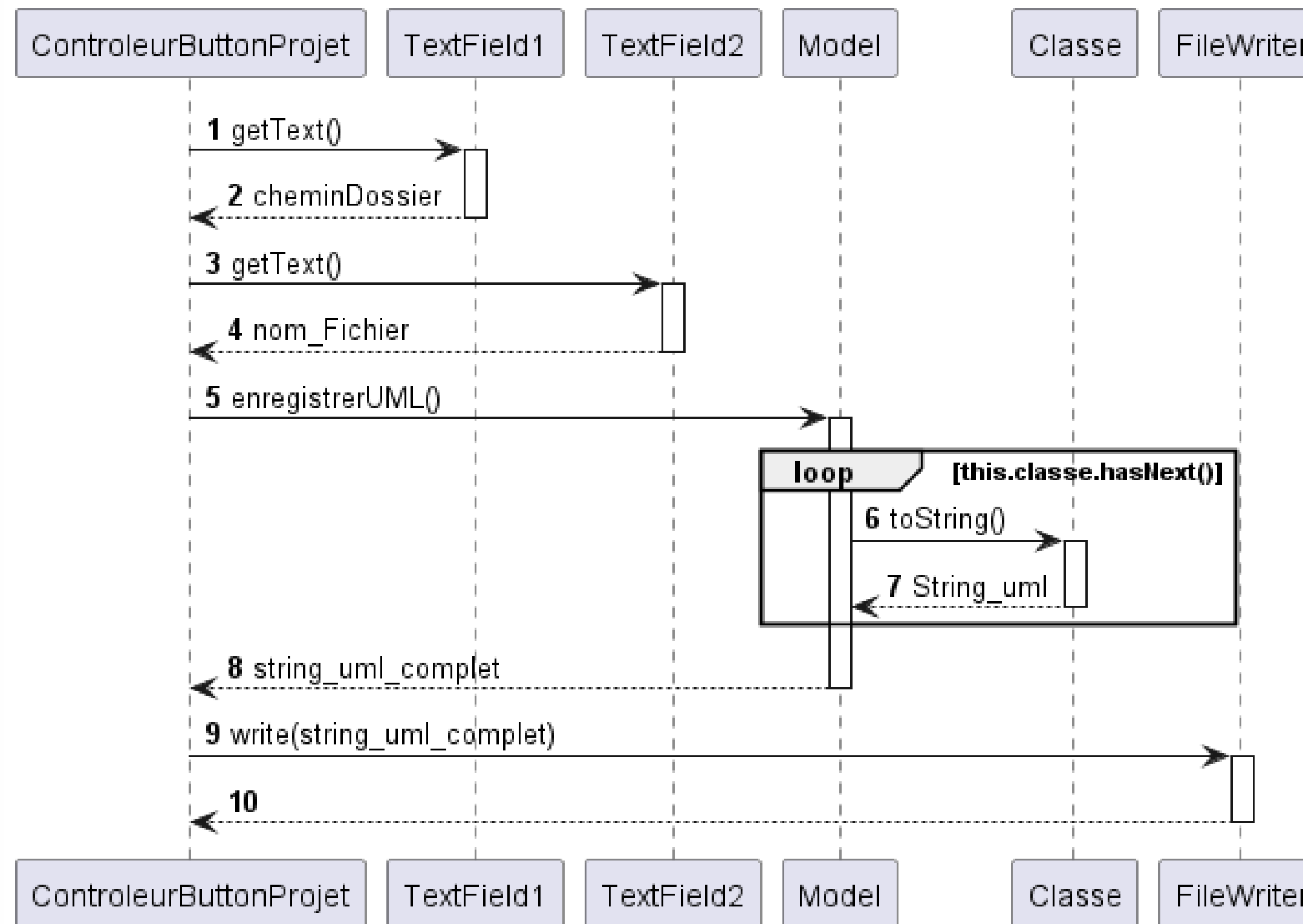
Enregistrer au format UML :

Analyse :



Réalisé :

Enregistrer au format UML :



CONCLUSION

- Respect de toutes les fonctionnalités prévues en analyse.
- Gestion des dépendances drastiquement différente.
- Pas de package d'état donc pas de patron Stratégie

