UML Quicksheet

EPITECH 2013

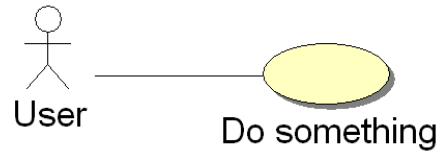
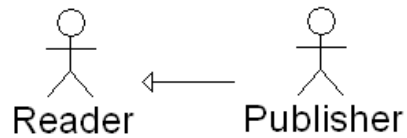
sebatien@migniot.com

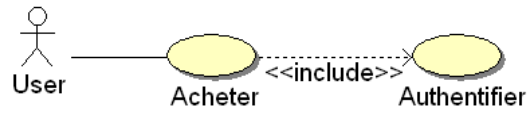
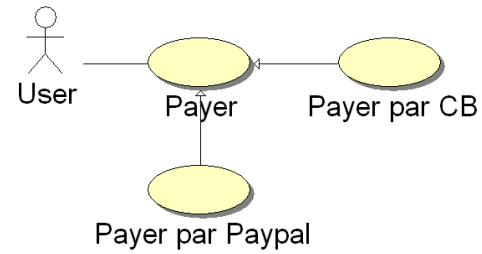
1. LA MÉTHODE

UML sert à représenter des logiciels. UML apporte une méthode

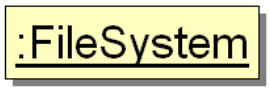
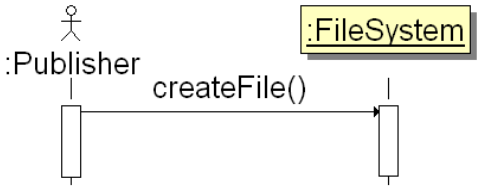
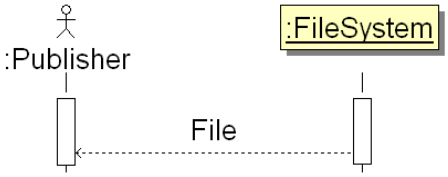
Etape	Description
Use Case	Scenario décrits par les **acteurs** du logiciel
Séquence	Enchainement de **messages** entre acteurs et classes
Classes	Diagrammes de classes, comme en C++

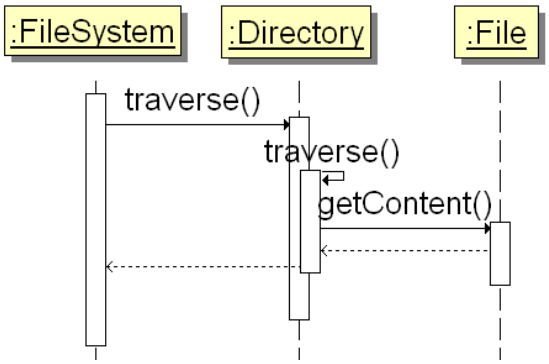
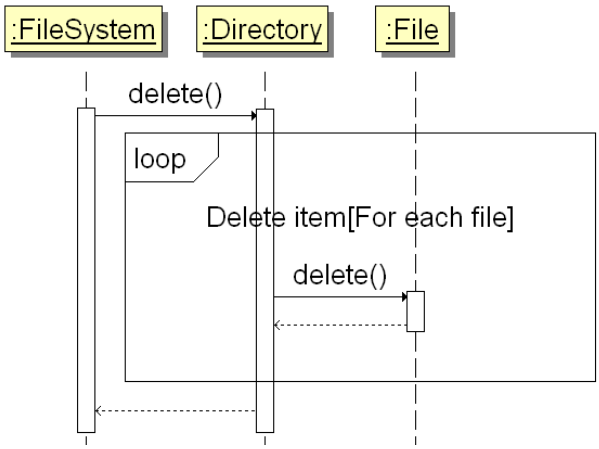
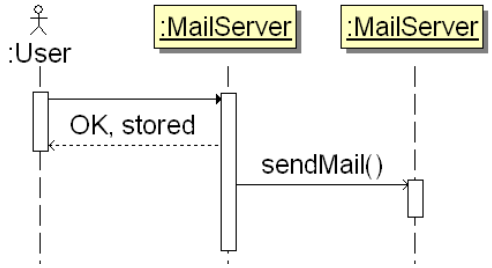
2. NOTATIONS USE CASE

Notation	Description
	**Acteur** : un participant au logiciel
	**Use Case** : un **comportement** accessible a un **acteur**
	**Association** : lien entre un **acteur** et son **usecase** accessible
	**Generalisation :** un **acteur hérite** d'une autre définition **d'acteur**. Ex : un contributeur est aussi un lecteur sur wikipedia

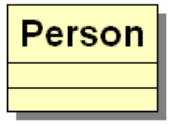
	**Include :** Le déroulement d'un **usecase inclue** *obligatoirement* le déroulement d'un autre
	**Extend :** Le déroulement d'un **usecase** peut être enrichi *optionnellement* par un autre ou **étendu**
	**Généralisation :** Un **usecase** fils *est* un **usecase** parent, par exemple un sous-cas

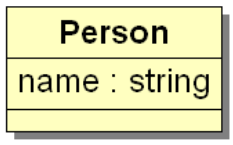
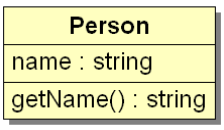
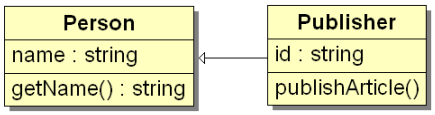
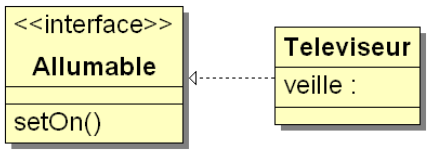
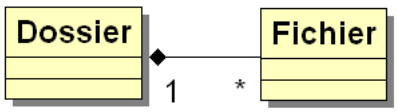
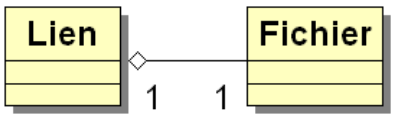
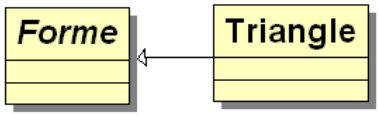
3. NOTATIONS SEQUENCES

Notation	Description
	**Acteur** : un participant au logiciel *issu d'un usecase*
	**Classe** : un **acteur** interne ou logiciel participant à la séquence d'appels
	**Message** : un **participant** appelle une **fonction** sur un autre. *Séquence système : ceux qui partent des acteurs*
	**Retour :** un **message** de *retour*, par exemple createFile() renvoie un File.

	**Message reflexif :** une fonction récursive ou une **fonction** *déjà décrite ailleurs*
	**Bloc** : bloc typé : • De type **loop** pour l'itération • De type **alt** pour if/then/else/switch • De type **opt** pout if simple • De type **assert** pour une condition obligatoire : une exception est lance si non remplie
	**Message asynchrone :** Flèche simple avec *extrémité non remplie* : Le retour peut arriver longtemps après

4. NOTATIONS CLASSES

Notation	Description
	**Classe :** Une **classe**, issue d'un **usecase** ou d'un diagramme de **sequences**

	**Attribut** : Un membre de la classe
	**Méthode** : Une fonction de la classe
	**Généralisation** : Héritage de classe
	**Généralisation** : Implémentation d'interface
	**Composition** : Contenance physique
	**Aggrégation** : Contenance par *pointeur*
	**Classe abstraite** : En italique