Diagramme de cas d’utilisation

• Décrit le comportement d’un système d’un point de vue utilisateur

• Permet de définir les limites du système et les relations entre le système et l’environnement

Acteur : entités qui interagit avec le système (dessin de bonhomme avec nom en dessous)

Cas d’utilisation : ensemble d’actions réalisées par le système en réponse à une action d’un acteur (bulle avec nom au milieu)

Système : définit l’application informatique (rectangle avec nom en haut)

Association : relation entre acteurs et cas d’utilisation qui représente la possibilité pour l’acteur de déclencher le cas (trait horizontal entre acteur et cas d'utilisation)

Inclusion : X inclut Y <=> X implique Y (flèche en V pointillée avec <<inclut>> au milieu) Extension : X étend Y <=> X peut être provoqué par Y (flèche en V pointillée avec <<étend>> au milieu)

Généralisation : X est une généralisation de Y <=> X est un cas particulier de Y (flèche fermée)

Diagramme de classes

• Montre la structure statique d’un système

• Explique ce qu’il faut réaliser, pas comment le réaliser

Classe : regroupement d’objets de même nature (tableau à une seule colonne avec nom en haut, puis attributs, puis méthodes)

Association : relation entre classes (trait entre deux classes), peut être nommée

Rôle : indique comment une classe voit une autre classe au travers de l’association (nom au-dessus de l'association du côté de la classe qui « voit »)

Multiplicité : montre le nombre d’objets liés par une association (nombre en dessous de l'association) • n : exactement n • n..m : de n à m • \* : de zéro à plusieurs • n..\* : de n à plusieurs

Associations multiple : plusieurs associations entre deux classes

Association n-aire : une association reliant plus de deux classes (carré tourné à 45° au milieu de l'association)

Agrégation : montre qu’une classe fait partie d’une autre classe (carré tourné à 45° du côté de la classe qui « contient »)

Héritage : désigne la relation de la classification entre un élément général et un élément plus spécifique (flèche fermée de classe héritante à classe parente) Classe abstraite : classe non instantiable (nom en italique)

Visibilité

• Public (+) : visible partout

• Protégé (#) : visible dans la classe et par tous ses descendants

• Privé (-) : visible uniquement dans la classe

• Aucun : visible uniquement dans le paquetage où la classe est définie Diagramme d'objets

• Une instance d’un diagramme de classe

• Montre l’état du système à un instant donné

Objet : instanciation d’un classe (rectangle contenant objet : Classe ou : Classe ou objet le tout souligné)

Lien : instance d’une association (trait entre objets)

Objet composite : objet classique, sauf que les attributs sont remplacés par des objets

Diagramme de séquence

• Montre les interactions entre objets/acteurs selon un point de vue temporel

• Illustre les cas d’utilisation Acteurs : acteurs d’un cas d’utilisation (même représentation que dans les diagrammes de cas d’utilisation et le diagramme d’objets)

Objets : objets qui communiquent entre eux ou avec les acteurs (même représentation que dans les diagrammes de cas d’utilisation et le diagramme d’objets)

Ligne de vie : précise l’existence d’un l’objet/acteur (trait pointillé en dessous de l'objet/acteur)

Période d’activité : temps pendant lequel un objet/acteur est en activité (rectangle remplace la ligne de vie)

Message : émission d’un événement d’un objet/acteur vers un autre objet/acteur (synchrone : flèche pleine, asynchrone : flèche en V, retour : flèche en V pointillée, avec le nom du message au dessus de la flèche)

Création : création d’un objet qui n’existe pas (flèche pleine avec <<créer>> au dessus) Destruction : destruction d’un objet qui n’existera plus (flèche pleine avec une croix à son extrémité et <<détruire>> au dessus)