

M1 : Ingénierie du Logiciel

UNIVERSITE PIERRE & MARIE CURIE (PARIS VI)

Examen Réparti 1ere partie

7 novembre 2014 (2 heures avec documents : tous SAUF ANNALES CORRIGÉES). Barème indicatif sur 20 points.

1. Questions de cours

[4 Pts]

Répondez de façon précise et concise aux questions.

Barème : VALABLE sur toutes les questions de cours : -25 à -50% si la réponse inclut la bonne idée, mais qu'elle est noyée dans des infos ou autres réponses fausses/inappropriées.

Q1.1 : Pourquoi n'utiliser que des types simples dans les signatures des opérations d'interface des composants ?

Portabilité, + ça n'induit que des dépendances fonctionnelles.

Echanger des classes viole le principe d'encapsulation.

Barème :

50% on cite la portabilité

50% on cite les dépendances fonctionnelles vs structurelles

Q1.2 : Qu'est-ce qu'une métrique de qualité logicielle ? Citez deux exemples.

Métrique : Un outil objectif qui fournit une mesure quantitative quand on l'applique à un sujet donné, ici on veut mesurer de la « qualité » et le sujet c'est donc un logiciel (ou un modèle, ou un procédé... la qualité logicielle c'est vaste)

La valeur absolue doit s'interpréter dans le contexte du projet (certains projets sont plus complexe ou gros intrinsèquement), souvent on surveille plus son évolution au fil du temps que sa valeur absolue, e.g. les variations rapides peuvent alerter sur des problèmes réels.

Exemples : nombre de lignes de code total, nombre de classes, profondeur des arbres d'héritage, nombre de cycles de dépendance, nombre d'opérations par classe, lignes de code par classe, lignes de code par méthode (moyen et max), diverses métriques de couplage interne des classes et inter classes existent...

Barème :

50% réponse qui montre qu'on sait ce qu'est une métrique de qualité

50% : 25% par métrique pertinente citée *2.

Q1.3 : Expliquez à quel moment(s) dans le projet il faut réaliser les étapes de l'Analyse (branche construction ET branche validation) du cycle en V présenté dans l'UE.

Début projet

Analyse + rédaction des tests de validation

Présentation TV au client (retour éventuel en analyse)

Conception et réalisation (TUNNEL)

Recette (basée sur TV)

Barème :

binaire 0/100% : on doit clairement comprendre qu'il y a un bout au début et un bout (la recette) à la fin du projet quand on remonte avec le produit intégré.

Q1.4 : Quels sont les alternatives possibles pour intégrer des éléments algorithmiques à un modèle UML ?

Alt 1 : modélisation à l'aide de Behavior UML, e.g. diagrammes d'activité

Alt 2 : notes de code, pas strictement UML.

Barème :

50% chaque alt.

2. Problème: Analyse de eProd [16 Pts]

On souhaite mettre en place un système informatique pour gérer la production d'une société produisant des vélos. L'application eProd doit permettre une gestion raisonnée des stocks et des commandes sur le site de production (usine).

Les vélos sont produits à partir de pièces détachées, assemblées sur le site de production. Par exemple,

- un vélo est constitué d'un cadre, de deux roues, d'une selle, de dix boulons etc... Le vélo est assemblé unité par unité (à la main par un ouvrier spécialisé) ce qui coûte 8,50 euros.
- Le boulon coûte 0,02 euros, s'achète chez Untel par paquets de 1000.
- Le tricycle est composé de trois roues, d'un cadre, d'une selle...l'assemblage (1 unité) coûte 7 euros.
- La roue est composée d'une jante, de 50 rayons, de 50 boulons, d'un moyeu... On assemble les roues par 10 sur la chaîne d'assemblage, ce qui coûte environ 12,43 euros par roue...

Le catalogue de eProd contient donc la description de l'ensemble des produits : chaque produit a un prix, une référence unique et une description.

Les pièces détachées sont commandées en bloc: pour chaque pièce détachée on sait le nombre d'unités dans un bloc. Chaque pièce détachée n'a qu'un seul fournisseur dont on connaît au moins le nom, l'adresse, et le téléphone du service commandes. Le prix d'une pièce détachée est le prix unitaire pratiqué par le fournisseur.

Pour chaque pièce assemblée, on dispose du prix d'assemblage unitaire qui reflète le temps nécessaire à produire l'assemblage. Comme la mise en route de la chaîne d'assemblage nécessite d'assembler en série, on dispose également du nombre d'unités produit par cycle de production. On a également la liste des pièces détachées et/ou assemblées (et la quantité de chacun) nécessaires pour produire la pièce. Le prix d'une pièce assemblée se calcule comme la somme des prix de ses constituants et du prix d'assemblage.

La réalisation d'une application permettant la construction du catalogue (saisie et édition de ces informations de catalogue et de fabrication) a été confiée à une autre branche de votre société.

eProd sera déployé sur les sites de production (usines). Chaque site de production dispose d'un stock, qui contient des pièces détachées et des pièces assemblées jusqu'aux produits finis. Les pièces du stock peuvent être « disponibles » ou « réservées » si une commande en cours de traitement va en faire usage.

Le gestionnaire du stock (GS) doit enregistrer la réception des livraisons des pièces détachées par les fournisseurs quand elles arrivent. Chaque fois qu'une pièce est assemblée, l'opération doit être enregistrée par le GS, ce qui met à jour la quantité de composants disponibles et ajoute les pièces assemblées au stock. Enfin quand les produits finis quittent l'usine pour honorer une commande, le GS l'enregistre ce qui fait basculer la commande à l'état « traitée » et met à jour le stock.

Les commandes à l'usine sont traitées par les gestionnaires commandes (GC), et concernent des pièces en nombre (>10 unités). Les commandes proviennent de la maison mère et doivent toutes être réalisées. Le GC commence par saisir la commande dans une interface graphique dédiée de eProd. Le GC peut visualiser la liste des commandes en cours et leur état. L'état des commandes est calculé et maintenu à jour en continu par eProd. La commande est à l'état

- « traitée » si la livraison a été enregistrée par le GS,
- « attente livraison » si les produits demandés sont immédiatement disponibles dans le stock,
- « en cours » si l'ensemble des pièces nécessaires à la production sont disponibles en stock même si les produits finis demandés ne le sont pas,
- ou « commande nécessaire » s'il faut recommander des pièces détachées aux fournisseurs pour mener la production à bien.

Dans le cas commande nécessaire, eProd doit aider le GC à établir des commandes auprès des fournisseurs : pour chaque fournisseur eProd devra lister les pièces à commander en quantité minimale (par bloc) pour honorer la commande. Le calcul des pièces détachées nécessaires doit prendre en compte les contraintes de production en série des pièces assemblées (on doit faire des cycles de production complets). L'état des commandes évolue au fil des actions enregistrées par le GS ; dès qu'une commande passe de « commande nécessaire » à « en cours » l'ensemble des pièces nécessaires à la production est réservé dans le stock.

Question 2.1 : (3 pts) Réalisez le diagramme de cas d'utilisation de la phase d'analyse. Vous justifierez tous vos choix, par un texte ou des annotations sur le diagramme.

Sur 100%.

Acteurs 10% : GS, GC. 0% si autre acteur primaire identifié. Acteur secondaire SICatalogue (l'autre filiale) toléré mais pas de points : l'intégration se fait hors use case.

UC GS 45% : enregistrer livraison pièces (15%), enregistrer production (15%), enregistrer livraison sortie usine (15%)

UC GC 45% :

- Saisir commande(15)
- Visualiser commandes (15),
- Gérer Commande fournisseur (10) (+ 5 si présenté en extends sur visualiser commandes)

Fautes fréquentes :

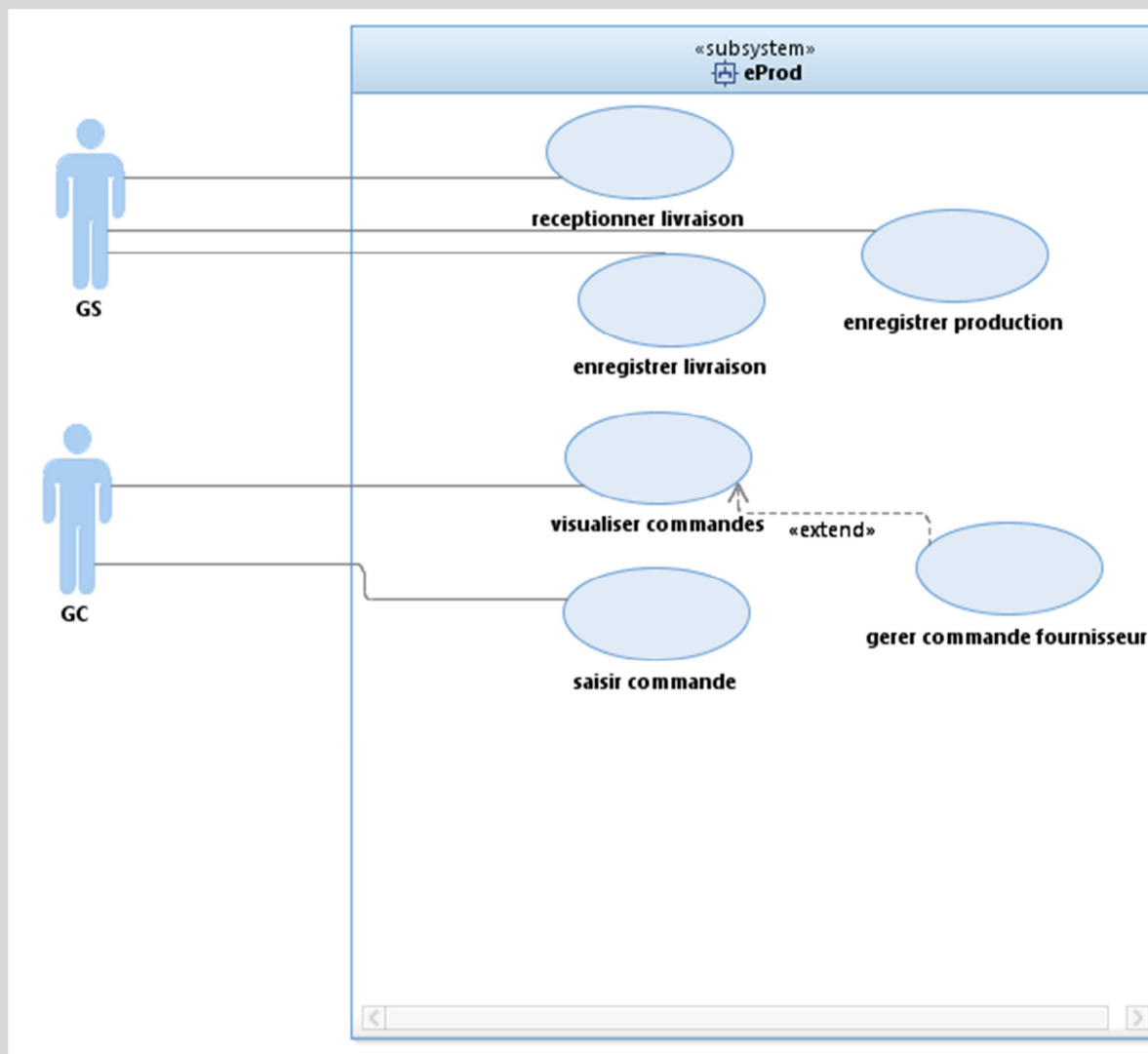
U.C : saisir catalogue, lié ou pas à un acteur « autre filiale » : hors sujet -10%.

U.C : maj stock, typiquement en include de un ou plusieurs use case : -20%

-5 à -15% pour les fautes

- 10 si tout est détaillé (calculs des pièces nécessaires en particulier) ça découle des use case, mais ce n'en sont pas.
- 10 par use case hors sujet identifié (saisie catalogue...)
- 5 par use case mal formulé : on veut un verbe qui exprime l'action du point de vue de l'acteur.

DIAGRAMME



- 10% aucun commentaire/aucun texte pour accompagner le diagramme, diagramme sec
- 20% par héritage, include ou extend injustifiable ou autre incohérence/mésusage d'UML.
- 10% si on ne précise pas qui fait l'action dans le scenario (use case sans acteur lié)

Question 2.2 : (3 pts) Précisez la feuille détaillée (acteurs concernés, hypothèses/pré-conditions, post-conditions, scénario nominal, alternatives, exceptions) du (ou des) cas d'utilisation(s) correspondant à la phase où le GS enregistre la fin d'une opération de production.

TODO : peu d'alternatives dans le workflow, mais pas mal de mise à jour alt à spécifier, au niveau du stock et de l'état des commandes.

Titre : Enregistrer production

Acteur : GS

Hypothèse : aucune

Pré : aucune

Post : Le stock est mis à jour. L'état des commandes est mis à jour si nécessaire.

Scénario :

1. Le GS choisit d'enregistrer une opération de production
2. Le système affiche un formulaire permettant de sélectionner la pièce produite
3. Le client saisit la référence produit de la pièce assemblée et valide
4. Le système met à jour le stock : les pièces intervenant dans la construction sont retirées du stock ($\text{nbUnitesParCycle} * \text{nombre intervenant dans la composition}$), les pièces construites ($\text{nbUnitesParCycle} \text{ unités}$) sont ajoutées au stock.

Alternative A1 : MAJ commande

En SN4, si les pièces nouvellement produites permettent de satisfaire les besoins d'une commande à l'état « en cours », celle-ci bascule à l'état « attente livraison ».

Exception E1 : Annulation

En SN3 l'utilisateur peut choisir d'annuler. Le système affiche alors l'écran d'accueil de l'utilisateur.

Barème : sur 100%

20% : post conditions, 10% maj stock, 10% maj etat commande

40 % : scenario :

- 10% : trigger initiative acteur
- 10% : affichage (SN 2)
- 10% : SN3
- 10% SN4 : maj stock

10% : la spécification de la maj du stock est précise, e.g. mentionne les contraintes production en série, ici ce texte est dans SN4, mais il peut être placé ailleurs (e.g. post-cond)

20% : ALT A1 ou autre spécification correcte des conditions de mise à jour des commandes

10% : E1 : on peut annuler à au moins certaines étapes

On peut éventuellement donner 10% pour des ALT ou exception bien spécifiées qui rattrapent d'éventuelles erreurs de saisie (ici le formulaire de SN2 empêche ce genre d'erreurs).

Cette question est très délicate à corriger. Il faut donc vérifier les points suivants.

-10 à +10% cohérence globale du texte, utilisation correcte des champs

Pré/Post/Scénario etc... En particulier, -10% si les préconditions/hypothèses sont testées

dans le scenario et +10% si les étapes sont bien affectées à acteur ou système et que la première action est à l'acteur

+10% la spécification de comment le stock est mis à jour est précise, et mentionne les contraintes production en série.

+10% E1 : on peut annuler à au moins certaines étapes

Erreurs fréquentes :

-10% le client n'a pas l'initiative (déclencheur)

-10% on ne sait pas clairement qui du système ou de l'acteur fait l'action dans une étape du scenario

-20% si on a découpé en plusieurs use case en question 1, mais que leur description détaillée n'est pas cohérente avec le diagramme. E.g. on a mis un include maj stock, mais pas de spéc cohérente (ce serait a priori du coup une étape du nominal) ici de ce comportement.

-15% :Spécification d'étapes hors système comme étapes du scenario (e.g. l'agent téléphone à l'employé) .

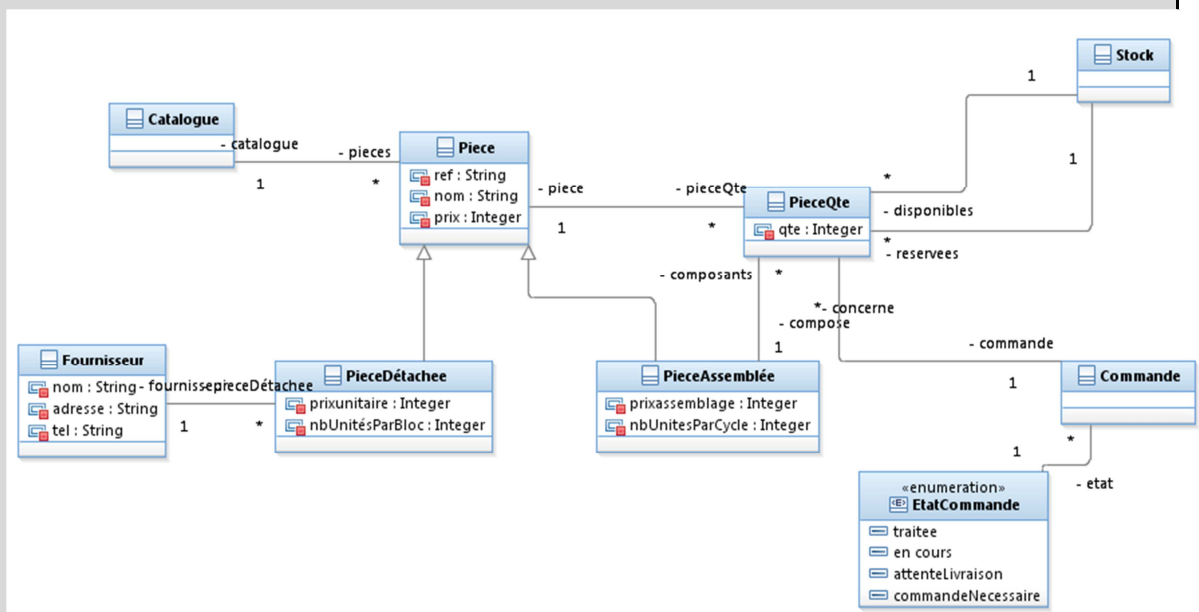
-10% à -30% les séquences sont mal expliquées/peu détaillée (ou compter 5 au lieu de 10% quand c'est mal expliqué la séquence)

Jusqu'à -50% si les pré et post condition sont incohérentes avec le scénario nominal

Jusqu'à -50% si le scénario fait apparaître des interactions entre des entités autres que les acteurs et le système

Question 2.3 : (5 points) Réalisez le diagramme de classes métier de la phase d'analyse. Vous justifierez tous vos choix, par un texte ou des annotations sur le diagramme. On ne représentera pas la classe représentant le « Système », introduite dans l'approche en V du module.

TODO : il est assez costaud quand même. On va sanctionner les classes vélo/tricycle mais ils risquent fortement d'en proposer, donc on sera pas trop durs.



Voici ma correction : 10% par élément, ne donner que 5 si la formulation est incomplète (e.g. cardinalités mal renseignées, il manque des attributs parmi ceux recherchés,...).

Composite (50%):

10% Piece, ref, nom, prix

10% PieceDétachée : extends Piece (5%) prix unitaire (ou pas, il peut être confondu avec le prix du produit), nbUnitésBloc

10% fournisseur avec attributs, lié à pièce détachée

20% PieceAssemblée : 10% attributs + extends, 10% lien sur Piece (children de Composite). Ne donner que 5% si les quantités ne sont pas explicitement représentées (on acceptera une simple assoc * --sans PieceQte du corrigé-- s'il y a au moins une note de commentaire sur cette association). On acceptera aussi les solutions avec un tableau d'entiers par exemple pour tracer les quantités de chaque composant.

Stock (20%) :

- 10% dispo, 10% réservées. D'autres modélisations sont possibles, mais on doit identifier la nature réservée ou non des pièces dans le stock.
- Donner 15% si les deux notions sont présentes mais que les Qte ne sont pas explicites.
- Donner seulement 10% pour un booléen ou enum sur Piece, vu que certaines occurrences peuvent être réservées et d'autres dispo, donc ça ne modélise pas de façon adéquate le CdC.

Commande (20%):

20% commande : 10% son état avec un enum, 10% concerne (5% si Qte pas explicite)

Catalogue : 10% avec son association sur pieces.

Malus :

-10% si associations orientées, compositions etc...

-10% si opérations sur les classes

-10% à 20% pour toute autre faute ou aberration

Les classes Velo, Tricycle etc... ne donnent pas de points, mais pas de malus non plus.

Question 2.4 : (2,5 pts) a) Réalisez un diagramme de séquence de niveau analyse présentant le déroulement (scénario **nominal**) de la saisie d'une commande concernant 20 vélos et 15 tricycles par le GC. b) Dessinez la classe « système » afin de préciser les opérations identifiées (signature, visibilité).

Essentiellement, on doit voir sur ce diagramme toute l'information circuler de l'acteur vers le système, sous une forme ou une autre. On ne doit pas nécessairement voir le « calcul pièces nécessaires » en self call, mais ça reste pertinent de le modéliser.

Plus de points si la modélisation structure les données passées, i.e. pas juste un string.

Donc :

a) :



b) :

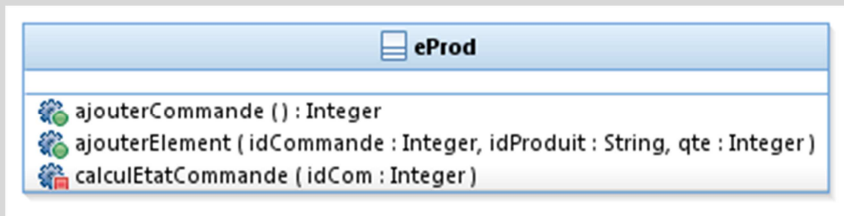


DIAGRAMME TODO

Barème :

A) 70%

+20% lignes de vie correctes : acteur vs système

Messages (30%):

- +10% creation commande
- +20% ajout des éléments à la commande
- On donne +10% (au lieu de 30%) si la signature est peu explicite mais qu'on voit les données transiter (String ou autre typage vague). Typiquement c'est le cas si on a une seule invocation pour créer la commande complète.

+20% modélisation (self loop ou commentaire) de la mise à jour état commande/recherche composants dans le stock/qqch de ce goût.

-10% on ne voit pas clairement que les lignes de vie sont des instances (notation o:Obj)

-20% si appel du système à une opération de l'acteur Agent (e.g. avec une demande de saisie par l'agent). L'envoi asynchrone d'un message, ou une note expliquant qu'on considère que Joe représente l'acteur et son IHM => -10%. Cela reste incorrect. On cherche les responsabilités du système, pas des acteurs (donc externes au système).

B)

30% : signatures cohérentes avec le diag de séquence et réalisables (j'ai vu un `<Int,String>[*]` par exemple, pas terrible), 0% dès qu'une incohérence est constatée. Les méthodes correspondant à d'autres use case sont tolérées mais ne donnent pas de points (hors sujet)

les self calls doivent être private (-10% si ce n'est pas le cas).

Question 2.5 : (2,5 pts) Ecrivez un test de validation couvrant l'enregistrement par le GS d'une livraison sortie d'usine pour honorer une commande.

TV042 : Test sortie usine

Contexte : A exécuter sur le poste GS après avoir exécuté le test 41 : saisie production 10 vélos. La commande « comT01 » est donc à l'état « attente livraison »

Entrée : numéro commande « comT01 »

Scenario :

1. Le GS sélectionne l'action « sortie usine »
2. Il saisit le numéro de commande « comT01 »
3. Il valide

Résultat attendu : le système affiche une confirmation. La commande est passée de « attente livraison » à « traitée ». Le stock contient 20 vélos et 15 tricycles en moins.

Moyens de vérification : visuel pour la confirmation. Sur le poste GC dans l'interface « visualiser commandes » on peut contrôler l'état de « comT01 » et sa mise à jour temps réel. On peut aussi constater la mise à jour du stock sur le poste GS.

Barème :

20% le contexte précise qu'une commande est « attente livraison ». Il doit être suffisamment précis pour être reproductible, ou on ne donne pas les 20%.

10% toutes les données saisies sont mentionnées dans la section « entrée » (id commande...)

20% scénario réalisable, qui ne mentionne que les actions utilisateur, précis (reproductible sans réfléchir)

30% résultats attendus : 10% confirmation, 10% maj stock, 10% maj etat commande

20% le moyen de vérification est raisonnable et permet de contrôler au moins un des deux : la maj stock ou la maj etat commande. « Visuel » ne donne pas de points.

-25% le scénario mentionne des actions du système (autre que résultat attendu)