

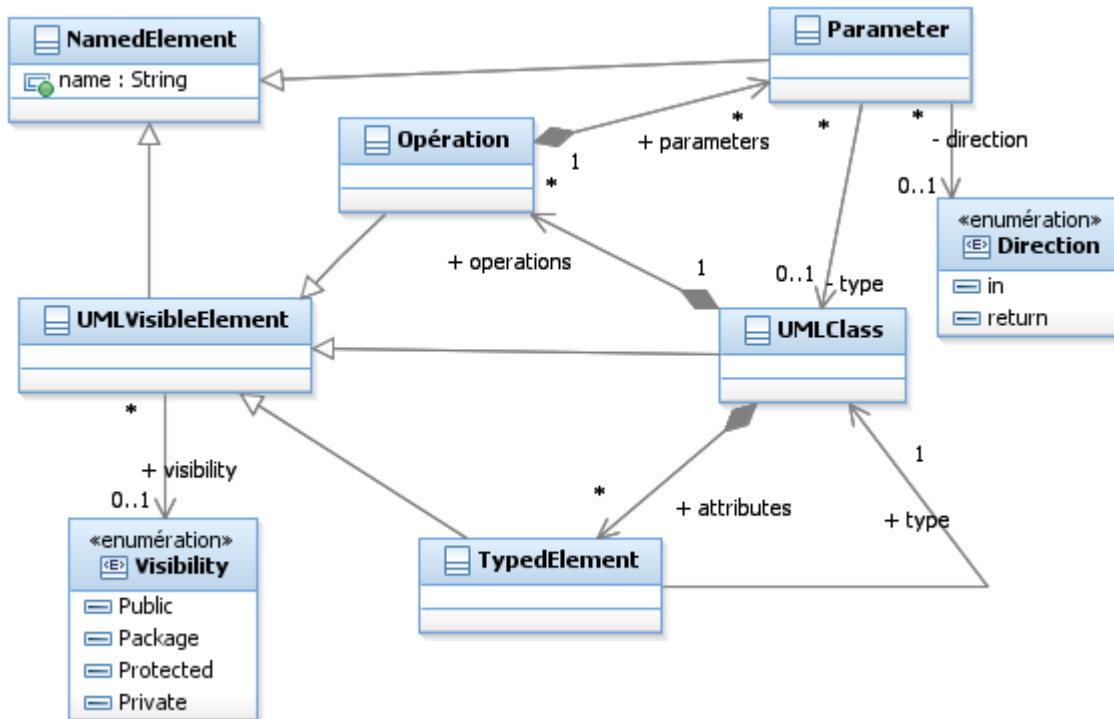
Examen Réparti 2eme partie

5 Janvier 2016 (2 heures avec documents : tous SAUF ANNALES CORRIGÉES).
Barème indicatif sur 22 points (donne le poids relatif des questions) (max 20/20).

Questions de cours

[3,5 Pts]

Répondez de façon précise et concise aux questions.
Soit le méta-modèle simplifié du diagramme de classe UML tel que présenté en TD.



Question Cours (QC):

a) (1,5 point) Représentez sous une forme textuelle arborescente (proche de xmi) le modèle instance suivant constitué de deux classes :



- b) Proposez une modification de ce métamodèle permettant de modéliser le concept de classe interne, définie au sein d’une autre classe.
- c) Proposez une façon de modéliser un package, c’est-à-dire un objet nommé (NamedElement) qui contient d’autres éléments nommés.

2. Problème: Conception Twitter [Barème sur 18,5 Pts]

Rappels d’analyse : Nous considérons l’application Twitter, permettant d’échanger des « Tweets » ou messages courts avec le monde entier. Par exemple, l’utilisateur @ytm peut envoyer le tweet « Love #UML and @GradyBooch » qui utilise le tag #UML et cite explicitement l’utilisateur « @GradyBooch ».

Tout utilisateur muni d’un compte est associé à un fil d’actualité, visible par tous, qui porte par ordre chronologique du plus récent au plus ancien tout tweet :

- qu'il a tweeté,
- qu'il a retweeté,
- qui le cite explicitement dans le corps du tweet,
- qu'un utilisateur auquel il est abonné à lui-même tweeté ou reweeté.

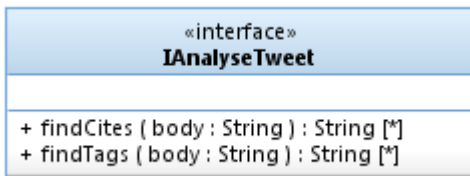
On peut également demander à afficher le « fil d'actualité » correspondant à un tag particulier, qui porte tous les tweets qui utilisent ce tag dans le corps du tweet.

Nous allons faire plusieurs hypothèses simplifiant le problème dans cet énoncé :

- Les tweets sont du texte pur, on ne peut pas attacher d'images ou fichier
- On ne peut pas rajouter du texte quand on « retweete » un message. Retweeter consiste simplement à ajouter le tweet en question à son propre fil d'actualité ainsi qu'à celui de tous ses abonnés potentiels.
- On ignore la fonctionnalité « aimer le tweet »
- On ne considère que le cas nominal, en particulier tous les noms de compte utilisés comme auteur ou cités existent par hypothèse
- Plus globalement, on ignore l'authentification et la création de compte (profil etc...) dans tout l'énoncé

I. Composant *CAAnalyseTweet*, analyse du contenu des Tweet (1,5 points)

Le composant bout de chaîne *CAAnalyseTweet* est responsable d'analyser les Tweet, pour déduire les personnes citées dans le tweet ainsi que les tags utilisés par un tweet. Les citations de personnes sont précédées d'une @ comme « @GradyBooch », les tags sont précédés d'un symbole # comme « #UML ». Ce composant offre *IAnalyseTweet* auquel on passe le corps du message à analyser et qui rend la liste des personnes citées ou des tags identifiés.



Question I : (1,5 pts)

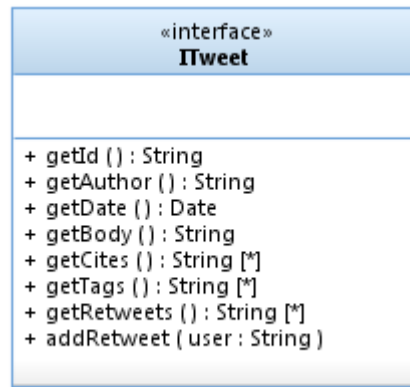
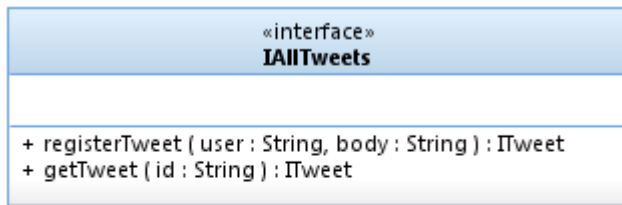
- Représentez sur un diagramme de composant le composant *CAAnalyseTweet* (interfaces offertes et requises).
- Proposez à l'aide d'un diagramme de classes une conception détaillée possible de ce composant.

II. Composant *CAAllTweets*, persistance et accès aux Tweets (6 points)

Ce composant de stockage est responsable d'assurer la persistance et l'accès au contenu des tweets. Pour des raisons d'efficacité et de temps d'accès il est typiquement répliqué en plusieurs instances dont l'état est maintenu synchronisé via des technologies du cloud qu'on n'abordera pas ici. On se contentera de supposer que pour les besoins de l'application c'est un composant singleton, dont l'unique instance est accessible de tout point de l'application.

Les tweets y sont stockés à l'aide de la méthode d'enregistrement, « registerTweet » à laquelle on passe le nom de l'auteur du tweet et le corps du texte tweeté ; la date du tweet est celle à laquelle il est soumis et les autres éléments sont déduits du tweet lui-même, à l'aide du composant *CAAnalyseTweet* décrit en Question I. L'on accède au contenu d'un tweet via un identifiant « id » unique.

Le tweet lui-même porte en lecture seule le nom de son auteur, le corps de texte, la liste des personnes citées dans ce tweet, la liste des tags présents dans le tweet, et la liste des noms des personnes ayant « retweeté » ce tweet. La seule opération permettant de modifier le tweet est « addRetweet » qui permet d'enregistrer qu'un retweet de ce message par « user » a été fait.



Question II.1: (2,5 pts)

- Représentez sur un diagramme de composant le composant CAllTweets (interfaces offertes et requises).
- Proposez à l'aide d'un diagramme de classes une conception détaillée possible de ce composant.

Question II.2: (3,5 pts)

On souhaite mettre en place des tests d'intégration de ce composant validant (entre autres) le scénario suivant : l'utilisateur @ytm tweete « Love #UML ».

- Représentez un ou plusieurs composants (testeur, bouchon(s)) permettant de mettre en place ces tests.
- Sur un diagramme de structure interne, représentez une configuration des composants pour le test.
- Sur un diagramme de séquence ou vous utiliserez une ligne de vie par instance identifiée en b), montrez les échanges nécessaires pour enregistrer le tweet donné en exemple puis s'assurer (résultat attendu) qu'il porte bien uniquement le tag #UML.

III. Composant CListeTweet (2 points)

Ce composant va permettre de matérialiser un ensemble de tweet correspondant soit à un fil d'actualité d'un utilisateur, soit à un mot clé ou tag comme #UML.



Il offre IListeTweet et doit donc permettre :

- D'ajouter de nouveaux tweets, à travers leur identifiant. Les tweets eux-même ne sont pas stockés par ce composant, ils restent la responsabilité de CAllTweets (cf. Question II).
- De lister les tweets, du plus récent au plus ancien. On passe pour cela un indice entre 0 et (size()-1). Par convention l'indice 0 correspond au plus ancien tweet de ce fil. Le détail des tweets est obtenu en s'appuyant sur le singleton CAllTweets.

Question III. : (2 pts)

- Représentez sur un diagramme de composant le composant CListeTweet (interfaces offertes et requises).
- Proposez à l'aide d'un diagramme de classes une conception détaillée possible de ce composant.

IV. Composant CAbonnements, gestion des abonnements (3,5 points)

Le composant bout de chaîne CAbonnements est responsable de savoir quels utilisateurs sont abonnés à un compte twitter donné. Un compte peut être abonné à un nombre arbitraire de comptes et réciproquement on peut avoir un nombre arbitraire d'abonnés.

On souhaite pouvoir :

- Abonner un utilisateur à un autre
- Désabonner un utilisateur d'un autre (sans effet si l'abonnement n'existe pas)
- Lister les noms des comptes des abonnés d'un compte donné
- Lister les noms des comptes auquel un utilisateur est abonné

Question IV. : (3,5 pts)

- Définissez une interface (avec signatures) supportant ces scenarios
- Sur un diagramme de séquence, modéliser l'interaction qui permet d'abonner l'utilisateur @bob à @alice. On suppose un composant CIHM représentant l'interface homme machine.
- Représentez sur un diagramme de composant le composant CAbonnements (interfaces offertes et requises).
- Proposez à l'aide d'un diagramme de classes une conception détaillée possible de ce composant.

V. Composant CTwitter, façade et contrôleur (5,5 points)

Le composant CTwitter va assembler l'ensemble des fonctionnalités. Il instancie et utilise les autres composants définis à ce stade.

Il offre ITwitter portant les opérations permettant d'obtenir les tweets constituant le fil d'actualité d'un compte, de trouver les tweets correspondant à un tag particulier, de tweeter un message (on fournit l'auteur et le texte du corps du tweet) et de retweeter un message (on fournit l'auteur du retweet et l'identifiant unique du tweet à retweeter).



Question V.1 : (4 pts)

On suppose que le composant CTwitter stocke et associe à chaque nom de compte et à chaque tag une occurrence de CListeTweet (listeAlice:CListeTweet, listeBob:CListeTweet...). On suppose aussi que le composant d'abonnement à enregistré que @bob est abonné à @alice (cf. question IV. b).

- Modéliser sur un diagramme de séquence où les lignes de vie représentent des instances des composants identifiés à ce stade ce qui se passe quand @alice tweete « Hello World ». On ne modélisera que les interactions réellement utiles ; les opérations testées en pratique mais qui rendent des listes vides dans cet exemple ne seront pas modélisées (e.g. il n'y a pas de tags ni de citation).
- Modéliser sur un nouveau diagramme de séquence le retweete par @charlie de ce même message. Charlie n'a pas d'abonnés mais il faut mettre à jour le tweet et l'ajouter à son propre fil d'actualité.
- Sur un nouveau diagramme de séquence, modéliser l'affichage du fil de @charlie par le composant CIHM. Il ne contient que le message d'Alice. On n'affichera que l'auteur et le corps du tweet.

Question V.2 : (1,5 pts)

Sur un diagramme de composant en déduire les interfaces requises et offertes de CTwitter et de CIHM.
