

BDD2

TP 1 : Modèle conceptuel de données

Pour réaliser vos TPs vous pouvez utiliser les logiciels suivants :

- StarUML (<https://staruml.io/download>)
- ArgoUML (<https://argouml-tigris-org.github.io/tigris/argouml/>)
- GenMyModel (<https://app.genmymodel.com/>)
- ... ou tout autre outil permettant de créer de diagramme UML (y compris Openoffice)

Le rendu sera effectué sous la forme d'une archive .zip ou .tar.gz. L'archive doit contenir :

- des captures d'écrans lisibles des modèles réalisés
- un fichier readme.txt décrivant votre avancement, les éventuelles précisions jugées nécessaires pour certaines décisions de modélisation.

Exercice 1/

Construisez la modélisation de manière itérative en traitant une à une les phrases suivantes. Il est possible de revenir sur les décisions de modélisation prises lors du passage d'une phrase à l'autre.

a) Des coureurs sont inscrits à des courses cyclistes.



Lecture

Coureur a ... (on liste des propriétés)

1 coureur peut s'inscrire à 1 ou plusieurs courses.

1 course peut accepter l'inscription de 1 ou plusieurs coureurs.

Validation

Est-ce que le texte dérivé du modèle correspond à ce qu'on demande dans a) ?

Oui -> Parfait

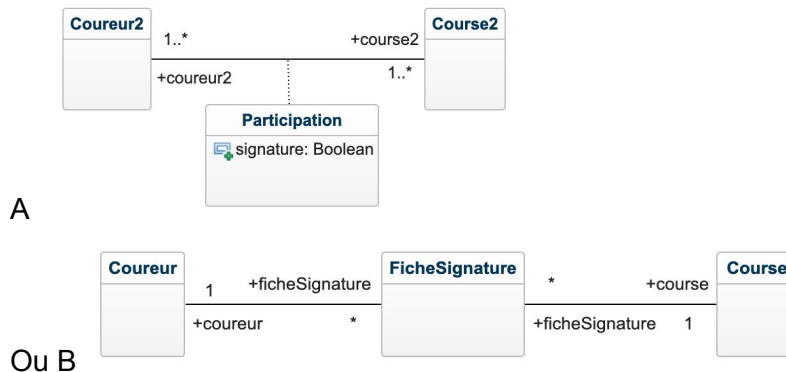
Non -> revoir...

b) La participation est actée par la signature d'une fiche de présence au départ.

Avant de procéder à la modélisation, il faut décider si la fiche de présence est spécifique à chaque Coureur, ou bien si elle est commune à tous les coureurs (un peu comme une fiche de DS). En ce qui suit, il y a plusieurs modélisations qui sont proposées.

Les deux premières A,B reflètent la situation où la fiche est individuelle.

La situation où la fiche est commune à tous les coureurs en étant unique pour la course est modélisée dans C.



Lecture A

Un coureur s'inscrit à une ou plusieurs courses. L'inscription comporte une signature pour valider la participation.

Une course peut accepter 1 ou plusieurs inscriptions de coureurs. Chaque inscription/participation est accompagnée par une signature dont la valeur devient true lorsque la course s'est déroulée.

Validation:

Est-ce que cela correspond au texte et à l'interprétation qu'on en fait (une fiche par coureur)?
Oui/Non

Lecture B

Coureur a ... props

Course a ... props

FicheSignature a ... props.

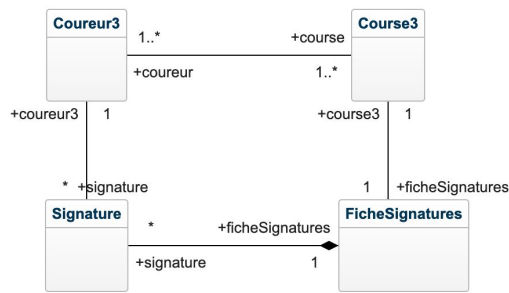
Un coureur signe 0 ou plusieurs Fiches de signature (car il peut participer à plusieurs courses).

Une fiche est signée par un seul coureur.

Une course comporte 0 ou plusieurs fiches de signatures. Une fiche concerne une unique Course.

Validation:

Est-ce que cela correspond au texte et à l'interprétation qu'on en fait (une fiche par coureur)?
Oui/Non



C

Lecture C

Coureur a ... props

Course a ... props

FicheSignatures a ... props

Signature a ... props.

Un coureur s'inscrit à une ou plusieurs courses. Remarque : cette relation peut être supprimée, car elle peut être dérivée de Course -> Fiche Signatures -> Signature -> Coureur

Une course comporte une et une seule fiche avec les signatures des coureurs. (pour le coup, je me dis qu'on devrait avoir 0..1 comme cardinalité, car avant que la course ne démarre pas forcément besoin d'avoir une fiche de signatures...)

Une fiche de signature est associée à une course. (A priori, on pourrait avoir une composition si on veut renforcer le fait que les fiches de signatures n'ont pas leur propre vie en dehors des courses);

Une Fiche de signatures comporte 0 ou plusieurs Signatures (individuelles)

Une Signature apparaît sur une unique fiche de signatures.

Une signature est réalisée par un coureur.

Un coureur peut réaliser une ou plusieurs signatures (car participant à plusieurs courses..)

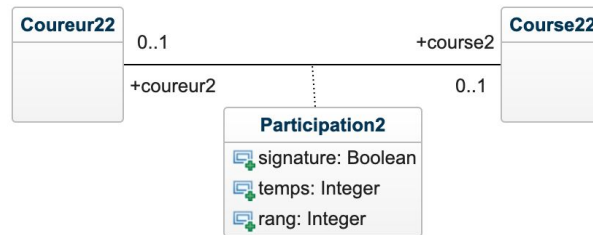
Validation:

Est-ce que cela correspond au texte et à l'interprétation qu'on en fait (une fiche par course)?

Oui/Non

La modélisation A est suffisante pour le moment. Les modélisations B et C qui permettent de disposer d'une entité à part entière pour la signature/participation deviennent intéressantes lorsque l'on veut caractériser davantage la participation en la mettant en lien avec d'autres entités, telles que les Bonus et/ou les Malus.

- c) Pour chaque course cycliste, on souhaite connaître le temps et le rang obtenus par chaque participant. Que pensez-vous du rang ? Doit-on le matérialiser ? Quel intérêt cela représente de le matérialiser ?



Commentaires :

Le rang peut se calculer à partir des temps de l'ensemble des coureurs (voir BDD1 syntaxe RANK).

Donc, quelque part, le rang associé à chaque participation peut être recalculé à tout moment.

En revanche, pour ce cas précis, il faut observer que :

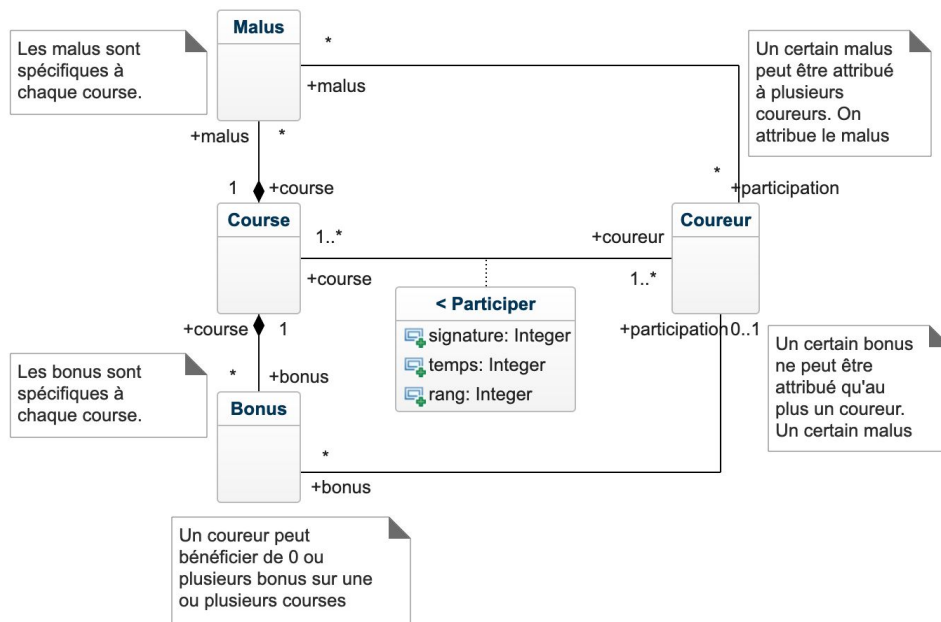
a) une fois la course finie, les rangs varient peu

b) une fois la course finie, on peut supposer que de nombreux accès aux résultats sont faits.

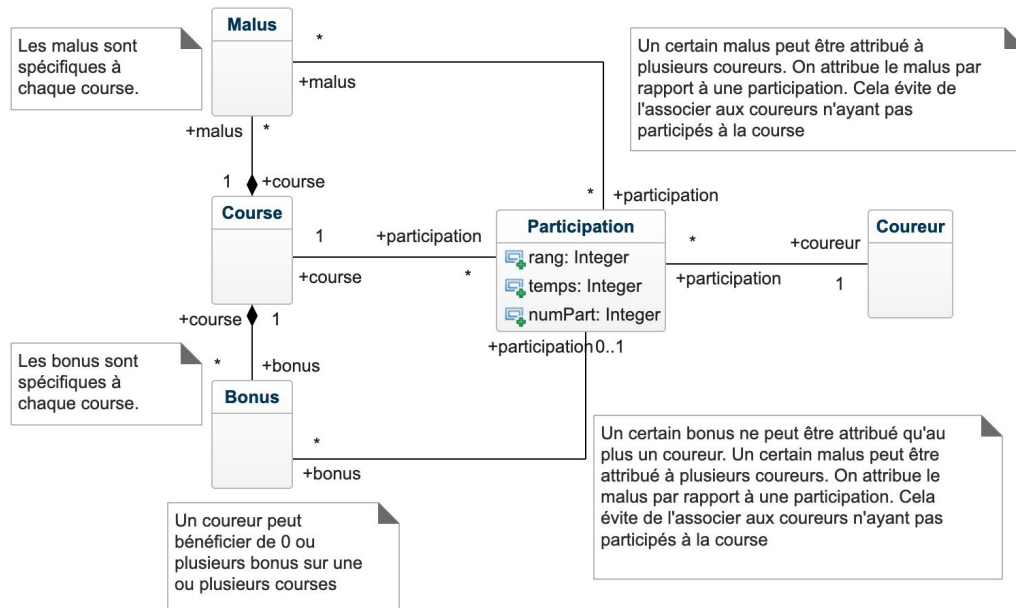
Dans ces conditions, il est intéressant de stocker le rang et ne pas le recalculer systématiquement.

- d) Certaines courses cyclistes accordent des bonus et des pénalités en termes de temps dont la nature et la valeur sont établies à l'avance et sont spécifiques à chaque course. Un bonus (par ex., premier arrivé) ne peut être accordé qu'une fois par course. Une même pénalité peut être attribuée à plusieurs coureurs.

Une première modélisation associe les bonus/malus directement aux coureurs, sans forcément avoir la garantie au niveau du modèle de leur participation à la course. La cardinalité côté Malus/Bonus dans la relation avec Course peut être * ou 1..* car pas explicitement précisé dans le texte. La cardinalité * est moins restrictive.



Une deuxième modélisation associe les bonus/malus directement à la participation des coureurs à la course. Ainsi, on peut associer des malus/bonus à travers leur participation aux coureurs inscrits dans une course.



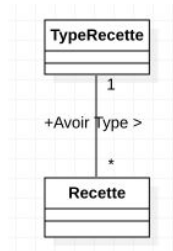
Exercice 2/

Construisez la modélisation de manière itérative en traitant une à une les phrases suivantes. Il est possible de revenir sur les décisions de modélisation prises lors du passage d'une phrase à l'autre.

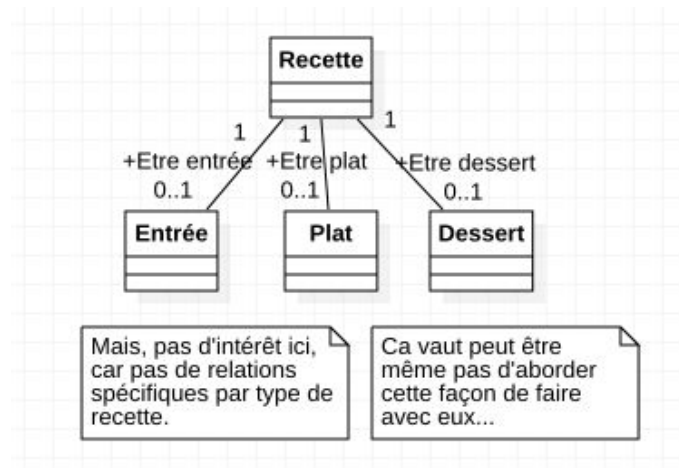
- a) On souhaite pouvoir gérer un catalogue de **recettes** de cuisine.

On n'inclut pas le catalogue comme entité car c'est le catalogue lui-même qui est modélisé. La base de données dans son ensemble correspond au catalogue.

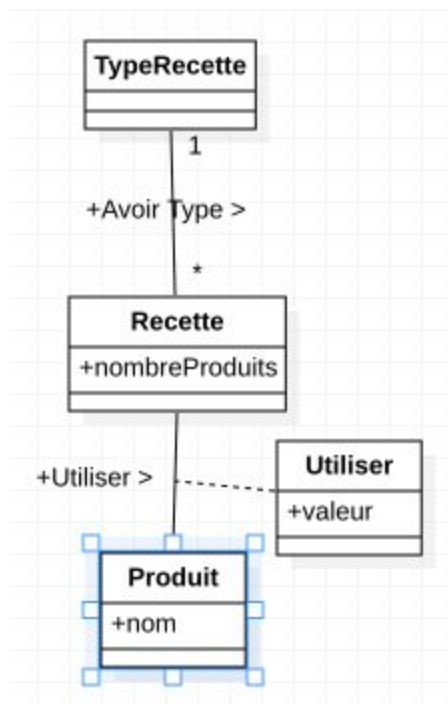
- b) Les **recettes** peuvent concerner les entrées, les plats, les desserts, les cocktails. On contraint les types possibles en associant l'entité *Recette* à une entité permettant de préciser les types de recettes connues dans le système.



Ou alors en plus compliqué, mais sans intérêt ici, car, pas de relation spécifique envisagée par type de plat dans la suite.

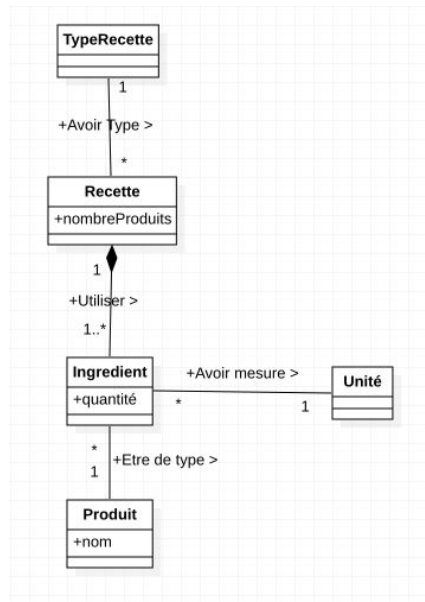


- c) Une **recette** indique les **produits** nécessaires, leur **nombre**, les **quantités**, ainsi que le **temps** approximatif pour la réalisation.

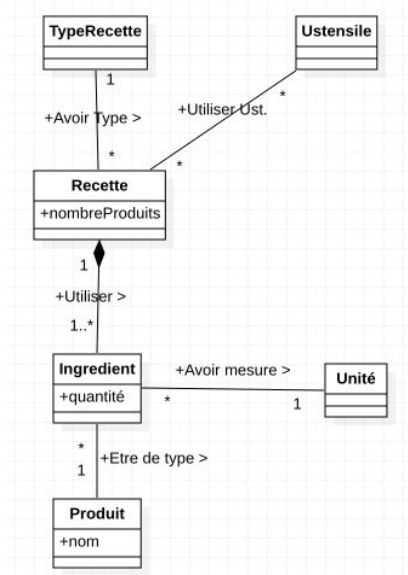


- d) Les quantités peuvent être exprimées en grammes, kilogrammes, litres, millilitres, cuillères à café, cuillères à soupe, etc.

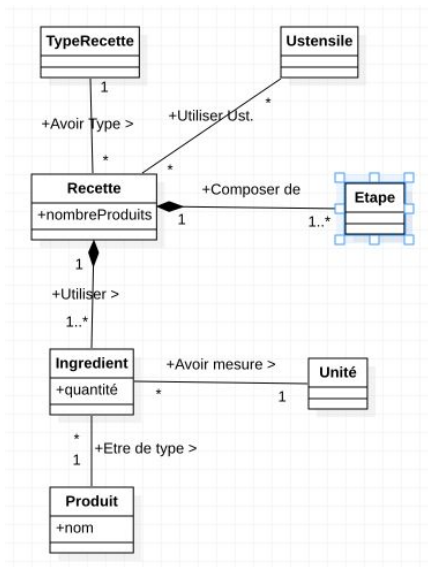
Comme on veut contraindre les unités de mesure de la quantité (valeur) on privilégiera une entité spécifique (ingrédient). Un ingrédient est donc une instance spécifique à une recette. (50 gr de beurre) est l'ingrédient I1 à mettre dans la réalisation d'une recette de Crêpe (R1). (50 gr de beurre) est l'ingrédient I2 à mettre dans la réalisation d'une recette de Cake (R2), $I1 \neq I2$. I1 et I2 sont tous deux reliés au produit Beurre (B1). Il est préférable de garder $I1 \neq I2$, car si on décide d'alléger les Crêpes (R1) en mettant moins de beurre, cela ne doit pas influencer sur le Cake (R2).



- e) Une recette précise les ustensiles nécessaires à sa réalisation.



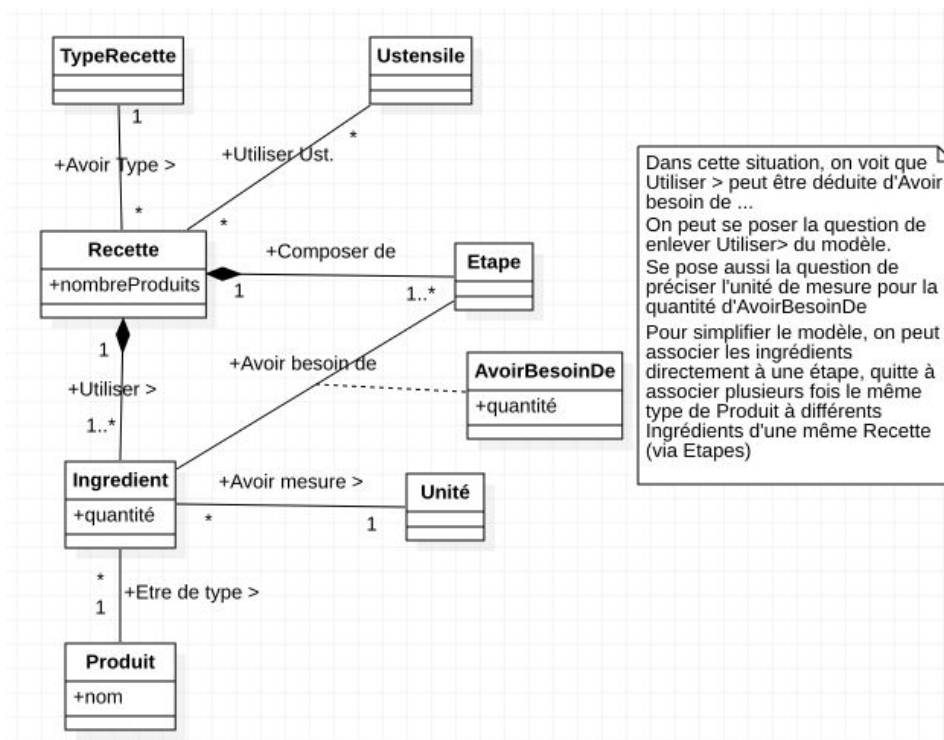
f) Une **recette** est composée de plusieurs **étapes** de préparation.



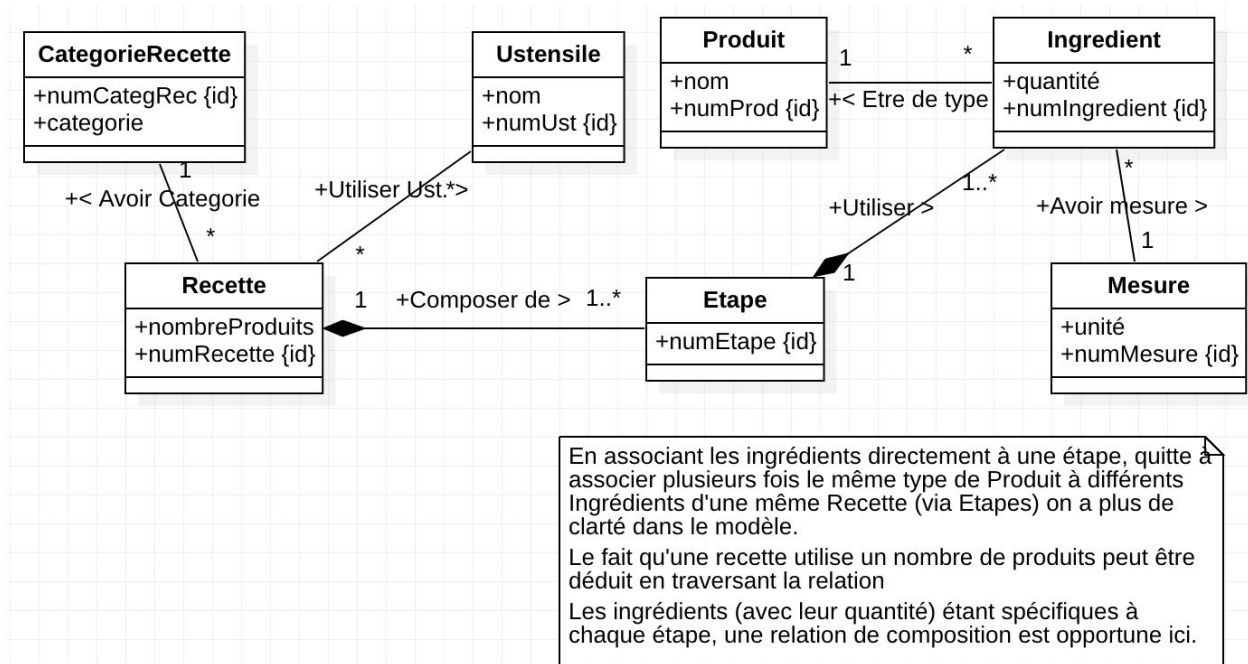
g) Chaque **étape précise** les ingrédients (et la quantité) nécessaires lors de cette étape.

Certains ingrédients peuvent être utilisés dans différentes étapes de la recette.

Ci-dessous une première solution, mais on voit de nombreuses redondances en termes de relations. On perd également l'unité de mesure pour les quantités associées aux différentes étapes.



Une version plus allégée sans redondance en termes de relations. On peut déduire les ingrédients d'une recette en parcourant les ingrédients associés aux différentes étapes.



h) Chaque **étape précise** les **ustensiles** nécessaires et le **temps** approximatif de réalisation.

On se retrouve exactement dans le même cas de figure que g)

Peut-on s'assurer au niveau du MCD que les quantités utilisées dans les étapes sont cohérentes avec la quantité globale renseignée au niveau de la recette ? Que faire ?

Avez-vous identifié d'autres propriétés qui peuvent se retrouver avec des valeurs incohérentes dans le modèle ?

Ici l'idée est de réduire les redondances en supprimant les relations et/ou propriétés qui peuvent être facilement ré-calculables.