

LOG2410 - Conception logicielle

Examen final - Hiver 2016

Documentation : aucune documentation permise.

Calculatrice non-programmable : non permise.

Date : 20 avril 2016.

Cet examen comprend 3 questions sur 3 pages.

Mise en contexte. Le texte qui suit décrit le contexte et les principales fonctionnalités du logiciel *PolyInfusion*.

Depuis quelques années, le thé gagne constamment en popularité et tend maintenant à entrer en compétition avec le café, le vin ou la bière parmi les boissons recherchées par des passionnés toujours plus nombreux. Ces amateurs sont aujourd'hui prêts à dépenser de fortes sommes pour acquérir des variétés particulières de thés et tout l'équipement nécessaire pour infuser chaque thé en tenant compte de ses caractéristiques propres. Afin de profiter de cet engouement pour le thé sous toutes ses formes, on vous mandate pour participer au développement d'un nouveau type d'appareil inspiré des cafetières expresso, mais spécifiquement destiné à infuser des thés. Ce nouveau type d'appareil comprendra entre autres un réservoir pour contenir de l'eau, une bouilloire interne et plusieurs réceptacles contenant chacun les feuilles d'une variété de thé. Lors du remplissage de chaque réceptacle de thé, l'utilisateur pourra faire lire par la machine un code barre permettant d'identifier le type de thé, ce qui permettra à la machine d'ajuster automatiquement les paramètres d'infusion. La machine sera contrôlée à l'aide d'une interface affichée sur un petit écran tactile à l'avant de l'appareil. Le fabricant prévoit mettre sur le marché plusieurs modèles de machines ayant différents prix et différentes capacités de stockage, mais dont les fonctionnalités de base seront identiques.

Vous êtes responsable de la conception préliminaire du logiciel de contrôle de ces machines à infuser. Le logiciel dédié sera installé directement sur la machine, et baptisé *PolyInfusion*. Ce logiciel sera développé de façon à utiliser les dernières technologies de l'internet des objets (Internet of Things ou IoT). Lors d'une analyse préliminaire des besoins, les fonctionnalités suivantes ont été identifiées :

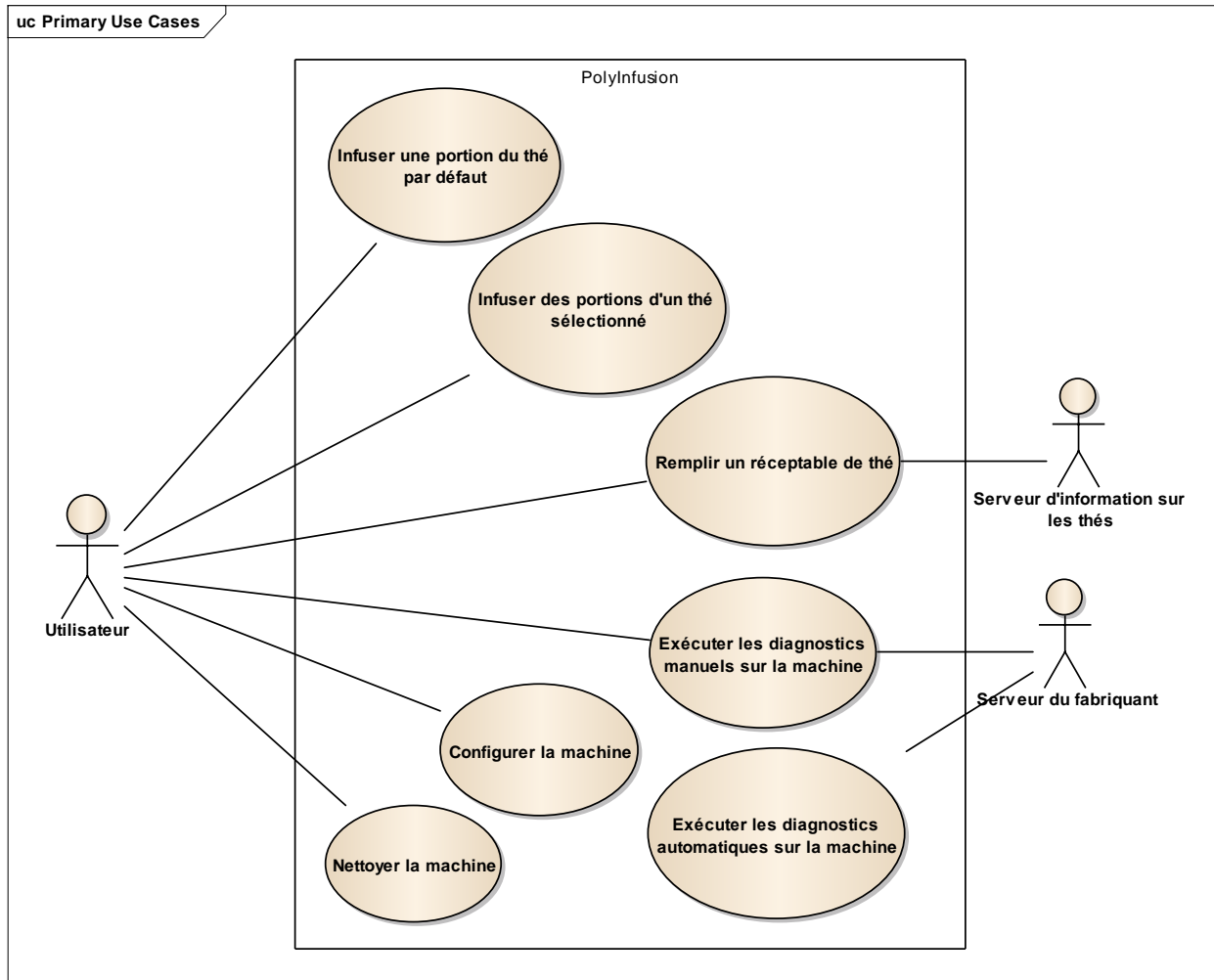
1. Lorsqu'un utilisateur touche à l'écran tactile, la machine à infuser passe du mode de veille au mode actif, dans lequel la machine est prête à accepter des commandes.
2. Si aucune commande n'est entrée à l'écran dans un délai spécifié dans les paramètres de configuration de la machine (30 minutes par défaut), la machine passe en mode veille.
3. En mode actif, la machine présente les options suivantes à l'écran :
 - a. Bouton « Infusion rapide » permettant d'infuser une seule portion du thé choisi par défaut.
 - b. Bouton « Sélection thé » permettant de sélectionner un thé parmi les thés disponibles dans la machine et d'infuser une ou plusieurs portions du thé sélectionné. S'il le souhaite, l'utilisateur pourra choisir le nombre de portions de thé à infuser. Par défaut, la machine infusera une seule portion.
 - c. Bouton « Remplissage » permettant d'ouvrir l'un des réceptacles à thé afin de le remplir. Lors du remplissage, l'utilisateur peut scanner un code barre qui sera lu par la machine afin que celle-ci puisse identifier le type de thé contenu dans le réceptacle. Afin de connaître les paramètres spécifiques d'infusion du thé (température de l'eau, durée d'infusion, etc.) la machine vérifiera dans une base de données locale, stockée sur la machine afin de déterminer si ce type de thé lui est connu. Dans le cas contraire, si la machine est branchée à internet, elle communiquera automatiquement avec un serveur distant afin de récupérer les paramètres d'infusion et conservera les paramètres récupérés dans sa base de données.
 - d. Bouton « Configuration » donnant accès à un menu de configuration de la machine. Le menu de configuration, dont l'accès sera protégé par un code de sécurité à 6 chiffres,

permettra à l'utilisateur de choisir les différents paramètres par défaut de la machine, dont : l'horaire de démarrage et de mise en veille automatique, le délai d'attente avant de passer au mode veille, le type de thé infusé par défaut, le nombre de portions infusées par défaut, la température de préchauffage de l'eau, etc. Ce menu permettra également de configurer certains aspects du logiciel, dont : la langue du menu, le code de sécurité, les paramètres de connexion à internet, etc.

- e. Bouton « Nettoyage ». Lorsque ce bouton est sélectionné, et après confirmation par l'utilisateur, la machine exécute une séquence de fonctions de nettoyage.
 - f. Bouton « Diagnostic ». Lorsque ce bouton est sélectionné, et après confirmation par l'utilisateur, la machine exécute une séquence de tests afin de vérifier le bon état de fonctionnement de la machine. Les tests incluent la vérification des différents senseurs (eau, température, thé), la vérification du système de chauffage, la vérification des numéros de version du logiciel, la vérification de la connexion réseau, etc. Une fois les tests effectués, un résumé des tests est affiché à l'écran. En cas de détection d'un problème, et après confirmation par l'utilisateur, la machine pourra soumettre par internet un rapport des tests au fabricant de la machine, qui pourra communiquer avec l'utilisateur si une réparation est nécessaire. Si une nouvelles version du logiciel est disponible, et après confirmation par l'utilisateur, la machine pourra télécharger le nouveau logiciel et procéder à son installation.
 - g. Bouton « Arrêt » qui permet d'éteindre complètement la machine.
- 4. Pendant que la machine est en train d'infuser et de servir un thé, l'écran affiche une seule option « Annuler » qui permette d'interrompre immédiatement l'opération. Lorsque l'utilisateur appuie sur le bouton « Annuler », la machine suspend l'opération en cours et permet à l'utilisateur de confirmer l'annulation ou de reprendre l'opération interrompue.
 - 5. Selon l'horaire de démarrage et de mise en veille programmé par l'utilisateur, la machine pourra passer du mode actif au mode veille et vice-versa de façon automatique, afin, par exemple, de s'assurer que la machine soit prête à être utilisée le matin au réveil.
 - 6. Une application séparée sera développée sous iOS et Androïd afin de permettre au propriétaire de contrôler la machine à infuser à distance.
 - 7. De façon automatique, grâce à ses différents senseurs, la machine effectuera une surveillance de son état afin de s'assurer qu'elle est bien approvisionnée en eau et en thé et qu'elle est prête à fonctionner. En cas de problème, la machine affichera un message à l'écran et, si le propriétaire à configurer le service sur le serveur du fabricant, transmettra l'information des tests au serveur du fabricant. L'utilisateur pourra alors recevoir une alerte sur son téléphone portable.

Question 1 - Processus de conception et cas d'utilisation (16 points)

- a) En considérant la description du système à concevoir, tracez le diagramme UML de haut niveau (diagramme de contexte) des cas d'utilisation du système *PolyInfusion* (5 points).



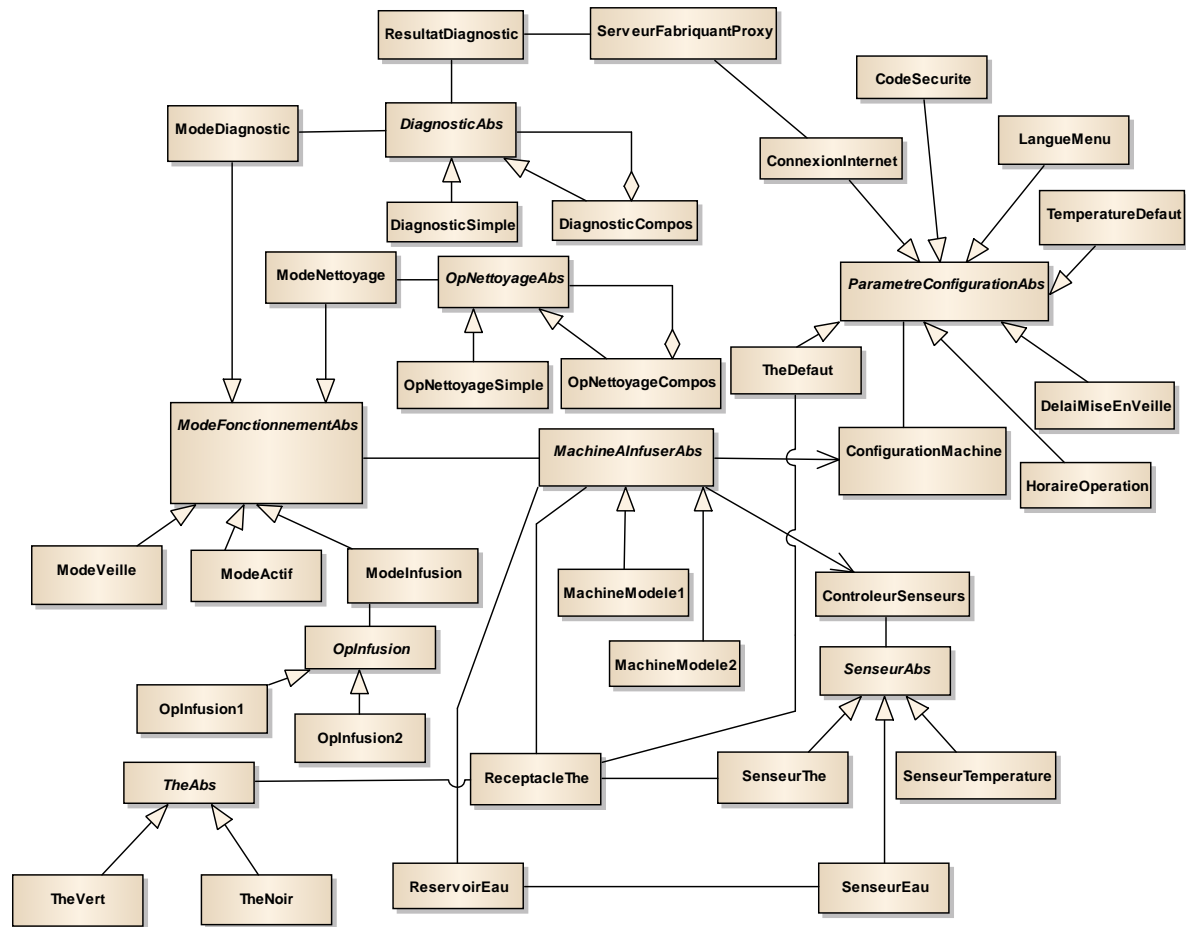
- b) Parmi les cas d'utilisation que vous avez modélisés en a), identifiez explicitement celui qui couvre la fonctionnalité de *vérification de la version du logiciel de contrôle de la machine* (1 points)

La fonctionnalité de *vérification de la version du logiciel de contrôle de la machine* fait partie du cas d'utilisation « Exécuter les diagnostics manuels sur la machine ».

- c) Spécifiez le cas d'utilisation identifié en b) dans sa version étendue, en vous limitant au scénario principal (4 points).

Cas d'utilisation	Exécuter les diagnostics manuels sur la machine
Système	PolyInfusion
Niveau	Objectif usager
Acteur primaire	Utilisateur
Parties prenantes	Utilisateur, fabricant, responsable de l'entretien
Préconditions	La machine est allumée, en mode actif et en attente
Garanties de succès	Les tests sont complétés, le logiciel est à jour, le résumé des résultats de test est disponible.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur choisit d'exécuter les diagnostics sur la machine 2. La machine demande à l'utilisateur de confirmer en entrant son code de sécurité 3. L'utilisateur entre son code. 4. La machine valide le code de sécurité. Si le code est correct, la machine demande à l'utilisateur de presser sur OK pour lancer les diagnostics 5. L'utilisateur presse OK. 6. La machine vérifie les différents systèmes. La machine vérifie si une connexion réseau est disponible. Si une connexion réseau est disponible, la machine vérifie si une nouvelle version du logiciel est disponible. Si une nouvelle version est disponible, la machine demande à l'utilisateur de confirmer le téléchargement et l'installation de la nouvelle version du logiciel. 7. L'utilisateur confirme l'installation du logiciel. 8. La machine affiche un résumé des tests effectués et lance l'installation du logiciel. La machine demande à l'usager de confirmer la transmission du résultat des tests vers le serveur du fabricant. 9. L'usager confirme la transmission des résultats.

- d) Proposez un diagramme de concepts pour le logiciel *PolyInfusion* (6 points).



Question 2 - Architecture logique (7 points)

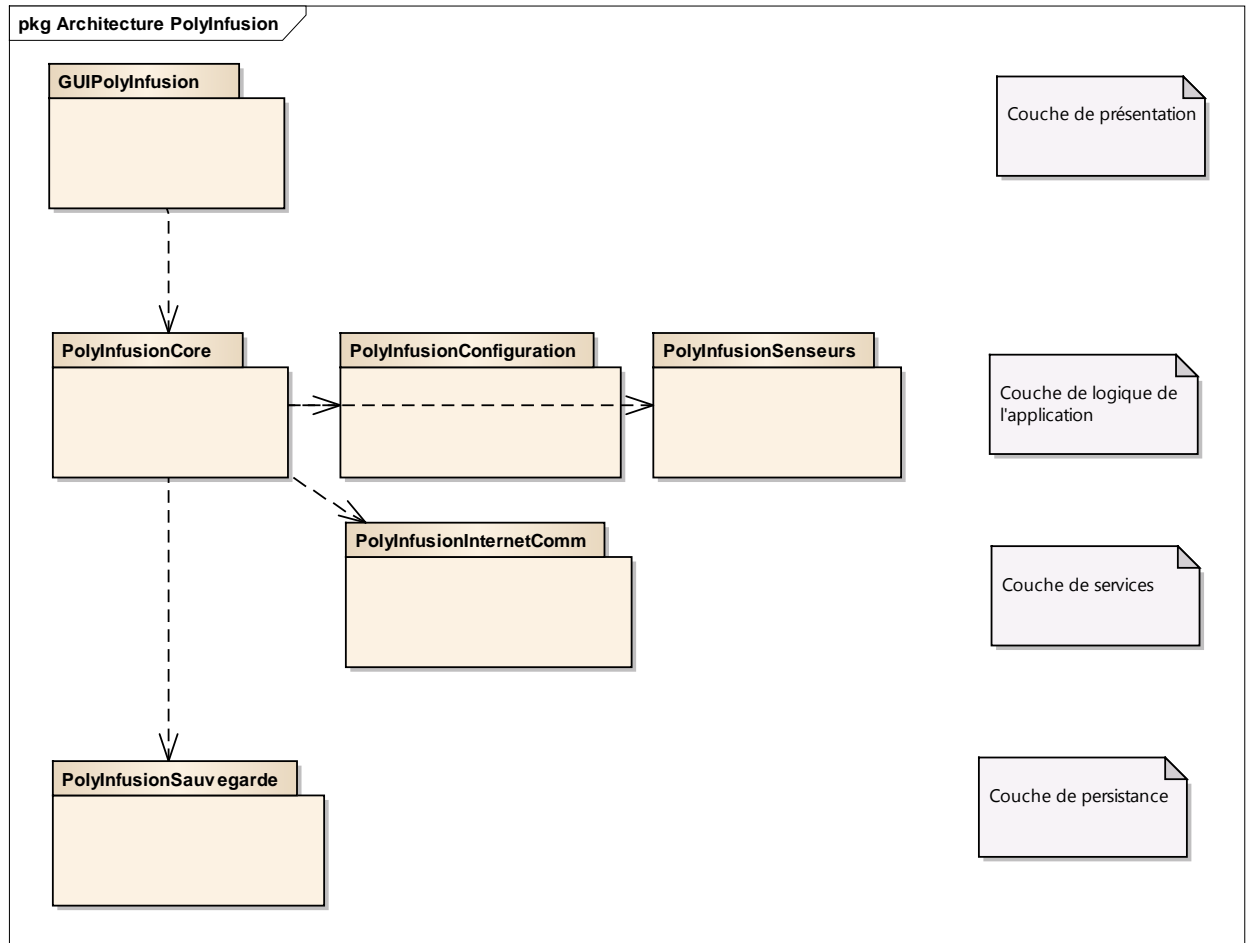
Les logiciels modernes sont développés selon des architectures logiques comportant plusieurs couches ou niveaux (*multi-tier architecture*).

- a) Indiquez deux avantages importants de la décomposition d'un logiciel en une architecture multi-niveaux (2 points),

Au moins deux parmi :

- La séparation de la logique d'application dans des composants séparées qui peuvent être réutilisé dans d'autres systèmes,
- La possibilité de répartir les niveaux sur différents nœuds de calcul, et dans différents processus,
- L'assignation de développeurs à la construction de chaque niveau: parallélisations des efforts et spécialisation des intervenants.

- b) Tracez, sous la forme d'un diagramme de paquetage, une première ébauche d'un diagramme d'architecture pour le système *PolyInfusion*, (3 points).



- c) En vous basant sur le diagramme tracé en b), expliquez la différence de modélisation entre la base de données locale et la base de données distante utilisées lors de la fonction de « remplissage » de la machine. Ces deux bases de données appartiennent-elles au même niveau architectural ? Justifiez votre réponse (2 points).

La base de données locale est modélisée comme un paquetage de la couche de persistance, donc au niveau le plus bas de l'architecture, alors que la base de données distante est intégrée dans un paquetage de communication dans la couche de service. Les deux bases de données ne sont donc pas au même niveau dans le modèle architectural.

Question 3 - Patrons de conception (22 points)

Durant la phase d'analyse et conception du logiciel *PolyInfusion*, voici quatre problèmes qui ont été identifiés et que l'on vous demande de résoudre en utilisant les patrons de conception :

- L'accès aux fonctionnalités de la machine dépend de l'état dans laquelle la machine se trouve. Par exemple, lorsque la machine est en mode veille, la seule fonctionnalité disponible est de mettre la machine en mode actif ; lorsque la machine est en train d'infuser un thé, la seule fonctionnalité disponible est d'interrompre l'opération, etc. On veut s'assurer en tout temps que les fonctions disponibles soient cohérentes entre elles et avec le mode dans lequel se trouve la machine.
- Pendant que la machine est en train d'infuser un thé, il est possible que le réservoir d'eau ou de thé choisi se vide complètement, ce qui empêcherait la machine de compléter l'opération. Dans ce cas, la machine doit afficher sur son écran tactile un message explicite indiquant que la machine manque d'eau ou de thé.
- Les opérations à effectuer pour infuser un thé varient en fonction du type de thé à infuser et des préférences personnelles de l'utilisateur. Certains thés doivent être humidifiés avant d'être infusés, d'autres doivent reposer avant d'être versés, le débit et la température de

l'eau peuvent également varier. On veut rendre interchangeables les différentes séquences d'opérations qui seront effectuées pour infuser un thé.

4. Lorsque l'on effectue un diagnostic de la machine, celle-ci doit chaque fois effectuer une série de tests, mais ces tests dépendent du modèle précis de la machine. On veut s'assurer que tous les tests soient effectués, tout en permettant de configurer chaque test selon les caractéristiques spécifiques de chaque modèle de machine.

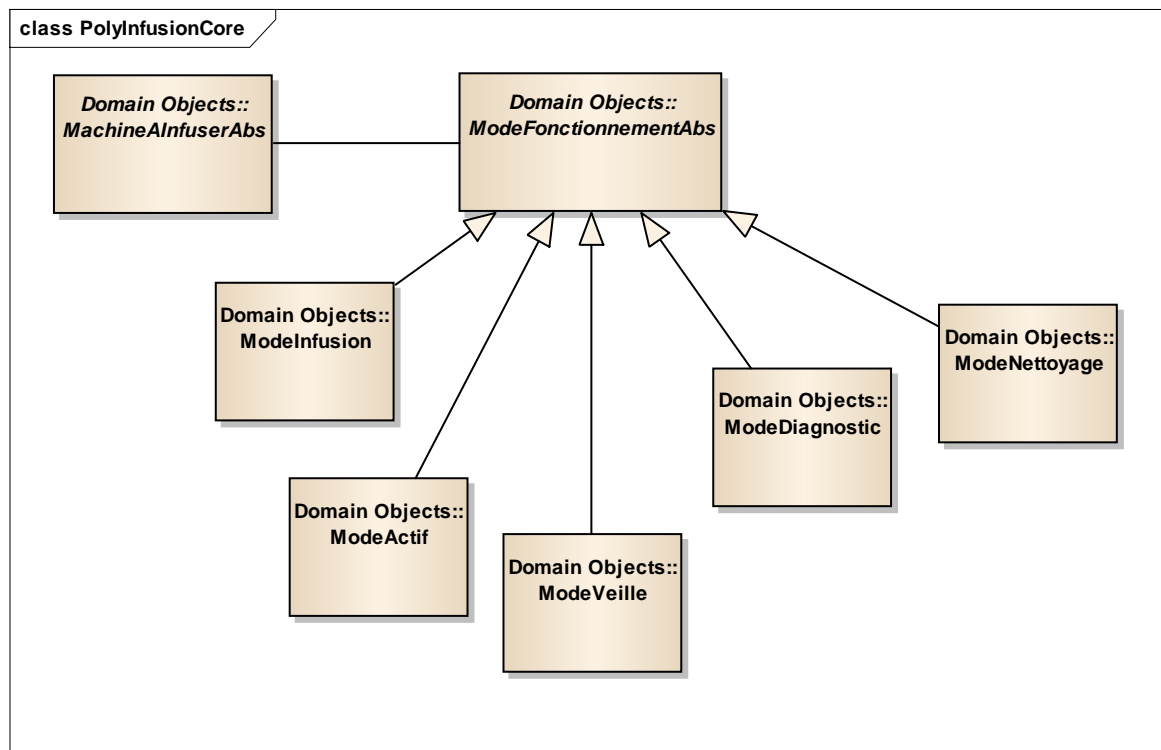
Le premier problème peut être résolu en appliquant le patron **State**. Le second problème peut être résolu en appliquant le patron **Observateur**. Le troisième problème peut être résolu en appliquant le patron **Stratégie**. Le quatrième problème peut être résolu en appliquant le patron **Template Method**.

Pour chacun des quatre problèmes de conception, on vous demande de répondre aux trois questions suivantes, en traitant un problème à la fois :

- a) Expliquer en vos propres mots l'intention du patron suggéré en expliquant pourquoi le patron désigné est applicable au problème posé (1 point pour chaque problème).
- b) Appliquer le patron suggéré en fournissant un diagramme de classes annoté, où les rôles identifiés dans le patron sont explicitement associés aux classes appropriées du système PolyInfusion (3 points pour chaque problème).
- c) Identifier, dans le contexte spécifique du système *PolyInfusion*, un avantage ou un inconvénient lié à l'utilisation du patron suggéré pour résoudre le problème de conception (1 point pour chaque problème).

Problème 1 – Patron State

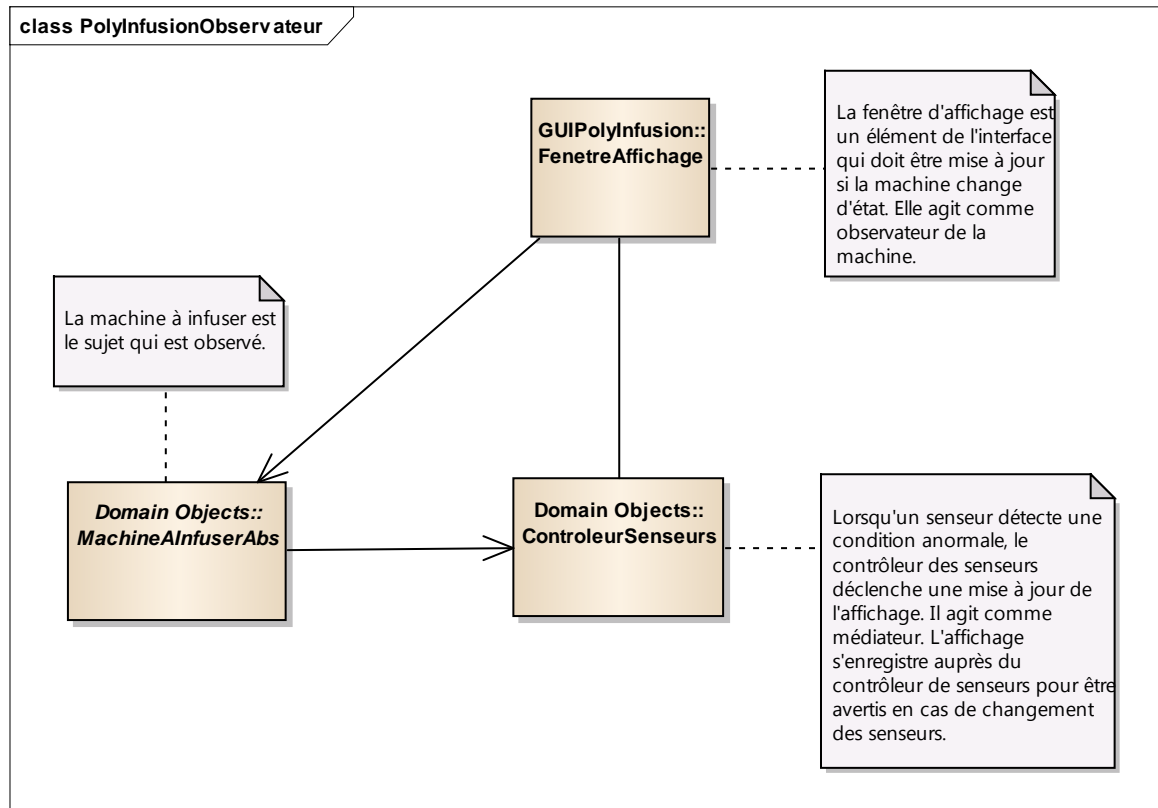
- a) L'intention du patron State est de permettre à un objet de changer son comportement en fonction de son état. L'objet se comporte comme s'il avait changé de classe. Ce patron permet de conserver la cohérence des fonctionnalités d'un objet lors d'un changement d'état. Quand la machine à infuser change de mode, ses réponses à des requêtes doivent toutes changer en bloc. Le patron State permet de modéliser le changement de mode de la machine.



- b)
- c) Un avantage du patron State est d'éviter les énoncés switch-case et d'assurer la cohérence des états. Il est facile d'ajouter un nouvel état au besoin.

Problème 2 – Patron Observateur

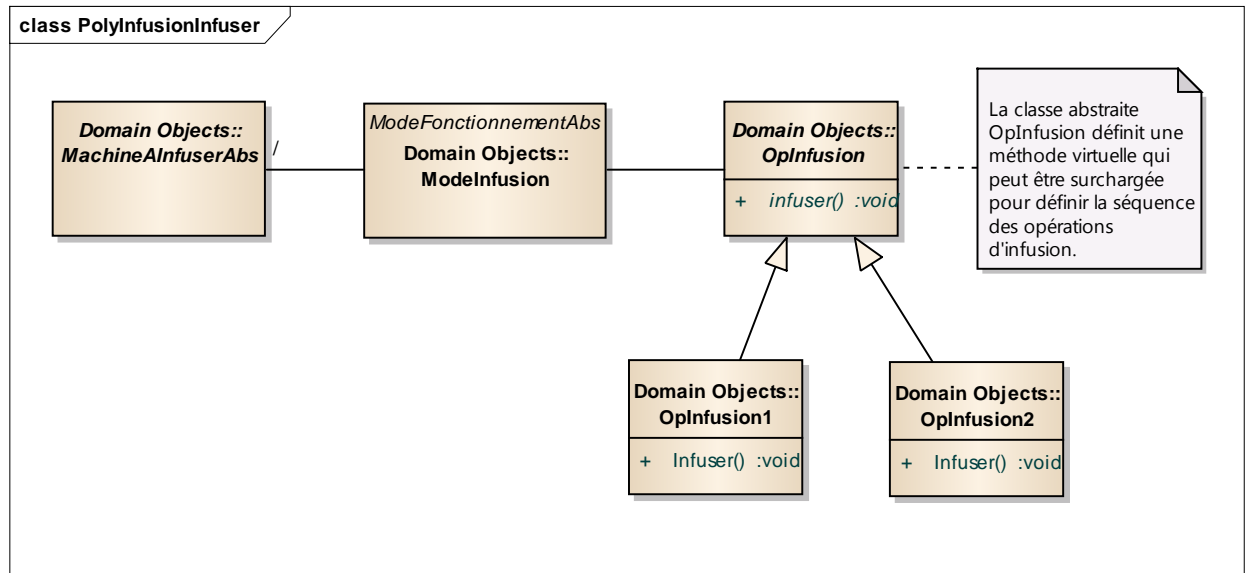
- a) Intention : Définit une relation un à plusieurs entre des objets de façon à ce que lorsqu'un objet change d'état, que tous ses dépendants soient avertis et mis à jour automatiquement. Dans ce cas-ci, lorsqu'un senseur détecte une condition particulière, comme le manque d'eau ou de thé, l'opération en cours doit être interrompue et l'affichage sur l'écran doit être mis à jour.



- b)
c) Un avantage important du patron Observateur est de faire en sorte que l'observateur n'est pas connu directement du sujet, mais sera mis à jour si le sujet change.

Problème 3 – Patron Stratégie

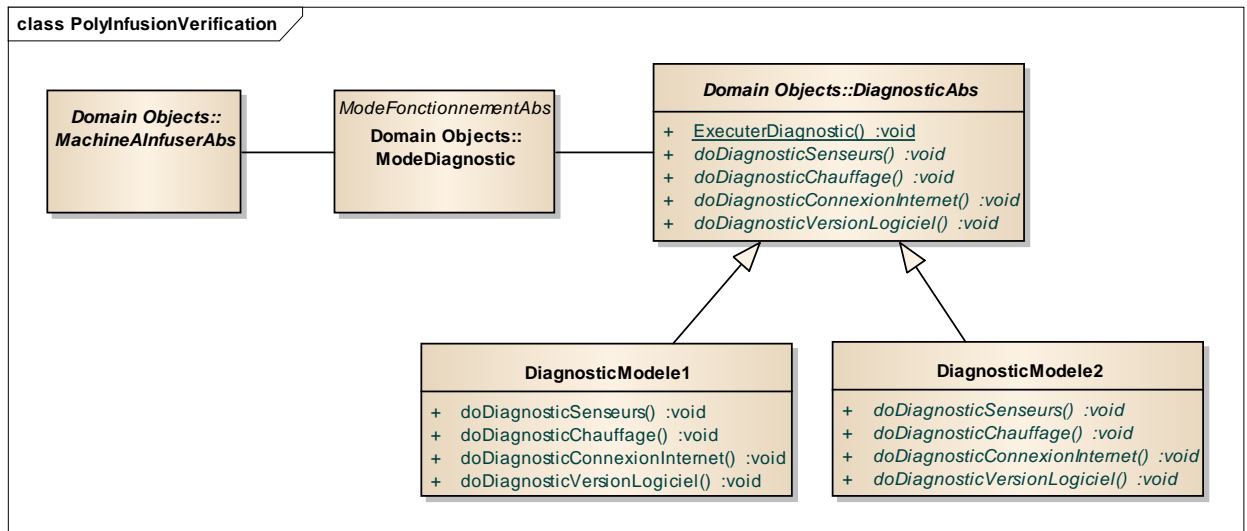
- a) Intention : encapsuler un algorithme dans une classe de façon à le rendre interchangeable. Le patron Stratégie permet de faire varier l'algorithme indépendamment du client qui l'utilise. Dans le cas de l'infusion des différents thés, la séquence des opérations peut varier en fonction du type de thé. En utilisant le patron stratégie, la machine à infuser peut être configurée pour changer d'algorithme d'infusion selon le type de thé en cours d'infusion.



- b)
c) Un grand avantage du patron Stratégie est qu'il est très facile d'ajouter de nouveaux algorithmes dérivés de la classe abstraite. Dans ce cas-ci, s'il y a une nouvelle séquence d'opération pour infuser une sorte particulière de thé, il suffit d'ajouter une nouvelle classe dérivée de OpInfusion.

Problème 4 – Template Method

- a) Intention : Le patron Template Method définit le squelette d'un algorithme, et laisse les sous-classes définir certaines étapes. Dans le cas de la vérification de la machine, on veut que toutes les étapes de vérification soient faites, mais chaque machine peut définir les détails spécifiques d'une étape de vérification.



- b)
c) Un avantage du patron Template Method est qu'il garantit que la séquence des étapes sera respectée, par contre, il force à créer une nouvelle sous-classe pour spécialiser le comportement.

Complément à la question sur les patrons de conception, on vous demande :

- d) D'expliquer en vos propres mots la différence entre le patron **State** et le patron **Stratégie** en expliquant les raisons pour lesquelles vous choisiriez l'un ou l'autre (2 points).
Le patron stratégie permet au contexte de configurer une opération ou un algorithme à la fois, tandis que le patron state sert à configurer toutes les opérations d'une classe simultanément.