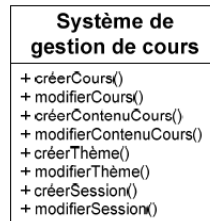


Travaux Dirigés n°3

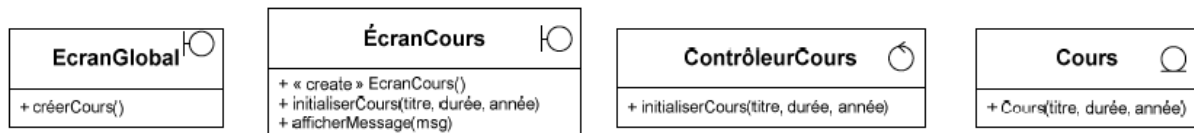
Diagrammes de séquences, Diagrammes de composants/déploiements

Exercice 1 : distribuer le comportement sur les classes participantes

Une phase d'analyse d'un système de gestion de cours a permis d'identifier les opérations suivantes :

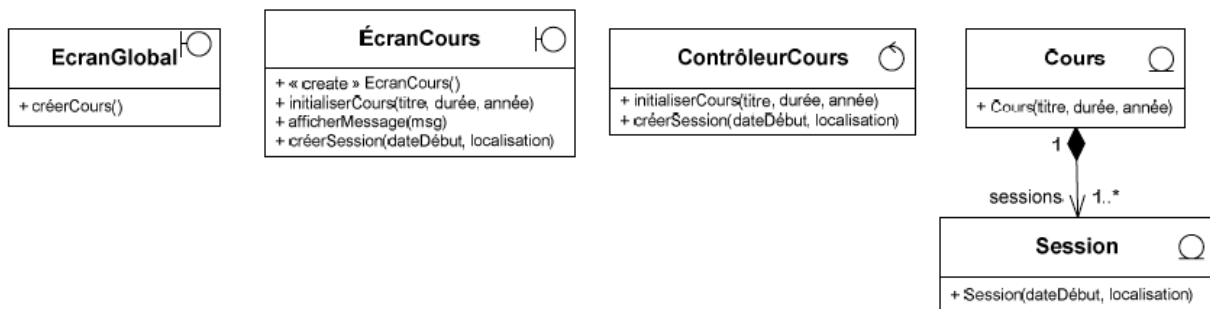


Les classes participantes de ce système sont :



- ✓ Représenter le diagramme de séquence de conception pour l'appel de l'opération « créerCours() » par un gestionnaire.

Le demandeur doit être informé de la bonne création du cours. Le modèle des classes participantes est modifié afin de permettre la prise en compte des sessions de cours :



- ✓ Représenter un diagramme de séquence de conception pour l'appel des opérations « créerCours() » et « créerSession() » par un gestionnaire.

Exercice 2 : Analyse de comportement, identification de composants

L'accès au système de gestion de cours nécessite l'authentification préalable du système d'authentification au moyen de la combinaison d'un login et d'un mot de passe.

Le système de gestion de droits fournit les droits qui conditionnent l'accès au bouton de création d'un cours. Sa sélection lance le formulaire de description d'un nouveau cours.

- ✓ Représenter le nouveau diagramme de séquence d'appel de l'opération « créerCours() » par un gestionnaire.

- ✓ Représenter le diagramme de composants modélisant le couplage entre le système de gestion de cours et les systèmes externes en illustrant les interfaces sous la forme d'icônes.
- ✓ Représenter le diagramme de composants modélisant le couplage entre le système de gestion de cours et les systèmes externes en illustrant les interfaces sous la forme de classe stéréotypée.

Exercice 3 : diagramme de déploiement

Le sous-système d'authentification du système gestion de configuration regroupe les classes d'écran d'authentification, d'informations des employés et de contrôle des identifiants.

Les modules de ce sous-système sont un fichier .sql qui permet de stocker les informations des employés, un fichier .jsp qui affiche un écran d'authentification et une classe .java de contrôle de l'authentification.

La base de données des employés est hébergée sur un serveur Windows de mémoire vive 4Go. L'application Web est-elle hébergée sur un serveur Tomcat qui communique avec le premier au moyen d'une liaison Ethernet 100b.

Représenter le diagramme de déploiement correspondant.

Diagrammes d'états-transitions, Diagrammes d'activités

Exercice 4 : diagramme d'états-transitions

Considérons une classe « Partie » dont la responsabilité est de gérer le déroulement d'une partie de jeu d'échecs. Cette classe peut être dans deux états :

- ✓ Gestion du tour des blancs
- ✓ Gestion du tour des noirs.

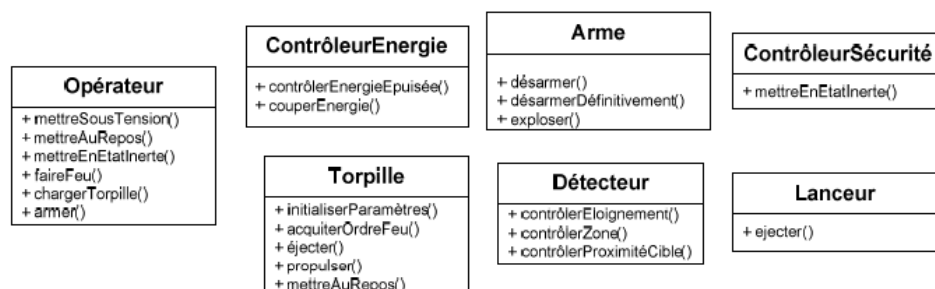
Les événements à prendre en considération sont

- ✓ un déplacement de pièces de la part du joueur noir ;
- ✓ un déplacement de pièces de la part du joueur blanc ;
- ✓ la demande de prise en compte d'un échec et mat par un joueur ;
- ✓ un échec et mat assure la victoire du dernier joueur. Dans ce cas, une activité « noirsGagnants » ou « blancsGagnants » est déclenchée ;
- ✓ la demande de prise en compte d'un pat qui mène à une fin de partie, avec une égalité. Dans ce cas, une activité « égalité » est déclenchée.

Représenter le diagramme d'états/transitions associé à la classe Partie.

Exercice 5 : Relation ente états-transitions et activités

Considérons un système de torpille dont les classes et fonctions sont décrites par le diagramme suivant :



Ci-après la description des entités du système recueilli auprès du maître d'ouvrage:

- ✓ Torpille :
 - si la torpille au repos est mise sous-tension par l'opérateur elle initialise ses paramètres et peut prendre en compte un ordre feu envoyé par l'opérateur une fois l'initialisation terminée ;
 - afin d'éviter le déchargement de la batterie de la torpille, si l'ordre feu n'est pas reçu 10 minutes après son attente, la torpille retourne au repos ;
 - lorsque la torpille reçoit un ordre feu envoyé par l'opérateur, elle acquitte de sa réception ;
 - lorsque la torpille est éjectée par le lanceur, elle démarre la propulsion ;
 - à chaque état de la phase de lancement, l'opérateur ou le contrôleur de sécurité peuvent ordonner un retour au repos ;
 - à tout moment, l'opérateur peut ordonner la mise en état inerte de la torpille, la charge est alors désarmée et l'énergie coupée. La torpille ne pourra plus jamais être mise sous-tension.
 - ✓ ContrôleurSécurité :
 - chaque seconde, le composant de contrôle de sécurité vérifie si la torpille est en zone interdite ;
 - dans le cas positif, il demande la mise en état inerte de la torpille.
 - ✓ Lanceur :
 - lorsque le lanceur est chargé et que la torpille a acquitté de la réception de l'ordre feu, la réception d'un ordre de lancement engendre l'éjection de la torpille.
 - ✓ Arme :
 - si la torpille est suffisamment éloignée et que l'opérateur a demandé l'armement de la torpille, elle s'arme ;
 - dans ce cas, la torpille peut exploser lorsque le détecteur indique qu'elle est à la bonne proximité de la cible ;
 - à tout moment, la détection d'un épuisement de l'énergie engendre le désarmement définitif de l'arme.
1. Représenter les diagrammes d'état-transitions des classes « Torpille », « ContrôleurSécurité », « Lanceur », et « Arme ».
 2. Traduire les diagrammes d'états-transitions en diagramme d'activités avec couloirs.
 3. Représenter les associations entre les classes ainsi que leur navigabilité.

NB : Le déclenchement d'une activité d'appel d'une opération externe à la classe du diagramme d'état représenté sera noté « Classe::Opération() ».