

Conception de machines à états et patron de conception State

Un système de planification de cycles de séminaires

Un centre culturel organise régulièrement des cycles de séminaires thématiques pour ses adhérents. Assister aux conférences d'un cycle est gratuit, il faut toutefois s'inscrire au préalable de manière à ce que le centre culturel puisse imprimer les éventuels documents fournis lors des séminaires. Quand un cycle de séminaires est complet (la capacité d'un cycle de séminaires, i.e. le nombre de places, est égale au nombre de places de la salle de séminaire choisie), les nouveaux inscrits sont placés sur liste d'attente dans l'espoir d'un éventuel désistement. Quand la période de réservation est échue, le centre culturel peut préparer le cycle de séminaires. Les inscrits d'un cycle de séminaires peuvent l'abandonner en cours de route. Si suite à des abandons, il reste moins que cinq inscrits, le cycle est abandonné. Les cycles de séminaires terminés (abandonnés ou terminés normalement) sont archivés dès que le lendemain de la date du dernier séminaire est atteinte.

Le centre culturel souhaite intégrer la planification des cycles de séminaires à son logiciel de gestion des adhérents.

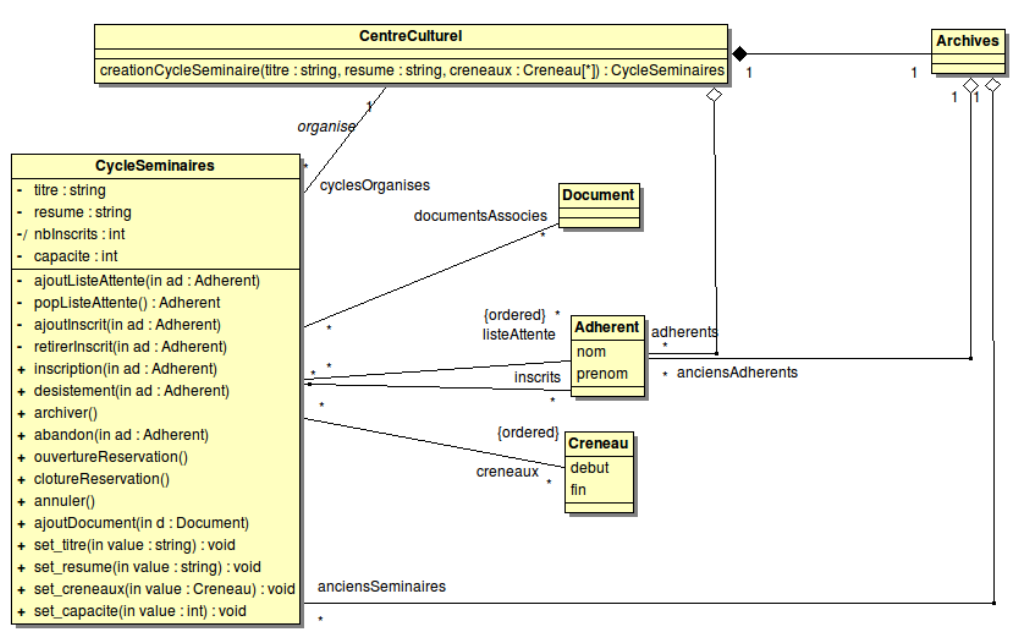


FIGURE 1 – Un diagramme de classes incomplet pour le centre culturel

Des diagrammes d'analyse incomplets vous sont donnés pour la planification de cycles de séminaires aux figures 1 (diagramme de classes) et 2 (machine à états de la classe `CycleSéminaires`).

Étude de la machine à états

Question 1. Décrivez les modifications à apporter à la machine à états de la figure 2 de manière à rendre les cycles de séminaires annulables. Les cycles sont annulables à tout moment, sous réserve qu'ils ne soient pas déjà terminés d'une manière ou d'une autre. On ne s'intéressera pas au traitement à effectuer pour prévenir les éventuels inscrits. Un cycle de séminaires est automatiquement annulé si quand on clôt la réservation, le nombre d'inscrits est strictement inférieur à 10. Les cycles annulés sont archivés de la même façon que les autres.

Question 2. Dans la machine à états de la figure 2, on ne voit pas la prise en compte du fait qu'un même adhérent ne peut pas s'inscrire deux fois au même cycle de séminaires. On suppose en effet que cette vérification est faite par les méthodes `ajoutInscrit` et `ajoutListeAttente`, qui ne font rien si l'adhérent passé en paramètre est déjà inscrit (sur liste principale ou sur liste d'attente). Sur cette base, la spécification de ce qu'il se passe lors de la réception de l'événement `inscription` dans les états `places disponibles` et `complet` est-elle correcte? Justifiez précisément votre réponse.

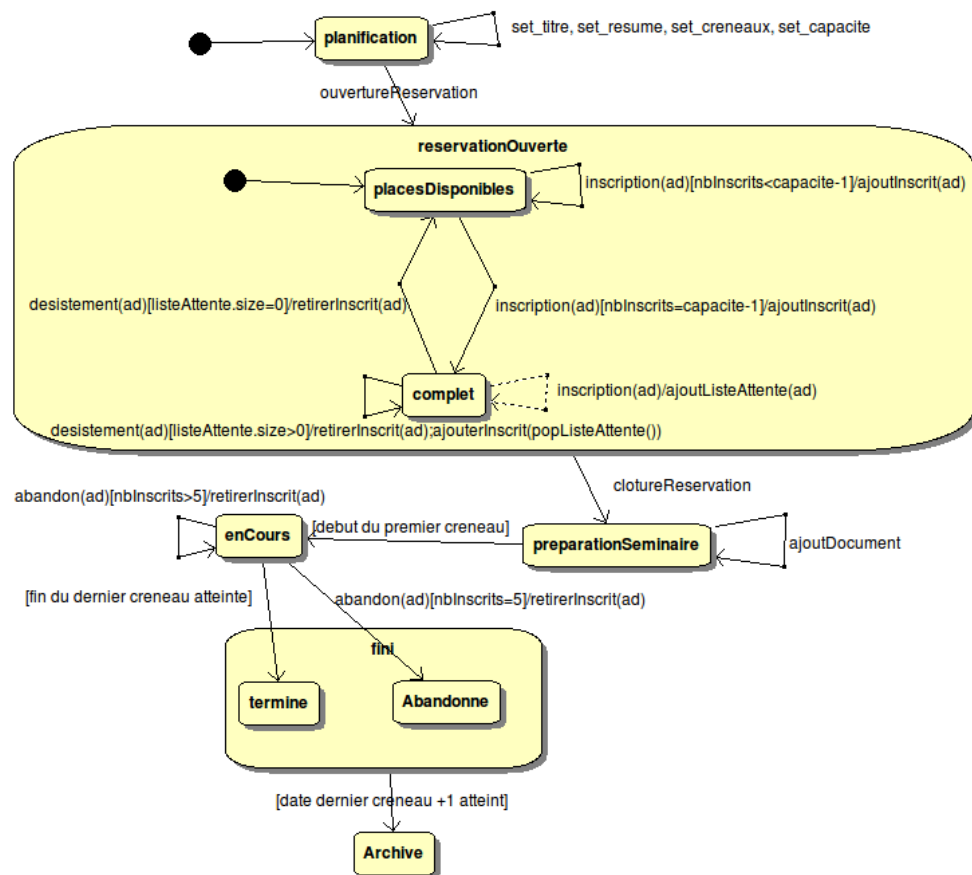


FIGURE 2 – Une machine à états pour la classe CycleSeminaires

Implémentation d'une machine à états avec le patron de conception State

Question 3. Comment implémenter la machine à état de la classe CycleSeminaires sans le patron de conception State ?

Question 4. Comment implémenter la machine à état de la classe CycleSeminaires avec le patron de conception State ? On réfléchira particulièrement aux deux points suivants :

- Comment un état a-t-il connaissance de l'objet représentant l'état le suivant ?
- Comment un état peut-il piloter le changement d'état d'un cycle de séminaires, i.e. comment a-t-il connaissance de son contexte ?

Question 5. Comment gérer les transitions spontanées ?