



Cours de Systèmes Multi-Agents et Intelligence Artificielle

Enseignant: Dr. NGUYEN Manh Hung

Master 1

Rapport de Projet

Réalisé par :

Hong G.

Tatiana Ngassa

Gervais Sikadie Sertillange F.

Ginel Dorleon

Août 2016

Quelques conventions utilisées dans ce document.

Les termes calendrier, emploi du temps, horaire, timetable, scheduling utilisés dans ce document sont tous équivalents.

Le terme participant utilise désigne en tout temps les acteurs , les entités du systèmes.

CSP : Constraint Satisfaction Problem

INTRODUCTION-

DESCRIPTION ET APPROCHE DU PROBLEME A RESOUDRE

Planifier un calendrier pour une activité, une école, ou gérer les heures de vol aérien dans une aviation où toutes les contraintes données sont satisfaites n'est pas une chose facile à implémenter par l'humain. Créer un tel calendrier est manuellement complexe et demande un travail de longue haleine.

Cependant, de nos jours, les systèmes intelligents sont construits pour agir à notre place et proposent des solutions satisfaisantes. Ainsi, dans le cadre du cours de Système Multi-Agents et Intelligence Artificielle, nous nous proposons de construire un système intelligent capable de générer le « Sheduling » d'une école donnée. . En automatisant un tel processus, notre système nous génère un calendrier en tenant compte des entrées pour les différentes entités du système. Un tel système permet ainsi aux gestionnaires d'établissement d'économiser beaucoup de temps soit pour la gestion des conflits d'horaire, soit pour gérer les horaires de cours ou soit pour planifier les examens.

Ensuite, le système que nous avons développé se base sur une approche pratique pour la construction des systèmes d'horaires de cours, qui aussi peut être personnalisé pour répondre à tous les problèmes d'horaire.

Même si la plupart des administrations de nos jours sont informatisées, le problème de la planification des horaires des cours dans les écoles, la planification des horaires de vol dans les aéroports, la planification dans le transport, certaines planifications sont encore principalement effectuées manuellement en raison de son degré de difficulté. Notre approche traite ainsi ce problème du côté des établissements scolaires. Les écoles sont encore confrontées au planning pour les demandes de rendez-vous, les calendriers tout en considérant le temps et les efforts. Ainsi, la programmation d'horaire est un problème de satisfaction contrainte à laquelle nous proposons une solution qui satisfait l'ensemble des données et contraintes reçues à l'entrée.

Ce problème de planification demande de trouver des créneaux horaires et les salles de classe qui peuvent satisfaire les contraintes imposées sur les cours offerts, la disponibilité et la préférence des professeurs, salles de classe et ainsi de suite. C'est un problème d'optimisation combinatoire de niveau très difficile, le temps de calcul par horaire tend à croître de façon exponentielle au fur et à mesure que le nombre de variables, c'est-à-dire nos entrées- augmente.

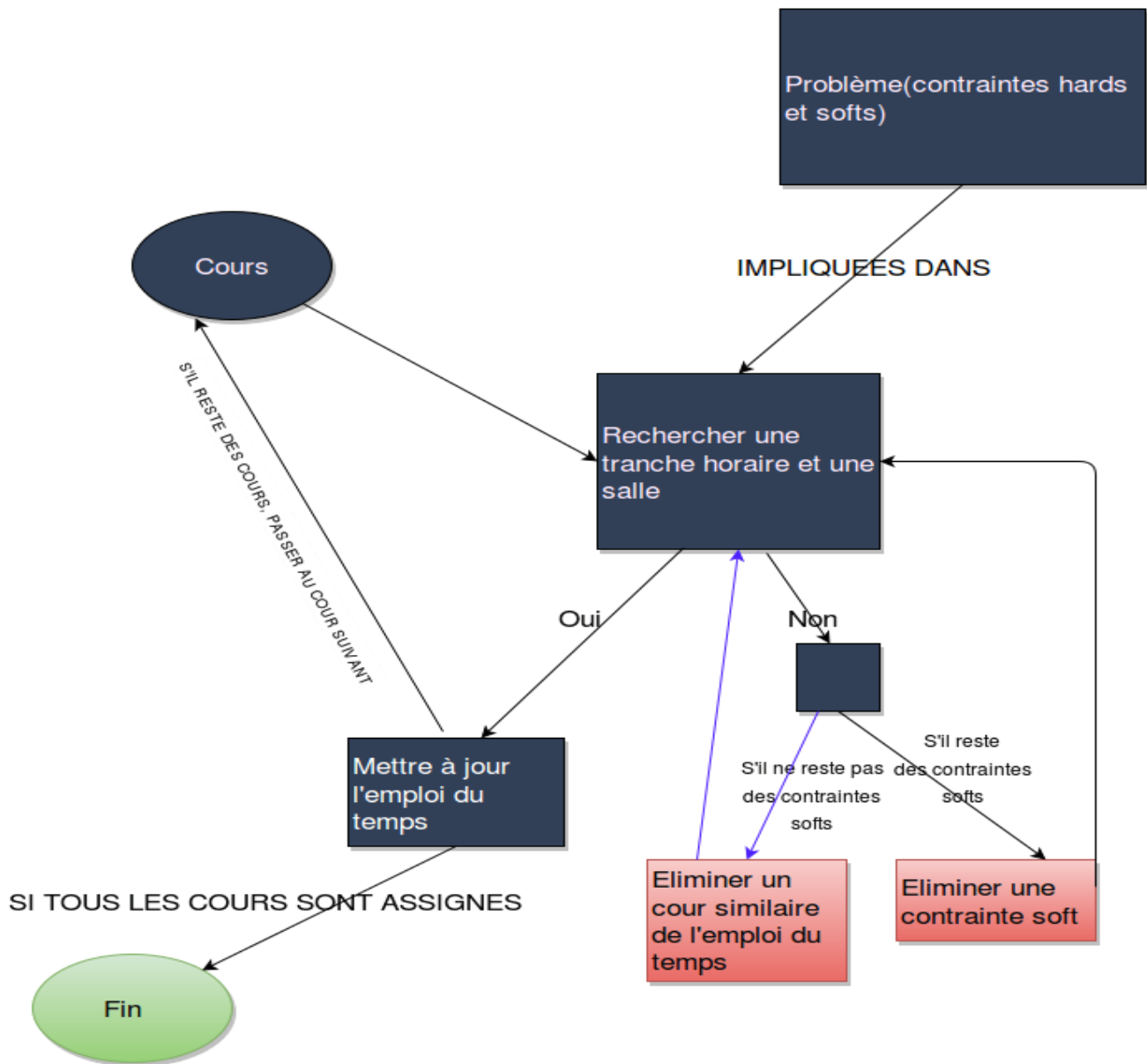
Pour résoudre un tel problème, différentes approches peuvent être utilisées comme le graphe coloré ou la programmation utilisant les procédures de recherche telles que le recuit simulé, le « LocalSearch » « Tabu Search », « Forward Search », « Backtracking », les algorithmes génétiques.

Ainsi, pour la constitution du système, nous adoptons l'approche de l'intelligence artificielle. Nous présentons une méthode hybride utilisant à la fois le « Back Tracking » et le « Constant Programming » permettant ainsi de surpasser les contraintes et produire les résultats attendus.

DESCRIPTION DE L'ALGORITHME UTILISE

Les contraintes étant divisées en deux groupes: les *hards* et les *softs constraints*. Les *hards constraints* doivent toujours être satisfaites mais les contraintes secondaires peuvent être violées. Les cours sont programmés à tour de rôle, s'il n'est pas possible d'assigner une tranche horaire et une salle sans violations de contrainte, le programme fait un retour en arrière, backtrack, relaxe ou élimine quelques contraintes *softs* une à une jusqu'à ce que le cour soit programmé. Si malgré ça le cour n'est pas programmé, un cours similaire sera déprogrammé et la recherche sera lancée à nouveau. Si à la fin de l'algorithme des cours restent non programmés, alors le programme n'a pas réussi sinon le Timetable est bon.

Schéma



PORTEE DU PROJET

Le « Système d'emploi du temps » génère un calendrier pour chaque classe, chaque cours et enseignant, en accord avec le calendrier de disponibilité des enseignants, la disponibilité et la capacité des ressources physiques. Par ressources physiques on entend: les salles de classe, laboratoires et salle informatique et les règles applicables aux différentes classes, semestres, enseignants et le niveau des sujets.

Tout en tenant compte de la négociation et la communication entre les agents, le système améliore énormément l'utilisation et l'optimisation des ressources disponibles.

ENVIRONNEMENT D'ENTREE

L'environnement considéré est un établissement qui administre ses cours comme suit :

- ✓ Les cours se font le jour.
- ✓ Les classes sont prévues pendant les jours de la semaine et le samedi matin.
- ✓ La configuration pour un cours est la suivante: Théorie -Révision-Pratique.
- ✓ En théorie, la classe peut contenir 40 à 60 étudiants, et de 15 à 25 pendant des heures de pratique.
- ✓ L'intervalle de temps minimum est de 1 heure. Ainsi, 1 heure pour les cours théoriques et 2h pour la révision et la pratique.
- ✓ Les heures de cours sont situées entre 08:30 à 17:00 en semaine.

Comme mentionné plus haut, de nombreux types de problèmes de planification de calendrier existent. Ces problèmes ont tous plusieurs propriétés en commun. L'un de leur point commun est que certaines entités doivent être programmées, par exemple, l'emploi du temps de l'établissement considéré est un problème à plusieurs entités telles que des professeurs, des cours, des pratiques, les classes et les laboratoires.

Toutes ces entités ont des propriétés, par exemple une classe est liée à la formation et les étudiants de cette classe sont enseignés. Ainsi, l'assignement ne peut pas être fait de façon arbitraire, mais de nombreuses contraintes sont à considérer. Une solution est possible si aucune des contraintes n'est violée et qu'il n'y a pas de conflits entre les agents qui interviennent.

LES CONTRAINTES POUR GERER LES CONFLITS

Il existe diverses contraintes auxquelles doivent tenir compte le système sur les créneaux horaires et les salles de classe.

Un calendrier qui rompt une contrainte n'est pas une solution réalisable, et doit être réparé ou rejeté par l'algorithme d'établissement des horaires. Des contraintes, dites « Hard Constraints » sont en effet définies pour permettre au système de gérer les "conflits". Par exemple :

- ✓ Une salle de classe ne peut pas être affectée à plus d'une activité dans le même temps.
- ✓ Un professeur ne peut pas enseigner plus d'une classe à la fois.
- ✓ Des cours pour une même année ou session d'un département ne peut pas avoir lieu en même temps.
- ✓ La salle de classe pour un cours doit avoir une capacité suffisante pour prendre tous les étudiants enregistrés au cours.
- ✓ La salle de classe doit être bien équipée avec des installations nécessaires pour les cours.

Ensuite, viennent s'ajouter des contraintes qui sont plus ou moins flexibles, les « soft constraints » et cela, c'est le système qui doit prendre cette décision et dire si oui ou non que c'est acceptable suivant un degré de priorité jugé par le système lui-même.

- ✓ Les cours ne sont pas affectés à des intervalles de temps interdits.
- ✓ Pendant les heures de cours, le professeur doit être limité à la durée maximale.
- ✓ Une pause de déjeuner doit être prévue.
- ✓ Les cours théoriques sont prévus en séance du matin, et les cours pratiques sont programmés en session de l'après - midi.
- ✓ Si possible, les heures pour un cours devraient être programmées consécutivement.
- ✓ Les salles de classe devraient être allouées de manière à minimiser les distances entre deux salles de cours voisines.

Cependant, il est généralement difficile de satisfaire à toutes ces contraintes quand on sait que toutes les entités du système ont leurs préférences et intérêts, d'où la nécessité pour le système de trouver une solution qui satisfait tous les acteurs du système.

LE SYSTEME PROPOSE,

DESCRIPTION

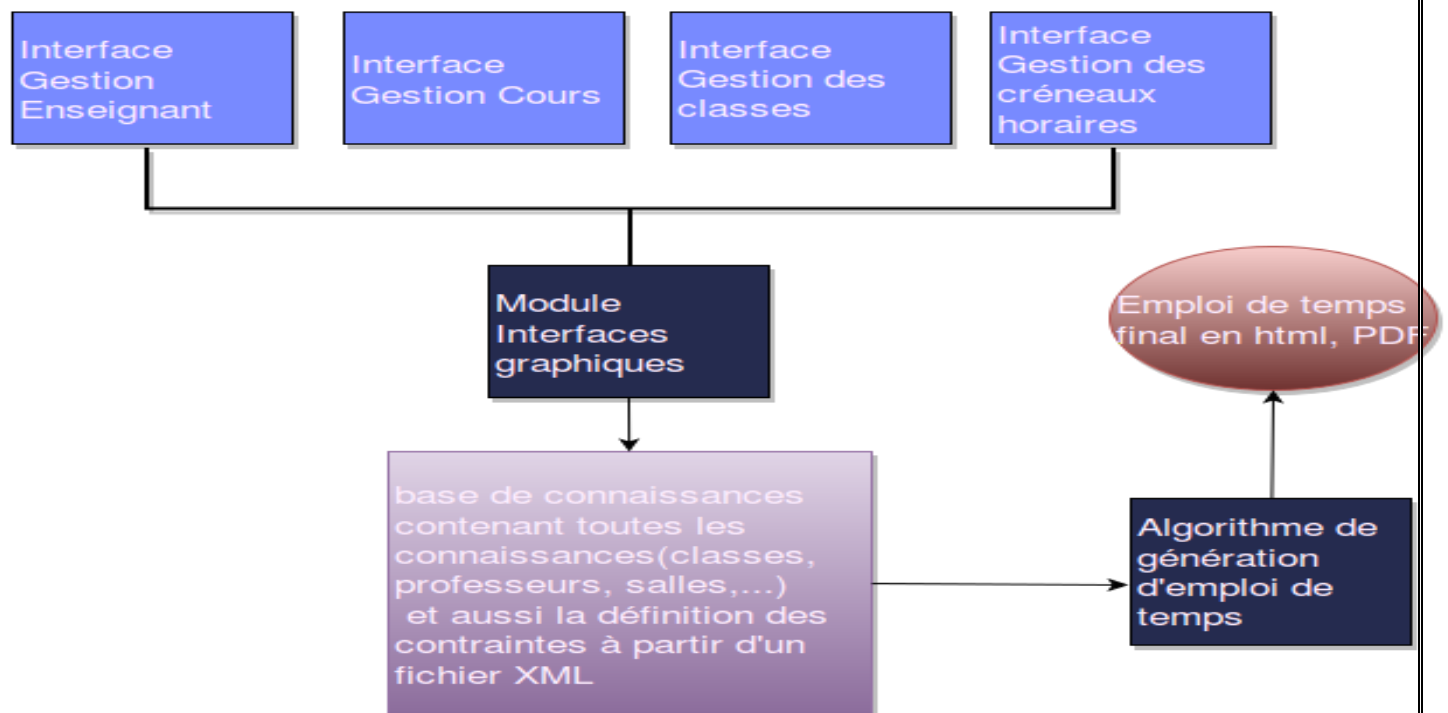
Notre système génère en sortie un calendrier pour une semaine, un mois, un semestre, etc... Le système prend en compte différentes entrées de l'utilisateur, stockées dans un fichier XML, telles que la liste des enseignants, la liste des cours, la liste des semestres la liste des salles de cours, la liste des jours et les heures de cours et autres règles, autres faits et contraintes à l'aide des formulaires dans l'espace « Entrée de Données »

Après avoir généré un calendrier, le système donne la possibilité l'exporter en format PDF ou HTML.

CARACTERISTIQUES

Le système est facile et simple à comprendre, avec une interface utilisateur intuitive avec des guides et des exemples.

Le système est rapide, il génère le calendrier en quelques minutes, ce qui permet d'économiser et les efforts des agents.



IMPLEMENTATION/ENVIRONNEMENT D'EXECUTION

COTE MATERIEL

La conception et l'exécution du système ont été réalisées sur deux ordinateurs utilisant le système d'exploitation Linux X64, version Ubuntu 16.04.1 LTS (Xenial Xerus)

1- Toshiba:

Processor: AMD E1-2100 APU with Radeon(TM) HD Graphics × 2

Carte Graphique: Gallium 0.4 on AMD KABINI (DRM 2.43.0, LLVM 3.8.0)

RAM: 4 GB

2- Dell

Processor: Intel® Core™ i3 – 3227U CPU @ 1,90GHz x 4

Carte Graphique: Intel® Ivy bridge Mobile

RAM: 4gb

LOGICIEL, LANGAGE, DESIGN ET ENVIRONNEMENT DE PROGRAMMATION

JAVA

XML :

JDK 1.8 +

Eclipse IDE, version Luna

DOM4J pour *parser* les fichiers XML

UML

ITEXT pour la génération des fichiers PDF.

DIAGRAMME DE PACKAGE

Ici, on peut voir le diagramme des packages contenant toutes les classes.

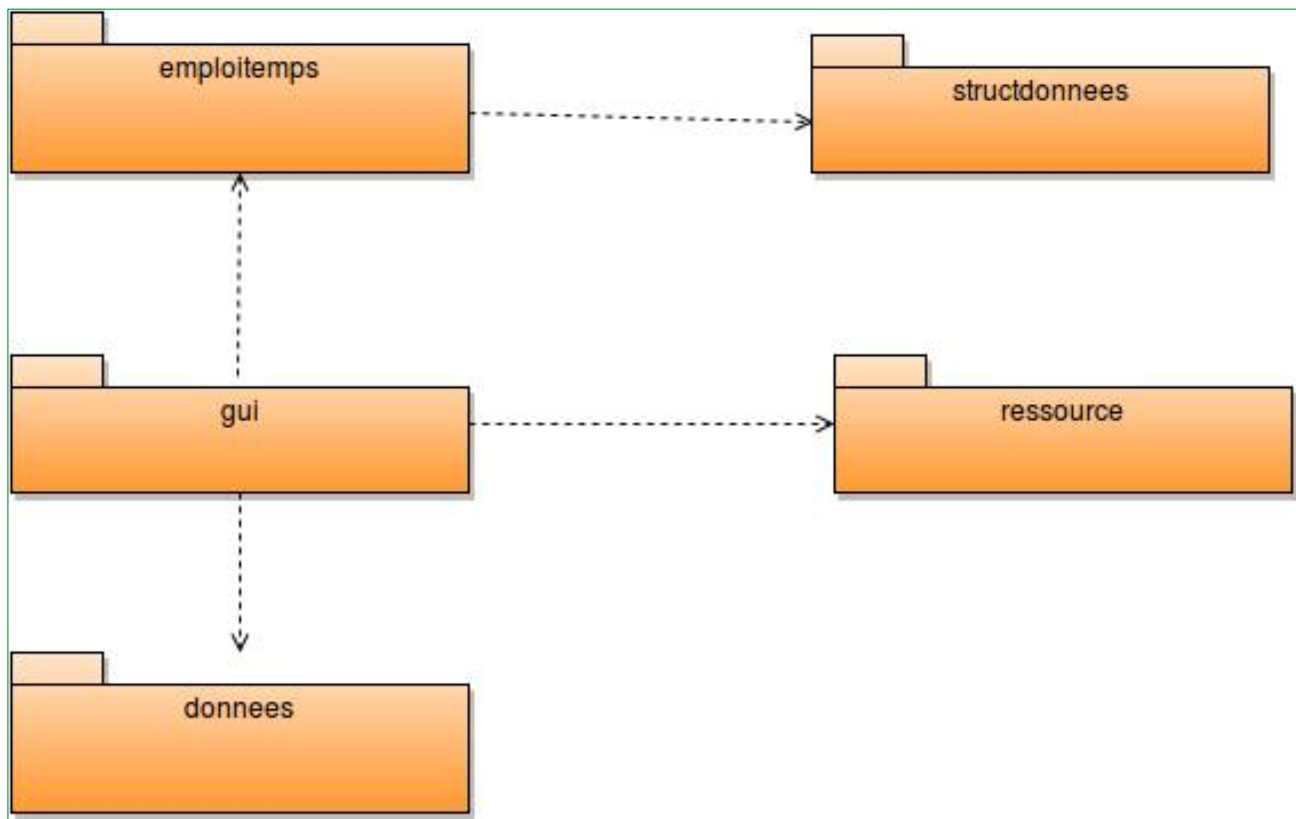


Diagramme de classe pour le package « *structdonnees* » contenant les différentes classes d'entrée.



Diagramme de classe pour le package « *gui* » qui contient les classes principales de lancée du système.

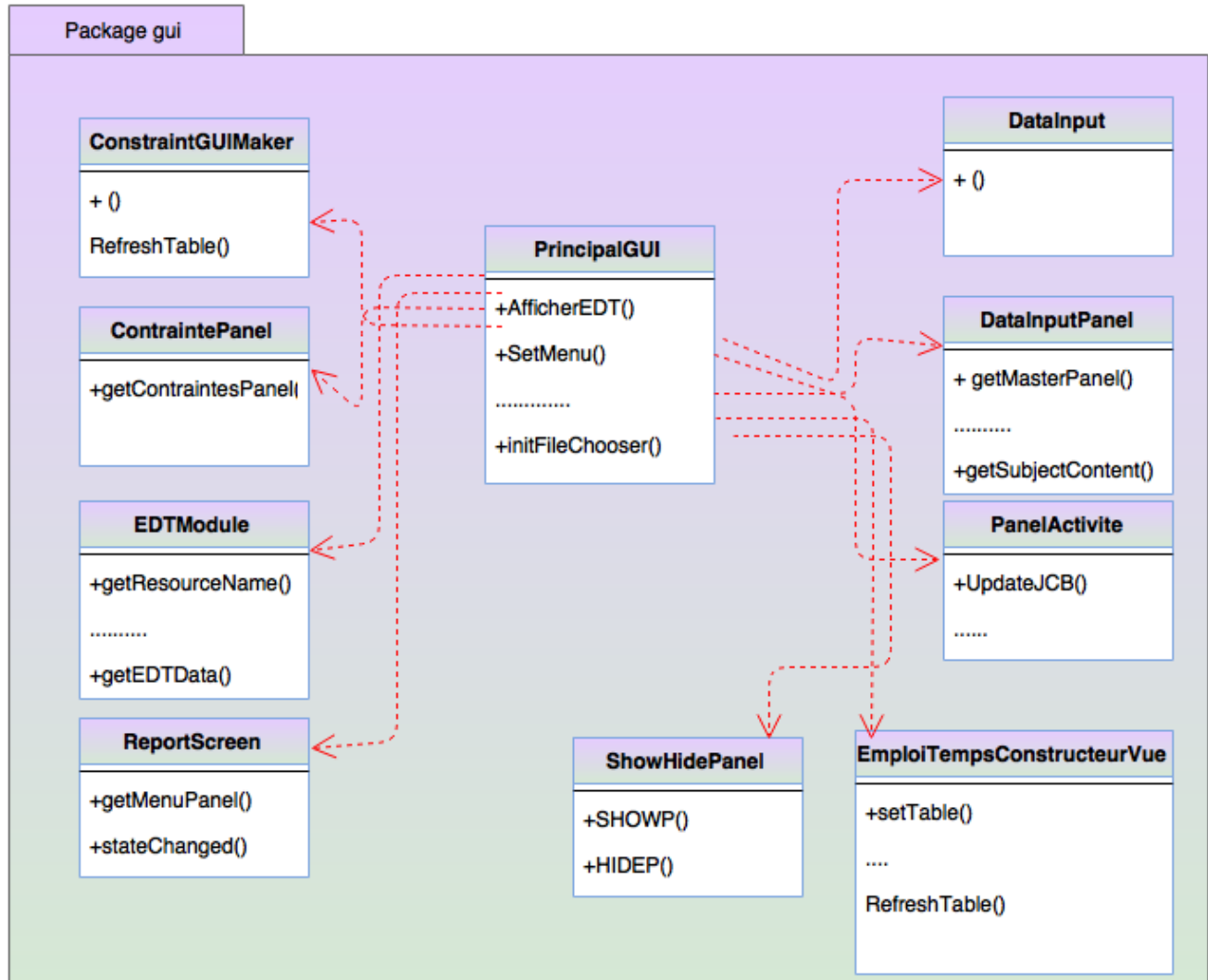


Diagramme de classe du package « *emploitemps* »

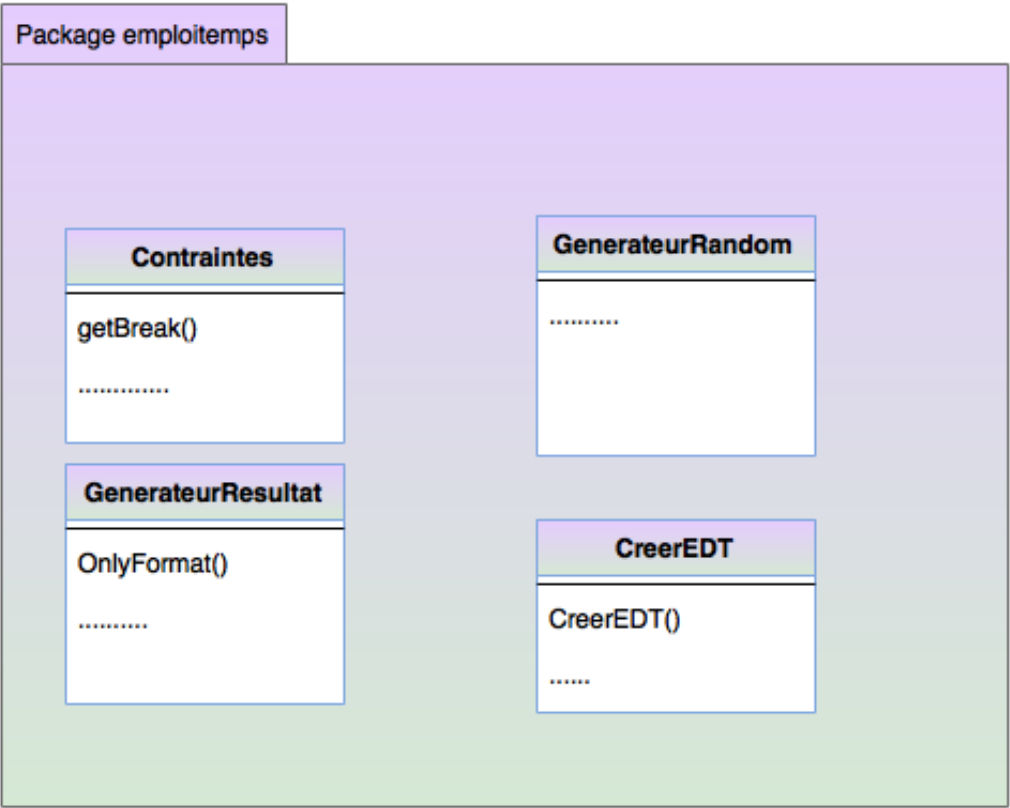


Diagramme de classe du package « donnees »

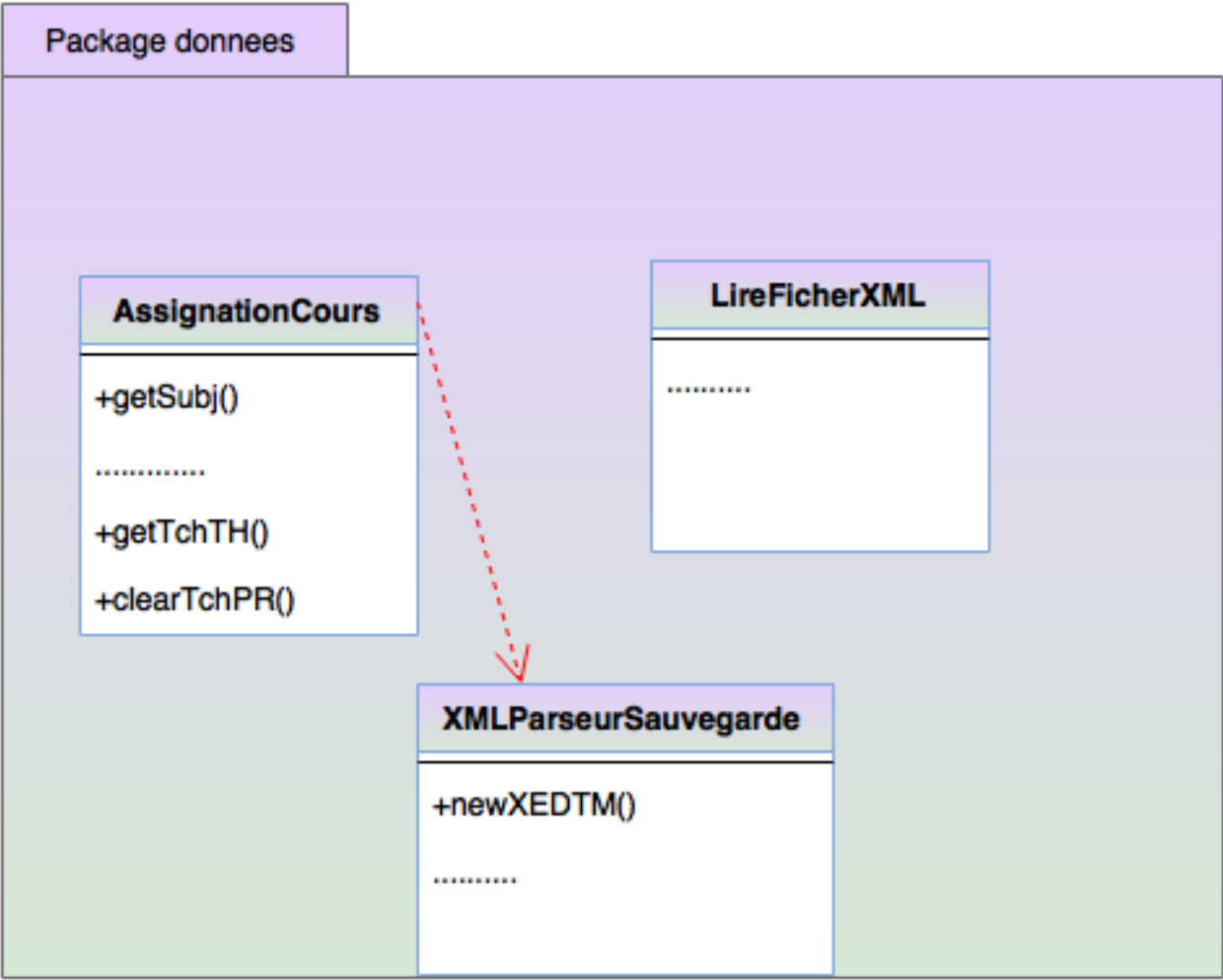
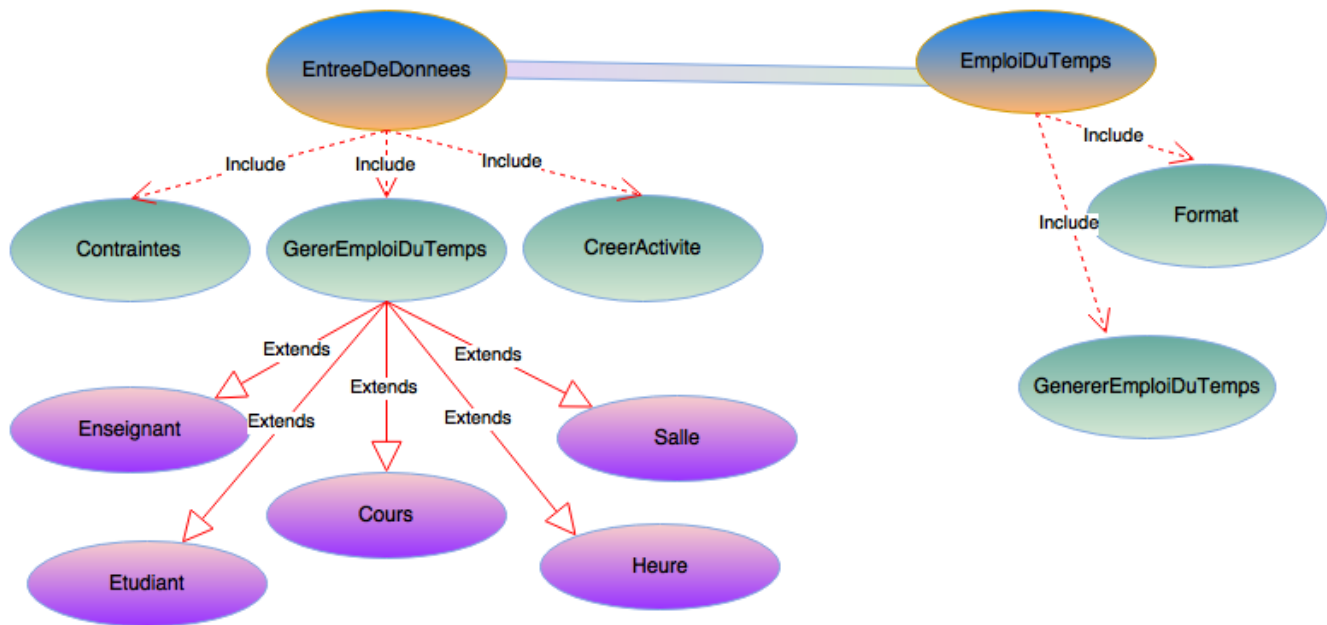


Diagramme de cas d'utilisation

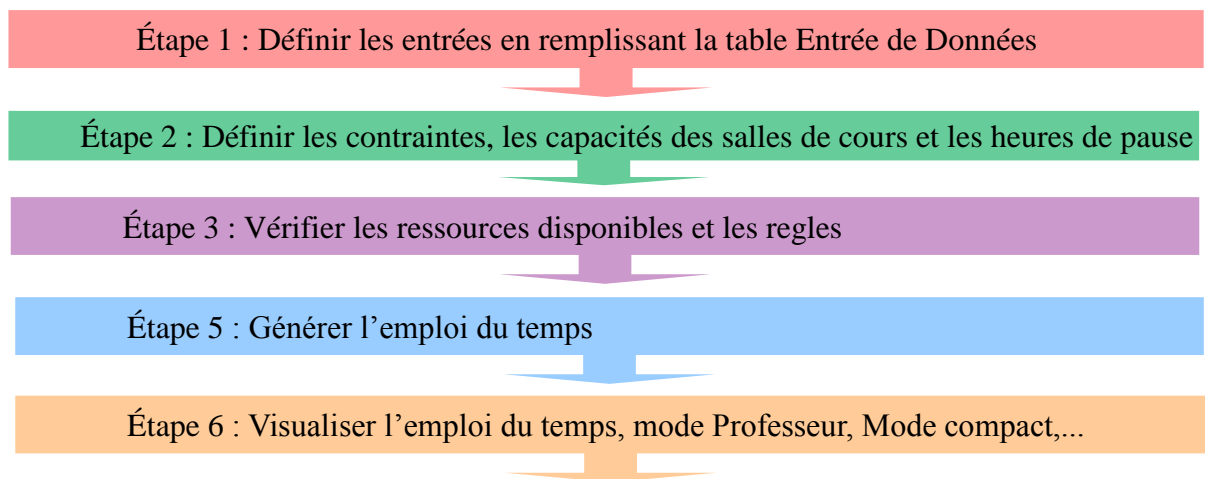


FONCTIONNEMENT DU SYSTEME

A l'initialisation du système, les agents/différentes entités se communiquent et définissent chacun leurs objectifs. Après que le système ait fini de prendre en compte toutes les préférences des agents ainsi que toutes les contraintes définies, il est maintenant de la part du système d'utiliser l'ensemble des données saisies pour générer un calendrier. Ainsi, en tenant compte de toutes les préférences des agents/de toutes les entités, le système génère un nouveau calendrier.

Pour générer un nouveau calendrier, il faut saisir un ensemble de données pour les différentes entités du système à savoir les enseignants, les cours, les salles de cours, les départements, les jours et les heures de cours de la semaine.

Le système indique les étapes à suivre pour générer un calendrier, voir figure suivante :



CAS DE TEST ET RESULTAT

Dans les pages suivantes, nous présentons quelques captures de quelques cas de test.

Vue de la page d'accueil du système



CONCLUSION

Utilisant l'approche de l'intelligence artificielle, nous proposons ce système pour la production d'un emploi du temps. Dans une nouvelle version de notre travail, nous pensons à d'autres fonctionnalités du système telles que le *Login* à partir d'une interface web, envoyer le calendrier par courriel, ajout de nouveaux agents pour un travail plus automatique.