

## Mini-projet d'Intelligence Artificielle

Planification des emplois du temps à Polytech

<b>Encadrant</b>	Hoël LE CAPITAINE
<b>Date</b>	2 janvier 2017

## Table des matières :

<b>Partie 1 : Résolution générale du problème</b>	<b>2</b>
Données nécessaires sous forme relationnelle	2
Données supplémentaires	3
Traduction du schéma d'analyse en Prolog	5
Explication des prédicats	6
Spécification et définition d'une solution	7

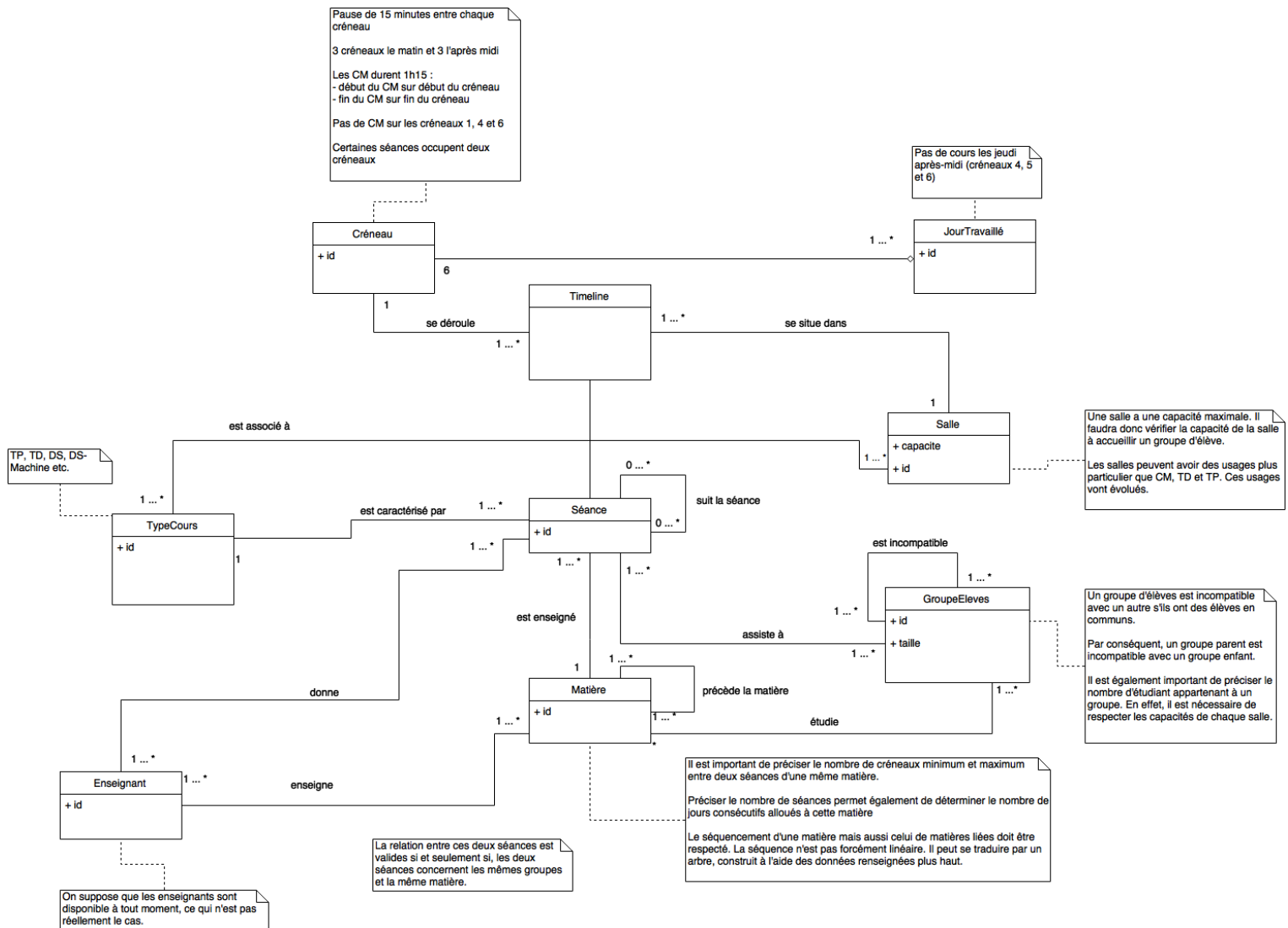
## Partie 1 : Résolution générale du problème

### 1) Données nécessaires sous forme relationnelle

Le problème paraît simple : il suffit d'assigner une salle ainsi qu'un créneau à chaque séance. L'ensemble de ces assignations nous donne notre emploi du temps qui est représenté par la relation "Timeline" sur le modèle.

Nous avons fait certains choix de modélisation :

- Les usages des salles et les types de cours étant similaires, nous les avons représentés par une unique relation
- Nous avons représentés les jours travaillés bien que ce ne soit pas nécessaire, les créneaux permettent de déduire implicitement le jour travaillé via une simple division



La représentation est également visible en cliquant sur ce [lien](#).

### Contraintes :

L'ensemble des contraintes présentes sur le schéma sont résumées ici :

- Pas de CM sur les créneaux 1, 4 et 6 de chaque journée
- La salle assignée doit avoir une capacité supérieure ou égale à la taille du ou des groupes qui assistent la séance
- Un enseignant n'est pas disponible pour une séance sur un créneau donné si et seulement si il administre déjà une séance sur ce même créneau
- Certaines matières sont liées : une matière ne peut être étudiée tant que la ou les matières qui doivent la précéder ne sont pas terminées
- Il existe également un ordonnancement des séances d'une même matière. Par exemple, il peut être nécessaire qu'un cours soit donné avant le début des travaux pratiques
- Deux groupes sont incompatibles si et seulement s'ils ont des élèves en commun. Par conséquent, deux séances distinctes destinées à deux groupes incompatibles ne peuvent avoir lieu sur le même créneau
- La salle doit être libre
- L'usage de la salle doit être adapté au type du cours

## **2) Données supplémentaires**

Un jeu de données est nécessaire afin d'établir une première solution partielle. Il ne doit pas être trop conséquent ainsi que respecter notre modèle relationnel.

Afin de rester cohérent, nous avons décidé de travailler sur un jeu de données que l'on connaît. Nous serons alors plus apte à juger de la pertinence de la planification réalisée par notre IA.

Nous nous sommes donc basés sur le département informatique de Polytech Nantes et plus particulièrement la quatrième année.

### Remarque :

*Les associations (autres que celles avec des séances) ne sont pas représentées ici. En effet, un simple id ne serait pas parlant. Elles sont cependant présentes dans sa traduction en Prolog.*

### Enseignants :

- Hoël LE CAPITAINÉ
- Pascale KUNTZ
- Guillaume RASCHIA
- José MARTINEZ
- Fabien PICAROUGNE
- Marc GELGON
- Julien COHEN
- Marcus BARKOWSKY
- Benoit PARREIN
- Rémi LEHN
- JP. GUEDON

Salles :

<b>Salle</b>	<b>Capacité</b>	<b>Usages</b>
B001	26	DS-Machine, TP, MP
B007	19	TP, Labo
A1	350	CM, DS
A2	160	CM, DS
E101	24	CM, TD
E102	24	CM, TD
E103	24	TD
E104	24	TD
D004	65	CM
D008	24	Innovation pédagogique
D009	24	TP
D010	48	CM
D012	24	TP
D013	110	CM, DS
D117	24	CM, TP

Groupes d'élèves :

- INFO
- SILR
- ID
- SILR1
- SILR2
- INFO4

Matières :

- Techniques de base de l'IA
- Mini-projet de systèmes à base de connaissances
- Optimisation et méta-heuristiques
- Réseaux 3
- Traitement du Signal
- Comptabilité
- Gestion organisationnelle des connaissances

Types de cours :

- CM
- TP
- TD
- DS
- DS-Machine

Séances :

- Séance 1 :
  - matière : Traitement de l'image
  - groupes : SILR1, SILR2
  - enseignants : JP. Guédon, M. Gelgon
  - salle : a1
- Séance 2 :
  - matière : Mini-projet de systèmes à bases de connaissance
  - groupes : ID4
  - enseignants : H. Lecapitaine
  - salle : d117
- Séance 3 :
  - matière : Mini-projet de systèmes à bases de connaissance
  - groupes : ID4
  - enseignants : H. Lecapitaine
  - salle : d117
- Séance 4 :
  - matière : Mini-projet de systèmes à bases de connaissance
  - groupes : SILR2
  - enseignants : G. Raschia
  - salle : c002

La semaine 1 comporte 28 séances, elles n'ont pas toutes été détaillées ici.

Ces données sont particulières car nous avons, tout simplement, choisis ces données. Bien qu'elles soient concrètes et réelles, elles ne sont cependant pas représentative de l'école. Enfin, la période temporelle est réduite à une semaine.

### **3) Traduction du schéma d'analyse en Prolog**

Lors de la question précédente, nous avons créé un jeu de données trop important. Comme l'a été évoqué dans le sujet, nous l'avons donc réduit et allons donc commencer par planifier uniquement la première semaine de la rentrée (Semaine 1 de 2017 : 02/01/17 - 08/01/17).

Nous l'avons ensuite traduit en Prolog. Il est important de préciser que l'ensemble des associations présentes sur le modèle relationnel sont traduits par des prédicats.

Le jeu de données est visible dans le fichier dataset.p1, présent également dans l'archive.

### *Explication des prédicats*

#### Entités :

- enseignants(nom) . Représente les enseignants.
- groupeEleve(nom) . Représente les groupes d'élèves.
- taille(groupeEleve/salle, entier) . Représente la taille d'un groupe d'élèves ou d'une salle.
- taille(salle, entier) . Représente la capacité d'une salle.
- typeCours(nom) . Représente les différents types de cours possibles.
- salle(nom) . Représente les salles de cours.
- matière(nom) . Représente les matières.
- seance(id) . Représente les séances.
- creneau(id) . Représente les créneaux.
- jours(id) . Représente les jours.

#### Relations entre ces entités :

- estDans(idCreneau, idJour) . Un certain créneau appartient à un certain jour.
- estIncompatible(groupeEleve, groupeEleve) . Représente l'incompatibilité entre 2 groupes d'élèves.
- usageSalle(nomSalle, typeCours) . Représente l'usage possible d'une salle pour un type de cours précis.
- enseigne(nomProf, nomMatière) . Représente le fait qu'un qu'un enseignant peut enseigner une matière.
- typeSeance(idSeance, typeCours) . Représente le fait qu'une séance est dédié à un certain type.
- suitSeance(idSeance1, idSeance2, creneauxMin, creneauxMax) . Représente le fait que la séance idSeance2 suit la séance idSeance1 (et donc qu'elle ne peut avoir lieu si la seance idSeance1 n'a pas encore eu lieu). creneauxMin et creneauxMax représente le nombre de créneaux minimum et maximum qu'il doit s'écouler entre ces 2 séances. N'ayant pas connaissance de la façon dont ils sont choisis, ils sont pour l'instant choisis arbitrairement.
- assiste(nomGroupe, idSeance) . Indique que le groupe nomGroupe assiste à la séance idSeance.
- anime(idSeance, nomProf) . Indique que le professeur nomProf anime la séance idSeance.
- estEnseigne(idSeance, nomMatière) . Indique que la séance idSéance est une séance de matière.
- precedeMatiere(nomMatière1, nomMatière2) . Indique qu'une matière précède une autre, et donc que les cours de nomMatière1 doivent être terminés avant les cours de nomMatière2. (Là aussi nos connaissances sur cette contrainte sont limitées donc nous nous contenterons de déductions implicites.)

#### 4) Spécification et définition d'une solution

Nous avons opté pour un algorithme non déterministe car nous avons abordé le problème similairement à la coloration de graphes vu en travaux dirigés.

En effet, il consiste à assigner une couleur à un sommet tout en respectant une unique contrainte : deux sommets voisins ne peuvent avoir la même couleur. Ici, ce n'est pas des couleurs que l'on va assigner mais une salle ainsi qu'un créneau à une séance. Il y a également un plus grand nombre de contraintes à prendre en compte. Cela est notamment dû à la complexité des "objets" que l'on manipule. Par exemple, une séance est caractérisé par une matière mais aussi un type etc. L'ensemble des contraintes est visible dans la première partie de ce compte rendu.

Enfin, nous avons tenté d'exprimer le problème sous la forme d'un sextuplet mais cela n'a pas été couronné de succès.

<b>Entrée :</b>	<ul style="list-style-type: none"> <li>- liste de séances <math>Seances(id, type, groupe(s) d'élèves, enseignant(s))</math></li> <li>- nombre de créneaux <math>k \in \mathbb{N}</math></li> <li>- liste de salles <math>Room(nom, usage(s), capacité)</math></li> </ul>
<b>Fonction :</b>	$ASSIGNER : (Seances \times \mathbb{N} \times Room) \rightarrow (s \rightarrow (\mathbb{N}, Room))$ avec $s$ une séance
<b>Préconditions :</b>	<ul style="list-style-type: none"> <li>- Si <math>k = 0</math> (aucun créneau n'est disponible) a une solution si et seulement si <math>Seances</math> est vide</li> <li>- Une liste de séance vide peut être planifiée</li> <li>- Le nombre de créneau disponible doit être suffisant pour accueillir toutes les séances</li> </ul>
<b>Postconditions :</b>	<ul style="list-style-type: none"> <li>- Créneau entre 1 et <math>k</math> compris :  <math>\forall (s, c, r) \in ASSIGNER(S, k, R), c \in \{1, \dots, k\}</math></li> <li>- La salle doit exister :  <math>\forall (s, c, r) \in ASSIGNER(S, k, R), r \in Room</math></li> <li>- Toutes les séances sont placées :  <math> PLACER(S, k)  =  S </math></li> </ul> <p><u>On trouve également l'ensemble des contraintes relevées lors de la modélisation :</u></p> <ul style="list-style-type: none"> <li>- Pas de CM sur les créneaux 1, 4 et 6 :  <math>\forall (s, c, r) \in ASSIGNER(S, k, R) \wedge typeCours(s) = cm, c \neq 6t \wedge c \neq 1 + 6t \wedge c \neq 4 + 6t (t \in \mathbb{N})</math></li> <li>- L'usage de la salle est adapté au type du cours :  <math>\forall (s, c, r) \in ASSIGNER(S, k, R), typeCours(s) = typeSalle(r)</math></li> <li>- La salle doit être assez grande pour accueillir le/les groupes d'élèves :  <math>\forall (s, c, r) \in ASSIGNER(S, k, R), taille(groupeEleve(s)) \leq taille(r)</math></li> <li>- Un enseignant ne doit pas administrer une autre séance sur ce créneaux :</li> </ul>



	$\forall (s_1, c_1, r_1) \in \text{ASSIGNER}(S, k, R),$ $\nexists (s_2, c_2, r_2) \in \text{ASSIGNER}(S, k, R) \mid c_1 = c_2 \wedge \text{enseignant}(s_1) = \text{enseignant}(s_2)$ <p>- La salle doit être libre :</p> $\forall (s_1, c_1, r_1) \in \text{ASSIGNER}(S, k, R),$ $\nexists (s_2, c_2, r_2) \in \text{ASSIGNER}(S, k, R) \mid c_1 = c_2 \wedge r_1 = r_2$ <p>- Le ou les groupes d'élèves concernés par la séance ne doivent pas suivre une autre séance pendant le créneau k :</p> $\forall (s_1, c_1, r_1) \in \text{ASSIGNER}(S, k, R),$ $\nexists (s_2, c_2, r_2) \in \text{ASSIGNER}(S, k, R) \mid (c_1 = c_2) \wedge (\text{assiste}(s_1) = \text{assiste}(s_2))$ <p>- Deux groupes incompatibles ne peuvent avoir cours sur un même créneau :</p> $\forall (s_1, c_1, r_1) \in \text{ASSIGNER}(S, k, R), \nexists (s_2, c_2, r_2) \in \text{ASSIGNER}(S, k, R) \mid$ $c_1 = c_2 \wedge \text{estIncompatible}(\text{groupeEleve}(s_1), \text{groupeEleve}(s_2))$ <p>- Un cours ne peut pas avoir lieu si les cours nécessaires à sa compréhension n'ont pas encore eut lieu (modélisé par le prédicat <code>suitSeance</code>) :</p> $\forall (s_1, c_1, r_1) \in \text{ASSIGNER}(S, k, R), \nexists (s_2, c_2, r_2) \in \text{ASSIGNER}(S, k, R) \mid$ $\text{suitSeance}(s_1, s_2) \wedge c_1 < c_2$
<b>Algorithme :</b>	<pre> fonction ASSIGNER(Seances, k, Room) :     Timeline ← {}     tant que  Seances  ≠ 0 :         s ← choix_nd(Seances)         r ← choix_nd(Room)         c ← choix_nd({1, ..., k})          /* La fonction test_post_cond() vérifie l'ensemble des postconditions référencées ci-dessus */         Si test_post_cond(Timeline ∪ ({s, r, c})) :             Timeline ← Timeline ∪ ({s, r, c})             Seances ← Seances \ s         Sinon :             retourne ⊥      fin tant que     retourne Timeline </pre>

Timeline est le résultat escompté. Cette liste sera composée de triplets, eux mêmes composés d'une séance, d'un créneau et d'une salle.

On part alors d'une liste de séances. Tant que celle-ci n'est pas vide, on choisit aléatoirement une séance, un créneau et une salle. Une fois l'assignation faite, une vérification est effectuée afin de garantir le bon respect des contraintes (ou postconditions). Si c'est validé, le triplet formé est ensuite inséré dans notre timeline

et la séance est ensuite retirée de la liste de séances passée en entrée. Dans le cas contraire, cela retourne un échec. On obtient alors une liste cohérente, où toutes les séances ont été attribuées.

Cet algorithme est très basique et nécessitera de nombreuses optimisations avant de pouvoir être réellement utilisé. Par exemple, il serait intéressant de prendre en compte la taille du ou des groupes qui suivent une séance afin d'assigner la salle qui correspond au mieux à celui-ci : il est inutile de monopoliser un amphithéâtre pour un groupe d'une trentaine d'étudiants. Il serait également très utile de classer les séances dans le "bon" ordre (selon l'ordonnancement des matières et des séances). On ne ferait alors plus de choix déterministe ni sur la séance, ni sur le créneau.

Enfin, des contraintes plus poussées peuvent également être prises en charge comme la gestion des emplois du temps des enseignants-chercheurs.