

Cahier des charges

Mise en contexte du problème

Ce projet a pour but la réalisation d'une application de production d'un emploi du temps, et ainsi de faciliter le travail de M. Maeso. En effet, chaque année, il doit créer manuellement l'intégralité des emplois du temps de l'école, en tenant compte de toutes les contraintes de disponibilités des professeurs, des salles, des élèves, etc.

Mais ce problème peut être généralisé à toutes les planifications horaires d'employés, comme dans les écoles, les hôpitaux et toutes autres entreprises ayant un roulement de personnel à organiser.

La planification sous contraintes est la discipline des mathématiques appliquées consistant à ordonner diverses séquences ou événements dans un espace-temps limité. Ces planifications impliquent un grand nombre de contraintes, qu'elles soient liées aux disponibilités de personnes, d'emplacements ou à tous autres problèmes divers et variés.

La réalisation d'une planification est à la fois longue et périlleuse. Il est très difficile de concilier l'ensemble des contraintes qui accompagnent cet ordonnancement et de pouvoir organiser cela de manière optimale. La mise en place d'un calendrier est souvent faite à la main ou de manière approximative avec d'autres applications.

L'objectif de ce projet est donc de réaliser une application, permettant de générer une planification tenant compte de toutes les contraintes que l'on aura indiquées au préalable. Dans un premier temps, nous nous contenterons de générer un emploi du temps propre à l'école, quitte à élargir notre champ d'action si nous en avons le temps.

Langage utilisé

Nous allons programmer ce projet en C++. En effet celui-ci est un langage objet et nous aidera à mieux organiser notre code grâce aux classes, au polymorphisme, la redéfinition et l'héritage. De plus, il s'agit du langage que nous avons le plus pratiqué durant notre cursus.

Les données d'entrée du problème

Les professeurs

Les professeurs vont être définis par :

- un nom
- des disponibilités
- la liste des cours qu'ils peuvent enseigner

Les classes

Les classes vont être définies par :

- un nom

- un nombre d'élève
- la liste des cours qu'elle va suivre

Les cours

Les cours vont être définis par :

- un nom
- un nombre d'heure de cours
- un type de salle dans laquelle il doit être donné

Les salles

Les salles sont définies par :

- un nom
- un nombre de place
- un type à savoir, une salle de cours, une salle de TP spécifique

Les données en sortie du problème

En sortie nous allons avoir les emplois du temps de tous les professeurs, de toutes les classes et de toutes les salles.

Les acteurs

Il existe deux principaux types d'acteur pour notre logiciel :

Premier acteur : Saisie des données

Le rôle du premier acteur est de saisir toutes les données relatives aux professeurs, salles, et élèves, et tous les paramètres nécessaires à une première simulation de l'emploi du temps.

Second acteur : Maintenance de l'emploi du temps

Etablir un emploi du temps est une bonne chose, mais il faut également pouvoir le modifier au cours de l'année si des événements imprévus doivent être rajoutés. Le rôle du deuxième acteur est donc de gérer la maintenance de l'emploi du temps au fur et à mesure que l'année avance. Il doit pouvoir rajouter des événements dans l'emploi du temps des professeurs et/ou des élèves.

Cas d'utilisation

Génération de l'emploi du temps

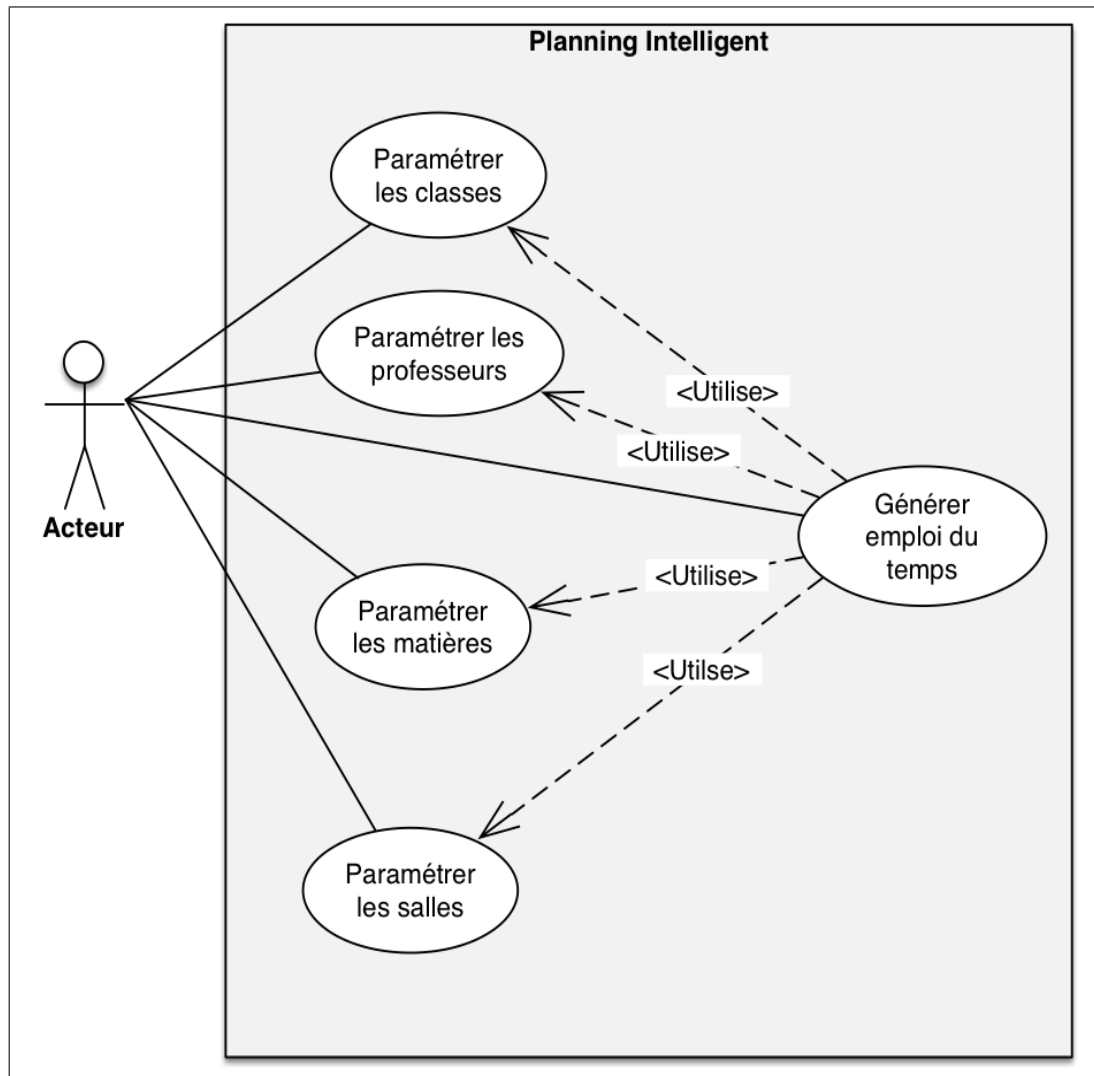


FIGURE 1 – Diagramme cas d'utilisation, génération de l'emploi du temps

Objectif

L'objectif est de créer un emploi du temps à partir de données brutes entrées par l'utilisateur.

Acteurs

L'acteur est celui qui est chargé de la saisie des données.

Données échangées et description des enchaînements

L'acteur en question entre les différentes données propres aux professeurs, classes, salles, et matières. Pour chaque élément, il devra remplir des critères bien spécifiques :

- Pour chaque professeur, il doit entrer les matières que celui-ci enseigne et ses disponibilités dans la semaine.
- Pour chaque promotion, il doit entrer le programme qui devra être dispensé au cours de l'année (matières et nombre d'heures).
- Pour chaque classe, il doit préciser la promotion à laquelle elle appartient.
- Pour chaque salle, il doit entrer leur contenance et leur type (salle de TP info, salle de cours, etc...)

Une fois que toutes les données ont été entrées, le logiciel génère automatiquement tous les emplois du temps de l'école.

Maintenance de l'emploi du temps

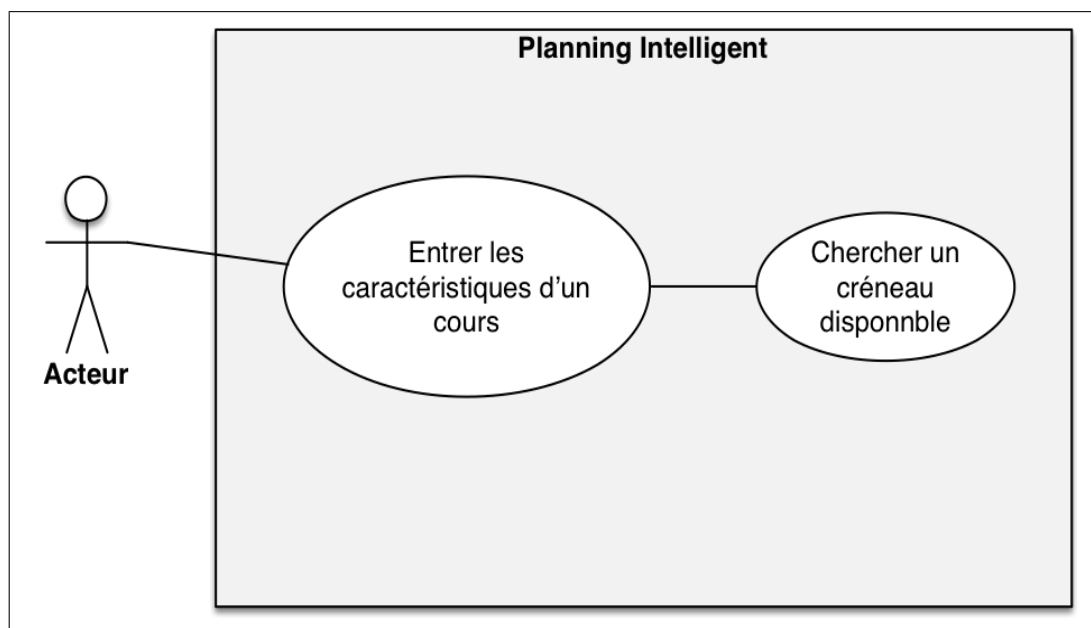


FIGURE 2 – Diagramme cas d'utilisation, maintenance de l'emploi du temps

Objectif

Pour une raison ou pour une autre, l'utilisateur peut être amené à rajouter un évènement dans l'emploi du temps. Le logiciel doit donc le permettre en tenant compte des évènements déjà placés et des contraintes que cela implique, sans avoir à générer à nouveau tout l'emploi du temps.

Acteurs

Le seul acteur à intervenir dans ce cas est celui chargé de la maintenance.

Données échangées

Pour placer un cours dans l'emploi du temps, le programme devra consulter les disponibilités de chaque élément :

- Les disponibilités des professeurs ou des intervenants
- Les disponibilités des élèves
- Les disponibilités d'une salle convenant à l'évènement, et dont la capacité est suffisante pour accueillir tous les élèves.

Description des enchaînements

Pré-condition

Lorsque l'utilisateur voudra rajouter un évènement, il lui faudra connaître les disponibilités de chacune des entités concernées. Il devra également savoir quelles classes sont concernées

par l'évènement, si elles doivent recevoir l'évènement par classe ou par promo, etc...

Séquence

L'utilisateur entre les disponibilités de chacun dans le programme, et celui-ci va chercher les différents créneaux possibles pour placer le cours. L'utilisateur aura alors deux possibilités : soit il laisse le programme placer automatiquement le cours, soit il demandera au programme d'afficher la liste des créneaux disponibles afin de placer manuellement le cours. Celui-ci n'aura qu'à choisir le créneau qui lui semble le plus adapté.

Organisation des données

Nous avons défini la semaine sur 22 créneaux de 2 heures chacun. Avec 2 créneaux le matin et 2 l'après-midi du lundi au samedi matin.

Les professeurs

Les disponibilités du professeurs vont être représenté par un mot binaire de 22 bits. 1 représentant une disponibilité et 0 une indisponibilité. Chacune de ces disponibilités vont être liés à une semaine.

Pour une semaine donnée, lorsque nous ajoutons un cours à un professeur, ses disponibilités changent et cela aura pour conséquence de modifier le mot binaire en faisant passer un bit à 0.

Chaque professeur aura une liste de cours qu'il sera susceptible de dispenser.

Les classes

Chaque classe va avoir une liste de semaines associée. Celles-ci vont être définies par un numéro définissant leur ordre dans l'année, et 22 créneaux assignés que l'on devra remplir avec les cours planifiés à la classe.

Chaque classe aura une liste de cours qu'elle devra obligatoirement suivre.

Méthode de résolution

Pré-traitement

Nous effectuons un pré-traitement pour éviter des calculs qui n'aboutiront pas. Nous comparons le nombre de cours qui doivent être donnés par matière sur une semaine toutes classes confondues, avec le nombre de créneaux disponibles pour l'ensemble des profs donnant cette matière.

- Si le nombre de créneaux total des professeurs est supérieur au nombre de créneaux du cours alors il n'y a pas de pré-traitement à effectuer pour ce cours.
- Si le nombre de créneaux total des professeurs est égal au nombre de créneaux du cours alors on peut directement associer les classes aux créneaux du ou des professeurs
- Si le nombre de créneaux total des professeurs est inférieur au nombre de créneaux du cours alors nous ne pouvons pas aboutir à une solution. Il faut donc ajouter des professeurs ou des créneaux.

Traitement

Dans un premier temps, on trie les professeurs de manière croissante par rapport à leurs disponibilités. Ensuite, nous remplissons de manière gloutonne les emplois du temps, c'est-à-dire qu'on place le cours sur le premier créneau libre.