

DOCUMENT DE LA SÉRIE  
«SPÉCIFICATIONS»  
MODÈLE DE DONNÉES  
**VARUNA**  
  
**VERSION 1301**

## 1. VERSION DU DOCUMENT

Date	Version du document	Intervenant	Action	Validation
01/01/13	Version 0.1 soumise aux inventeurs	M. Sherlock Olm	Rédaction	OUI

## 2. DOCUMENTS RÉFÉRENCÉS

Date	Document	Auteur	Version
01/03/13	Modélisation et résolution Varuna	M. Sherlock Olm	0.1

*Ce document est sous une licence libre (FDL) et peut être diffusé librement et largement sans l'accord explicite de ses auteurs*

### 3. INTRODUCTION

Ce document décrit le modèle de données utilisé pour décrire les objets métiers servant à établir l'emploi du temps d'une structure universitaire ou scolaire.

### 4. SOMMAIRE

1. Version du document.....	1
2. Documents référencés.....	1
3. Introduction.....	2
4. Sommaire.....	2
5. Le contexte.....	2
6. Les composants de varuna.....	3
7. Les objets.....	3
7.1. Autour du cours.....	3
Le cours.....	4
Le type de cours.....	4
La famille de cours.....	5
7.2. Les ressources.....	5
L'enseignant.....	6
La salle.....	6
Les autres ressources.....	7
7.3. Autour des contraintes.....	7
7.4. Autour de la disponibilité.....	8
7.5. Autour du temps.....	9
L'instant.....	9
L'horaire.....	9
Le moment.....	10
Le créneau.....	10
La durée.....	10
La zone.....	10

### 5. LE CONTEXTE

Varuna est un logiciel édité par la société abstract en France entre 1993 et 2006. Son code<sup>1</sup> a été libéré sous licence MIT en 2006 par décision des associés avant la cessation d'activité de la société.

Varuna est un ensemble de programmes permettant de générer de façon automatique ou assistée des emplois du temps pour les grandes écoles et les universités.

Varuna a été installé et utilisé dans de nombreuses universités et grandes écoles françaises : assas, jussieu, joseph fourier, talence, esiee...

Le modèle de données et l'heuristique sont les deux atouts de ce logiciel.

---

<sup>1</sup> <https://github.com/sherlock38/varuna>

## 6. LES COMPOSANTS DE VARUNA

Varuna est organisé en plusieurs composants :

- *varuna moteur*, écrit en lelisp<sup>2</sup>, reposant sur une heuristique inventée par Pascal Rivière et le moteur de propagation par contraintes pecos (le lisp et pecos sont des bibliothèques éditées par Ilog) ;
- *varuna interface*, écrite en lelisp et utilisant les bibliothèques aida et masai (aida et masai sont des bibliothèques éditées par ilog) ; l'interface est composée elle-même de plusieurs modules : réservation de salles, annulation de cours, rattrapage de cours, gestion multi-sites...
- *vgraphe*, un composant écrit en c++ permettant de générer des graphes de contraintes autour des emplois du temps ;
- *varnet*, un composant écrit en c++ d'affichage web des emplois du temps ;
- *vtrans*, un composant écrit en c++ de transfert de données d'une année sur l'autre ;
- *vtraceur*, un composant écrit en c++ de traçage des modifications ;
- *vimport*, *vexport*, des composants écrits en c++ d'import et d'export des données ;
- *vprint*, un composant écrit en c++ de conversion des éditions varuna en postscript générique ;
- *varutils*, des programmes écrits en lelisp utiles pour l'analyse des problèmes pendant la modélisation.

Tous ces composants utilisent le modèle de données décrits ci-dessous.

## 7. LES OBJETS

Nous allons étudier les objets en distinguant :

- le cours, l'objet de la *production*
- les salles, les enseignants, les *ressources* non temporelles consommées pour produire les cours
- les contraintes
- la notion de disponibilité, qui est une liberté ou une contrainte selon le point de vue
- l'utilisation du temps, la ressource « *employée* »

### 7.1. AUTOUR DU COURS

Le cours est l'objet produit par les établissements de type universitaire ou grandes écoles.

---

<sup>2</sup> [http://en.wikipedia.org/wiki/Le\\_Lisp](http://en.wikipedia.org/wiki/Le_Lisp)

## LE COURS

Le cours se décrit par :

Nom attribut	description	F/O
instant-trouve	Instant de débt de cours trouvé par la dernière résolution	F
code	Identifiant du cours, forcément unique	O
libelle	Intitulé du cours	O
duree	Durée du cours en unité de temps (généralement la minute)	O
type	Type du cours, parmi une liste définir par l'utilisateur	O
type-salle	Type de la salle nécessaire	O
audience	Audience attendue du cours	O
nb-seances	Nombre de séances successives du cours (les cours ayant lieu une semaine sur deux sont générés par une famille spécifique avec des contraintes spécifiques)	O
dispo	La disponibilité du cours, c'est à dire les plages horaires pendant lequel le cours peut avoir lieu.	F
fige	Booléen valant true si le cours est figé. Dans ce cas, l'instant trouvé doit être renseigné.	F
salle-figee	Booléen valant true si la salle est figée. Dans ce cas, le champ salle-figee-reelle doit être renseigné.	F
commentaires	Des commentaires sur le cours, ne servant pas pour le moteur	F
salle-trouve	L'identifiant de la salle trouvée après résolution	F
salle-figee-reelle	L'identifiant de la salle figée quand une salle est figée.	F

Par exemple, en format varuna, voici la représentation d'un cours.

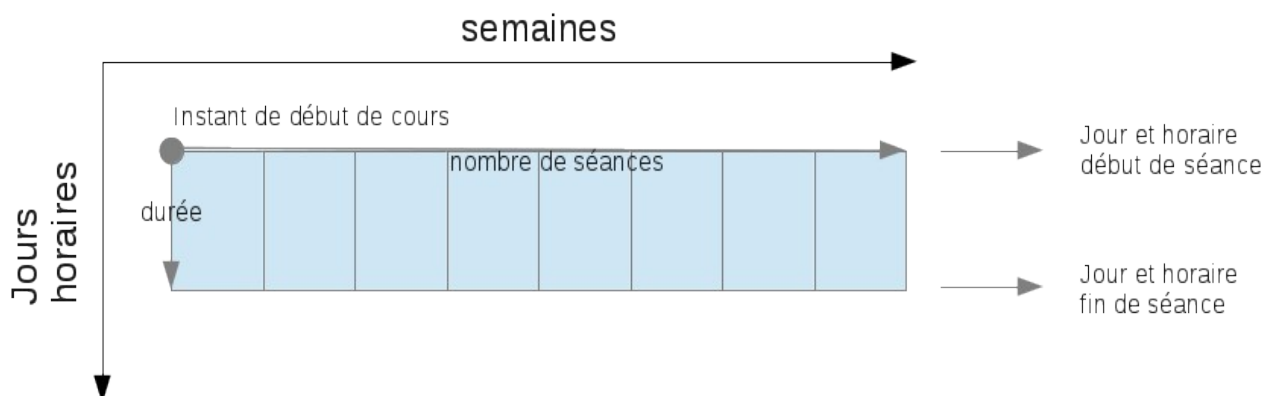
```
(cours
  (instant-trouve 1296)
  (code 'c0)
  (libelle "T3-GENT:COURS:(1à6)")
  (duree 240)
  (type 'cours-1)
  (type-salle 'amphi)
  (audience 70)
  (nb-seances 1)
  (dispo '(((480 . 1200) (1920 . 2640) (3360 . 4080) (4800 . 5520) (6240 . 6960) (7680 . 8340))
(2 . 4)))
  (fige 't)
  (salle-figee ())
  (commentaires '(""))
  (salle-trouvee 's1)
  (salle-figee-reelle 's1)
)
```

Le cours est lié à l'objet salle pour deux attributs : *salle-trouvee* et *salle-figee-reelle*.

On peut définir un cours varuna comme « une séquence de n séances (attribut nb\_seance) de même durée (attribut duree), n et d étant des entiers positifs non nuls. Les séances du cours commencent toutes le même jour de semaine

et la même heure sur n semaines consécutives ».

On peut donc représenter un cours comme une sorte de rectangle



un cours est associé à un rectangle :

- de hauteur la durée en minutes de chacune des séances, appelée durée du cours. Cette hauteur est une valeur discrète.
- de largeur le nombre de séances du cours. Cette largeur est une valeur discrète.

Le rectangle est placé à un instant de début de cours (la notion d'instant est défini plus loin).

**Remarque** : les cours ayant lieu 1 semaine sur N ne sont donc pas des cours varuna mais sont modélisés à l'aide de familles.

## **LE TYPE DE COURS**

Le type de cours permet de catégoriser les cours et affecter à ces catégories des horaires de début de cours différents.

## **LA FAMILLE DE COURS**

La famille de cours est une notion ensembliste récursive.

Sa définition pourrait être : « une famille de cours est un ensemble de cours et de familles de cours, appelées dans ce contexte sous-famille ».

Par exemple, en format varuna, voici la représentation d'une famille de cours.

```
(groupe
  (code 'gr410-st)
  (libelle "CT1-PRB:COURS:(1)")
  (cours '(c9676 c9649 c9648 c9577 c9576 c9551 c9535 c9534 c9400 c9399 c8938 c8937 c7998 c7997
c7974 c7973 c7849 c7848 c7816))
  (groupes ())
  (contraintes ())
  (dispo '(((480 . 1200) (1920 . 2640) (3360 . 4080) (4800 . 5520) (6240 . 6960) (7680 . 8340))
(1 . 48)))
  (type-macro "STANDARD")
  (type-cours ()))
```

```
(type-salle ())
(commentaires '(""))
(nb-seances ())
(audience ())
(durees ())
(profs ())
(nb-td ())
(duree ())
)
```

La famille est utile essentiellement pour trois actions différentes :

- la famille permet de générer des cours, comme une famille de groupes de tds ; la famille est dite *génératrice* ;
- la famille permet de poser des contraintes (implicite ou explicite) sur l'ensemble formés par les cours et les sous-famille ; nous les appelons des familles *naturelles* ;
- la famille permet de regrouper un ensemble de cours sans lien précis avec la modélisation avant génération automatique, c'est à dire sans aucune influence sur l'heuristique. Ces familles ont souvent du sens en dehors de la modélisation ; cela peut-être, par exemple, tous les cours de première année. Nous les appelons des familles *neutres*.

## 7.2. LES RESSOURCES

Pour produire les cours et « employer » le temps, il faut aussi consommer d'autres ressources : des salles, des enseignants et un peu de matériel comme de la craie, des tableaux, des rétroprojecteurs...

### L'ENSEIGNANT

L'enseignant se décrit par :

Nom attribut	description	F/O
code	Identifiant de l'enseignant, forcément unique	O
libelle	Intitulé de l'enseignant	O
civilite	Civilité de l'enseignant	O
cours	La liste des codes de cours enseignés	F
dispo	La disponibilité de l'enseignant, c'est à dire les plages horaires pendant lequel l'enseignant peut dispenser un cours.	O
commentaires	Des commentaires sur l'enseignant, ne servant pas pour le moteur	F

Par exemple, en format varuna, voici la représentation d'un enseignant.

```
(prof
  (code 'p23)
  (libelle "BASKIOTIS D.")
  (civilite "")
  (cours '(c9048 c7943 c7942 c7944 c8546 c8545 c8544 c8543 c6930 c6931 c6932 c6929 c6874 c6873
c6872 c6351 c6823 c6352 c6350 c6349))
)
```

```
(dispo '(((480 . 1200) (1920 . 2640) (3360 . 4080) (4800 . 5520) (6240 . 6960) (7680 . 8340))
(1 . 48)))
(commentaires '(""))
)
```

L'enseignant porte en lui une contrainte *implicite* de « non ubiquité » sur les cours qu'il enseigne.

**Remarque** : nous préférons le mot enseignant à professeur. Comme en 1993, les noms de fichiers étaient sous windows de 8 lettres et qu'il n'y avait pas d'abréviation courte pour enseignant, le mot « professeur » a été choisi.

L'enseignant est donc lié à une liste d'objets cours par l'attribut *cours*.

L'enseignant porte également une autre contrainte implicite par sa disponibilité (attribut *dispo*) qui restreint d'autant les domaines de chacun de ces cours.

## LA SALLE

La salle se décrit par :

Nom attribut	description	F/O
code	Identifiant de la salle, forcément unique	O
libelle	Intitulé de la salle	O
capacite	Capacité en nombre de sièges de la salle, à rapprocher de l'audience du cours	O
type	Le type de la salle, à rapprocher du type de salle attendue pour un cours	O
dispo	La disponibilité de la salle, c'est à dire les plages horaires pendant laquelle on peut dispenser un cours dans la salle..	F
commentaires	Des commentaires sur la salle, ne servant pas pour le moteur	F

Par exemple, en format varuna, voici la représentation d'une salle :

```
(salle
  (code 's5)
  (libelle "Math B")
  (capacite 400)
  (type 'amphi)
  (commentaires '(""))
)
```

Une salle étant une ressource, elle n'est pas liée par elle-même à un autre objet.

La catégorisation proposée (type de salle) est une catégorisation simple, non hiérarchique.

La salle porte en elle une contrainte *implicite* de « non ubiquité » sur les cours produit dans la salle, sauf quelques rares cas (salles se séparant en deux, grande salle, notion de plateau de plusieurs salles sans affectation de salles et recevant plusieurs cours).

## **LES AUTRES RESSOURCES**

Les autres ressources ne sont pas prises en compte dans varuna, notamment le matériel spécifique ou l'organisation hiérarchique des ressources.

Ainsi les notions de matériel sont modélisées si besoin par un type de salle spécifique.

L'organisation en plusieurs sites est proposée mais sa gestion est relativement exogène à la résolution.

**Remarque** : de même, le consommateur de la production, à savoir *l'apprenant*, n'a pas été modélisé dans varuna. Il est caché *collectivement* derrière certaines familles.

## **7.3. AUTOUR DES CONTRAINTES**

La plupart des contraintes sont *implicites* car portées par les objets.

Ainsi, un professeur ne pouvant pas assurer plusieurs cours en même temps (non ubiquité de l'être humain), une contrainte de non-chevauchement s'applique à tous les cours qu'il enseigne.

De même, deux cours ne pouvant avoir lieu dans la même salle en même temps, une contrainte de non-chevauchement s'applique à tous les cours d'une même salle.

**Remarque** : si un cours n'a pas de professeur ou de salle déjà affectés avant résolution, alors ces contraintes ne s'appliquent pas.

Il est possible de poser une contrainte explicite entre plusieurs cours ou au sein d'une famille de cours.

Ces contraintes explicites sont :

- *non-chevauchement*, exprimant le fait que les cours ne se chevauchent jamais (aucune minute en commun sur aucune séance)
- *simultaneite*, exprimant que les cours ont le même début de cours (même jour, même horaire, même semaine)
- *simultaneite-fin-seance*, exprimant que les cours se terminent tous ensemble (même jour, même horaire, même semaine)
- *meme-jour*, exprimant que les séances de ces cours doivent toutes avoir lieu le même jour de semaine
- *jours-differents*, exprimant que les séances de ces cours ne doivent jamais être placées le même jour de semaine
- *meme-horaire*, exprimant que les séances de ces cours doivent toujours être placées au même horaire de début de cours, quelques soient la semaine et le jour de semaine
- *meme-semaine*, exprimant que les cours commencent tous la même semaine



- *precedence*, exprimant que chaque cours doit avoir fini dans l'année avant que le suivant ne commence
- *proximite-semaine*, exprimant que chaque cours démarre dans l'année peu après le cours précédent (au plus n semaines, n paramètre de la contrainte)

Varuna ne définit que ces contraintes explicites.

De fait, les deux dernières contraintes n'ont jamais été testées avec plus de deux cours mais la généralisation à n cours est assez évidente.

**Remarque** : varuna ne définit pas la contrainte chevauchement-maximum sur les cours, qui pourrait s'expliquer en « parmi la liste de ces cours, au plus n peuvent se chevaucher ». Cette contrainte est assez cruciale pour résoudre des emplois du temps avec des ressources limitées mais non encore affectées (professeurs ou salles).

## 7.4. AUTOUR DE LA DISPONIBILITÉ

La disponibilité peut être vue comme une contrainte ou une liberté, selon le point de vue.

La disponibilité s'applique à la plupart des objets de varuna : le cours, la famille, la salle, le professeur, l'école.

Elle se définit toujours comme l'ensemble des périodes de temps pendant les quelles une ressource est disponible ou un cours peut se dérouler.

La disponibilité définit donc le domaine de liberté temporel dans lequel le cours pourra être placé.

La disponibilité dans varuna ne permet pas de définir des niveaux de souhait mais précise très finement les périodes de disponibilité.

La disponibilité se définit comme une liste d'intervalles de semaine croisée avec une liste de créneaux de disponibilité dans la semaine.

## 7.5. AUTOUR DU TEMPS

Le temps est l'objet même de varuna puisque le logiciel se propose de :

1. modéliser l'emploi qu'il en est fait au sein d'une école ou une université
2. d'assister le responsable des emplois du temps dans une génération automatique ou assistée de solution au modèle créé

Le temps est une notion difficile à appréhender notamment car le temps d'y penser, il est déjà passé. L'objet des emplois du temps est justement d'organiser à l'avance son emploi.

Il est souvent représenté par une dimension unique à sens unique mais l'utilisation quotidienne met plutôt en avant des cycles reposant sur des unités et des sous-unités dans un système de mesure non décimal.

La notion pivot est le *jour* qui est à la fois *englobé* et *englobant*.

L'organisation des emplois du temps universitaires repose généralement sur la semaine, regroupée en trimestre, semestre et/ou année.

Le système calendaire *{Année, mois, jour}* est alors souvent remplacé par le système *{Période étalon, numéro de semaine, jour de semaine}*. L'avantage du second système est qu'il repose sur une séquence de séquence de 7 jours différents.

Le jour englobe généralement un système fondé sur *{heure, minute, seconde}* mais il existe aussi *{demi-journée, heure, minute, seconde}*

Dans les emplois du temps modélisés par varuna, le système ne descend pas en dessous de la minute. Varuna retient le système *{heure, minute}*.

Le vocabulaire utilisé est souvent ambigu car :

- il s'emploie sur différents systèmes
- il confond la durée et l'instant.

Nous avons donc eu besoin, pour spécifier varuna, de préciser ces notions, notamment en définissant le vocabulaire.

### **L'INSTANT**

L'instant est un point dans le temps universel, par exemple le 21 juillet 1969 à 02h56 UTC.

L'instant de début de chacun des cours est l'un des objets de la résolution de varuna.

### **L'HORAIRE**

L'horaire précise un point précis temporel dans une journée, par exemple 10h00 ou 23h59. Il est défini dans varuna en heures et minutes.

Il varie entre 0h00 et 23h59.

### **LE MOMENT**

Le moment est un horaire associé à un jour de semaine, par exemple, le mardi 13h35.

### **LE CRÉNEAU**

Un créneau est un intervalle semi-ouvert à droite entre deux horaires, deux moments, deux instants selon le contexte.

### **LA DURÉE**

La durée est la longueur en nombre de minutes d'un créneau. Comme le créneau est semi-ouvert à droite, il n'y a pas d'ambiguïté sur le mode de calcul de la durée.

Ainsi [8h00;9h00[ est un créneau d'horaires de durée 60 (minutes).

## **LA ZONE**

La zone est une liste de créneaux disjoints. Elle peut donc être d'horaires, de moments ou d'instant.