Université Catholique de Louvain
Louvain School of Engineering
Department of Computer Engineering

Automatisation de la gestion des programmes de cours

Auteur
Xavier Crochet

Promoteurs
Kim Mens
Chantal Poncin
Lecteurs
John Smith
John Doo

Mémoire présenter dans le cadre du *Master 120* en *Sciences Informatiques* option *sécurité et réseaux*

Louvain-la-Neuve
Juin 2014

Un grand merci à Chantal
Poncin et Kim mens pour
leur support, conseils et
excellente disponnibilité.

Contents

Chapter 1

Introduction

1.1 Contexte

Chaque année à l'Université catholique de Louvain, et dans toute les autres universités, un étudiant est amené à devoir effectuer des choix au niveau du programme de cours qu'il va suivre. Même si ces choix demeurent plus restreint en bac qu'en master le processus actuel représente une lourde tâche de travail pour le personnel en charge de la vérification de ces programmes.

En bachelier, cela se limite au choix d'une mineure de manière générale. La tâche est déjà plus compliquée lorsqu'il faut traiter le programme d'un étudiant qui recommence son année.

En master par contre, l'ensemble des choix est plus vaste. Le catalogue des programmes est plus versatile (Master 120, Master 60) et plus modulable. En effet, un étudiant doit choisir une ou plusieurs options. De plus, chacune d'entre elles est composée de cours obligatoires et optionnels. Ensuite, pour les programmes étalés sur deux ans, chacun des cours peut être suivi durant la première ou la deuxième année. Enfin, il faut prendre en compte les diverses équivalences lorsqu'un étudiant de notre faculté part en Erasmus, ou lorsqu'un étudiant étranger vient étudier dans notre université. Le processus de validation est donc plus complexe et nécessite considérablement plus de temps pour chacune des deux parties, à savoir la commission de programme du département d'informatique de le l'école polytechnique de Louvain (**La commission INFO**) et ses étudiants.

En outre, la fédération Wallonie - Bruxelles (FWB) a réalisé une réforme des programmes de nos universités. Le but de cette réforme est de proposer des programmes plus à la carte, afin que la structure des formations enseignées chez nous se calque sur celle des grandes universités anglo-saxonnes.

Cette réforme a pour conséquence de complexifier les programmes de master, mais surtout ceux de bachelier, offrant plus de liberté aux étudiants. Alors que le gros du travail se fait actuellement à l'aide d'un formulaire

papier, il est urgent de passer à une version automatique pour simplifier la tâche au personnel et aux étudiants.

Qui plus est, ce type de plate-forme pourrait être un point de départ à d'autre services proposés aux étudiants ainsi qu'au professeurs, pour gérer les horaires des cours par exemple.

1.2 Problème

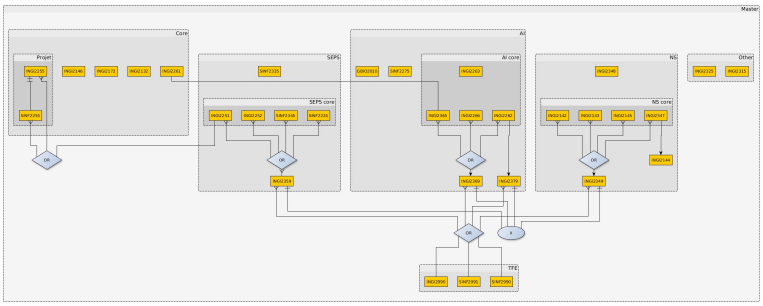
Le problème est, une fois découpé, relativement simple à comprendre. Nous avons deux acteurs; *la commission INFO* et les étudiants. La *commission INFO* propose un catalogue de cours aux étudiants et plusieurs types de contraintes s'exercent sur chacun des cours de ce catalogue. L'étudiant doit se construire un programme de cours à partir du catalogue et *la commission INFO* doit vérifier l'intégrité du programme de chaque étudiant. Il y a donc un processus de *négociation* entre les deux parties, durant lequel *la commission INFO* souligne les erreurs éventuelles pour être, par après, corrigées par l'étudiant jusqu'à la validation de son programme.

Cependant, il faut garder à l'esprit que l'état du catalogue, des contraintes ou même des cours peut évoluer à tout moment. Comme mentionné plus haut, la *FWB* peut émettre de nouvelles réformes, de nouvelles lois dont l'impact peut bouleverser de façon relativement importante la structure des programmes. De plus, il est fréquent de voir certaines contraintes comme les crédits d'un cours, le semestre ou même le cycle durant lequel il est dispensé changer au cours des années. (Une version plus détaillée des différentes contraintes sera présentée dans les sections qui suivent.)

L'interprétation, la connaissance et la vérification de l'ensemble de ces contraintes, qui ne sont parfois pas très explicites, sont à la charge de *la commission INFO*.

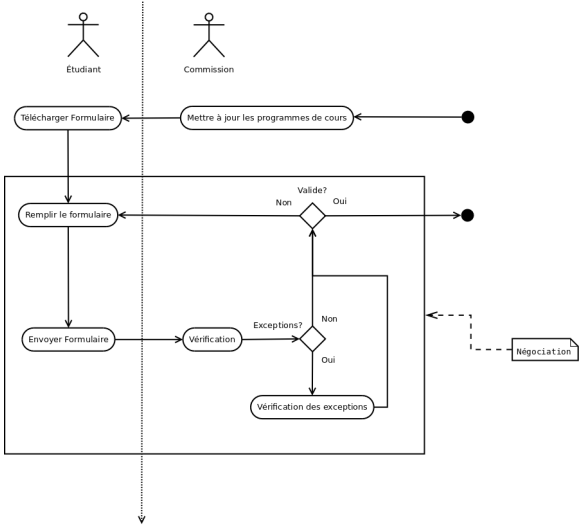
L'image 1.1 représente l'ensemble des cours, modules et contraintes du programme de **master** en informatique. Les nœuds représentent les différents cours, modules et programmes disponibles, tandis que les liens entre ces différents nœuds correspondent aux dépendances entres les cours. La complexité du graphe, malgré qu'il a été retravaillé et simplifié pour augmenter sa lisibilité, est assez évidente.

Figure 1.1: Programme de Master



Présentement, le processus de construction du programme de cours pour un étudiant est assez rudimentaire. Ce processus est représenté sur le diagramme 1.2

Figure 1.2: Processus actuel



La commission INFO commence par mettre à jour ses différents programmes de cours, puis créée formulaire excel contenant le programme de cours et enfin le mets en ligne sur le portail de la faculté (Afin qu'il soit téléchargeable par les étudiants par après)

L'étudiant télécharge le formulaire contenant le programme de cours

Ici commence la phase de négociation entre *la commission INFO* et les étudiants. Les deux parties vont s'échanger un formulaire que l'étudiant va compléter et *la commission INFO* vérifier. L'échange va se poursuivre jusqu'à ce que le programme soit valide.

L'étudiant complète d'abord son formulaire en tenant compte des spécifications du programme. S'il est en bac par exemple, il y a des cours obligatoire à prendre chaque année. S'il est en master, il y a une option à

choisir, des cours obligatoires, etc. Il peut cependant exister des exceptions dans les différents programmes qui permettre d'enfreindre ces spécifications. Par exemple, un étudiant en provenance d'une autre université ou d'une autre faculté peut avoir déjà suivit l'équivalent d'un cours obligatoire dans son université ou sa faculté d'origine. Il va devoir rendre un formulaire qui ne respecte pas les contraintes initiales, justifier ces exceptions et négocier leur acceptation avec *la commission INFO*.

Il envoie son formulaire par mail à *la commission INFO*.

La commission INFO réceptionne le formulaire. Elle vérifie la validité du programme. Si l'étudiant revendique des exceptions dans son programme, elle vérifie si elle sont fondées ou non. Si le programme ou ces exceptions ne sont pas validées, il est demandé à l'étudiant de compléter sont formulaire avec les informations qu'il manque.Le processus de négociation recommence ainsi à zero. Si le programme est valide, il est encodé.

Le processus actuel, que ce soit en bac ou en master, se déroule essentiellement à l'aide d'une feuille de papier et d'un bic. L'étudiant recherches les informations dont il a besoin sur le site de l'UCL, auprès de ses collègues ou encore sur les forums de cours, complète le formulaire et le dépose au secrétariat.

La commission INFO quant à elle, effectue une vérification à la main de ces formulaires, et contacte les étudiants si besoin est. Informatiser ce processus à l'air du web 3.0 est une nécessite plus qu'absolue.

1.3 Motivation

Reprenons le diagramme 1.2 représentant le processus de création de programme tel qu'il se déroule actuellement.

Le mot d'ordre est **automatisation**.

Comme présenté dans la section précédente, certaines étapes, comme la **mise à jour** et la plupart de celles contenue dans le bloc de **négociation**, sont sources de beaucoup de problèmes pour *la commission INFO*.

Avant toute chose, il y a beaucoup d'informations qui sont échangées entre les deux parties, que ce soit implicitement ou explicitement.

Les données implicites correspondent aux informations relatives aux étudiants, comme l'historique de leur parcourt universitaire, qui pourrait par exemple justifier certaines des exceptions mentionnées précédemment.

Les données explicites correspondent aux informations relatives aux programmes de cours mais aussi aux choix faits par les étudiants. Ces données concernent les

- programmes de cours
- les modules et différents options

- cours: leur sigle, le semestre et les années académiques durant lesquels ils sont dispensé, le nombre de crédits, ...

- les contraintes qui agissent sur ces différents programmes, modules et cours.

Il faut aussi garder une trace de ce que l'étudiant a suivi les années précédentes. En effet, il est nécessaire de savoir quel cours un étudiant à réussi l'année précédente pour attribuer les différentes dispenses lorsqu'il recommence son année par exemple.

Il est donc indispensable d'inclure dans la solution une base de donnée pour y stocker toute ces informations, afin qu'elles soit à disposition des deux parties à tout moment.

Différents points du processus actuel 1.2 doivent automatisés.

Le premier point identifié se situe au niveau du support utilisé par l'étudiant et *la commission INFO*, le formulaire Excel, pour ajouter de l'information sur le programme de cours. Tout d'abord, *la commission INFO* doit générer manuellement ce formulaire, en y incluant les cours et différentes options du programme de cours en question. Ajouter les différents blocs, cours et leur crédits respectifs est assez contraignant sur un simple tableur. De plus, beaucoup d'erreurs peuvent êtes laissées par l'étudiant lorsqu'il complète celui-ci. Certes, il y a certains moyen à disposition sous Excel (En utilisant des macros), pour vérifier certaines contraintes du programmes (Comme le nombre de crédits d'un module par exemple) mais ceux-ci peuvent être ignorés par l'étudiant, et doivent de toute façon être revérifiées par après par *la commission INFO*. Le premier point de la solution proposée est donc d'offrir une plate-forme permettant aux acteurs d'échanger ces différentes informations.

Deuxièmement, l'étape de complétion du formulaire par l'étudiant doit être améliorée. Il n'est pas possible de mettre des informations concernant les contraintes autre que numéraires dans le formulaire excel utilisé pour le moment. La plate-forme doit donc inclure des vues représentant de façon claires et concises les différentes contraintes des programmes de cours.

Troisièmement, l'étape de vérification des contraintes, pour valider un programme d'étudiant est coûteuse en temps pour *la commission INFO*, alors que cela ne prendrait que quelques secondes pour un ordinateur. Un module pour vérifier ces contraintes doit être inclus dans la plate-forme

Quatrièmement, il faut s'attaquer au processus de négociation qui amène l'étudiant, en relation avec *la commission INFO*, à construire un programme valide. Tant que le programme de l'étudiant n'est pas valide, le programme de l'étudiant doit être corrigé par *la commission INFO*, celle-ci doit le contacter en pointant les parties non correctes de son programme et l'étudiant doit à son tour comprendre les requêtes de *la commission INFO*, puis tenter de les corriger. Par mail ou par papier, cela peut être très long.

1.4 Objectifs

De manière générale, le but de cette application est d'automatiser la gestion des programmes de cours.

Le *pré-objectif* est d'être indépendant de la base de données EPC. Pour des raisons institutionnelles, l'équipe EPC est surchargée. Ils sont réticents à donner un accès au données. Qui plus est, il n'ont pas le temps de changer leur processus et ils s'intéressent peu, par manque de ressources, à la facilité d'utilisation. Mieux vaux éviter l'utilisation direct de EPC pour le moment, jusqu'à ce que notre outil aie un certain succès. Après quoi, peut être l'outil peut échanger des données avec EPC ou pourrait être intégré, mais ce ne sera pas pour immédiatement. C'est pourquoi il faut pouvoir importer les données de façon efficace et intuitive.

La solution doit être maintenable et évolutive. En effet, la structure des programmes est en constante évolution. De plus, il est probable que la commission de programme découvre de nouveaux besoins qui devront être implémentés à l'avenir. Il est donc primordial de structurer l'application intelligemment pour que celle-ci soit modulaire et qu'on ne doive pas repartir de zero lors de développement ultérieurs.

La *commission INFO* doit pouvoir apporter des catalogues de cours sur l'application. Un catalogue de cours est un ensemble de programmes de cours, contenant les différents modules, cours et dépendances. Un programme de cours est un cursus qu'il est possible de suivre dans la faculté, comme par exemple le programme de MASTER ou celui de BAC.

Les informations des programmes de cours doivent pouvoir être téléchargées depuis l'application ainsi qu'être mises à jour

Les données doivent être visibles de manière synthétique par la commission (vue *admin*)

Il doit y avoir un historique des différentes versions des programmes de cours mis en ligne par la *commission INFO* tout aux long des années académiques,pour gérer l'évolution de ceux-ci et pouvoir permettre aux étudiants (à qui cela est permis) de choisir dans leurs programmes des cours d'anciens catalogues, en cas de report de note par exemple.

Les étudiants doivent pouvoir construire leur programmes. L'application doit leur dire si leur programme est cohérent ou non.

Au niveau de ces contraintes, il doit y avoir une certaine souplesse. Il n'est pas possible d'avoir une vision *manichéenne* à ce niveau

Il doit être possible aux étudiants d'attirer l'attention sur certaines parties de leur programme en y ajoutant un commentaire pour poser une question, ou pour justifier un choix.

En tant qu'étudiant il doit être possible de

- se créer un compte utilisateur avec mon adresse mail UCL

- sélectionner la version du catalogue de cours avec laquelle je vais travailler/
- se créer un programme en choisissant un des programmes de cours disponibles à suivre.
- choisir les différents modules de cours à suivre. (Option, tronc commun, ...)
- configurer son programme par année académique, en choisissant les cours que l'on va suivre durant les différents semestres.
- voir les contraintes qui ne sont pas respectées. Par exemple, le nombre de crédits manquants pour valider un module, ou encore les dépendances d'un cours
- pouvoir soumettre à la validation son programme, même s'il ne respecte pas toutes les contraintes.
- pouvoir communiquer avec la commission info à travers l'application. Par exemple justifier une contrainte non respectée par écrit. ("J'ai déjà suivit un cours très semblable durant mon cursus dans la faculté X à l'université Y)

La *commission info* doit pouvoir

- importer les différents programmes de cours dans l'application
- mettre à jour les données relatives à ces programmes
- être notifié lorsqu'un étudiant envoie son programme à la validation
- accéder aux programmes des étudiants.
- communiquer avec les étudiants à travers l'application.
- marquer les années précédentes des étudiants comme réussies ou ratées.

Chapter 2

Énoncé du problème

La gestion des programme de cours n'est pas une chose aisée. La situation est complexe essentiellement pour deux raisons.

1. La *commission INFO* s'occupe d'un nombre assez élevés de programmes
2. Il existe des contraintes de différentes sortes qui restreignent les étudiants dans les choix qu'ils peuvent faire lorsqu'ils configurent leur programme de cours

Ces deux points vont être présentés en détail dans les sections qui suivent.

2.1 Programmes proposés

La liste des programmes proposé dans le département d'informatique est la suivante:

Bachelier en sciences informatiques - SINF1BA [**?**] - C'est un programme de 180 crédits. Comme dans tout programme de bac, l'étudiant est amené à devoir choisir une mineure dans ce programme. Une mineure intitulée *Approfondissement en sciences informatiques* en est aussi disponible pour les étudiants qui suivent ce programme. La durée de ce programme est de trois ans.

Bachelier en sciences de l'ingénieur, orientation ingénieur civil - FSA1BA (Majeure & Mineure en informatique) - [**?**]. Ici il n'est pas question du programme FSA1BA dans son entièreté mais de la majeure ou mineure que l'étudiant en sciences de l'ingénieur est amené à choisir lorsqu'il suit se programme

Master [120] en sciences informatiques - SINF2M [**?**] - Ce programme est destiné aux étudiants en provenance du programme *SINF1BA*. Il comporte un module obligatoire (le tronc commun) et la possibilité de choisir une ou plusieurs modules optionnels (Génie logiciel, systèmes de programmation, intelligence artificielle, réseaux et sécurité). La charge du travail de fin d'étude est de 28 crédits. La durée de ce programme est de deux ans

Master [60] en sciences informatiques - SINF2M1 [?] - Ce programme est destiné aux étudiants en provenance du programme *SINF1BA*. Il comporte un module obligatoire (le tronc commun) la possibilité de choisir quelques cours au choix mais pas d'options. La charge du travail de fin d'étude est plus petite que celle de son homologue *SINF2M*: 15 crédits.

Master [120] : ingénieur civil en informatique - INFO2M [?] - Ce programme est destiné aux étudiants en provenance du programme *FSA1BA* ayant suivit la mineure ou la majeure en informatique. Il comporte un module obligatoire (Le tronc commun) ainsi que la possibilité de choisir un ou plusieurs modules optionnels (Génie logiciel, systèmes de programmation, intelligence artificielle, réseaux et sécurité). La charge du travail de fin d'étude est de 28 crédits. La durée de ce programme est de deux ans.

Année d'études préparatoire au master en sciences informatiques - SINF1PM [?] - Ce programme est destiné aux étudiants en provenance de Hautes écoles d'informatique. Il permet d'accéder aux programmes *SINF2M* et *SINF2M1*. C'est un programme *à la carte* qui dépend du *background* de l'étudiant. Les cours sont choisi parmi ceux proposé dans le programme *SINF1BA*. Ce programme affiche entre 46 et 60 crédits. La durée de ce programme est d'un an.

Outre ces programmes, *la commission INFO* doit s'occuper du cas des étudiants erasmus. La principale difficulté, que cela soit un étudiant immigrant ou émigrant, est de trouver des équivalences entre les cours suivit dans université d'origine et ceux proposé à ucl ou l'inverse.

De plus, les intersections entre ces cours sont nombreuses. Beaucoup de cours sont disponibles pour une partie voir la totalité des programmes cités ci-dessus.

2.2 Contraintes

Les contraintes sont un point important du problème. En plus d'être nombreuses et diversifiées, elles requièrent beaucoup de travail au niveau de leur vérification du coté de *la commission INFO*. En outre, elles sont difficiles à exprimer avec les moyens mis à la disposition de l'équipe. La conséquence directe de ceci est qu'il est difficile pour les étudiants de comprendre le pourquoi du comment de ces contraintes. Ils n'en tiennent donc pas compte à 100% lorsqu'ils construisent leur programme de cours.

Voici une liste qui en présente brièvement les différentes sortes que l'on peut rencontrer.

1. **Dépendances entre les différents cours** - Ces contraintes sont de deux types :

(a) Les prérequis : Les prérequis d'un cours sont les cours qu'il faut avoir réussi (dans des années académiques antérieure) afin de pouvoir suivre ce cours

(b) Les corequis : Les corequis d'un cours sont les cours qu'il faut avoir suivit **au plus tard** durant la même année académique que ce cours.

L'image 2.1 illustre ce type de contrainte. On peut voir que le cours *INGI2365* a pour prérequis le cours *SINF1121* et pour corequis le cours *INGI2261*. Il est donc nécessaire d'avoir **validé** *SINF1121* ainsi que de suivre (au plus tard) **durant la même période** le cours *INGI2261* (s'il n'a pas été suivit précédemment) pour avoir accès à *INGI2365*.

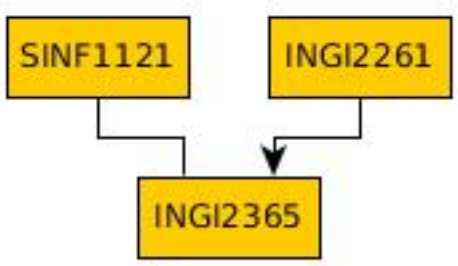


Figure 2.1: Dépendances du cours INGI2261

2. **Contraintes induites par les programmes** - Ce sont les différents cours ou ensembles de cours qu'il est obligatoire de suivre avant de valider un programme. En master, il y a, par exemple, le module intitulé *Tronc Commun* qu'il est obligatoire de suivre, ainsi que le mémoire. Certains modules optionnels, comme les options de master, sont constitués de sous-modules dont il est obligatoire de suivre la totalité des cours qu'ils contiennent.

3. **Contraintes temporelles** - Ce sont les contraintes les plus basiques. Elle représente la période de temps durant laquelle il est possible de suivre le cours en question. Initialement, elles sont exprimés en terme de semestre. Pour des raisons variables, comme un professeur qui part à la retraite, ou qui prend une année sabbatique, elle peuvent très bien s'exprimer en terme d'années académiques. Un autre exemple sont les cours bisannuels qui sont des cours se donnant deux fois sur l'année.

4. **Contraintes sur les propriétés** - Ces contraintes portent sur les propriétés des cours, programmes ou modules. Principalement, elles portent sur les crédits minimum et maximum d'un programme ou d'un module.

L'objectif est de développer une application pour que la charge des différents problèmes présentés plus haut soit à la charge d'une machine. L'idée est de pouvoir

- Importer et mettre à jour les données relatives à ce catalogue de cours.
- Permettre aux étudiants de conserver un historique des cours et programmes qu'ils ont déjà suivi au cours des années précédentes
- Visualiser ces données, que l'on soit étudiant ou membre de la commission de programme.
- Effectuer une vérification en temps réel de la validité des programmes de cours créés par les étudiants
- Permettre à la commission de communiquer avec les étudiants via l'application. Pour par exemple attirer l'attention de l'étudiant sur une partie de son programme qui ne semble pas cohérente, ou dans l'autre sens, demander des explications à la commission sur certains points.
- Porter l'application aisément en production (Via heroku par exemple)

De manière plus générale, le but de ce mémoire est de développer une application conviviale, maintenable et évolutive afin qu'elle puisse être utilisée par les étudiants et la commission de programme.

Ce mémoire sera typiquement un projet que pourrait rencontrer un informaticien dans la vie active, où la *commission INFO* joue le rôle de client, et le promoteur, celui de chef de projet.

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography