

IA01 : Conduite d'expertise d'un SE d'ordre 0+

TP03 - 28/11/2017 - Le Gauche Valentin & Danous Natan

1 Choix de notre problématique et source d'expertise

Nous avons choisi de bâtir un système expert d'ordre 0+ permettant de savoir si un étudiant ira ou n'ira pas en cours. Pour cela, nous nous sommes d'abord basés sur des articles afin de connaître les raisons d'absence de motivations de certains étudiants et fournir un cadre général à notre étude. Ensuite, nous avons élaboré un sondage pour adapter notre expertise au cadre UTCéen. Vous trouverez le détail du sondage ainsi que les articles consultés en annexe. Les questions de ce sondage sont basées selon les sources d'expertises trouvées sur internet.

Nous avons fait le choix de bâtir notre propre source d'expertise afin qu'elle puisse être plus proche de la réalité UTCéenne. Ainsi notre source d'expertise rassemble ce que l'on a pu déduire des articles scientifiques et certaines caractéristiques propre à l'UTC. Un exemple concret, dans l'article [1], l'une des raisons pour ne pas aller en cours est qu'un étudiant a à sa disposition des ressources autres que le cours magistral. Dans le sondage nous demandons donc au étudiants qui ne vont pas en cours si ils disposent de ressources autres que le cours magistral, si oui, nous demandons des précisions : polycopié, Moodle, Memora, Mattermost ... De même, si la raison principale pour laquelle un étudiant ne va pas aller en cours est qu'il sort la veille, nous pouvons relier à la réalité UTCéenne en lui demandant si il est allé au Pic, à une "estu" etc Cela nous permet de cibler précisément ce qui peut décourager un étudiant à aller en cours.

2 Formalisation

- Modélisation d'un fait : (`opérateur attribut valeur`). Par exemple, (`= contenu theorique`)
- Modélisation d'une règle : (`identifiant liste_de_premisses conséquent`). L'identifiant est un symbole, la liste des prémisses est une liste de faits, le conséquent est un fait. Une liste de prémisses équivaut à la conjonction des faits qui la compose. Nous traitons la disjonction en séparant les faits et donc en créant plusieurs règles. Voici l'exemple d'une règle : (`R51 ((= salle amphi) (= contenu theorique)) (= type coursMagistral)`).
- Modélisation de la base de faits : (`fait1 fait2 ... faitn`). La base de faits est une liste de l'ensemble des faits dont nous disposons. Nous l'implémentons sous forme d'une variable globale : (`defparameter *BF* (list '(= contenu theorique) '(= tailleSalleDuCours 100) ...)`).
- Modélisation de la base de règle : (`regle1 regle2 ... reglen`). Nous procédons de la même manière que pour la base de faits : (`defparameter *BR* (list '(R1 ((= sortie PIC)) (>= heureCouchage 0)) '(R2 ((= sortie evenementUTC)) (>= heureCouchage 0)) ...)`).

3 Programmation du moteur d'inférences

Le chaînage le plus logique du moteur d'inférences est un chaînage avant. En effet on cherche à savoir, à partir d'une base de faits, si un utilisateur va aller en cours ou non. C'est un moteur qui recherche en largeur.

3.1 Fonctions outils

Plusieurs fonctions outils ont du être faites avant de programmer le moteur d'inférences. La plus intéressante et utile est la fonction **declenchable**. Elle même repose sur la fonction **eval-fait** qui évalue un fait face à son prémisses supposé de même type (on teste le type dans **declenchable**). Avec un prémisses donné, **declenchable** parcourt une base de faits et si elle trouve un fait de même type, elle l'évalue avec le prémisses et renvoie ce résultat.

Pour exemple, prenons comme prémisses (\geq **tailleSalleDuCours** 50) et comme base de fait, ((= **heure** 11) (= **tailleSalleDuCours** 100) ...). La fonction va donc trouver lors de sa deuxième itération l'attribut **tailleSalleDuCours** qui est le même que celui du prémisses. Elle va donc appeler la fonction **eval-fait** qui va détecter l'opérateur \geq et qui saura correctement faire l'évaluation $100 \geq 50$. La fonction **declenchable** retournera donc T.

Une règle déclenchable est donc une règle qui est vérifiée par un fait de la base de faits.

3.2 Moteur d'inférences

Le moteur d'inférences que nous avons programmé fonctionne de manière itérative. On l'appelle au début avec le but final recherché (par exemple, ici (= **allerEnCours** NIL)). On effectue une copie de la base de règles et de la base de faits. Si le but recherché est présent dans la base de faits, on a finit, l'utilisateur risque fort de ne pas aller en cours. Sinon, on parcourt la base de règles, on parcourt les prémisses de la règle courante, si l'un des prémisses de la règle courante n'est pas déclenchable, les prémisses de la règle ne sont pas validés. Si les prémisses sont valides, la règle est stockées dans la mémoire de travail EC et elle est retirée de la copie de la base de règles.

Si EC est non vide, on retire de EC la dernière règle mise puis on ajoute à BF la conclusion de cette règle. Ensuite la boucle recommence jusqu'à ce qu'on atteigne une valeur de retour. Si EC est vide, c'est que l'on a trouvé aucune règles qui puisse être déclenchée avec les faits dont nous disposons, on ne peut donc pas conclure sur le but.

Scénario d'utilisation :

```
BF : ((= CONTENU THEORIQUE) (= TAILLESALLEDUCOURS 100) (= HEURE 11)
      (= HEURECOUCHAGE 1))
```

Le moteur détecte trois règles :

```
R51 : (((= SALLE AMPHI) (= CONTENU THEORIQUE)) (= TYPE COURSMAGISTRAL))
R50 : (((>= TAILLESALLEDUCOURS 50)) (= SALLE AMPHI))
R49 : (((<= HEURE 10) (>= HEURECOUCHAGE 0)) (= ETAT FATIGUE))
```

Puisque le moteur parcourt en largeur, il peut directement ajouter R51 grâce à la BF et le résultat de R50 :

```
BF : ((= TYPE COURSMAGISTRAL) (= CONTENU THEORIQUE) (= TAILLESALLEDCOURS 100)
      (= HEURE 11) (= HEURECOUCHAGE 1))
```

R51, R50 sont successivement supprimés de la BF car elle ont été déclenchées, leur conclusions sont ajoutées à la BF :

```
BF : ((= SALLE AMPHI) (= TYPE COURSMAGISTRAL) (= CONTENU THEORIQUE)
      (= TAILLESALLEDCOURS 100) (= HEURE 11) (= HEURECOUCHAGE 1))
```

On déduit (= ETAT FATIGUE) de R49 que l'on supprime de EC. Grâce à cette conclusion et le fait (= TYPE COURSMAGISTRAL) on peut déclencher R52 qui mène à (= VOLONTE NIL). Cela nous permet de déclencher R53 qui mène à (= ALLERENCOURS NIL), notre but est atteint, il se trouve dans la BF, le moteur d'inférences retourne :

```
"Vous risquez fort de sécher ce cours"
```

4 Conclusion

4.1 Pistes d'améliorations

Résultat non booléen

Il a été proposé lors de la présentation orale de pondérer les règles afin d'obtenir non pas un résultat booléen comme présenté ici mais un pourcentage, "Vous avez 30% de chance de ne pas aller en cours". Cela aurait en effet été plus intéressant pour l'utilisateur et plus réaliste.

Nous avons néanmoins choisis de ne pas intégrer cette idée afin de rester fidèle à notre source d'expertise. En effet, nous ne pourrions pas attribuer un "poids" à une règle sans "tricher" vis à vis des données et des résultats de sondage dont on dispose. Si nous avions eu cette idée avant de faire le sondage auprès des étudiants, nous aurions pu en conséquence, adapter le sondage.

Interaction avec utilisateur

Un autre point d'amélioration serait de développer un mini programme qui constituerait une base de fait en fonction de questions posées à l'utilisateur.

4.2 Réflexions finales

Malgré certains points d'amélioration énoncés plus haut, nous sommes tout de même content d'avoir pu mener ce projet jusqu'aux objectifs que nous nous étions fixés. De plus nous pensons que ce projet pourrait véritablement être utile aux étudiants d'abord, afin qu'ils prennent plus conscience des raisons pour lesquelles ils ne vont pas en cours. Il pourrait éventuellement être utile aux professeurs afin qu'ils puissent mieux cibler les raisons pour lesquelles leurs étudiants ne vont pas à leur cours.

L'objectif de constituer notre propre source d'expertise a été rempli (Plus de 500 réponses). Il y eu une certaine curiosité chez les étudiants et certains nous ont demandé de publier les résultats. Nous savions dès le début, que nous devions mener un sondage auprès des étudiants sur les raisons de

leur absentéisme. En effet, les articles que nous avons trouvés ne nous permettaient pas à eux seuls d'avoir un sujet suffisamment riche, de plus nous voulions considérer certaines caractéristiques propre à l'UTC. Les résultats ont été très intéressent et ont confirmé les articles cités en Annexe.

5 Annexe

5.1 Articles

[1] : Friedman, Paul, Fred Rodriguez, and Joe McComb. "Why Students Do and Do Not Attend Classes: Myths and Realities." (https://teach.its.uiowa.edu/sites/teach.its.uiowa.edu/files/docs/docs/Why_Students_Attend_Class_ed.pdf)

[2] : Class120. "College Students Reveal Why They Skip Class In 140 Characters Or Less." (<https://www.prnewswire.com/news-releases/college-students-reveal-why-they-skip-class-in-140-characters-or-less-300125051.html>)

[3] : Tom Clay and Lori Breslow. "Why Students Don't Attend Class." (<http://web.mit.edu/fnl/volume/184/breslow.html>)

[4] : Nichole Pulgar. "Why Do Students Skip Class?" (<http://wgssgmn.com/why-do-students-skip-class/>)

5.2 Sondage et résultats

Suivre le lien ci-après et cliquer sur « afficher les réponses précédentes » :

<https://goo.gl/forms/jjZr6KCUMQnNAupL2>.