

IA01 – TP3 – Rapport

15 novembre 2020 – A20 – Branly Stéphane – Fall Mohamed

Contents

Avant-propos	2
Expertise	2
Représentation des connaissances	3
Fonctionnement	4
Avantages et limites de ce Système Expert	9
Conclusion	10

Avant-propos

Dans cette situation particulière, vous cherchez de nouvelles activités dans le but de passer le temps, faire de nouvelles connaissances (avec le respect des gestes barrières) et vous changer les idées. Mais il se trouve qu'il existe un nombre incalculable d'activités possibles. Vous avez du mal à trouver une idée d'activité tout en respectant vos critères. Et si l'on vous disait qu'il existait un guide vous permettant de rendre le choix de votre activité aussi facile que possible ?

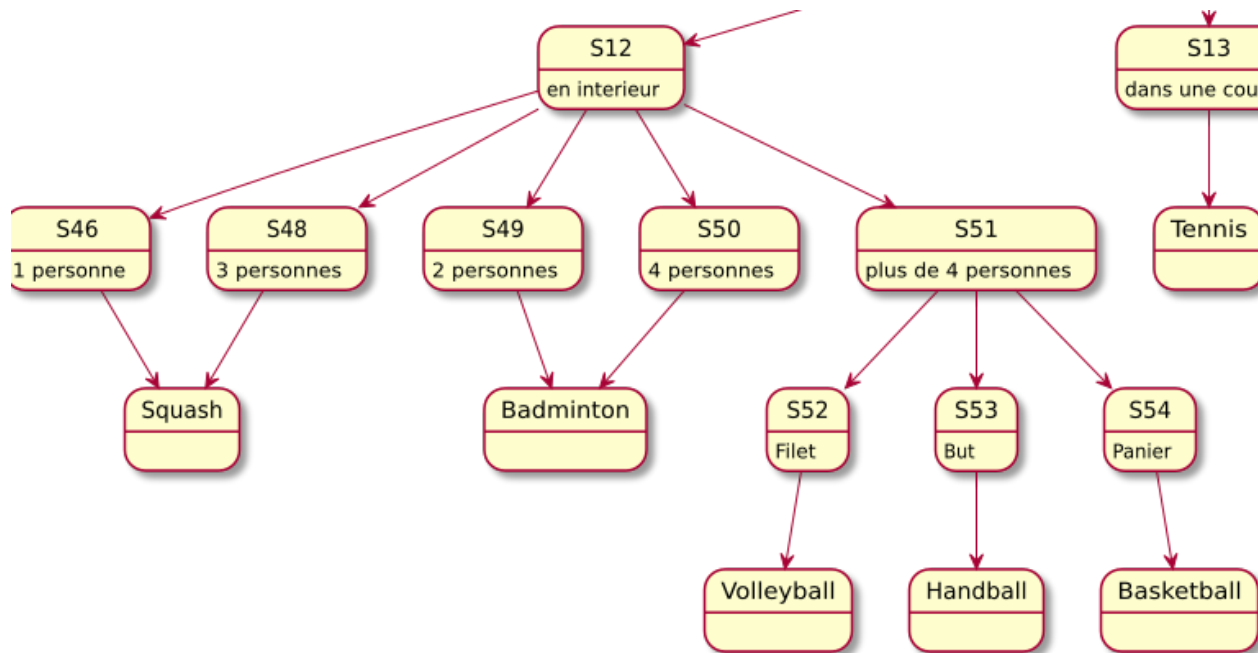
Le but de ce TP a donc été de réaliser un système expert dans ce but précis. Un système expert est un outil conçu pour simuler le savoir-faire d'un spécialiste, dans un domaine précis et bien délimité, grâce à l'exploitation d'un certain nombre de connaissances fournies explicitement.

Le SE que nous avons conçu est d'ordre 0+, c'est à dire que les faits permettant de valider les règles sont des couples attribut-valeur.

Expertise

Notre système expert a pour objectif de mener un élève vers l'activité qui lui est adaptée au fur et à mesure qu'il détaille ses préférences qui lui sont demandées de manière dynamique.

Pour ce faire, il fallait répertorier un ensemble d'activités classées en différentes catégories. Nous avons donc recherché les disciplines qui pouvaient correspondre au contexte scolaire, avec comme distinction initiale les occupations physiques et celles de nature exclusivement intellectuelles. Après avoir constitué l'ensemble des activités, on obtient une arborescence de catégories et sous-catégories, représentant avec précision une expertise en matière de classification des différentes activités (à retrouver ci-dessous). Après avoir filtré les préférences de l'utilisateur, le système expert retourne l'activité correspondant à ces critères.



extrait de l'arbre d'activités (arbre complet en annexe).

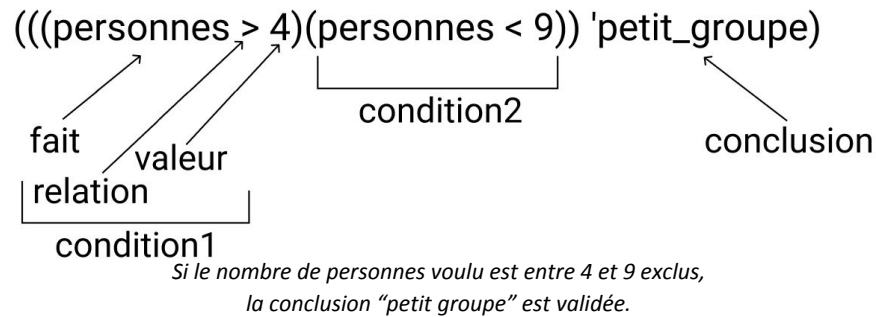
Pour établir notre catalogue d'activités sportives et les sous-types, nous nous sommes principalement aidé de cette source :

- <http://web.ftvs.cuni.cz/eknihy/jazyky/terminologiedesportpourlesetudiantseneps/Texts/3-Classification.html>

Représentation des connaissances

Base de règles:

Pour la base de règles de notre système expert, nous avons opté pour la forme d'une liste de règles (*R1 R2 R3 ...*). Chacune des règles étant sous la forme ((*condition1 condition2 ..*) *conclusion*). Une condition est elle-même sous la forme (*variable_fait relation valeur*), une règle n'est ainsi remplie que quand toutes ces conditions sont remplies. La conclusion sera ainsi validée.



Nous pouvons noter qu'une règle ne contient que des conditions liées par un ET et se terminant par une seule conclusion. Néanmoins, cela n'empêche pas d'avoir des OU en prémisse et des ET en conclusion.

Ainsi :

- si A ou B **alors** C se traduira par 2 règles : **si A alors C** ; **si B alors C**
- si A **alors** B et C se traduira par 2 règles : **si A alors B** ; **si A alors C**

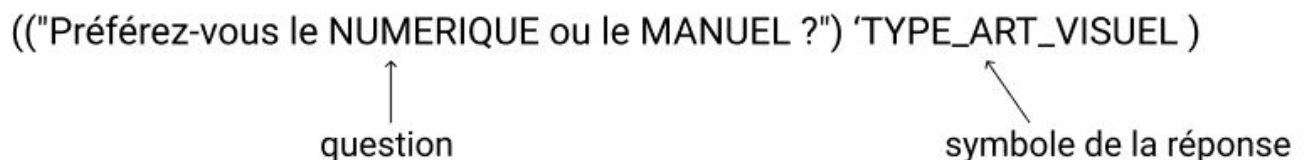
Base de faits:

contient des faits dans R (dans la valeur du symbole correspond à la valeur répondue par l'utilisateur). Les faits valeurs sont uniquement ajoutés par réponse à une question et des faits Booleans (leur présence dans la base de faits correspond à TRUE). Les faits booléens sont uniquement ajoutés par activation de règles.

Base de questions et d'activités:

Nous avons décidé de créer une base de questions ainsi qu'une base d'activités, ces deux entités permettront au système expert d'affiner le nombre d'activités en fonction de nos réponses aux questions.

Un élément de la base de question (Q1 Q2 ...) est sous la forme:



Un élément de la liste d'activité (A1 A2 ...) se présente sous la forme:



On notera que pour déduire une activité, des conditions doivent être validées, celles-ci peuvent être liées par des ET et des OU (distingués par une profondeur différente de liste).

Fonctionnement

Fonctions de services

Sur les règles:

- ***add-rule***: prend des conditions ainsi qu'une conclusion. Une liste est générée et ajoutée à la base de règles.
- ***get-conditions-rule***: accès aux conditions.
- ***get-conclusion-rule***: accès à la conclusion.
- ***desactive-rule***: supprimer une règle de la base de règle (utilisée quand une règle ne peut plus être validée (au moins une condition fausse, ou lorsque la règle a été exécutée)).
- ***execute-rule***: valider la règle (i.e. ajouter la conclusion à la base de faits).
- ***check-condition***: effectuer le test de la condition d'une règle.

Sur les questions:

- ***add-question***: prend une question et une réponse et ajoute aux questions la liste générée.
- ***get-question***: accéder à une question.
- ***get-response-var***: accéder à une réponse.
- ***ask-question***: affiche la question et insère dans la base de faits le symbole de la réponse ayant pour valeur celle renseignée par l'utilisateur).

Sur les activités:

- ***add-activity***: ajouter une activité à la liste d'activité.
- ***ask-activity*** : permet de retrouver une activité dans la base en demandant à l'utilisateur

- **get-conditions-activity**: accéder aux conditions à valider de l'activité.
- **get-activity-activity**: accéder au nom de l'activité.
- **get-description-activity**: accéder à la description de l'activité.
- **desactive-activity**: supprimer une activité de la liste d'activité (est utilisée lorsqu'elle n'est plus atteignable (au moins une condition fausse)).

Inférence

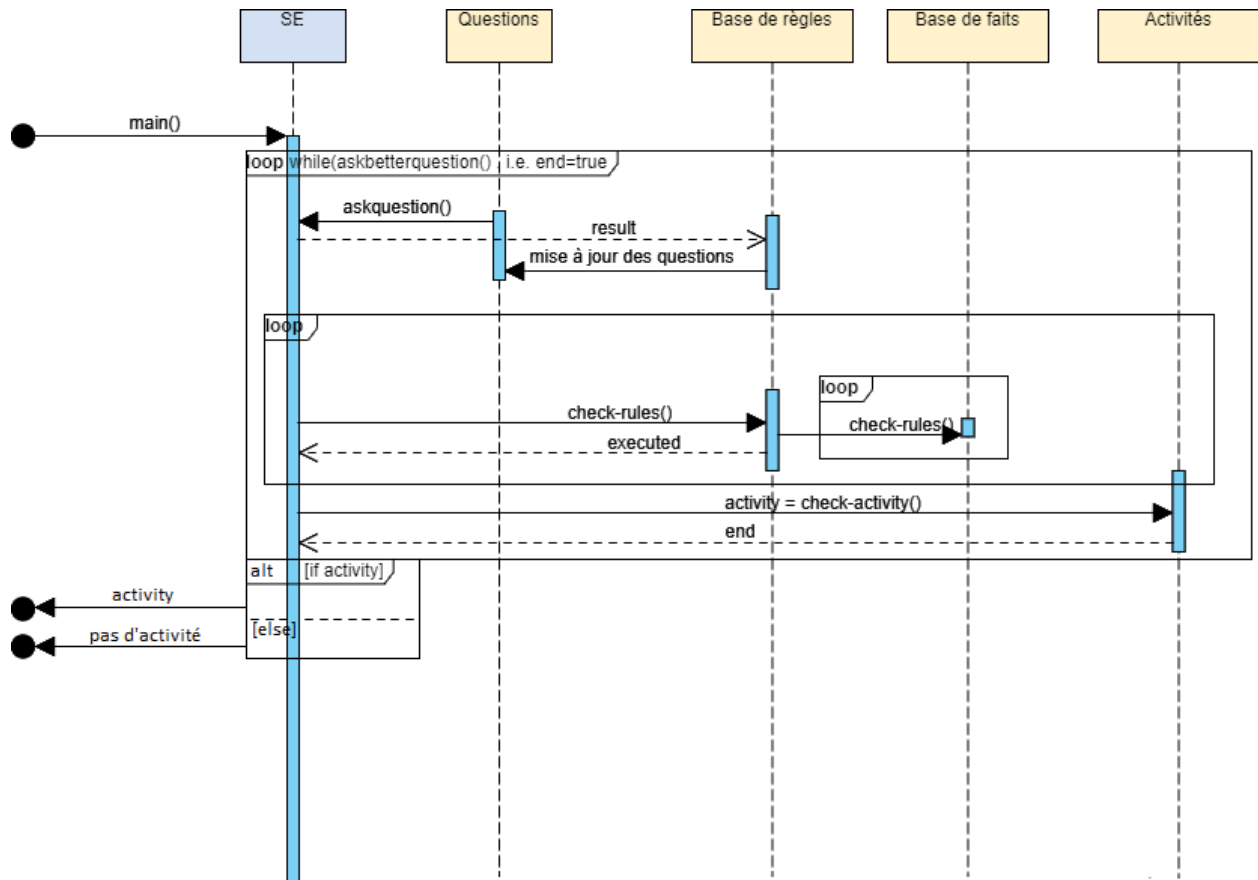
Toute l'organisation de ces structures de données nous permet de créer un moteur d'inférence capable de trouver le prochain critère à déterminer pour occasionnellement repérer la sous-catégorie suivante d'activités.

Après avoir établi la base de faits, de règles, les activités et les questions, le moteur d'inférence par chaînage avant est lancé à partir de la fonction chaînage-avant. Celle-ci continue de poser les questions jusqu'à trouver l'activité ou jusqu'à l'échec (pas d'activité trouvée, ou plus assez de questions/paramètres pour différencier les activités). Il s'agit d'un parcours en largeur étant donné que l'on regarde toutes les règles que l'on peut activer avant de parcourir à nouveau toutes les règles si au moins une a été activée.

Principales fonctions et déroulement du chaînage-avant:

- **ask-better-question**: continue à être appelée tant qu'il y'a des questions disponibles et qu'une unique activité n'est pas trouvée. On parcourt l'ensemble des activités, et pour chaque condition d'une activité on incrémente un compteur attaché à la question qui valide la règle rattachée à la condition (pour *intellectuelle*, par exemple, la règle qui lui est rattachée est celle concluant *type-activité*). La question avec le compteur le plus élevé est celle dont la règle à valider est le plus commun dénominateur de toutes les activités présentes, et ainsi celle qui sera posée (avec *ask-question*).
- **check-rules**: continue de vérifier toutes les règles jusqu'à avoir validé celles dont les conditions sont remplies, et supprime celles qui ne pourront plus être activables (au moins une condition ET est fausse).
- **check-activites**: mise à jour de la liste d'activité en ne gardant que celles présentant des conditions actuellement possibles. S'il ne reste qu'une seule activité, celle-ci est renvoyée et sera considérée comme le résultat final que le système expert renverra à l'utilisateur.

Diagramme de séquence:



Examples:

```

CG-USER(26): (chainage-avant)
"Vous cherchez une activité pour combien de personnes ?"
> 5
"Quel type d'activité cherchez-vous ? (intellectuelle ou physique)"
> PHYSIQUE
"Quel type d'activité physique cherchez-vous ? (sur TERRAIN, en INTERIEUR (et cours), SPECIALE (aquatique et hivernale) ou en EXTERIEUR)"
> TERRAIN
"Indiquez le nombre d'heures que vous pouvez consacrer à l'activité (par session)"
> 3

#####
Nous avons trouvé une activité qui pourrait vous convenir !
Il s'agit de l'activité "Baseball"
"Défoulez-vous avec une bonne partie de baseball !"

```

Principales fonctions et déroulement du chaînage-arrière:

Nous avons également décidé d'implémenter une fonctionnalité de chaînage arrière. Celle-ci permet de vérifier si une activité choisie correspond bien aux critères de l'utilisateur.

- **ask-activity** : nous permet de retrouver l'activité dans la base d'activité à partir d'une entrée de l'utilisateur

Ensuite nous parcourons toutes les conditions et posons les questions associées pour vérifier les faits. La boucle s'arrête quand toutes les conditions sont valides ou quand au moins une est non valide. Se traduisant ainsi par une validation ou non de l'activité répondant aux critères de l'utilisateur.

Exemple:


```

CG-USER(19): {chainage-arriere}
Indiquez le nom de l'activité dont vous voulez vérifier si elle vous convient
> "Cyclisme"
"Quel type d'activité cherchez-vous ? (intellectuelle ou physique)"
> Physique
"Quel type d'activité physique cherchez-vous ? (sur TERRAIN, en INTERIEUR (et cours), SPECIALE (aquatique et hivernale) ou en EXTERIEUR)"
> EXTERIEUR
"Vous cherchez une activité pour combien de personnes ?"
> 1

#####
L'activité que vous avez indiqué semble vous convenir !

```

On remarquera que le chaînage arrière possède presque les mêmes fonctions que le chaînage avant. Néanmoins, la complexité en est réduite étant donné que nous avons uniquement 1 seule activité dans la base d'activités retenue (celle indiquée par l'utilisateur). Les conditions d'arrêt de la boucle principale du chaînage arrière varient par rapport à celles du chaînage avant car l'objectif n'est pas le même. Nous ne cherchons plus à ressortir une seule activité de la base d'activités. Mais nous cherchons plutôt à s'assurer que l'activité présente dans la base d'activités correspond bien aux critères de l'utilisateur.

Avantages et limites de ce Système Expert

Tout ingénieur doit avoir conscience du résultat qu'il fournit. C'est pourquoi nous désirons expliciter notre point de vue sur le système expert proposé.

Avantages :

- structure modulable : il est facile d'ajouter des paramètres, des questions, des règles et des activités grâce aux fonctions add-.... . Le moteur d'inférence gère lui-même l'arbre virtuel de questions qui seront posées.
- lisibilité du code : le code est globalement lisible, découpé en de petites fonctions élémentaires qui permettent d'alléger l'ensemble et de tester les fonctions individuellement.
- interface : pour un utilisateur, il est facile d'utiliser ce système expert grâce à notre interface qui est simple et efficace

Limites :

- Peu d'activités : le domaine des activités est très très très vaste et dense. Nous avons fait le choix de nous limiter à quelques activités pour répondre au sujet.
 - Solutions : Il serait possible d'ajouter des activités dans le code en "dur". Ou de demander à l'utilisateur, quand il y a un échec, de donner une activité qui pourrait correspondre aux paramètres fournis, afin d'étoffer la liste d'activités. Ou demander à spécialiser les activités en concurrence (ayant presque les mêmes conditions d'activation) en ajoutant des paramètres et des questions associées.

- Ordre des questions : actuellement, les questions posées sont celles qui ont une réponse menant à un paramètre permettant de préciser le plus d'activités.
 - Ce qui est préférable : le mieux, ce serait de trouver les questions qui permettent de séparer le plus d'activités en plusieurs groupes avec le même nombre d'activités par groupe.
- Risque d'échecs : étant donné que l'ordre des questions est ainsi, il se peut que des actions mènent à un échec.
 - Solution : pour éviter ces échecs, il faudrait corriger le défaut précédent mais aussi étoffer la base de données en précisant au maximum des paramètres pour les activités et ajouter des activités.

Conclusion

Pour conclure, nous pouvons dire que ce TP nous a fait beaucoup réfléchir sur comment concevoir un système expert. En effet, désirions avoir un système expert robuste et répondant aux besoins tout en faisant en sorte qu'il soit adaptable et simple d'utilisation. Le code, les commentaires, la présentation et le rapport guident un utilisateur ou un informaticien dans son utilisation et modification.

De plus, il a été très enrichissant de mettre en pratique les notions de système expert et plus globalement d'intelligence artificielle vues en cours afin d'implémenter notre propre système expert avec un chaînage avant et arrière.

Nous tenons personnellement à remercier l'équipe enseignante qui a su nous partager leurs connaissances avec passion lors de ce semestre particulier.

<https://imgur.com/A7VndEA>

