

Ontologie

Programmes d'échecs

9 JANVIER

IA03

Vivien Leclercq



Introduction

Dans le cadre de l'UV IA03, il nous a été demandé de réaliser une ontologie en lien avec la veille technologique que nous avons dû faire au préalable sur un sujet de notre choix. Cette dernière portait sur les différents programmes d'échecs.

Les échecs sont un jeu de société qui a été introduit au Xe siècle par les Arabes bien qu'on ignore l'origine exacte de sa création. Le jeu d'échecs est un des jeux de réflexion les plus populaires au monde. Cela est en partie lié à des séries qui ont fait l'objet de succès populaires comme *Le jeu de la dame* sur Netflix.

Nous verrons dans cette veille technologique que les programmes d'échecs ont beaucoup évolué avec le temps et sont devenus de plus en plus performants. Il s'agit également d'un des champs de bataille entre l'Homme et la machine. D'ailleurs, de nos jours, la machine a déjà surpassé l'Homme dans les programmes d'échecs et nous verrons comment.

Contexte

Cette ontologie est basée sur le travail de veille technologique que j'ai réalisé au début du semestre dans le cadre de l'UV IA03. De ce fait, cette ontologie vise à décrire le domaine des programmes d'échecs.

La méthodologie utilisée pour cette ontologie est celle détaillée par Natalya F. Noy et Deborah L. McGuinness de l'Université de Stanford, la même que nous avons vu en cours.

Table des matières

Introduction	2
Contexte	2
Contexte	4
Domaine et portée.....	4
A quels types de questions l'ontologie devra-t-elle fournir des réponses ?	4
Qui va utiliser et maintenir l'ontologie ?	5
Pourquoi utiliser une ontologie ?	5
Envisager une éventuelle réutilisation des ontologies existantes	5
Travaux de base	5
Concept clé	6
Travail à compléter	8
Sources de données pour les fichiers PGN	9
Méthodologie suivie	9
Etapas de construction de l'ontologie	11
Enumérer les mots importants de l'ontologie	11
Définir les classes et la hiérarchie des classes.....	12
Définir les propriétés des classes	16
Définir les facettes des attributs	22
Créer des instances	22
Complément sur les classes	24
Classes disjointes	24
Complément sur les attributs.....	25
Convention de nommage.....	25
Utilisation du vocabulaire SKOS	26
Utilisation du Linked Open Data	29
Interrogation de l'ontologie au moyen de SPARQL	31
Conclusion	43

Contexte

Domaine et portée

Avant de commencer à élaborer une ontologie, il convient de se poser les bonnes questions et de délimiter le domaine et la portée que va couvrir l'ontologie.

Quel est le domaine que va couvrir l'ontologie ?

Le domaine de l'ontologie concerne les programmes d'échecs. Elle vise à analyser le fonctionnement des programmes d'échecs et à représenter la manière dont fonctionne une partie d'échecs. Il ne s'agit pas de représenter l'ensemble des stratégies à adopter aux échecs comme l'ensemble des ouvertures possibles (ce qui demande une vie entière à être assimilée) mais bien à proposer une approche permettant d'appréhender simplement les concepts de base des échecs.

Il est important de noter que les sous-domaines de l'ontologie se complètent car il est inconcevable de parler de programme d'échecs ou de tournois d'échecs sans parler des échecs en général.

Dans quel but utiliserons-nous l'ontologie ?

L'ontologie a pour but de permettre aux novices de mieux appréhender les échecs en apportant une représentation symbolique des règles. Elle vise aussi à permettre de catégoriser les différents programmes d'échecs présents à ce jour et d'en évaluer les potentielles forces ou faiblesses.

Elle pourra fournir des statistiques sur des parties d'échecs et étudier les comportements des joueurs d'échecs (humain ou programme), leur niveau, leur style de jeu.

A quels types de questions l'ontologie devra-t-elle fournir des réponses ?

L'ontologie apportera des réponses quant à la signification des termes techniques relatifs aux échecs ainsi que des réponses sur la structure d'une partie d'échecs. Elle fournira des statistiques sur des joueurs à partir de parties d'échecs déjà réalisées.

Exemple de questions auxquelles devra répondre l'Ontologie :

- Quelles sont les ouvertures existantes aux échecs ?
- Quelles sont les phases d'une partie d'échecs ?

-
- Quels sont les programmes d'échecs existants ?
 - Quels programmes d'échecs utilisent des réseaux de neurones ?
 - Quel est le niveau aux échecs d'un individu X ?
 - Quel est le programme d'échecs le plus puissant en termes de niveau ?
 - Quels ont été les déplacements joués lors de la partie XXX ?
 - Quels sont les tournois d'échecs qui ont eu lieu ?
 - Quelles parties ont été jouées lors du tournoi XXX ?
 - Combien de fois le joueur X a-t-il gagné contre le joueur Y ?
 - Quelles sont les cases qui composent l'échiquier ?
 - Quelles sont les pièces majeures aux échecs ?
 - Quelles parties d'échecs ont été jouées en ligne ?
 - Quels ont été les rapports de partie écrits par l'individu X ?
 - Quelles parties ont utilisés l'ouverture XXX ?
 - Quelles parties se sont jouées le DD/MM/YYYY ?

Qui va utiliser et maintenir l'ontologie ?

L'ontologie sera entièrement réalisée par moi-même, Vivien Leclercq étudiant en IA03. Elle pourra être consulté librement en ligne et interrogée dans le cloud.

A termes, l'ontologie sera mise à jour automatiquement par un programme qui s'occupe de récupérer les données des parties d'échecs ayant lieu dans le monde.

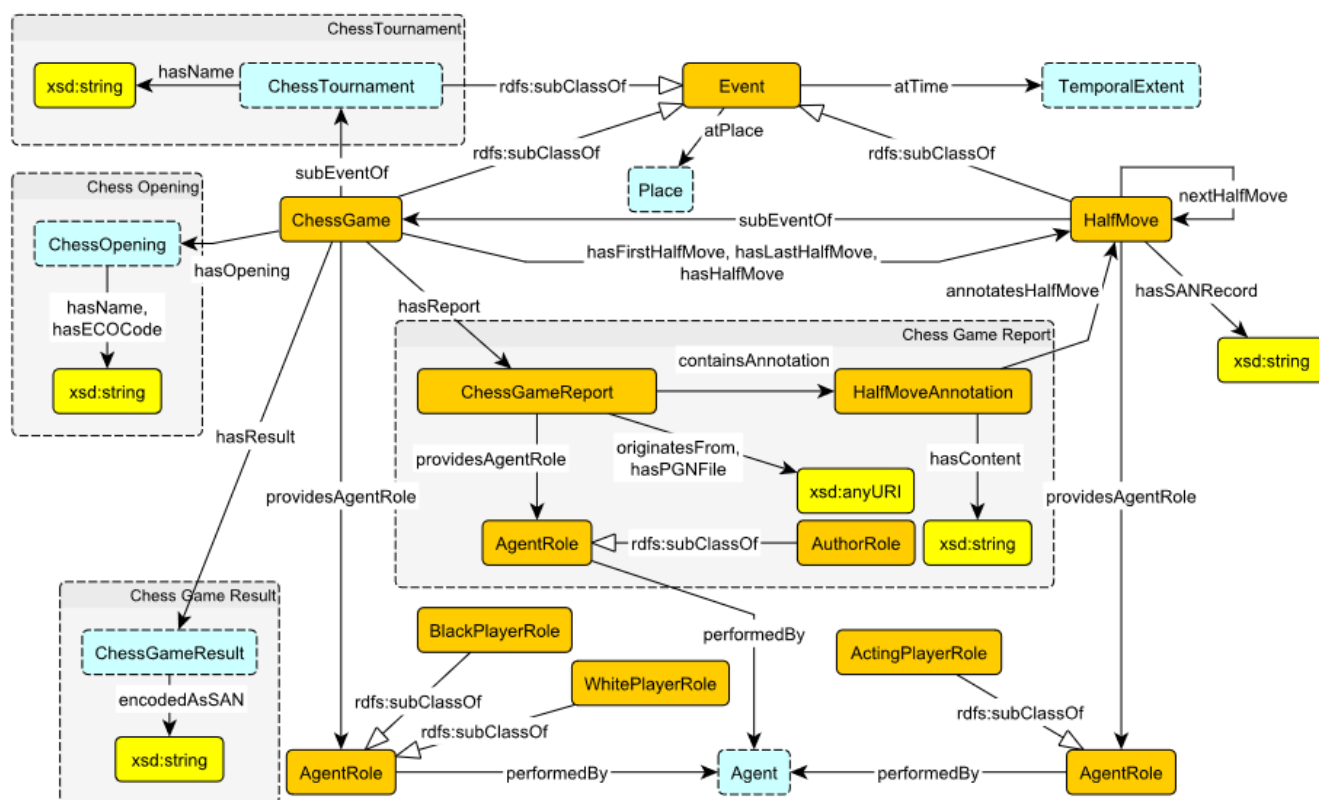
Pourquoi utiliser une ontologie ?

L'objectif n'est pas seulement de stocker des connaissances et de faire des requêtes classiques comme on pourrait le faire avec une base de données, mais bien de représenter les concepts liés aux échecs et les interactions entre ces concepts avec d'élaborer un langage commun et une compréhension commune du domaine d'application de l'ontologie. L'existence de ces interactions entre ces concepts permet d'élaborer des requêtes beaucoup plus riches et ainsi de répondre à des questions beaucoup plus complexes que dans une base de données classique.

Envisager une éventuelle réutilisation des ontologies existantes

Travaux de base

Après avoir effectué des recherches sur Internet, j'ai trouvé des travaux sur l'élaboration d'une ontologie concernant les échecs : Modeling With Ontology Design Patterns: Chess Games As a Worked Example de Adila Krisnadhi et Pascal Hitzler à Wright State University. Cette ontologie, reprenant une bonne partie des concepts que je cherche à modéliser, servira de base solide à la construction de ma propre ontologie.



Représentation complète de l'ontologie de base

Lien de l'ontologie existante : <http://people.cs.ksu.edu/~hitzler/pub2/01-chess-example.pdf>

Dans un souci de cohérence, cette ontologie sera traduite en français.

Concept clé

Avant d'aller plus loin, il est impératif de vous présenter un concept clé qui est au centre de l'ontologie que je cherche à construire : les fichiers PGN.

En effet, un fichier PGN (Portable Game Notation) est un fichier texte avec l'extension .pgn. Il permet de retranscrire fidèlement le déroulement d'une partie d'échecs en proposant une manière de formater : les déplacements, le nom des

joueurs, la date de la partie, le lieu de la partie, le tournoi de la partie, le round de la partie.

```
[Event "F/S Return Match"]
[Site "Belgrade, Serbia JUG"]
[Date "1992.11.04"]
[Round "29"]
[White "Fischer, Robert J."]
[Black "Spassky, Boris V."]
[Result "1/2-1/2"]
```

```
1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 Nf6 5. O-O Be7 6. Re1 b5 7. Bb3 d6 8. c3
O-O 9. h3 Nb8 10. d4 Nbd7 11. c4 c6 12. cxb5 axb5 13. Nc3 Bb7 14. Bg5 b4 15.
Nb1 h6 16. Bh4 c5 17. dxe5 Nxe4 18. Bxe7 Qxe7 19. exd6 Qf6 20. Nbd2 Nxd6 21.
Nc4 Nxc4 22. Bxc4 Nb6 23. Ne5 Rae8 24. Bxf7+ Rxf7 25. Nxf7 Rxe1+ 26. Qxe1 Kxf7
27. Qe3 Qg5 28. Qxg5 hxg5 29. b3 Ke6 30. a3 Kd6 31. axb4 cxb4 32. Ra5 Nd5 33.
f3 Bc8 34. Kf2 Bf5 35. Ra7 g6 36. Ra6+ Kc5 37. Ke1 Nf4 38. g3 Nxh3 39. Kd2 Kb5
40. Rd6 Kc5 41. Ra6 Nf2 42. g4 Bd3 43. Re6 1/2-1/2
```

Structure d'un fichier PGN complet

On remarque en effet la présence d'en-têtes apportant des informations sur le contexte de la partie suivie d'une suite de caractères qui représentent les déplacements. Il s'agit d'un format international, c'est-à-dire que tous les joueurs expérimentés de la planète comprennent ce langage et sont capables de visualiser une partie d'échecs entière à partir de cette suite de caractères.

A propos des en-têtes, on remarque que certaines sont parfois incomplètes, ce qui peut arriver.

```
[Event "Interclubs FRA"]
[Site "?"]
[Date "?????.???.???"]
[Round "?"]
[White "Calistri, Tristan"]
[Black "Bauduin, Etienne"]
[Result "1-0"]
```

```
1.e4 c5 2.Nf3 e6 3.d4 cxd4 4.Nxd4 Nc6 5.Nc3 a6 6.Be2 Qc7 7.O-O Nf6 8.Be3 Bb4
9.Na4 O-O 10.c4 Bd6 11.g3 Nxe4 12.Bf3 f5 13.Bxe4 fxe4 14.c5 Be7 {Les Noirs ont
un pion d'avance mais de gros problèmes pour mettre leur Fc8 et leur Ta8 en jeu}
15.Qg4 Ne5 16.Qxe4 d5 17.cxd6 Bxd6 18.Rac1 Qa5 19.Nb3 {Les blancs ont
récupéré leur pion et toutes leurs pièces sont mobilisées}
19...Qb4
    (19...Qd5 20.Qxd5 exd5 21.Nb6 Bh3 22.Nxa8 Nf3+ 23.Kh1 Bxf1 24.Rxf1 Rxa8 25.Rd1)
    (19...Nf3+ 20.Kg2 Qh5)
20.Qxb4 Bxb4 21.Nb6 $18 {Les noirs n'arriveront jamais à sortir leur Fc8}
21...Rb8 22.Bc5 Bxc5
    (22...Nd3 23.Bxf8 Nxc1 24.Rxc1 Bxf8 25.Rxc8 Rxc8 26.Nxc8)
23.Nxc5 Rd8 24.Rfd1 Re8 25.Ne4 Nf7 26.Rc7 Kf8 27.Rdc1 1-0
```

Structure d'un fichier PGN incomplet

Ce deuxième fichier est beaucoup plus complexe et contient des commentaires écrits par un auteur. Il est même parfois possible de retrouver des annotations sur des fichiers PGN qui correspondent à des commentaires qu'à ajouté un expert aux échecs afin de mettre l'accent sur l'avantage que possède un joueur par rapport à l'autre. Chaque commentateur d'échecs dispose de son propre système d'annotation utilisant une suite de symboles qui lui est propre. Cependant, on note une certaine convention sur la manière de noter certains coups (?? : très mauvais coup, !? : coup intéressant, !! : coup excellent).

Enfin, il est à noter qu'il existe également des en-têtes optionnels à chaque fichier PGN permettant de donner davantage d'informations sur le contexte d'une partie. Parmi celles-ci, on retrouve :

- WhiteElo, BlackElo : classement Elo des Blancs et des Noirs
- Opening : ouverture
- Variation : variante
- ECO : code ECO (code international de l'ouverture)

Travail à compléter

Je viens de vous présenter, ce qui me servira de base dans la construction de mon ontologie. Cependant, il faut garder à l'esprit que ces éléments ne suffiront pas et qu'il faudra ajouter des éléments notamment pour inclure le fait de représenter les règles d'échecs et la manière dont fonctionnent les programmes d'échecs.

A titre d'exemple, voici un fichier PGN qui devra pouvoir être pris en charge dans l'ontologie que je cherche à modéliser :

```
[Event "CCRL Blitz"]
[Site "CCRL"]
[Date "2021.09.15"]
[Round "604.4.788"]
[White "Fat Fritz 2 (in SF) 64-bit 8CPU"]
[Black "Lc0 0.28.0 w752187 RTX2080"]
[Result "1/2-1/2"]
[ECO "E32"]
[Opening "Nimzo-Indian"]
[Variation "classical variation"]
[PlyCount "105"]
```

```
[WhiteElo "3720"]  
[BlackElo "3674"]
```

```
1. d4 Nf6 2. c4 e6 3. Nc3 Bb4 4. Qc2 O-O 5. a3 Bxc3+ 6. Qxc3 b6 7. e3 d5 8. Nf3  
Bb7 9. b3 Nbd7 10. Be2 c5 11. O-O Rc8 12. Rd1 Ne4 13. Qb2 Qf6 14. a4 a5 15. Qc2  
Rfd8 16. Bb2 cxd4 17. exd4 Nd6 18. Rac1 dxc4 19. d5 Qe7 20. dxe6 fxe6 21. Ba3  
Nc5 22. bxc4 Nf5 23. Bxc5 bxc5 24. Rxd8+ Rxd8 25. Rb1 h6 26. Qc3 Be4 27. Re1  
Qc7 28. Bf1 Bc6 29. Qa3 Qd6 30. Ne5 Nd4 31. Bd3 Rf8 32. Qa1 Qd8 33. Qb1 Be8 34.  
Qb7 Qf6 35. f3 Bxa4 36. Qe4 Qf5 37. Qxf5 exf5 38. Ra1 Re8 39. f4 Bc2 40. Bxc2  
Nxc2 41. Rxa5 Nd4 42. g3 g5 43. Kf2 gxf4 44. gxf4 Ne6 45. Ke3 Rb8 46. Nd3 Re8  
47. h3 Ng5+ 48. Ne5 Ne4 49. Ra7 Rb8 50. Nd3 Re8 51. Ne5 Rb8 52. Nd3 Re8 53. Ne5  
1/2-1/2
```

Exemple de fichier PGN à prendre en charge

Sources de données pour les fichiers PGN

Afin de compléter mon ontologie, il est nécessaire de trouver des sources de données libre d'accès de fichiers PGN.

Depuis plusieurs années, le niveau entre les programmes d'échecs et les humains s'est détaché, à tel point que les humains et les programmes d'échecs ne s'affrontent plus lors de tournois officiels du fait qu'aucun humain ne veuille être humilié face à une machine. De ce fait, les tournois se faisant s'affronter les humains et les programmes d'échecs sont différents et les sources de données pour récupérer les fichiers PGN sont différentes.

Pour ce qui est des humains, la source de données utilisées sera :
<https://www.pgnmentor.com/files.html#players>

Pour ce qui est des programmes d'échecs, la source de données utilisée sera :
<http://www.computerchess.org.uk/ccrl/404/games.html>

Méthodologie suivie

Du fait du grand nombre de parties d'échecs disponibles au format PGN, toutes ne seront pas directement implémentées au sein de l'ontologie. Pour l'exemple, seules quelques parties seront implémentées afin de vérifier la fonctionnalité de l'ontologie.

Par la suite, un programme pourra automatiser l'intégration de ces données facilement.

Etapes de construction de l'ontologie

Enumérer les mots importants de l'ontologie

Voici une liste de mots relatifs aux programmes d'échecs et qui devront être présents dans l'ontologie :

- Echecs
- Partie d'échecs
- Echiquier
- Blancs/Noirs
- 64 cases
- Cases nommées
- Pièce
- Pièces majeures/mineures
- Pion
- Fou
- Cavalier
- Tour
- Roi
- Reine
- Echec et mat
- PAT
- Egalité
- Grand roque
- Petit roque
- Prise en passant
- Cadence
- Classement Elo
- 2 joueurs
- Garry Kasparov
- Stockfish
- Leela Chess Zero
- Komodo
- Valeur des pièces
- Ouverture
- Milieu

-
- Finale
 - Jeux électroniques
 - Algorithme MinMax
 - Elagage Alpha-Bêta
 - Programmes d'échecs
 - Mephisto
 - Réseaux de neurones
 - Supercalculateurs
 - Deep Blue
 - Jeu de Go
 - Méthode de Monte Carlo
 - AlphaGo
 - Alpha Zero

Définir les classes et la hiérarchie des classes

J'ai décidé d'adopter une approche de haut en bas. En effet, il m'a semblé plus légitime de commencer par des concepts globaux et généraux avant d'aborder des concepts plus spécifiques.

Voici la hiérarchie des classes choisie :

- AnnotationDeDeplacement
- Case
- Couleur
 - Blanc
 - Noir
- Echiquier
- Endroit
- Evenement
 - Deplacement
 - PartieDEchecs
 - TournoiDEchecs
- Joueur
 - Humain
 - Programme
 - PremierProgramme
 - ReseauDeNeurone

-
- SuperCalculeur
 - ObjetTemporaire
 - PhaseDePartie
 - Ouverture
 - Milieu
 - Finale
 - Piece
 - PieceMajeure
 - PieceMineure
 - RapportDePartie
 - RegleSpeciale
 - ResultatDePartie
 - RoleDeJoueur
 - Auteur
 - JoueurBlanc
 - JoueurNoir
 - StructureDeProgramme
 - TypeDeMouvement

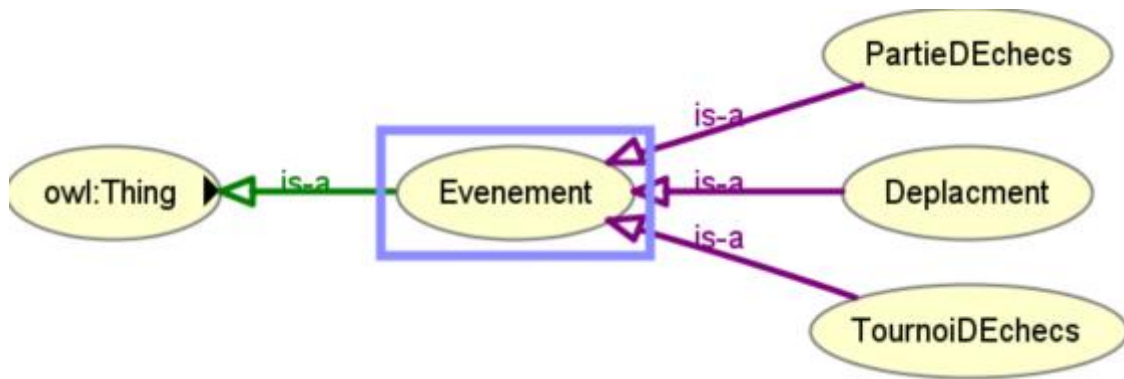
Le cœur de l'ontologie, les classes Evenement, Déplacement, AnnotationDeDeplacement, Joueur, Endroit, ObjetTemporaire, RoleDeJoueur, RapportDePartie, sont directement issus de l'ontologie pré-existante sur laquelle je me base.

D'un point de vue graphique :



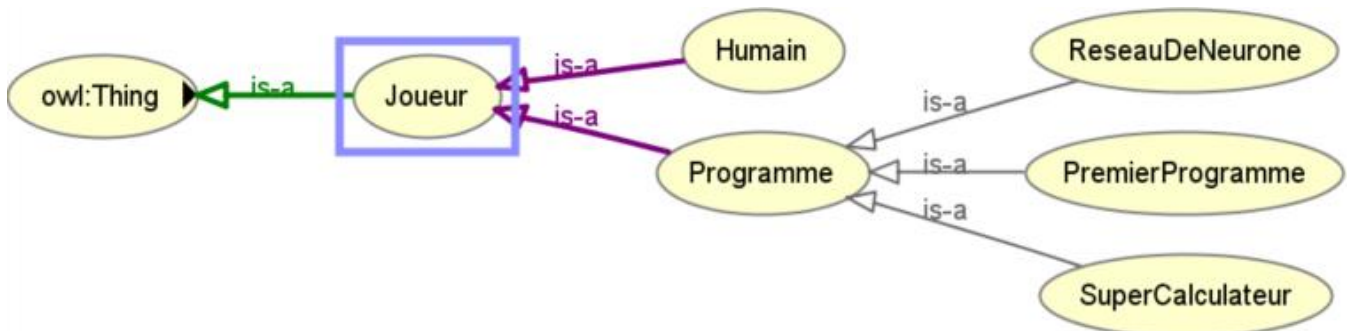
Hiérarchie des classes de l'ontologie sur les programmes d'échecs

Il est intéressant ici de se focaliser sur des parties de l'ontologie et de vous décrire ce qui a guidé l'adoption de cette hiérarchie.



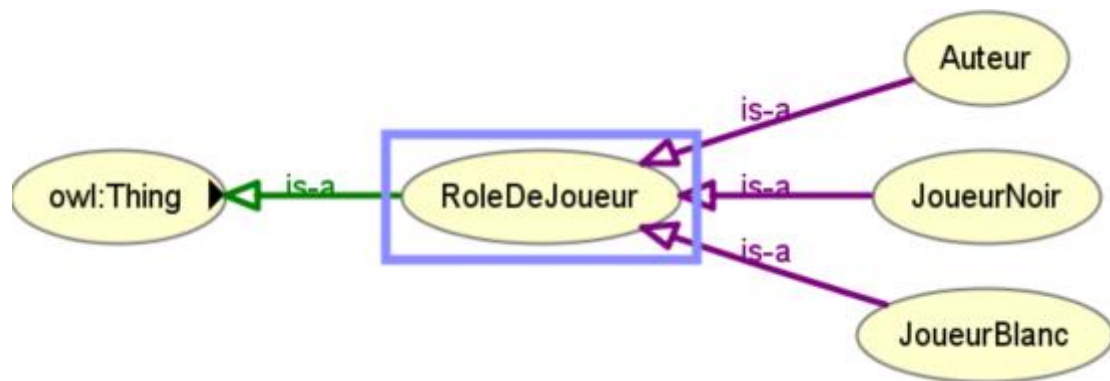
Hiérarchie de la partie « Evenement » de l'ontologie

On remarque que pour cette partie de l'ontologie, ce qui guide la hiérarchisation, c'est le fait que relation de subsomption soit une relation de type « est-un ». En ce sens toutes les classes filles doivent être une spécialisation de la classe mère. En effet, ce qui caractérise un événement est le fait que ce soit un concept qui a lieu à un temps donné et à un endroit donné. En cela, une partie d'échecs ou un tournoi d'échecs ou un déplacement d'une partie d'échecs sont des spécialisations d'un événement.



Hiérarchie de la partie « Joueur » de l'ontologie

A présent, voici une analyse de la partie « Joueur » de l'ontologie. Le concept de joueur représente un individu qui participe à une partie d'échecs. Ce concept peut être spécialisé en deux parties : joueurs humains et programmes joueurs. Il est donc normal d'avoir voulu créer deux sous-concepts du concept de joueur. Ensuite, le concept de joueur programme, dans le contexte de programme d'échecs, peut être lui-même spécialisé en différent type de programme d'échecs : les réseaux de neurones, les supercalculateurs et les programmes classiques.



Hiérarchie de la partie « Rôle de joueur » de l'ontologie

Enfin, cette partie de l'ontologie vise à représenter le concept de rôle que peut jouer un joueur aux échecs. Il s'agit de regrouper au sein d'un même concepts tous les rôles que peut accomplir un joueur aux échecs : il peut jouer les blancs ou les noirs ou bien même être l'auteur d'un rapport de partie. Il est important de comprendre ici que ces sous-concepts ne sont pas disjoints car il est possible qu'un joueur ait joué les blancs dans une partie et les noirs dans une autre. C'est le moteur d'inférence qui calcule directement les types supplémentaires que peut avoir un joueur.



Exemple de typage dynamique de « Rôle de joueur »

On remarque ici que l'individu « stockfish-nnue » est primordiallement de type ReseauDeNeurone qui est un sous-type de Programme qui est un sous-type de Joueur. Du fait qu'il au moins une partie dans laquelle il existe une relation binaire de type « aPourJoueurBlanc » entre cette partie et cet individu, le Reasoner a calculé que l'individu était de type JoueurBlanc (Range de la relation). Il en va de même avec le type JoueurNoir.

Définir les propriétés des classes

Une fois les classes définies, il est important de définir les interactions qui peuvent exister entre les classes ainsi que les éléments qui les composent.

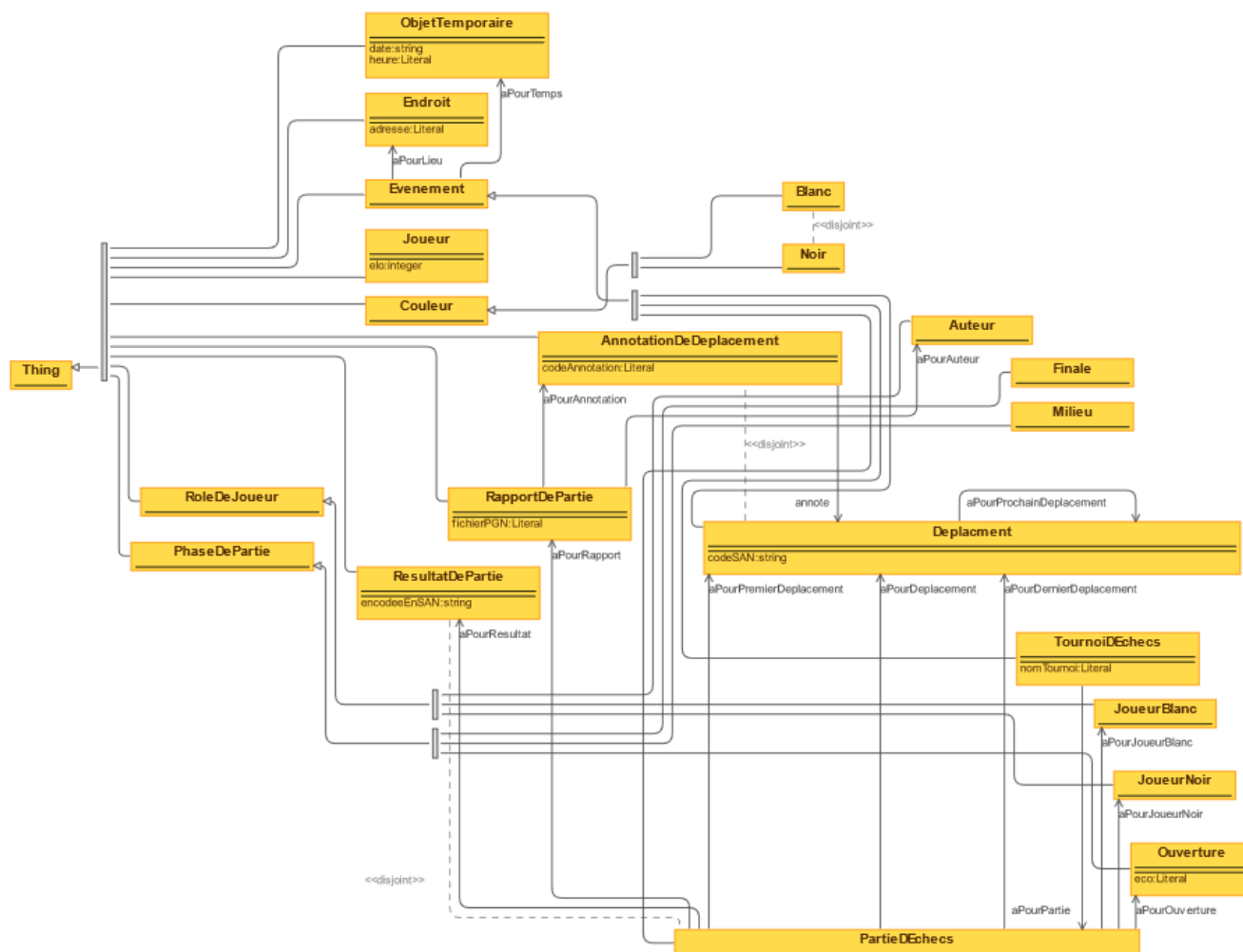
Voici la liste des propriétés existantes au sein de mon ontologie :

-
- AnnotationDeDeplacement
 - `annotate` (Deplacement)
 - `codeAnnotation` (xsd:string)
 - Case
 - `aPourCouleur` (Couleur)
 - `nomCase` (xsd:string)
 - Couleur
 - Blanc
 - Noir
 - Echiquier
 - `estComposeDe` (Case)
 - `estConstitueDe` (Piece)
 - Endroit
 - `adresse` (xsd:string)
 - Evenement
 - Deplacement
 - `codeSAN` (xsd:string)
 - PartieDEchecs
 - `aPourDeplacement` (Deplacement)
 - `aPourPremierDeplacement` (Deplacement)
 - `aPourDernierDeplacement` (Deplacement)
 - `aPourOuverture` (Ouverture)
 - `aPourMilieu` (Milieu)
 - `aPourFinale` (Finale)
 - `aPourJoueurBlanc` (JoueurBlanc)
 - `aPourJoueurNoir` (JoueurNoir)
 - `aPourLieu` (Endroit)
 - `aPourTemps` (ObjetTemporaire)
 - `aPourRapport` (RapportDePartie)
 - `aPourResultat` (ResultatDePartie)
 - `seJoueSur` (Echiquier)
 - TournoiDEchecs
 - `nomTournoi` (xsd:string)
 - `aPourPartie` (PartieDEchecs)
 - Joueur
 - `elo` (xsd:integer)

-
- Humain
 - nom (xsd:string)
 - prenom (xsd:string)
 - Programme
 - nomProgramme (xsd:string)
 - implemente (StructureDeProgramme)
 - PremierProgramme
 - ReseauDeNeurone
 - SuperCalculateur
 - ObjetTemporaire
 - date (xsd:string)
 - heure (xsd:string)
 - PhaseDePartie
 - Ouverture
 - eco (xsd:string)
 - Milieu
 - Finale
 - Piece
 - estDeCouleur (Couleur)
 - aPourCase (Case)
 - aPourValeur (xsd:double)
 - aPourMouvement (TypeDeMouvement)
 - aPourRegleSpeciale (RegleSpeciale)
 - RapportDePartie
 - fichierPGN (xsd:anyURI)
 - aPourAnnotation (Annotation)
 - aPourAuteur (Auteur)
 - RegleSpeciale
 - ResultatDePartie
 - encodeeEnSAN (xsd:string)
 - RoleDeJoueur
 - Auteur
 - JoueurBlanc
 - JoueurNoir
 - StructureDeProgramme
 - TypeDeMouvement

Chaque propriété est indiquée en rouge. Le domaine est la classe mère du paragraphe et le range est toujours précisé entre parenthèses.

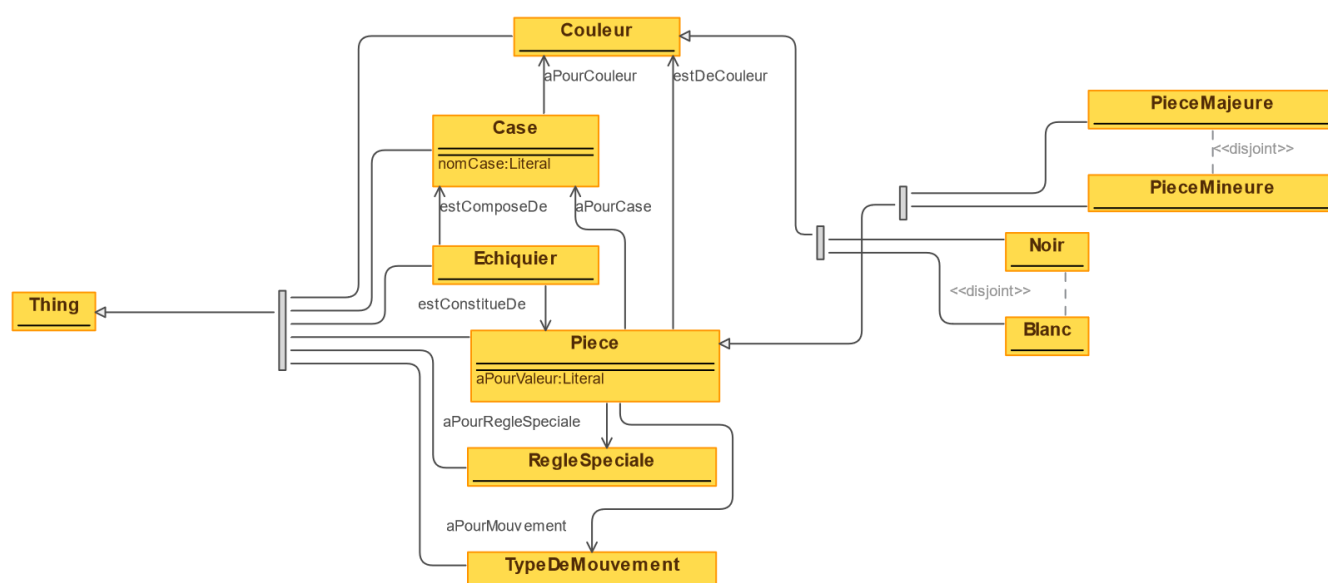
Afin d'être lisible, voici des représentations successives de morceaux de l'ontologie constituée.



Représentation de la partie de l'ontologie concernant les parties d'échecs.

Cette partie de l'ontologie représente les parties d'échecs de la même manière que dans l'ontologie de base utilisée pour cette ontologie. Il est mieux de commencer la lecture de cette ontologie en partant de la classe PartieDEchecs qui est vraiment centrale. Cette partie d'échecs possède des déplacements liés par une liste chaînée, ainsi qu'un premier et un dernier déplacement. Cette partie donne lieu à un rapport de partie écrit par un auteur qui contient des annotations sur les déplacements. Une partie d'échecs possède une ouverture particulière, un milieu de partie et une finale. Elle a toujours un joueur noir et un joueur blanc. Une partie d'échecs appartient à un tournoi d'échecs qui est un événement et possède donc un

endroit et un objet temporel. Enfin, la partie possède un résultat de partie qui lui est propre.



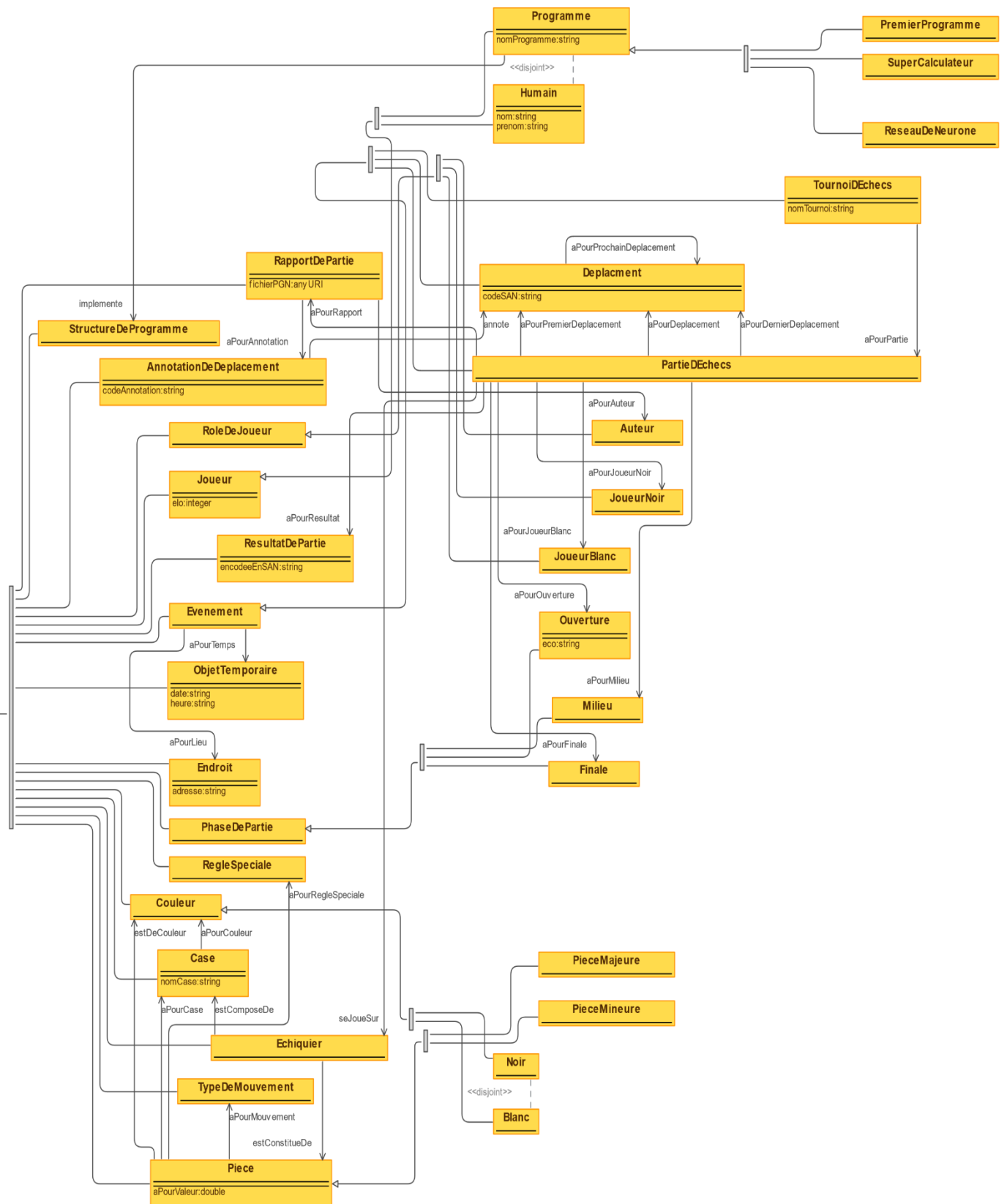
Représentation de la partie de l'ontologie concernant les échecs.

Cette partie de l'ontologie a pour but de représenter les règles aux échecs. Il est mieux de commencer la lecture par la classe **Pièce** qui est spécialisée en deux types : les pièces majeures et les pièces mineures. Chaque pièce a pour mouvement de mouvements qui lui suit accordés et des règles spéciales. Les pièces sont disposées sur un échiquier lui-même composé de cases. Les pièces et les cases de l'échiquier peuvent être soit blanches soit noires.

Enfin, il est important de comprendre qu'il ne s'agit pas de parties complètement séparés de l'ontologie, mais de parties qui peuvent interagir entre elles. Ici cette séparation en deux parties distinctes a été faite de manière à produire une représentation lisible et compréhensible de l'ontologie.

Le lien entre les deux parties se fait en pratique par le biais de la classe **Echiquier** car une partie d'échecs est jouée sur un échiquier standard ou un échiquier en ligne. A partir de cet échiquier il est possible de lui attribuer les cases et les pièces qui le composent.

Vous trouverez page suivante une représentation de cette ontologie sans les individus.



Représentation de l'ontologie sans les individus

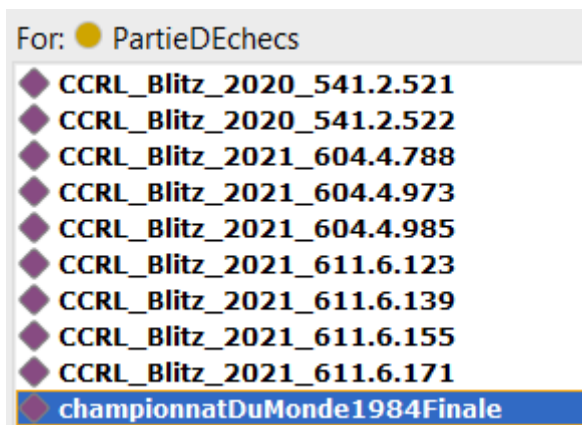
Définir les facettes des attributs

Les facettes des attributs ont déjà été en partie définie à l'étape précédente lors de l'élaboration du domaine et du range de chaque propriété.

Créer des instances

Afin de peupler l'ontologie, des exemples de parties ont servi de base comme individus. Ainsi, une dizaine de parties d'échecs ont été représenté et parmi elles, ont été décrite avec leur déplacement.

Il n'est pas raisonnable d'ajouter plus d'individus car la base de connaissance est déjà assez pleine pour pouvoir être interrogée. De plus, il est fastidieux d'ajouter tous les déplacements un par un à la main. Cette tâche pourrait être faite automatiquement par un programme qui pourra, à l'avenir, s'occuper de peupler l'ontologie à partir de fichiers PGN. Néanmoins, quelques exemples de parties ont été ajoutées à la main dans la base de connaissance.



Individus de PartieDEchecs

For: ● Déplacement

- ◆ CCRL_Blitz_2020_541.2.521D1
- ◆ CCRL_Blitz_2020_541.2.521D10
- ◆ CCRL_Blitz_2020_541.2.521D11
- ◆ CCRL_Blitz_2020_541.2.521D12
- ◆ CCRL_Blitz_2020_541.2.521D13
- ◆ CCRL_Blitz_2020_541.2.521D14
- ◆ CCRL_Blitz_2020_541.2.521D15
- ◆ CCRL_Blitz_2020_541.2.521D16
- ◆ CCRL_Blitz_2020_541.2.521D17
- ◆ CCRL_Blitz_2020_541.2.521D18
- ◆ CCRL_Blitz_2020_541.2.521D19
- ◆ CCRL_Blitz_2020_541.2.521D2
- ◆ CCRL_Blitz_2020_541.2.521D20

Quelques individus de Déplacement

En plus de ces parties d'échecs, l'échiquier et ce qui le compose (Cases et Pièces) ont été entièrement modélisé. Afin d'ajouter de la structure, les règles de bases qui composent les échecs ont aussi été modélisés.

For: ● Case

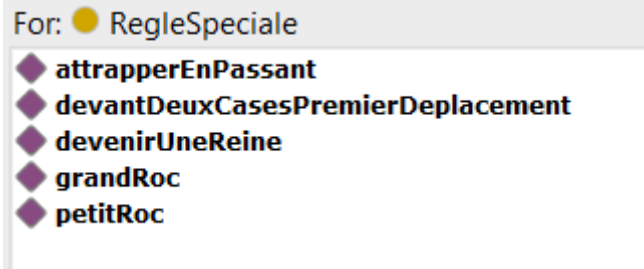
- ◆ a1
- ◆ a2
- ◆ a3
- ◆ a4
- ◆ a5
- ◆ a6
- ◆ a7
- ◆ a8
- ◆ b1
- ◆ b2
- ◆ b3
- ◆ b4
- ◆ b5

Quelques individus de Case

For: ● TypeDeMouvement

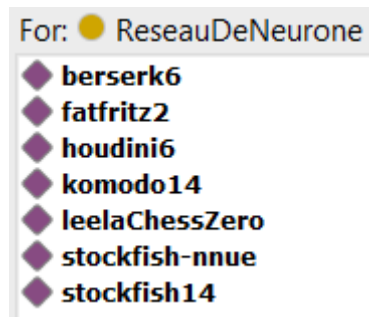
- ◆ attaquerSurLesCotesUneCase
- ◆ devantUneCase
- ◆ diagonale
- ◆ enL
- ◆ ligne
- ◆ partoutUneCase

Individus de TypeDeMouvement



Individus de RegleSpeciale

Enfin, des exemples de programmes d'échecs et de joueurs humains ont aussi été modélisés ainsi que les principales structures de programme que compte un programme d'échecs.



Individus de ReseauDeNeurone (sous-classe de Programme)

Complément sur les classes

Classes disjointes

Afin d'éviter des anomalies dans des données qui composent notre base de connaissance, il est nécessaire d'ajouter explicitement des disjonctions. Cela afin de garantir que l'ontologie reste cohérente et également afin d'éviter la perte de sens sémantique.

Ainsi les classes suivantes ont été déclaré comme disjointes :

- AnnotationDeDeplacement et Deplacement.
- Blanc et Noir (il est inconcevable d'être à la fois blanc et noir).
- PartieDEchecs et ResultatDePartie (l'un compose l'autre).
- Humain et Programme (il est inconcevable d'être à la fois un humain et un programme).
- PieceMajeure et PieceMineure (il est inconcevable d'être à la fois une pièce majeure et une pièce mineure).

Complément sur les attributs

Il n'y a pas eu la nécessité de créer des propriétés symétriques, transitives ou réflexives. Ce qui est une bonne nouvelle car cela permet de ne pas augmenter la complexité de l'ontologie.

A noter que les types des Data properties n'ont pas pu être définis correctement du fait que le Reasoner se soit montré incapable d'importer XSD.

Convention de nommage

Comme dans la plupart des ontologies existantes, des conventions de nommages ont été établis.

Nom des classes : elles commencent toujours par une majuscule, chaque nouveau mot commence par une majuscule, il n'y a pas d'espace entre les mots. Les accents sont interdits.

Exemple : HypnoseDeRue

Nom des instances : elles commencent toujours par une minuscule, chaque nouveau mot commence par une majuscule, il n'y a pas d'espace entre les mots. Les accents sont interdits.

Exemple : premierVolontaire

Nom des attributs : ils commencent toujours par une minuscule, chaque nouveau mot commence par une majuscule, il n'y a pas d'espace entre les mots. Les accents sont interdits. En plus de cela, la plupart des attributs ont pour préfixe « aPour- » suivi du nom de l'objet en range de la relation.

Exemple : aPourCouleur

Utilisation du vocabulaire SKOS

Comme demandé, l'ontologie contient une partie utilisant le vocabulaire SKOS afin de décrire le vocabulaire des échecs. Les principaux concepts y ont été représentés à l'intérieur du Concept Schema « Vocabulaire des Echecs ».

Voici une représentation de ce vocabulaire générée à partir de SKOS Play ! :

Vocabulaire des Echecs



Case

Couleurs

- L Noir
- L Blanc

Echiquier

Partie

Joueur

Pièces

- L Cavalier
- L Fou
- L Roi
- L Pion
- L Reine
- L Tour

Structures des programmes d'échecs

- L Algorithme MinMax
- L Elagage Alpha-Bêta
- L Méthode de Monte-Carlo
- L Réseau Neuronal
- L Table de Finales

[Représentation de SKOS Play ! du Vocabulaire des Echecs](#)

De même, voici une autre représentation de ce même vocabulaire de manière plus graphique avec encore une fois l'application SKOS Play ! :



Représentation de SKOS Play ! du Vocabulaire des Echecs

Il est important de rappeler que ces représentations ne permettent de visualiser qu'une partie du vocabulaire. Ainsi les définitions pour chaque concept propre au vocabulaire des échecs ne sont pas représentées. De même seuls les labels préférés en

français sont visibles ici alors qu'il existe des labels alternatifs ou cachés en français et en anglais pour certains concepts.

Le vocabulaire a été construit en utilisant une Collection « Pièces » qui regroupe toutes les pièces existantes aux échecs. De même il existe aussi une autre Collection « Structures des programmes d'échecs » qui regroupe toutes les structures de programmation que peuvent intégrer les programmes d'échecs.

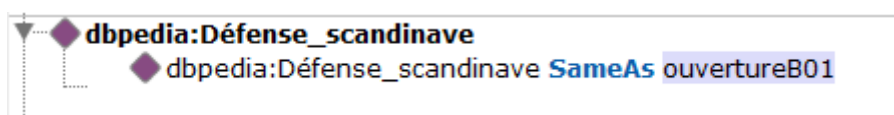
Utilisation du Linked Open Data

Afin de rendre l'ontologie connectée avec des concepts appartenant au monde extérieur, l'ontologie a été mise en lien avec des éléments appartenant au Linked Open Data. Ces relations ont été faites de la même manière que celle vue en TD.

En effet, cette mise en relation s'est faite exclusivement avec des éléments appartenant au Linked Open Data de dbpedia.org.

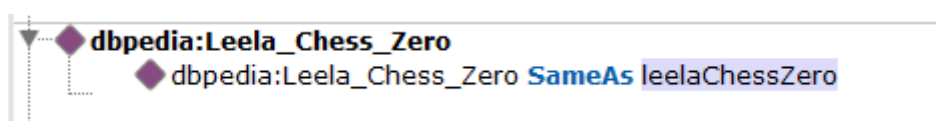
Dès que cela a été possible, les individus ont été mis en lien, de la manière que celle vue en TD, avec des individus appartenant à dbpedia.org.

C'est le cas pour les ouvertures qui ont été mises en lien avec les individus de dbpedia.org. A titre d'exemple nous l'individu ouvertureB01 dont le code ECO est « B01 » est le même individu que l'individu Défense_scandinave de la base de connaissance dbpedia.org. Ainsi, si l'utilisateur de l'ontologie souhaite en savoir plus sur l'ouverture « B01 » il peut consulter l'individu Défense_scandinave sur dbpedia.org et ainsi en apprendre plus. Dans Protege, cela est visible de la manière suivante :



Exemple de mise en relation de l'ontologie avec un individu de dbpedia.org

En plus des ouvertures, certains individus de type Joueur ont été mis en relation avec les individus correspondant sur la base de connaissance dbpedia.org. Cela a été fait dès que cela était possible, c'est-à-dire que certains joueurs (humains ou programmes) n'ont pas pu être mis en relation avec des individus de la base dbpedia.org du fait que leur page correspondante n'existe pas du côté de dbpedia.org. C'est ainsi que l'individu leelaChessZero a été mis en relation avec l'individu correspondant Leela_Chess_Zero de la base de connaissances dbpedia.org. Dans Protege, cela est visible de la manière suivante :



Deuxième exemple de mise en relation de l'ontologie avec un individu de dbpedia.org

Enfin, les couleurs blanches et noires sont également des individus qui existent sur Wikipédia et ont pu mettre en lien avec les individus correspondants de l'ontologie.

D'un point de vue complet, voici la liste des individus de la base de connaissance qui ont été mis en lien avec l'ontologie :

- ◆ **dbpedia:Anatoli_Karpov**
- ◆ **dbpedia:Blanc**
- ◆ **dbpedia:Deep_Blue**
- ◆ **dbpedia:Début_Larsen**
- ◆ **dbpedia:Défense_Benoni**
- ◆ **dbpedia:Défense_scandinave**
- ◆ **dbpedia:Gambit_du_roi**
- ◆ **dbpedia:Garry_Kasparov**
- ◆ **dbpedia:Komodo_(programme_d'échecs)**
- ◆ **dbpedia:Leela_Chess_Zero**
- ◆ **dbpedia:Mephisto_(jeu_d'échecs)**
- ◆ **dbpedia:Noir**
- ◆ **dbpedia:Ouverture_anglaise**
- ◆ **dbpedia:Ouverture_Bird**
- ◆ **dbpedia:Ouverture_d'échecs_irrégulière**
- ◆ **dbpedia:Stockfish_(programme_d'échecs)**
- ◆ **dbpedia:Variante_Lasker_du_gambit_dame_refusé**

Liste des individus de dbpedia.org en relation avec l'ontologie

Cette liste pourra évoluer en fonction du peuplement à venir de l'ontologie.

Interrogation de l'ontologie au moyen de SPARQL

Il s'agit dans cette partie de présenter les requêtes SPARQL capable de répondre aux questions que nous nous sommes posées au début de ce rapport.

- Quelles sont les ouvertures existantes aux échecs ?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>
```

```
SELECT DISTINCT ?eco WHERE {
    ?xouverture1 rdf:type db:Ouverture .
    ?ouverture1 db:eco ?eco .
}
ORDER BY ?eco
```

?eco
A00
A01
A02
A03
A04
A09
A10
A43
B01
B12
B19
C19
C37
D30
D55
E32

Il ne s'agit pas de toutes les ouvertures disponibles mais seulement celles contenues dans la base de connaissance.

- Quelles sont les phases d'une partie d'échecs ?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>
```

```
SELECT DISTINCT ?phase WHERE {
    ?phase rdfs:subClassOf db:PhaseDePartie .
}
ORDER BY DESC(?phase)
```

?phase
owl:Nothing
db:PhaseDePartie
db:Ouverture
db:Milieu
db:Finale

- Quels sont les programmes d'échecs existants ?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>
```

```
SELECT DISTINCT ?elo ?nom WHERE {
    ?programme a db:Programme .
    ?programme db:elo ?elo .
    ?programme db:nomProgramme ?nom .
}
ORDER BY ?nom
```

?elo	?nom
3554	Berserker 6.0 64Bit
2853	Deep Blue
3518	Fat Fritz 2
3389	Houdini 6
3361	Komodo 14 MCTS 64-bit
3371	Leela Chess Zero
1875	Mephisto Phantom
3545	Stockfish 14
3583	Stockfish+NNUE 150720 64-bit

- Quels programmes d'échecs utilisent des réseaux de neurones ?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```



```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```

```

SELECT DISTINCT ?elo ?nom WHERE {
    ?programme a db:Programme .
    ?programme db:elo ?elo .
    ?programme db:nomProgramme ?nom .
    ?programme db:implemente db:reseauNeuronal
}
ORDER BY ?nom

```

?elo	?nom
3554	Berserker 6.0 64Bit
3518	Fat Fritz 2
3389	Houdini 6
3361	Komodo 14 MCTS 64-bit
3371	Leela Chess Zero
3545	Stockfish 14
3583	Stockfish+NNUE 150720 64-bit

- Quel est le niveau aux échecs d'un individu X ?

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```

```

SELECT DISTINCT ?elo ?nomProgramme ?nom ?prenom WHERE {
    ?joueur a db:Joueur .
    ?joueur db:elo ?elo .
    OPTIONAL { ?joueur db:nomProgramme ?nomProgramme }.
    OPTIONAL { ?joueur db:nom ?nom . ?joueur db:prenom ?prenom } .
}
ORDER BY ?elo

```

?elo	?nomProgramme	?nom	?prenom
1000		Leclercq	Vivien
1875	Mephisto Phantom		
2705		Karpov	Anatoly
2715		Kasparov	Gary
2853	Deep Blue		
3361	Komodo 14 MCTS 64-bit		
3371	Leela Chess Zero		
3389	Houdini 6		
3518	Fat Fritz 2		
3545	Stockfish 14		
3554	Berserker 6.0 64Bit		
3583	Stockfish+NNUE 150720 64-bit		

- Quel est le programme d'échecs le plus puissant en termes de niveau ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX dbo: <http://dbpedia.org/resource/>

PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT DISTINCT ?elo ?nomProgramme WHERE {
    ?programme a db:Programme .
    ?programme db:nomProgramme ?nomProgramme .
    ?programme db:elo ?elo .
}
ORDER BY DESC(?elo)
```

?elo	?nomProgramme
3583	Stockfish+NNUE 150720 64-bit
3554	Berserker 6.0 64Bit
3545	Stockfish 14
3518	Fat Fritz 2
3389	Houdini 6
3371	Leela Chess Zero
3361	Komodo 14 MCTS 64-bit
2853	Deep Blue
1875	Mephisto Phantom

Ici le Reasoner ne connaît pas la commande « LIMIT 1 », il faudrait l'ajouter pour garder seulement le premier résultat.

- Quels ont été les déplacements joués lors de la partie XXX ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
 PREFIX dbo: <http://dbpedia.org/resource/>
 PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT ?deplacement ?san WHERE {
  db:championnatDuMonde1984Finale db:aPourDeplacement ?deplacement .
  ?deplacement db:codeSAN ?san
}
ORDER BY ?deplacement
```

?deplacement	?san
db:championnatDuMonde1984FinaleD1	Nf3 d5
db:championnatDuMonde1984FinaleD10	Bxc4 Qa5
db:championnatDuMonde1984FinaleD11	O-O Bxc3
db:championnatDuMonde1984FinaleD12	Qxc3 Qxc3
db:championnatDuMonde1984FinaleD13	bxc3 Nd7
db:championnatDuMonde1984FinaleD14	c6 bxc6
db:championnatDuMonde1984FinaleD15	Rab1 Nb6
db:championnatDuMonde1984FinaleD16	Be2 c5
db:championnatDuMonde1984FinaleD17	Rfc1 Bb7
db:championnatDuMonde1984FinaleD18	Kf1 Bd5
db:championnatDuMonde1984FinaleD19	Rb5 Nd7
db:championnatDuMonde1984FinaleD2	d4 Nf6
db:championnatDuMonde1984FinaleD20	Ra5 Rfb8
db:championnatDuMonde1984FinaleD21	c4 Bc6
db:championnatDuMonde1984FinaleD22	Ne1 Rb4
db:championnatDuMonde1984FinaleD23	Bd1 Rb7
db:championnatDuMonde1984FinaleD24	f3 Rd8
db:championnatDuMonde1984FinaleD25	Nd3 g5
db:championnatDuMonde1984FinaleD26	Bb3 Kf8
db:championnatDuMonde1984FinaleD27	Nxc5 Nxc5
db:championnatDuMonde1984FinaleD28	Rxc5 Rd6
db:championnatDuMonde1984FinaleD29	Kc2 Kc7

- Quels sont les tournois d'échecs qui ont eu lieu ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
 PREFIX dbo: <http://dbpedia.org/resource/>
 PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT ?nom WHERE {
  ?tournoi a db:TournoiDEchecs .
  ?tournoi db:nomTournoi ?nom
}
ORDER BY ?nom
```

?nom
CCRL BLitz 2020
CCRL BLitz 2021
Championnat du Monde 1984

- Quelles parties ont été jouées lors du tournoi XXX ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
 PREFIX dbo: <http://dbpedia.org/resource/>
 PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```

SELECT DISTINCT ?partie ?date ?lieu ?nomBlanc ?nomNoir ?score WHERE {
  db:CCRL_Blitz_2021 db:aPourPartie ?partie .

  ?partie db:aPourLieu ?lieu .

  ?partie db:aPourTemps ?temps .
  ?temps db:date ?date .

  ?partie db:aPourJoueurBlanc ?joueurBlanc .
  ?joueurBlanc db:nomProgramme ?nomBlanc .
  ?partie db:aPourJoueurNoir ?joueurNoir .
  ?joueurNoir db:nomProgramme ?nomNoir .

  ?partie db:aPourResultat ?resultat .
  ?resultat db:encodeeEnSAN ?score .
}
ORDER BY ?partie
  
```

?partie	?date	?lieu	?nomBlanc	?nomNoir	?score
db:CCRL_Blitz_2021_604.4.985	16/09/2021	db:CCRL	Leela Chess Zero	Stockfish 14	1/2-1/2
db:CCRL_Blitz_2021_611.6.123	04/11/2021	db:CCRL	Berserker 6.0 64Bit	Leela Chess Zero	1/2-1/2
db:CCRL_Blitz_2021_611.6.139	04/11/2021	db:CCRL	Leela Chess Zero	Berserker 6.0 64Bit	1-0
db:CCRL_Blitz_2021_611.6.155	04/11/2021	db:CCRL	Berserker 6.0 64Bit	Leela Chess Zero	1/2-1/2
db:CCRL_Blitz_2021_611.6.171	05/11/2021	db:CCRL	Leela Chess Zero	Berserker 6.0 64Bit	1-0

- Combien de fois le joueur X a-t-il gagné contre le joueur Y ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX dbo: <http://dbpedia.org/resource/>
 PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT DISTINCT ?partie WHERE {
  {
    ?partie db:aPourJoueurBlanc db:leelaChessZero .
    ?partie db:aPourJoueurNoir db:berserk6 .

    ?partie db:aPourResultat ?resultat .
    ?resultat db:encodeeEnSAN ?san .
    FILTER (?san = "1-0")
  }
  UNION
  {
    ?partie db:aPourJoueurBlanc db:berserk6 .
    ?partie db:aPourJoueurNoir db:leelaChessZero .

    ?partie db:aPourResultat ?resultat .
    ?resultat db:encodeeEnSAN ?san .
    FILTER (?san = "0-1")
  }
}
ORDER BY ?partie
```

?partie
db:CCRL_Blitz_2021_611.6.139
db:CCRL_Blitz_2021_611.6.171

Ici le résultat correspond au nombre de ligne. Le Reasoner ne prend pas en charge la fonction COUNT.

- Quelles sont les cases qui composent l'échiquier ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
 PREFIX dbo: <http://dbpedia.org/resource/>
 PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT ?nomCase WHERE {
  db:echiquierStandard db:estComposeDe ?case .
  ?case db:nomCase ?nomCase
}
ORDER BY ?nomCase
```

?nomCase
A1
A2
A3
A4
A5
A6
A7
A8
B1
B2
B3
B4
B5
B6
B7
B8
C1
C2
C3
C4
D1
D5

- Quelles sont les pièces majeures aux échecs ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX dbo: <http://dbpedia.org/resource/>

PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT DISTINCT ?x ?valeur ?nomCase WHERE {
    ?x a db:PieceMajeure .
    ?x db:aPourValeur ?valeur .
    ?x db:aPourCase ?case .
    ?case db:nomCase ?nomCase
}
ORDER BY ?x
```

?x	?valeur	?nomCase
db:dameBlanche	8.8	D1
db:dameNoire	8.8	D8
db:tourBlanche	5.1	H1
db:tourBlanche	5.1	A1
db:tourNoire	5.1	A8
db:tourNoire	5.1	H8

On observe ici qu'il y a bien 4 tours et 2 dames au début d'une patie.

- Quelles parties d'échecs ont été jouées en ligne ?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>
```

```
SELECT DISTINCT ?partie ?date ?lieu ?nomBlanc ?nomNoir ?score WHERE {
    ?partie a db:PartieDEchecs .
    ?partie db:seJoueSur db:echiquierEnLigne .

    ?partie db:aPourLieu ?lieu .

    ?partie db:aPourTemps ?temps .
    ?temps db:date ?date .

    ?partie db:aPourJoueurBlanc ?joueurBlanc .
    ?joueurBlanc db:nomProgramme ?nomBlanc .
    ?partie db:aPourJoueurNoir ?joueurNoir .
    ?joueurNoir db:nomProgramme ?nomNoir .

    ?partie db:aPourResultat ?resultat .
    ?resultat db:encodeeEnSAN ?score .
}
ORDER BY ?partie
```

?partie	?date	?lieu	?nomBlanc	?nomNoir	?score
db:CCRL_Blitz_2020_541.2.521	20/07/2020	db:CCRL	Stockfish+NNUE 150...	Komodo 14 MCTS ...	1/2-1/2
db:CCRL_Blitz_2020_541.2.522	20/07/2020	db:CCRL	Komodo 14 MCTS 6...	Stockfish+NNUE 1...	0-1
db:CCRL_Blitz_2021_604.4.788	15/09/2021	db:CCRL	Fat Fritz 2	Leela Chess Zero	1/2-1/2
db:CCRL_Blitz_2021_604.4.985	16/09/2021	db:CCRL	Leela Chess Zero	Stockfish 14	1/2-1/2
db:CCRL_Blitz_2021_611.6.123	04/11/2021	db:CCRL	Berserker 6.0 64Bit	Leela Chess Zero	1/2-1/2
db:CCRL_Blitz_2021_611.6.139	04/11/2021	db:CCRL	Leela Chess Zero	Berserker 6.0 64Bit	1-0
db:CCRL_Blitz_2021_611.6.155	04/11/2021	db:CCRL	Berserker 6.0 64Bit	Leela Chess Zero	1/2-1/2
db:CCRL_Blitz_2021_611.6.171	05/11/2021	db:CCRL	Leela Chess Zero	Berserker 6.0 64Bit	1-0

- Quels ont été les rapports de partie écrits par l'individu X ?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
```

PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT DISTINCT ?rapport WHERE {  
    ?rapport a db:RapportDePartie .  
    ?rapport db:aPourAuteur db:vivienLeclercq  
}  
ORDER BY ?rapport
```

?rapport
db:CCRL_Blitz_2020_541.2.521Rapport
db:CCRL_Blitz_2020_541.2.522Rapport
db:CCRL_Blitz_2021_604.4.788Rapport
db:CCRL_Blitz_2021_604.4.973Rapport
db:CCRL_Blitz_2021_604.4.985Rapport
db:CCRL_Blitz_2021_611.6.123Rapport
db:CCRL_Blitz_2021_611.6.139Rapport
db:CCRL_Blitz_2021_611.6.155Rapport
db:CCRL_Blitz_2021_611.6.171Rapport
db:championnatDuMonde1984FinaleRapport

- Quelles parties ont utilisés l'ouverture XXX ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX dbo: <http://dbpedia.org/resource/>

PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```
SELECT DISTINCT ?partie ?date ?lieu ?nomBlanc ?nomNoir ?score ?eco  
WHERE {
```

```
    ?partie a db:PartieDEchecs .
```

```
    ?partie db:aPourOuverture ?ouverture .
```

```
    ?ouverture db:eco ?eco .
```

```
    FILTER (?eco = "C19") .
```

```
    ?partie db:aPourLieu ?lieu .
```

```
    ?partie db:aPourTemps ?temps .
```

```
    ?temps db:date ?date .
```

```
    ?partie db:aPourJoueurBlanc ?joueurBlanc .
```

```
    ?joueurBlanc db:nomProgramme ?nomBlanc .
```

```
    ?partie db:aPourJoueurNoir ?joueurNoir .
```

```
    ?joueurNoir db:nomProgramme ?nomNoir .
```

```
    ?partie db:aPourResultat ?resultat .
```

```
    ?resultat db:encodeeEnSAN ?score .
```


}

ORDER BY ?partie

?partie	?date	?lieu	?nomBlanc	?nomNoir	?score	?eco
db:CCRL_Blitz_202...	04/11/2021	db:CCRL	Berserker 6.0 64Bit	Leela Chess Zero	1/2-1/2	C19
db:CCRL_Blitz_202...	04/11/2021	db:CCRL	Leela Chess Zero	Berserker 6.0 64Bit	1-0	C19

Exemple ici avec l'ouverture C19.

- Quelles parties se sont jouées le DD/MM/YYYY ?

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX dbo: <http://dbpedia.org/resource/>

PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

SELECT DISTINCT ?partie ?date ?lieu ?nomBlanc ?nomNoir ?score ?eco

WHERE {

 ?partie a db:PartieDEchecs .

 ?partie db:aPourOuverture ?ouverture .

 ?ouverture db:eco ?eco .

 ?partie db:aPourLieu ?lieu .

 ?partie db:aPourTemps ?temps .

 ?temps db:date ?date .

 FILTER(?date = "20/07/2020") .

 ?partie db:aPourJoueurBlanc ?joueurBlanc .

 ?joueurBlanc db:nomProgramme ?nomBlanc .

 ?partie db:aPourJoueurNoir ?joueurNoir .

 ?joueurNoir db:nomProgramme ?nomNoir .

 ?partie db:aPourResultat ?resultat .

 ?resultat db:encodeeEnSAN ?score .

}

ORDER BY ?partie

?partie	?date	?lieu	?nomBlanc	?nomNoir	?score	?eco
db:CCRL_Blitz_202...	20/07/2020	db:CCRL	Stockfish+NNUE 1...	Komodo 14 MCTS...	1/2-1/2	A43
db:CCRL_Blitz_202...	20/07/2020	db:CCRL	Komodo 14 MCTS ...	Stockfish+NNUE 1...	0-1	C37

- Quelles sont les pièces existantes aux échecs ?

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```

```

SELECT ?label ?definition WHERE {
    db:pieces skos:member ?piece .
    ?piece skos:definition ?definition .
    FILTER(lang(?definition) = "fr") .
    ?piece skos:prefLabel ?label .
    FILTER(lang(?label) = "fr")
}

```

?label	?definition
Roi@fr	Pièce qu'il faut à tout prix protéger pour ne pas perdre et qu'il faut mettre en échecs et mat pour gagner. Le roi est capable de se déplacer d'une case dans tou...
Cavalier@fr	Pièce qui permet de se déplacer en "L". Au nombre de 2 pour chaque joueur au début de la partie.@fr
Fou@fr	Pièce qui permet de se déplacer en diagonale. Au nombre de 2 pour chaque joueur au début de la partie.@fr
Pion@fr	Pièce qui permet de se déplacer en ligne droite. Au nombre de 8 pour chaque joueur au début de la partie.@fr
Tour@fr	Pièce majeure qui permet de se déplacer en ligne. Au nombre de 2 pour chaque joueur au début de la partie.@fr
Reine@fr	Pièce majeure qui permet de se déplacer en ligne ou en diagonale.@fr

Cette requête utilise le vocabulaire SKOS et produit les résultats en français seulement.

- Quelles sont les structures des programmes d'échecs existantes ?

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbo: <http://dbpedia.org/resource/>
PREFIX db: <http://www.semanticweb.org/vivien/programmesdechecs#>

```

```

SELECT ?label ?definition WHERE {
    db:structuresDesProgrammesDEchecs skos:member ?struct .
    ?struct skos:definition ?definition .
    FILTER(lang(?definition) = "fr") .
    ?struct skos:prefLabel ?label .
    FILTER(lang(?label) = "fr")
}

```

?label	?definition
Réseau Neuronal@fr	Structure appartenant au domaine de l'apprentissage statistique ou de l'apprentissage profond qui cherche à imiter le comportement du cerveau...
Table de Finales@fr	Structure interne à un algorithme qui regroupe un ensemble de coup à jouer pour terminer une partie étant donnée une situation donnée.@fr
Elagage Alpha-Bêta@fr	Concept qui cherche à éliminer des parties d'un arbre de possibilités d'un jeu afin d'éviter d'avoir à calculer des situations abérrantes.@fr
Méthode de Monte-Carlo@fr	Méthode d'optimisation qui a pour but de jouer aléatoirement la fin d'une partie afin de trouver le meilleur coup possible.@fr
Algorithme MinMax@fr	Algorithme cherchant à trouver le meilleur chemin dans un arbre de jeu dans le but de maximiser le score d'un joueur et de minimiser le score ...

Cette requête utilise le vocabulaire SKOS et produit les résultats en français seulement.

Conclusion

Ainsi se termine l'élaboration de l'ontologie sur les échecs et les programmes d'échecs. L'ontologie permet de répondre aux questions que nous nous sommes posées au début de ce rapport et rempli ainsi le cahier des charges.

Pour aller plus loin, il sera intéressant d'élaborer un outil capable de collecter des fichiers PGN et de les intégrer directement dans la base de connaissance afin de faire de réelles statistiques sur le monde des échecs. Il s'agira simplement de peupler automatiquement l'ontologie et non de modifier sa structure.