

RAPPORT

NETLINK

Alexandre AUDA

Table des matières

RAPPORT	1
Introduction:	3
1- Explication générale et objectif du projet:.....	4
1.1-Objectif général du projet NetLink:	4
1.2- Mise en œuvre sur un exemple:	4
Premier scénario:	5
Deuxième scénario:	6
1.3- Spécificité du projet NetLink:	7
2- Explication du modèle:.....	8
2.1-Explication générale du modèle:.....	8
2.1.1-Description générale des Classes:.....	8
2.1.2-Description générale des propriétés:.....	9
2.2-Utilisation de l'inférence:	9
2.2.1-Utilisation générale de l'inférence dans NetLink:	9
2.2.2-Illustration de l'utilisation de l'inférence dans NetLink:	11
Premier scénario:	17
Deuxième scénario:	18
3- NetLink online:	19
3.1-Explication générale de l'alimentation de la base de connaissance de NetLink par le Web:	19
3.2-Illustration de l'utilisation du Web dans NetLink:	20
4. Implémentation des stratégies de recommandation dans NetLink:	24
Premier scénario:	26

Deuxième scénario:	29
5. Conclusion:.....	31

Introduction:

Dans ce projet, l'objectif est de construire une application de notre choix en utilisant la puissance des techniques issues du Web-Sémantique. En outre, nous attacherons une attention particulière au fonctionnement de ces diverses techniques et mettrons en lumière les différentes possibilités qu'elles offrent.



1- Explication générale et objectif du projet:

1.1-Objectif général du projet NetLink:

En premier lieu, l'objectif du projet NetLink est de fournir une plateforme offrant un service d'aide à la recommandation professionnelle. Plus précisément, cette application a pour but de fournir un outil d'aide à la décision pour parvenir à se faire recommander professionnellement en jouant sur les connaissances et la nature des relations existantes entre les personnes.

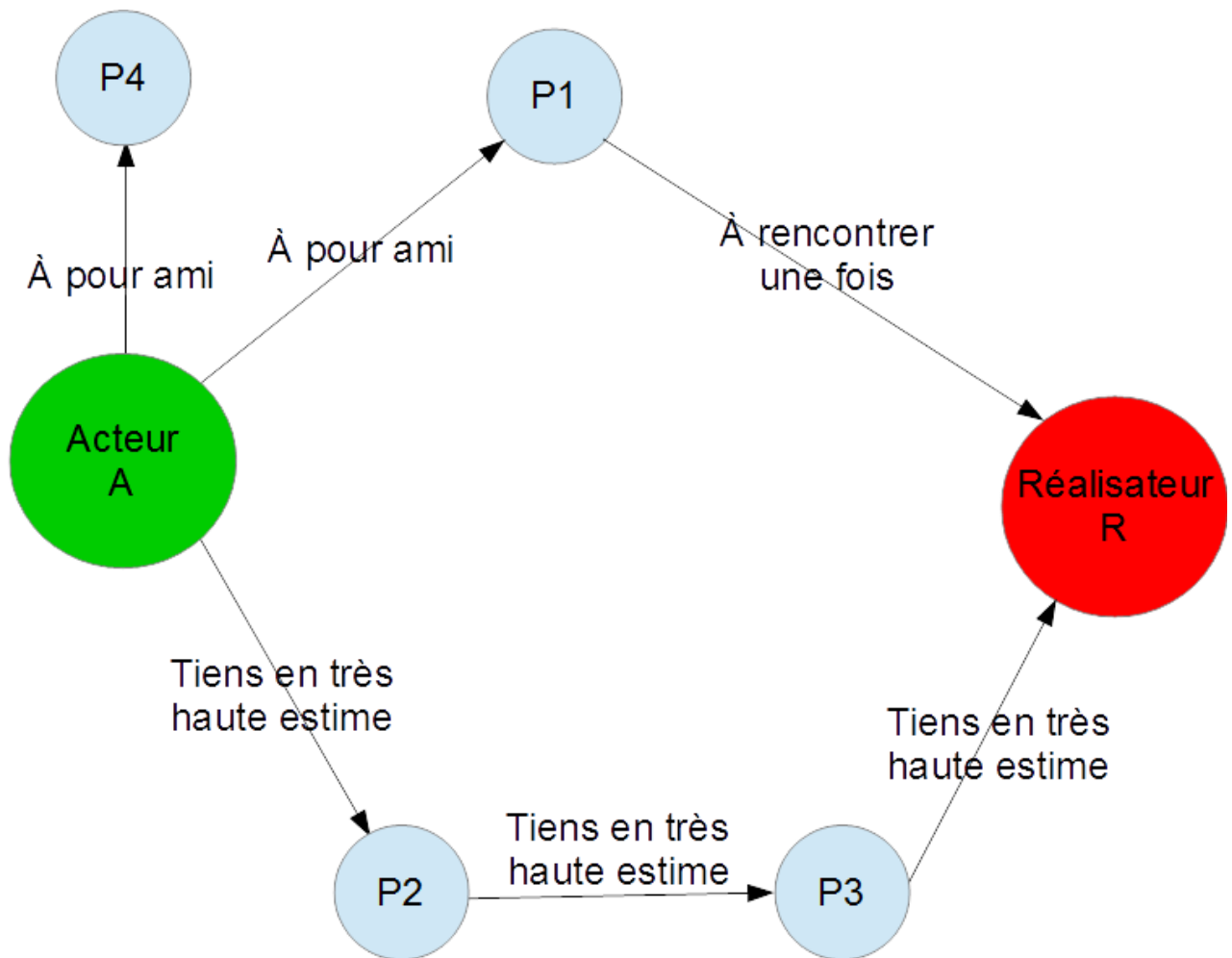
De plus, l'utilisateur sera en mesure de paramétrer la stratégie de recommandation souhaitée et l'application fournira à l'utilisateur les réponses en fonction de cette dernière.

1.2- Mise en œuvre sur un exemple:

Prenons un scénario simple pour illustrer cela:

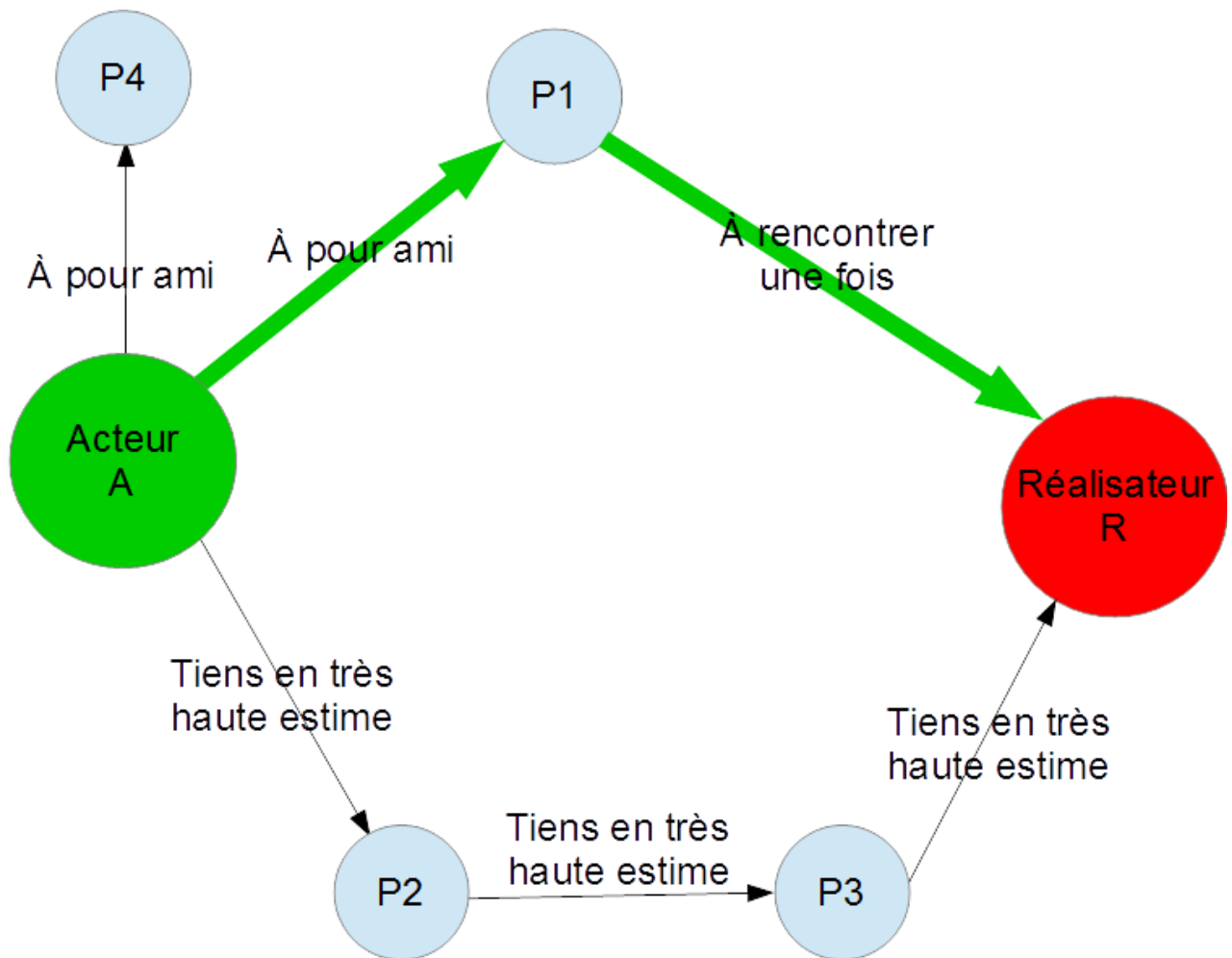
-Un acteur A souhaite jouer le premier rôle dans un film d'un grand réalisateur R. Pour cela, il souhaite se faire recommander par des connaissances du réalisateur pour augmenter ses chances d'obtenir le rôle.

Or, les relations dans l'entourage de l'acteur A et du réalisateur R sont les suivantes:



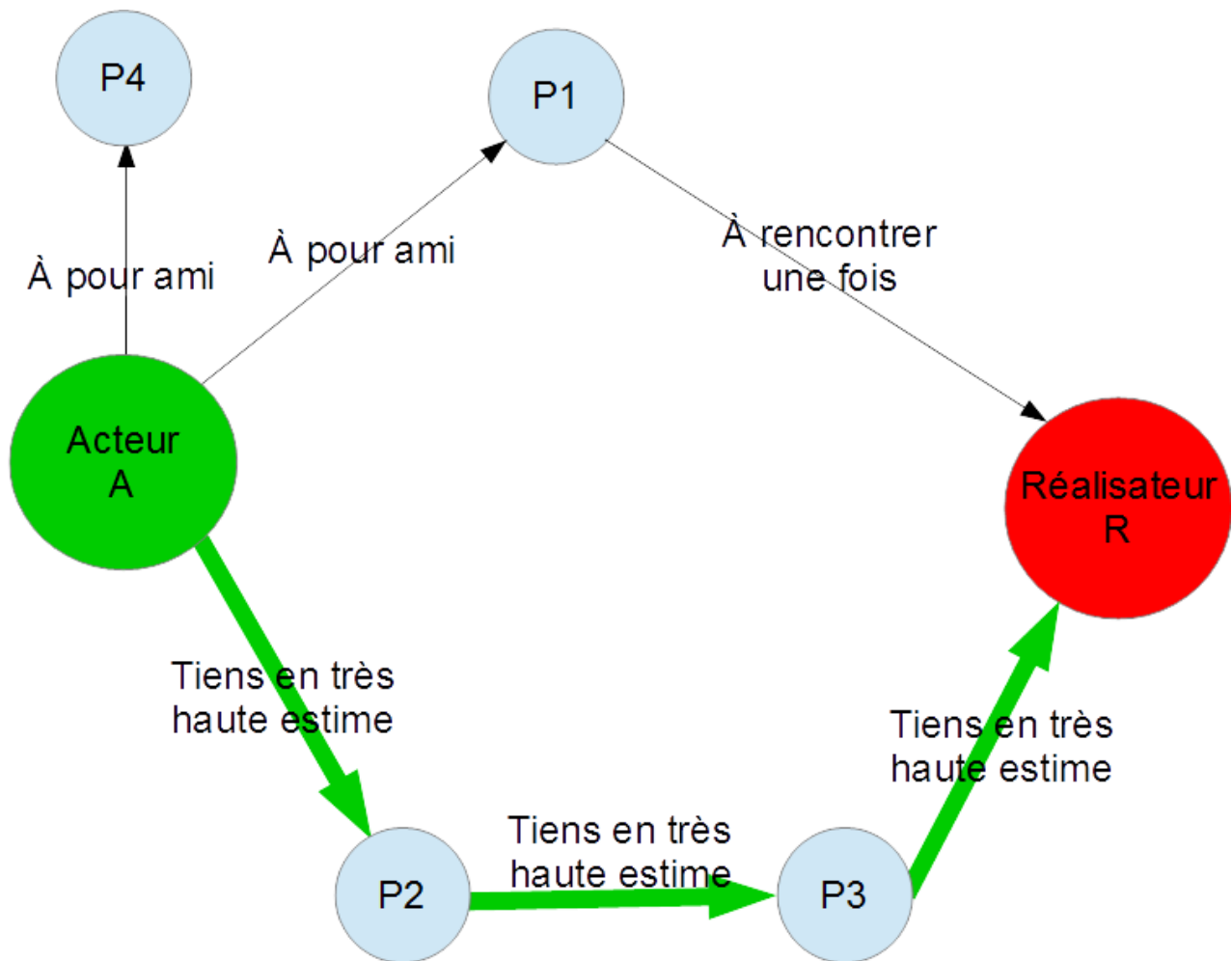
Premier scénario:

-Dans un premier scénario, l'acteur A adopte une stratégie de recommandation consistant à passer par le moins d'intermédiaire possible sans se préoccuper de la nature des relations qui lient les personnes. Dans ce cas, l'application lui conseillera le chemin suivant (chemin représenté en vert):



Deuxième scénario:

-Dans un deuxième scénario, l'acteur A adopte une stratégie de recommandation consistant à passer par le chemin où les relations sont les plus fortes. Dans ce cas, l'application lui conseillera le chemin suivant (chemin représenté en vert):



1.3- Spécificité du projet NetLink:

Ainsi, l'application sera en mesure de s'adapter dynamiquement en fonction des choix de l'utilisateur et le conseillera au mieux en fonction de ses besoins.

En effet, les possibilités de stratégie de recommandation sont infinis et l'utilisateur sera en mesure d'effectuer des requêtes avancées telles que: « *Trouve un chemin de recommandation passant par le moins d'intermédiaires possible mais dont la force de la dernière liaison est au moins égale à "tenir en très haute estime" ».*

De plus, la nature des relations entre les différentes personnes pouvant varier au cours du temps, la gestion de ces changements devra s'effectuer de manière dynamique.

Ainsi, si la nature d'une relation se modifie avec le temps ou si de nouvelles relations se créent, ces modifications devront être prises en compte sans aucune intervention extérieure quelconque.

Aussi, l'application aura recours de manière importante à la puissance de l'inférence pour permettre d'effectuer ses recherches même sur un jeu de données très incomplet. Pour plus de détails, veuillez-vous reporter à la section [correspondante](#).

Enfin, la base de connaissance de l'application NetLink pourra également être relié et alimenté par le Web, en l'occurrence ici par DBpedia, de sorte que l'application puisse adapter sa recherche et se mettre à jour en temps réel. Pour plus de détails, veuillez-vous reporter à la section [correspondante](#).

2- Explication du modèle:

2.1-Explication générale du modèle:

2.1.1-Description générale des Classes:

Dans un premier temps, nous nous limiterons à des Personnes et notamment des Personnes en lien avec le monde du cinéma.

Ainsi, nous aurons des classes telles que des « Acteurs », des « réalisateurs », des « producteurs »,

2.1.2-Description générale des propriétés:

Nous associons ensuite des propriétés à ces classes.

Ces propriétés sont principalement relatives à la description de la nature d'une relation qui lie deux personnes.

Nous pouvons distinguer dans ces propriétés deux groupes principaux:

- Les propriétés décrivant de manière concrète la nature d'une relation qui lie deux personnes.
- Les propriétés permettant d'associer à la nature d'une relation, un poids. Les propriétés décrivant de manière concrète la nature d'une relation héritent de ces propriétés.

Ce modèle va nous permettre de pouvoir traduire de façon quantifiable pour un programme la force d'une relation et ainsi pouvoir « réfléchir » en fonction de la force d'une relation, chose qu'un programme ne pourrait effectuer en temps normal.

Pour plus de détails concernant le fonctionnement de ce modèle, veuillez-vous reporter à la section [suivante](#).

2.2-Utilisation de l'inférence:

2.2.1-Utilisation générale de l'inférence dans NetLink:

Dans cette section, nous allons détailler notre utilisation d'un concept fondamental pour notre projet: l'inférence.

En effet, notre utilisation importante du concept de l'inférence dans cette application s'effectue à plusieurs niveaux et pour plusieurs raisons.

La première provient de la probabilité importante d'un jeu de données initiale très incomplet.

En effet par l'utilisation importante de propriété Owl par exemple, nous extrayons un nouveau savoir à partir d'un savoir existant ce qui nous permet de compléter un graphe qui serait très incomplet. Pour une illustration concrète de ce cas de figure, veuillez-vous reporter à la section [suivante](#).

La seconde utilisation importante de l'inférence provient de la modélisation de la force d'une relation à l'aide de poids.

En effet, à partir d'un savoir existant, ici la nature d'une relation, nous en déduisons un savoir nouveau, ici une traduction mathématique de la force de cette relation.

Ainsi, si une personne x a rencontré une fois la personne y , nous allons associer à cette relation un poids de 1.

Cela peut se traduire de la manière suivante :

$$x R_1 y \rightarrow x R_2 y$$

Avec R_1 la propriété « a rencontré une fois la personne » et R_2 la propriété « a une relation de poids 1 avec la personne ». Ici, R_1 et R_2 sont des propriétés symétriques.

De même, si une personne x est amie avec la personne y , nous allons associer à cette relation un poids de 3.

Cela peut se traduire de la manière suivante :

$$x R_1 y \rightarrow x R_2 y$$

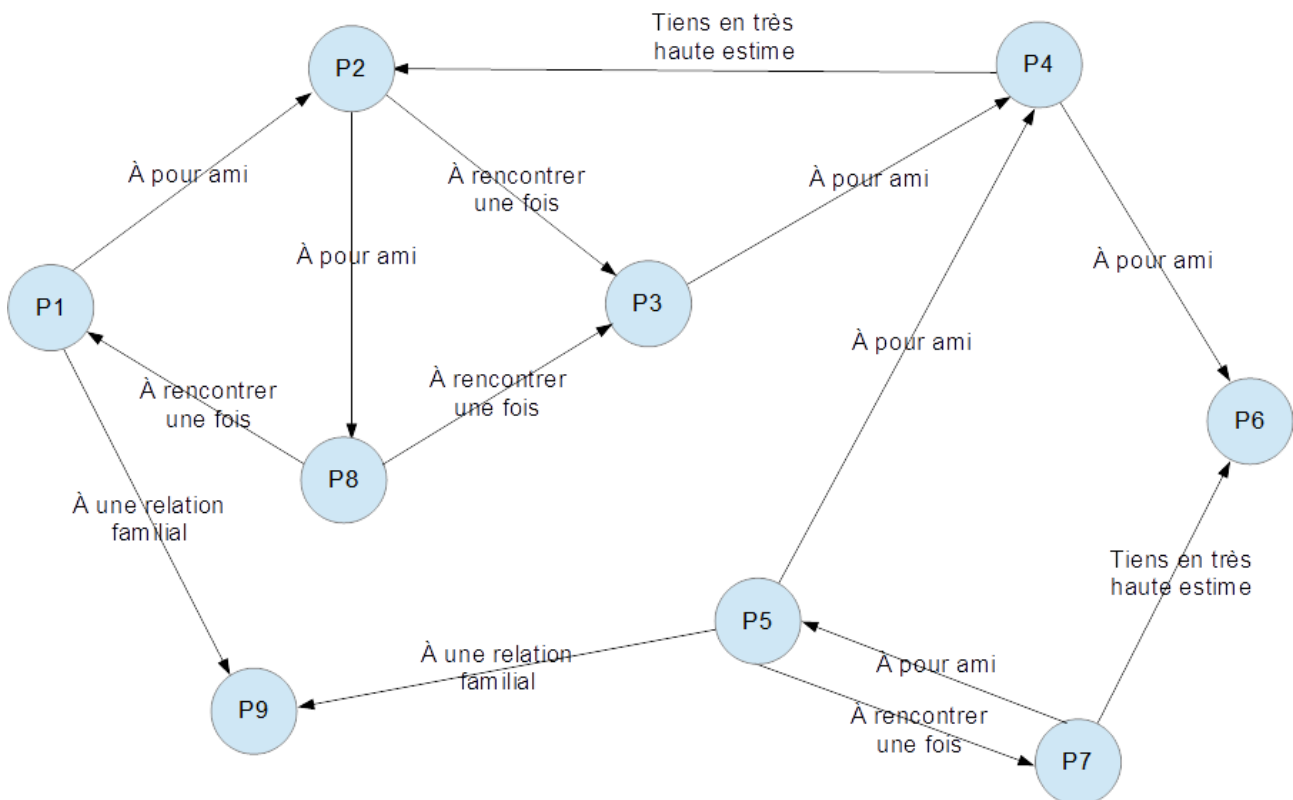
Avec R_1 la propriété « est ami avec la personne » et R_2 la propriété « a une relation de poids 3 avec la personne ». Ici, R_1 et R_2 sont des propriétés symétriques.

Pour une illustration concrète de ce cas de figure, veuillez-vous reporter à la section [suivante](#).

2.2.2-Illustration de l'utilisation de l'inférence dans NetLink:

Dans cette partie, nous allons présenter une illustration du fonctionnement de NetLink sur un exemple simple.

En effet, supposons que l'on est initialement le graphe orienté suivant:



Les nœuds du graphe P_1, P_2, \dots, P_n sont des personnes.

Les arêtes représentent la nature des relations entre ces personnes.

Supposons maintenant que l'acteur P1 souhaite se faire recommander professionnellement auprès du réalisateur P7.

Avec le graphe actuel, l'acteur P1 ne possède pas de chemin sur son réseau de connaissance lui permettant de se faire recommander auprès du réalisateur P7.

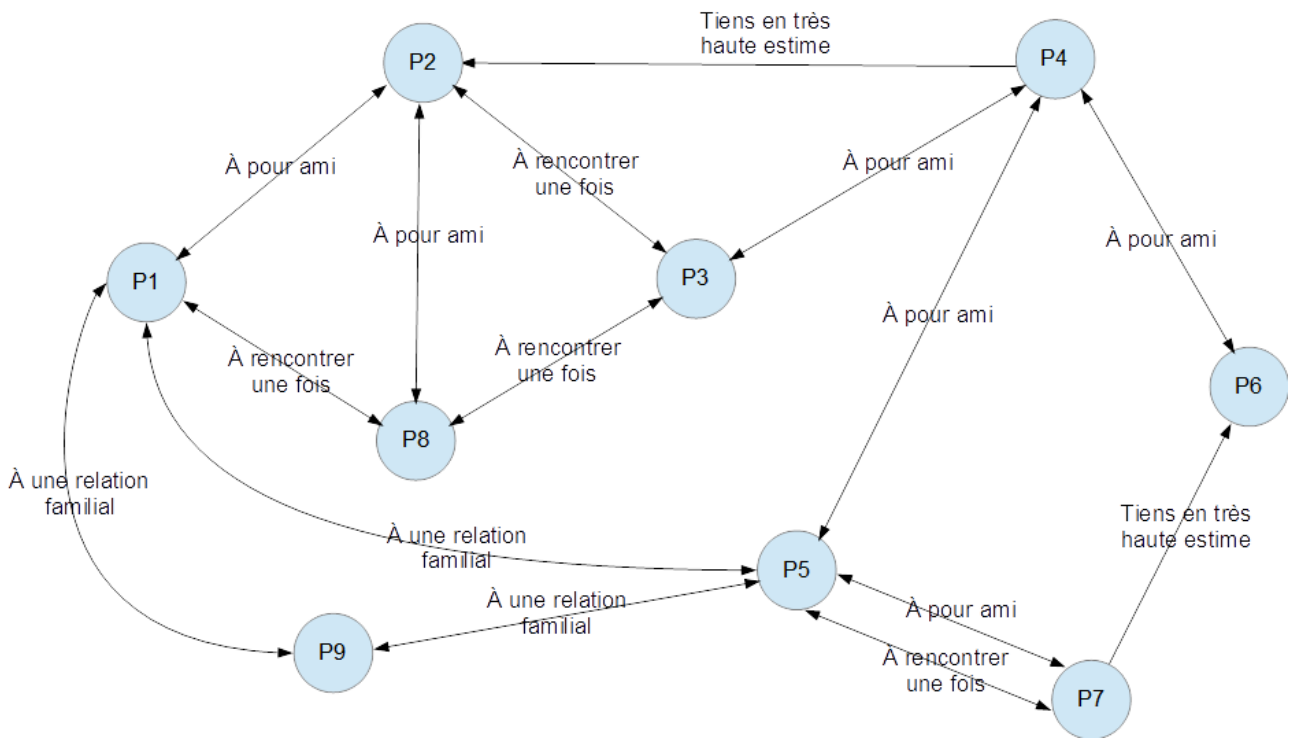
Nous allons donc ici faire usage de la puissance de l'inférence pour compléter notre graphe orienté, difficilement exploitable en l'état.

En effet, nous savons par exemple que si une personne x est ami avec une personne y alors y est ami avec la personne x . On dit alors que la propriété « est ami » est *symétrique*.

Il en va de même pour les propriétés « a rencontré une fois » ou « a une relation familiale ». En revanche, nous supposons que la propriété « tiens en très haute estime » n'est pas forcément symétrique.

De plus, nous savons également que la propriété « a une relation familiale » est une propriété transitive. En effet, si une personne x « a une relation familiale » avec la personne y et que la personne y « a une relation familiale » avec la personne z alors la personne x « a une relation familiale » avec la personne z .

Ainsi, nous pouvons donc, par inférence, transformer le graphe précédent de la manière suivante:



Nous pouvons voir ici que nous avons pu transformer et compléter notre graphe orienté initiale grâce à l'inférence.

En effet, à partir d'un savoir initial nous avons pu en extraire un savoir nouveau, permettant ainsi de rendre notre modélisation plus exploitable.

Cependant, nous pouvons poursuivre cette démarche pour aller plus loin.

Nous avons rendu, grâce à l'inférence, notre graphe initial beaucoup plus exploitable, mais nous voulons maintenant parvenir à quantifier la force des relations qui lient les personnes en nous basant sur des postulats.

Il est en effet très probable qu'il sera beaucoup plus facile d'obtenir des recommandations très favorables de la part d'amis que de personnes que l'on a rencontré qu'une fois par exemple.

Nous allons donc traduire cette notion en nous servant une nouvelle fois du concept d'inférence.

Ici, nous allons pondérer notre graphe orienté pour exprimer la force d'une relation et ainsi obtenir de nouvelles connaissances à partir d'un savoir initial.

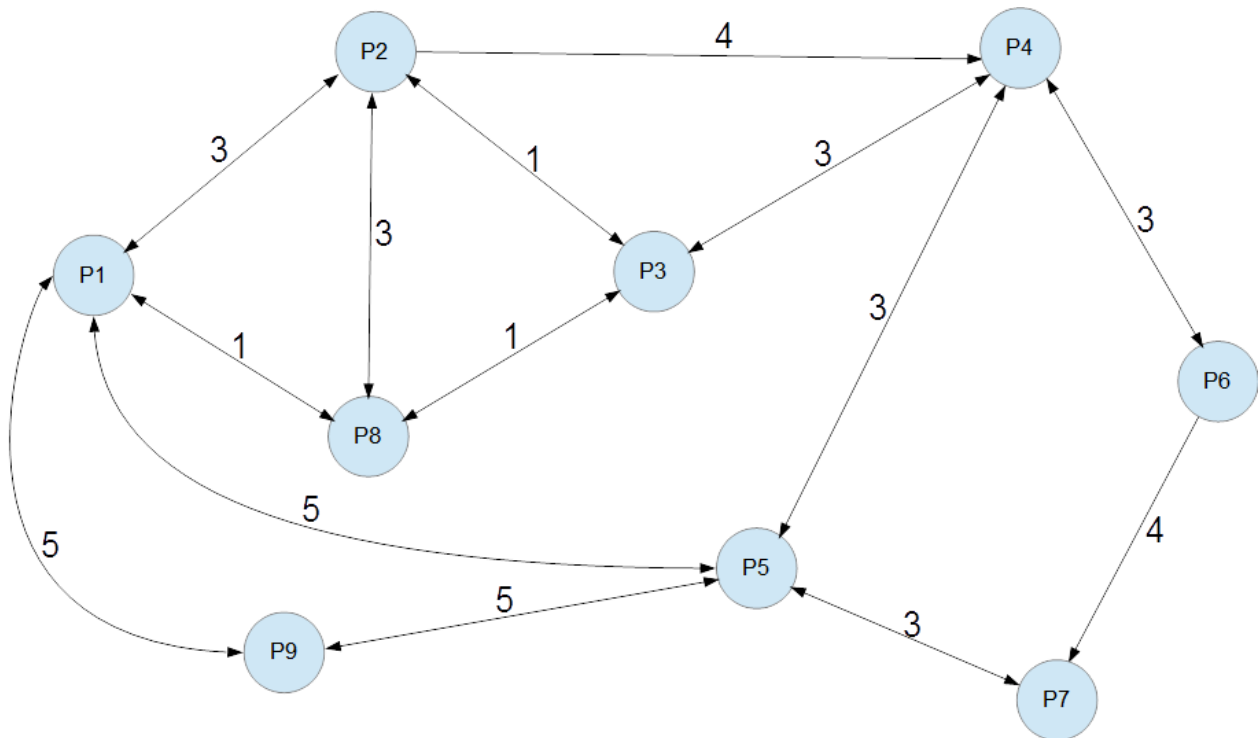
Nous traduirons dans cet exemple la force des relations de la manière suivante:

- La relation « a rencontré une fois » implique la relation symétrique « a une relation de poids 1 »
- La relation « a pour ami » implique la relation symétrique « a une relation de poids 3 »
- La relation x « tiens en très haute estime » la personne y implique la relation y « a une relation de poids 4 » avec la personne x
- La relation « a une relation familial » implique la relation symétrique « a une relation de poids 5 »

Ici plus les poids sont fort, plus cela traduit une relation forte.

Il est à noter que l'on aurait très bien pu prendre l'inverse et considérer que plus les poids sont faible, plus cela traduit une relation forte. En pratique, nous avons pris en compte les deux possibilités, ce qui nous permet de choisir indifféremment une stratégie de pondération du graphe orienté pour nous permettre une plus grande souplesse lors de l'application ultérieure d'algorithmes de graphe.

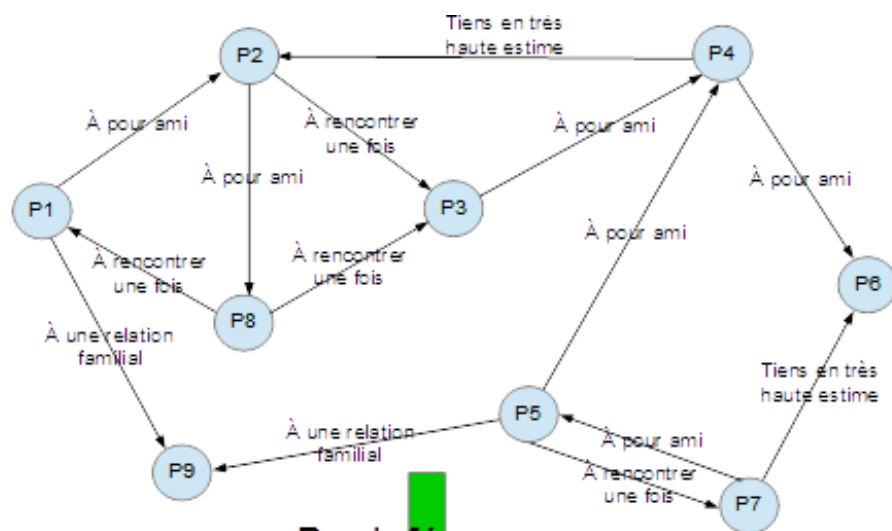
Nous obtenons ainsi par inférence le graphe orienté pondéré suivant:



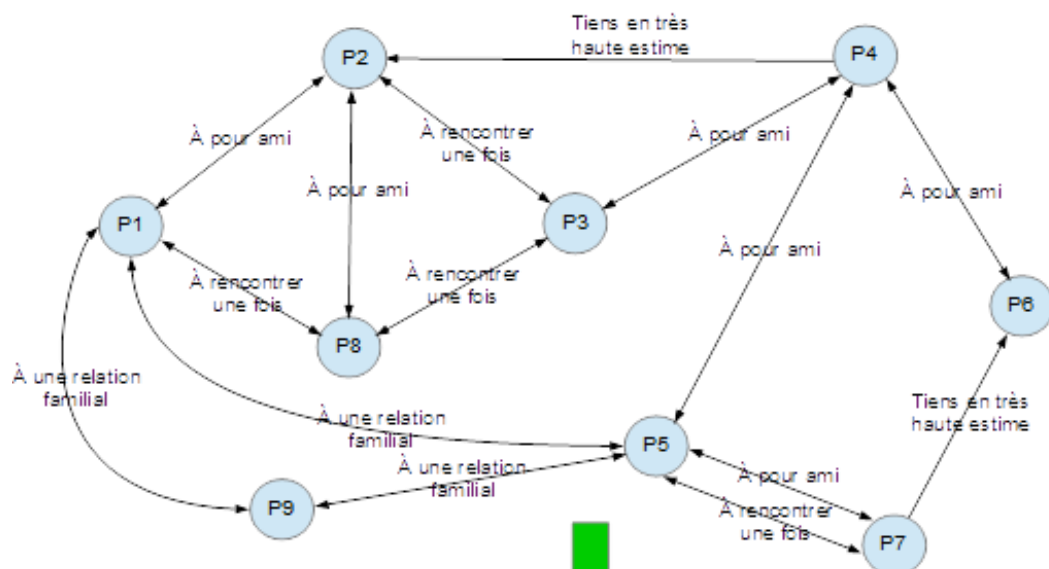
Ainsi, nous pouvons voir que par l'utilisation de l'inférence nous avons pu successivement extraire un savoir nouveau à partir d'un jeu de données initiales très incomplet.

Aussi, nous obtenons par ce biais une représentation facilement exploitable et dont nous disposons d'un grand nombre d'algorithmes se basant sur cette représentation à partir d'un modèle inexploitable initialement.

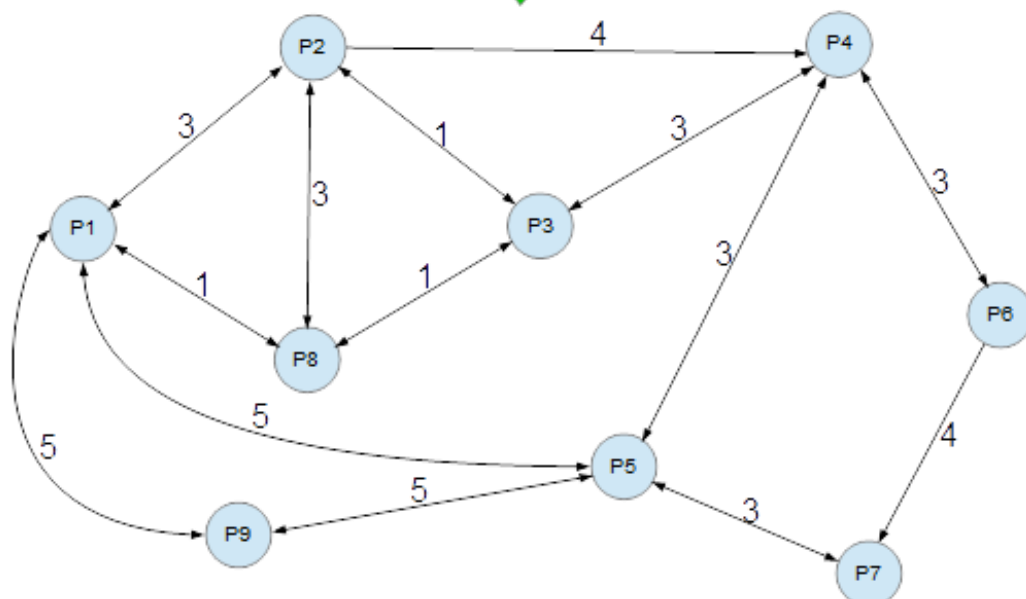
Nous pouvons donc récapituler les étapes successives de la manière suivante:



Par inférence



Par inférence



Il est à noter que sur le dernier graphe, les relations présentent sur le graphe précédent n'ont pas disparu et la pondération du graphe vient s'ajouter aux relations précédentes et non les substituer. Cependant, par soucis de clarté, nous ne présentons sur ce dernier graphe que les nouvelles informations déduites par inférence, à savoir la pondération du graphe permettant de traduire la force des relations entre les personnes.

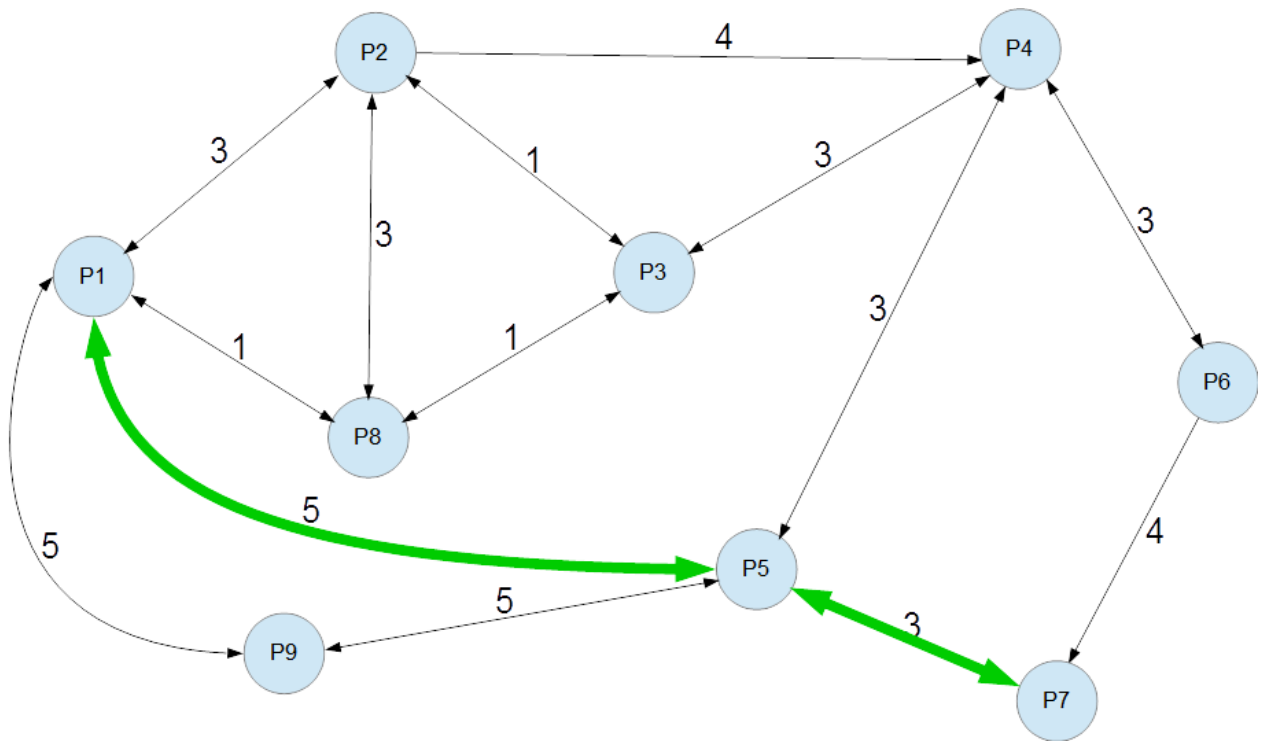
Avec ce modèle obtenu, nous sommes maintenant en capacité d'utiliser toute la puissance des algorithmes sur les graphes orientés pondérés pour pouvoir proposer différentes solutions en fonction de la stratégie de recommandation de l'utilisateur.

Ainsi, pour en revenir à notre exemple où l'acteur P1 souhaite se faire recommander professionnellement auprès du réalisateur P7, nous pouvons avoir divers scénarios:

Premier scénario:

Supposons que l'acteur P1 adopte une stratégie de recommandation consistant à passer par le moins d'intermédiaires sans se soucier de la nature des relations qui existe entre les personnes.

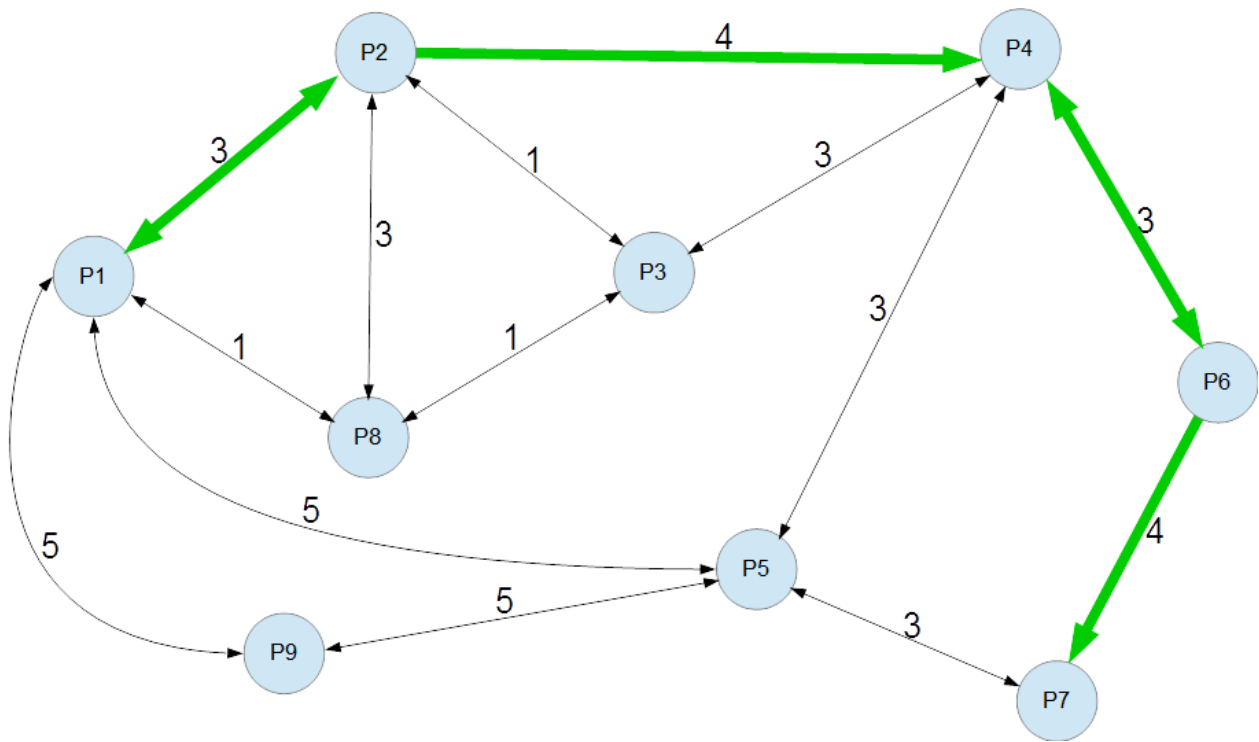
Dans ce cas, l'application proposera à l'acteur P1 le chemin suivant (chemin présenté en vert):



Deuxième scénario:

Supposons maintenant que l'acteur P1 adopte une stratégie de recommandation consistant à passer par le moins d'intermédiaires possible et où la dernière personne recommandant l'acteur P1 doit avoir la relation la plus forte avec le réalisateur P7.

Dans ce cas, l'application proposera à l'acteur P1 le chemin suivant (chemin présenté en vert):



Ainsi, nous pouvons voir que l'application NetLink parvient à s'adapter en fonction de la stratégie de recommandation professionnelle choisie par l'utilisateur.

3- NetLink online:

3.1-Explication générale de l'alimentation de la base de connaissance de NetLink par le Web:

La base de connaissance de l'application NetLink a également pour ambition d'être reliée et alimentée par le Web, en l'occurrence ici par DBpedia, de sorte que l'application puisse adapter sa recherche et se mettre à jour en temps réel.

En effet, la nature des relations entre les différentes personnes pouvant varier au cours du temps, la gestion de ces changements devra s'effectuer de manière dynamique.

Ainsi, si la nature d'une relation se modifie avec le temps ou si de nouvelles relations se créent ou disparaissent, ces modifications devront être prises en compte sans aucune intervention extérieure quelconque et l'utilisation du Web comme base de connaissance sera une caractéristique puissante.

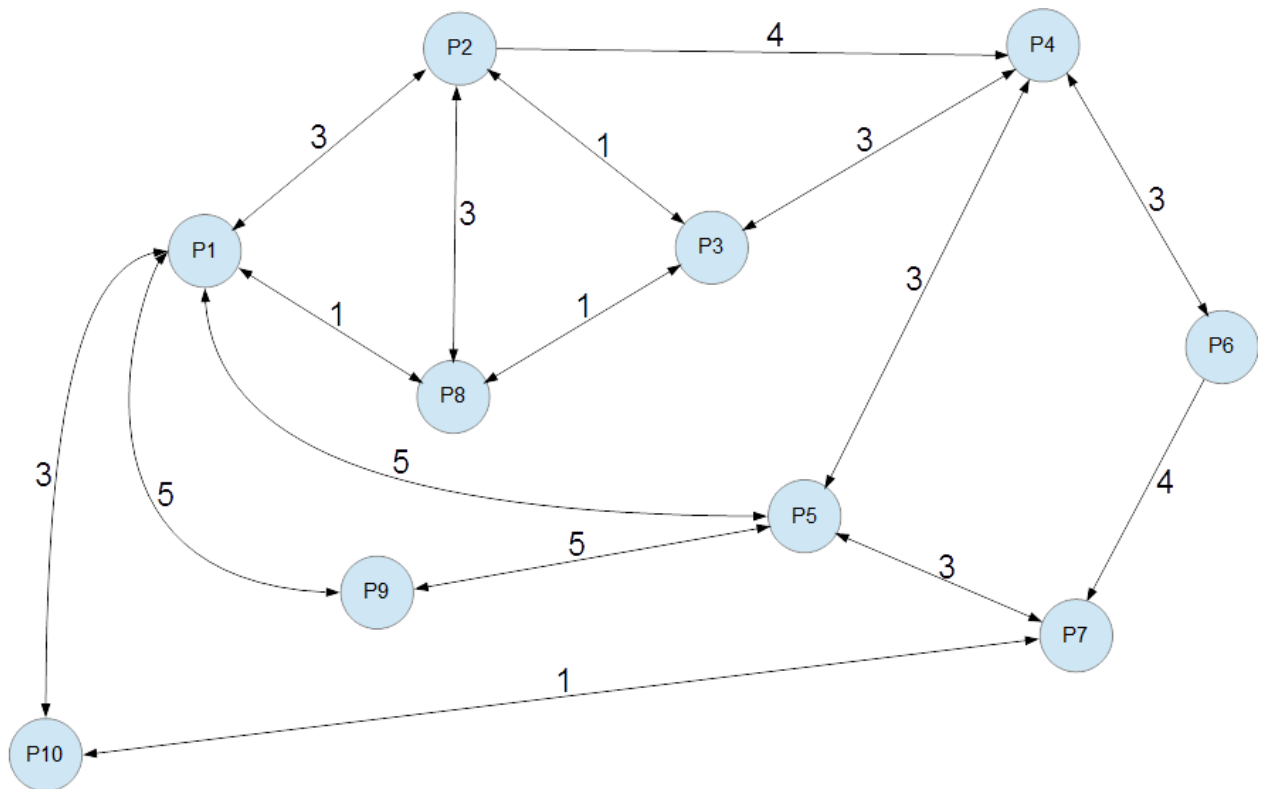
3.2-Illustration de l'utilisation du Web dans NetLink:

Pour mieux comprendre la gestion du Web dans l'application NetLink, considérons le scénario suivant:

En reprenant notre exemple où l'acteur P1 souhaite se faire recommander professionnellement auprès du réalisateur P7, supposons maintenant qu'un utilisateur quelconque apprend qu'une personne P10 « est ami » avec l'acteur P1 et que P10 « a rencontré une fois » le réalisateur P7.

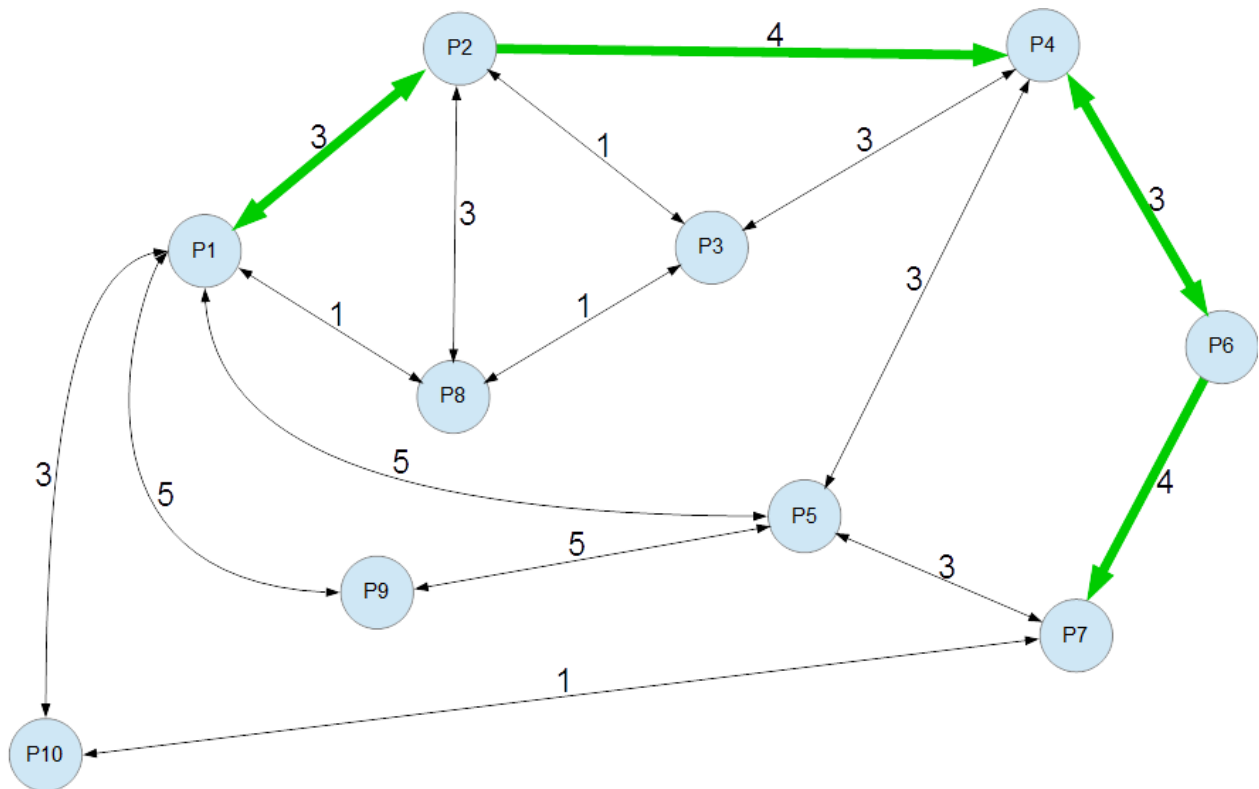
Cet utilisateur met donc à jour ces relations dans DBpedia.

Dans ce cas, l'application NetLink étant reliée à la base de connaissance de DBpedia, prendra automatiquement en compte ces nouvelles relations sans aucune autre intervention et le graphe prendra donc la forme suivant:



En supposant que l'acteur P1 adopte la stratégie de recommandation professionnelle consistant à passer par le moins d'intermédiaires possible et où la dernière personne recommandant l'acteur P1 doit avoir la relation la plus forte avec le réalisateur P7.

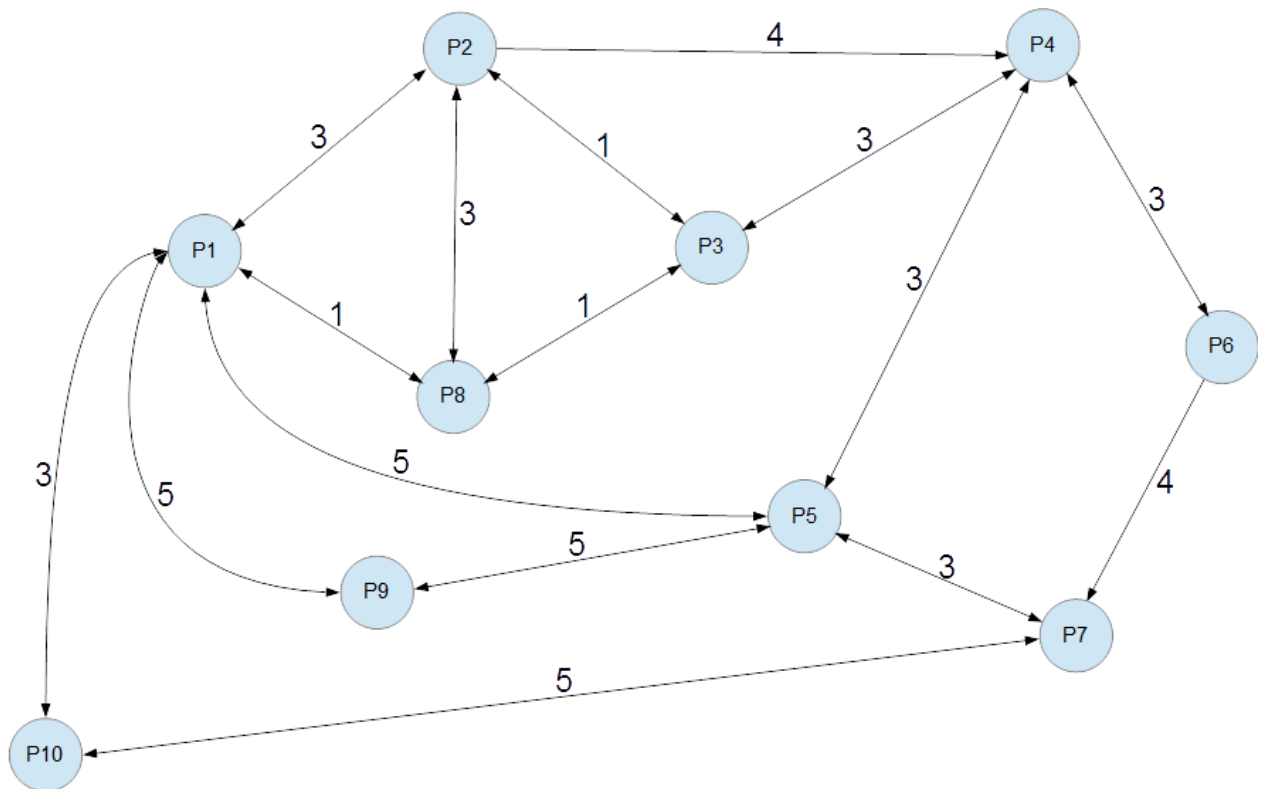
Dans ce cas, l'application proposera à l'acteur P1 le chemin suivant (chemin présenté en vert):



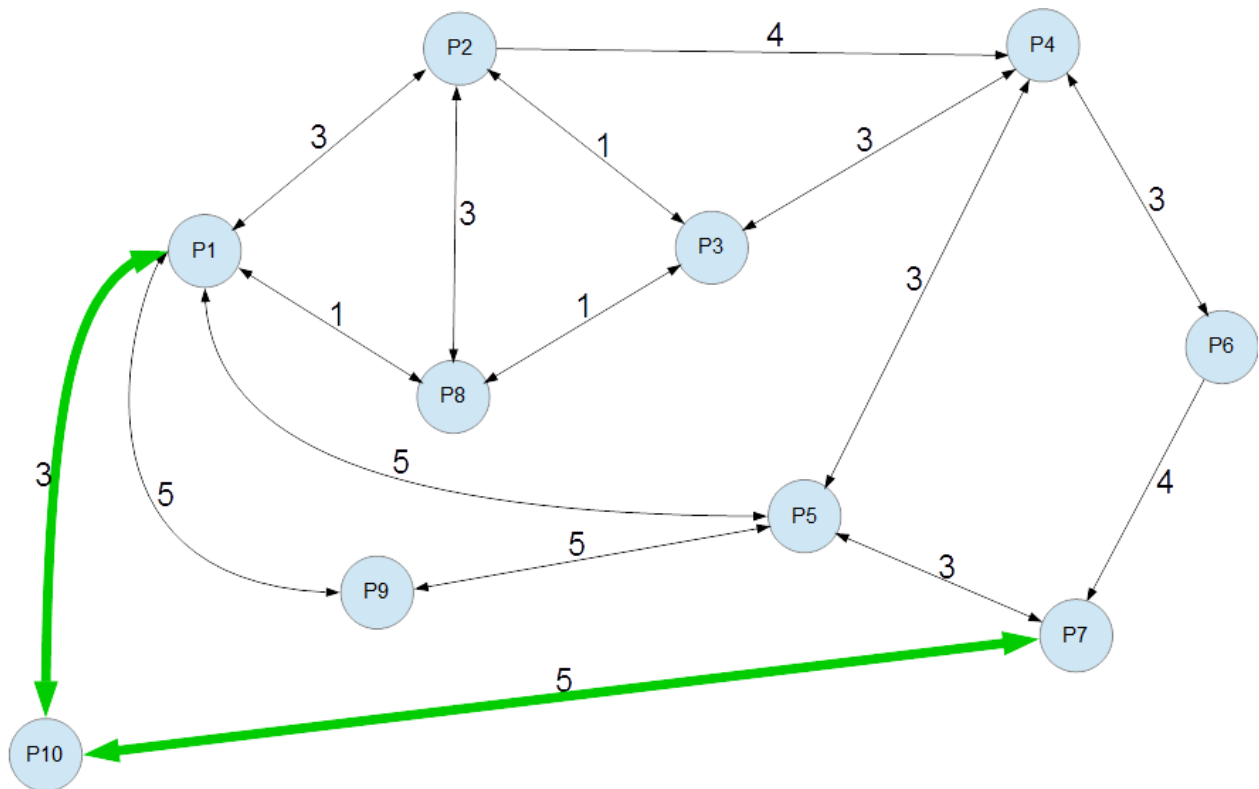
Supposons maintenant qu'un autre utilisateur de DBpedia apprend que la personne P10 « a épousé » le réalisateur P7. Nous avons donc à présent la personne P10 qui « a une relation familiale » avec le réalisateur P7.

Par conséquent, cet utilisateur met à jour DBpedia en ce sens.

Ainsi, l'application NetLink prend automatiquement en compte ce nouveau changement et le graphe prend alors la forme suivante:



Si nous reprenons l'hypothèse que l'acteur P1 adopte la stratégie de recommandation professionnelle consistant à passer par le moins d'intermédiaires possible et où la dernière personne recommandant l'acteur P1 doit avoir la relation la plus forte avec le réalisateur P7, l'application proposera alors à l'acteur P1 le chemin suivant (chemin présenté en vert):



Nous pouvons donc voir ici tout l'intérêt du modèle et sa capacité à pouvoir alimenter sa base de connaissance par le Web.

4. Implémentation des stratégies de recommandation dans NetLink:

Nous avons vu précédemment comment les techniques issues du Web-Sémantique nous ont permis de construire une représentation facilement exploitable sous forme de graphe, dont nous possédons des outils puissants et performants pouvant travailler sur de telles représentations.

De plus, nous avons montré comment l'application NetLink ne pourrait fonctionner sans l'utilisation de concepts avancés du Web-Sémantique comme l'inférence et comment le Web-Sémantique rendait possible des fonctionnalités qui seraient

impossible sans ce dernier, comme par exemple profiter du Web pour alimenter dynamiquement notre base de connaissance.

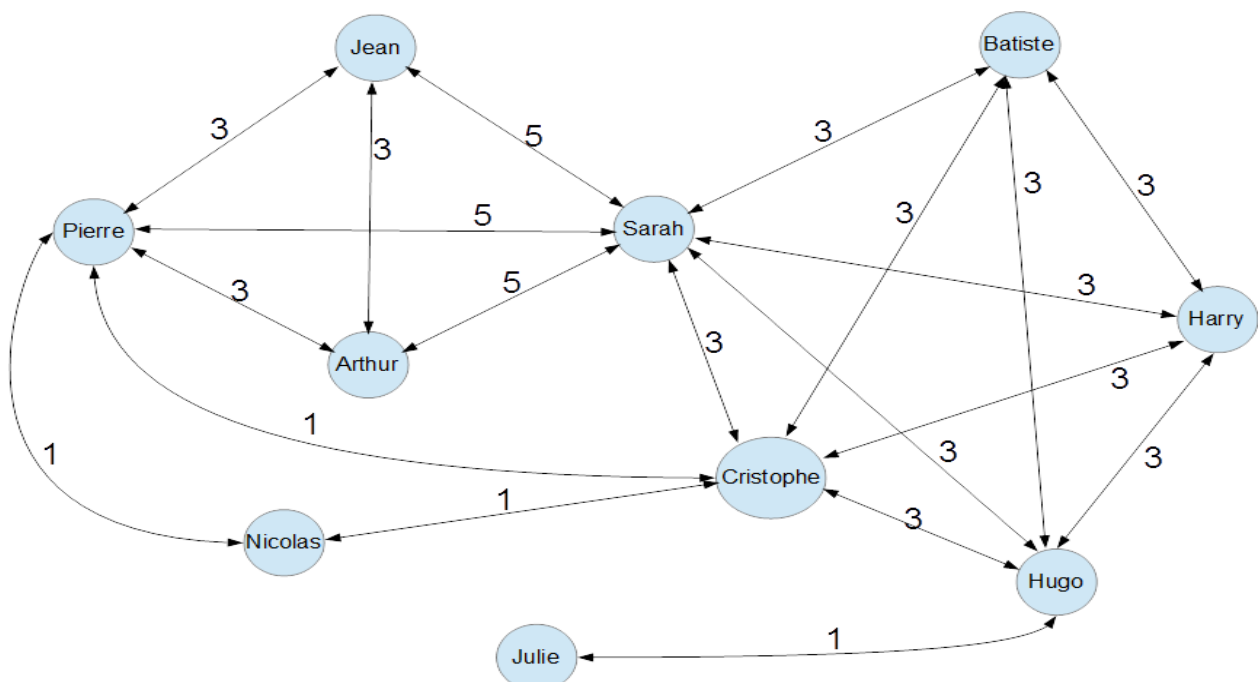
Maintenant, nous souhaitons savoir comment nous utilisons le graphe obtenu pour permettre de proposer différentes solutions en fonction d'une stratégie de recommandation spécifiée par l'utilisateur.

Dans cette section, nous expliciterons donc l'implémentation des différents algorithmes permettant d'exploiter le graphe obtenu. L'application NetLink étant programmé en Java (en utilisant la librairie Jena pour les requêtes SPARQL), le vocabulaire utilisé utilisera la terminologie du langage Java.

Tout d'abord, la première étape consiste à transformer le graphe obtenu à l'aide d'une structure de donnée facilement utilisable pour l'application ultérieure d'algorithme d'intelligence artificielle. Il s'agira ici d'algorithmes performants de recherche dans des graphes.

Pour cela, nous pouvons transformer notre graphe pour la représenter à l'aide d'une matrice d'adjacence.

Par exemple pour le graphe ci-dessous:



Cela donnerait la matrice d'adjacence M suivante:

$$M = \begin{pmatrix} 0 & 3 & 5 & 0 & 1 & 0 & 0 & 3 & 1 & 0 \\ 3 & 0 & 5 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 5 & 5 & 0 & 3 & 3 & 3 & 3 & 5 & 0 & 0 \\ 0 & 0 & 3 & 0 & 3 & 3 & 3 & 0 & 0 & 0 \\ 1 & 0 & 3 & 3 & 0 & 3 & 3 & 0 & 1 & 0 \\ 0 & 0 & 3 & 3 & 3 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 3 & 3 & 0 & 0 & 0 & 1 \\ 3 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Pour des raisons de performance et d'adaptabilité du code, notre représentation est légèrement différente d'une matrice d'adjacence classique mais fonctionne de la même manière.

A ce stade, nous avons une représentation du graphe.

Prenons maintenant deux exemples de stratégie de recommandation pour illustrer ces implémentations:

Premier scénario:

Si nous souhaitons par exemple adopter la stratégie de recommandation consistant à passer par le moins d'intermédiaire possible sans se préoccuper de la nature des relations liant les personnes entre elles alors nous utiliserons un algorithme de graphe tel que le « *Breadth First Search* », « *Iterative Deepening Search* », ou encore le « *Bidirectionnal Search* » (référence utilisé: *Artificial Intelligence: Moderne Approach*) consistant à rechercher efficacement le plus court chemin dans un graphe non pondéré.

Ainsi, nous transformons la matrice d'adjacence précédente ainsi:

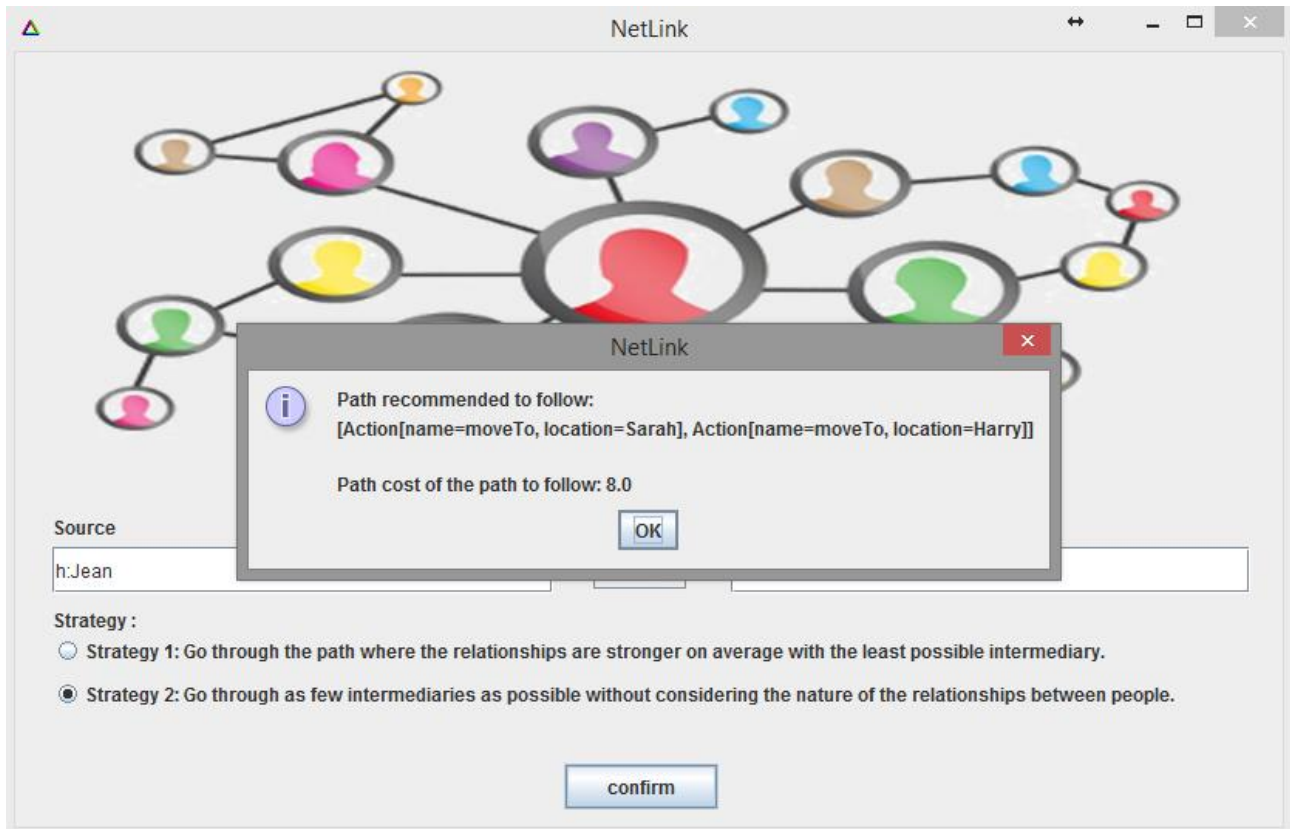
$$M = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Et nous appliquons un des algorithmes cités plus haut pour rechercher le plus court chemin dans un graphe non pondéré, ce qui nous donnera un chemin de recommandation passant par le moins d'intermédiaires possible.

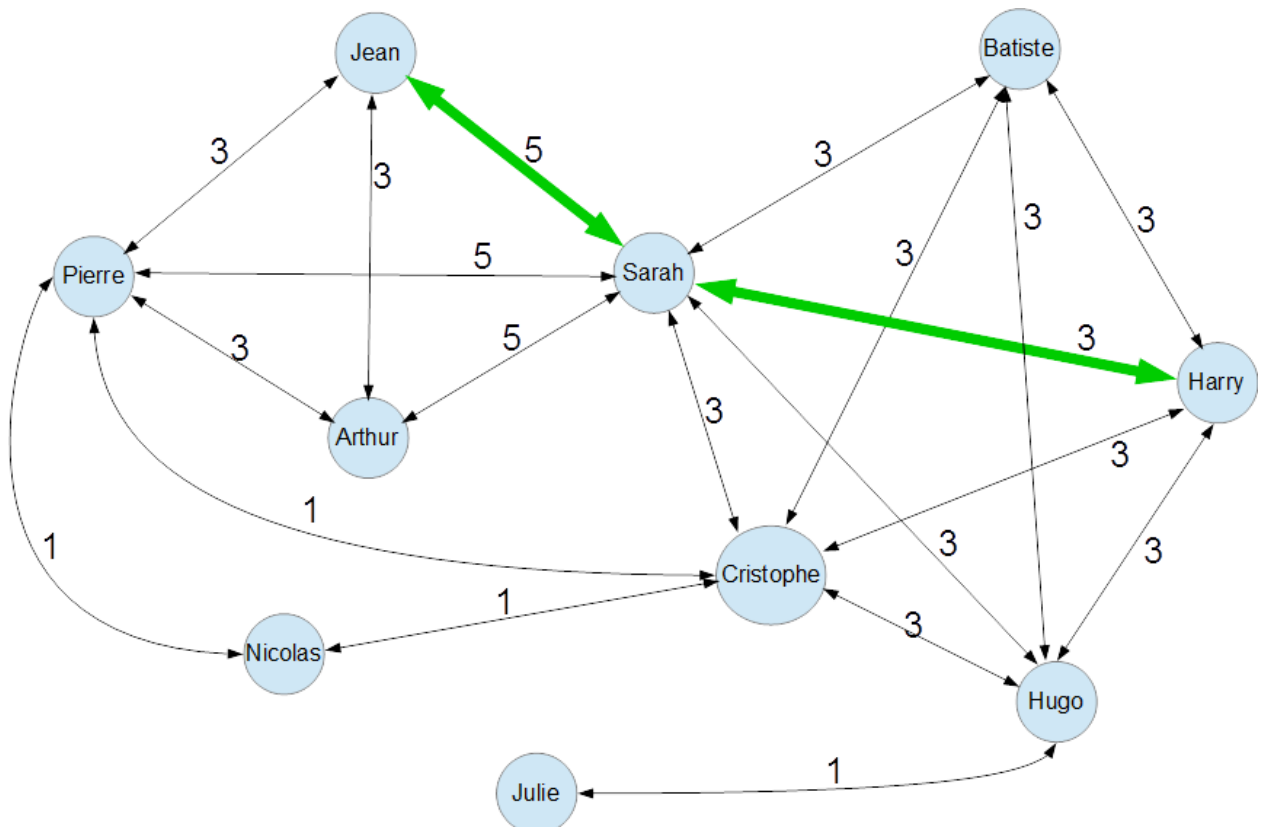
Illustration de cette stratégie dans NetLink:



Résultats:



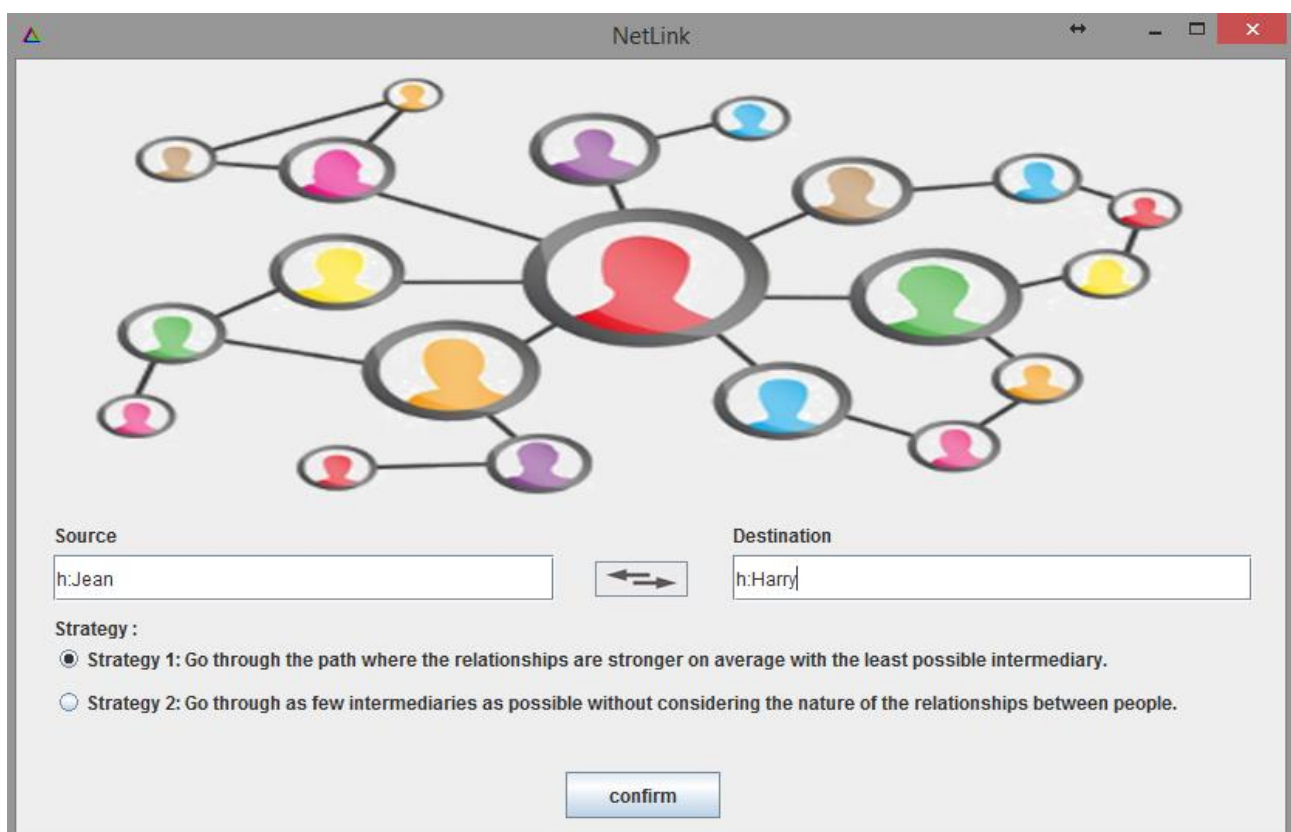
Nous vérifions bien la validité des résultats sur le graphe initial:



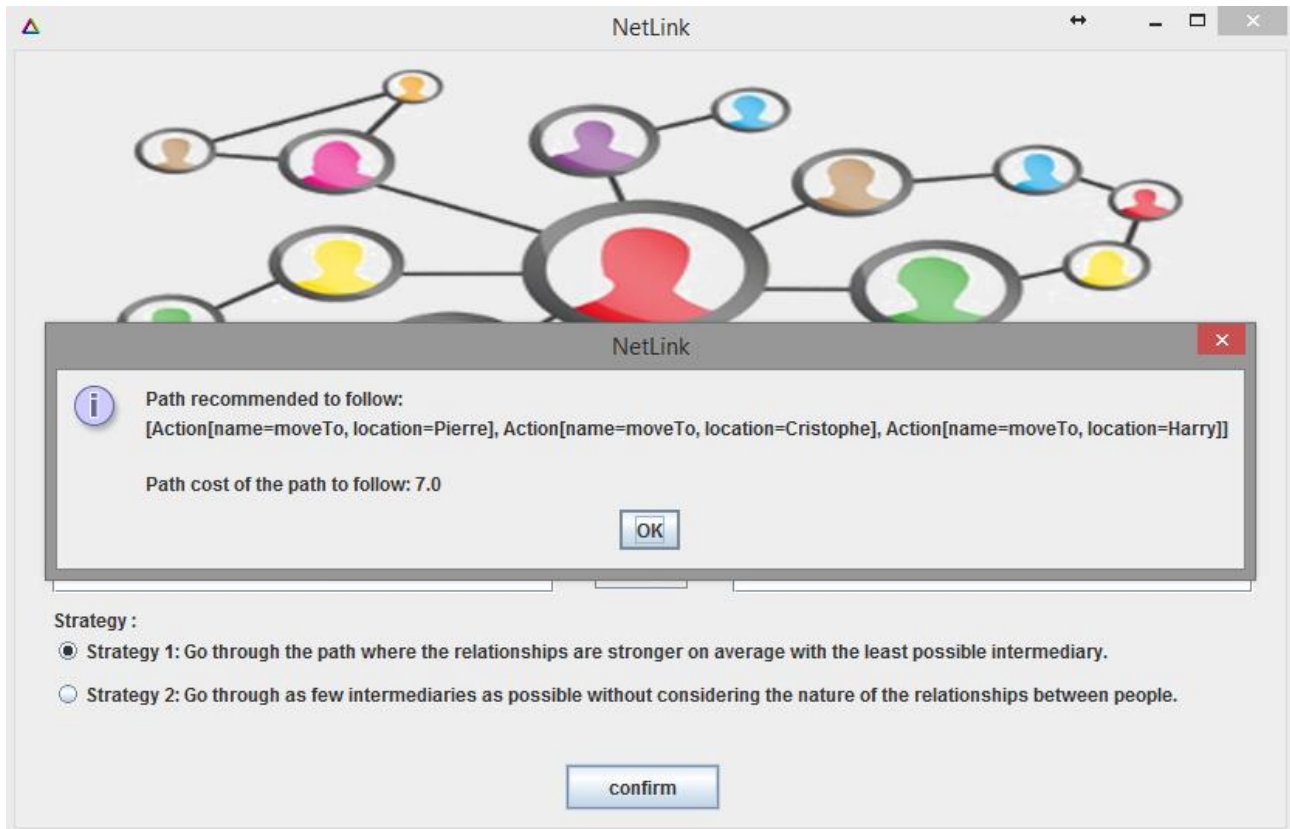
Deuxième scénario:

Dans ce scénario, nous souhaitons par exemple adopter la stratégie de recommandation consistant à passer par le moins d'intermédiaire possible et où les relations sont les plus fortes entre les personnes alors nous utiliserons un algorithme de graphe tel que le « *Uniform Cost Search* » (référence utilisé: *Artificial Intelligence: Moderne Approach*) consistant à rechercher efficacement le plus court chemin dans un graphe pondéré en minimisant les coûts. Nous rapelons que nous avons utilisé le prédicat selon laquelle plus la pondération est faible, plus la relation est forte.

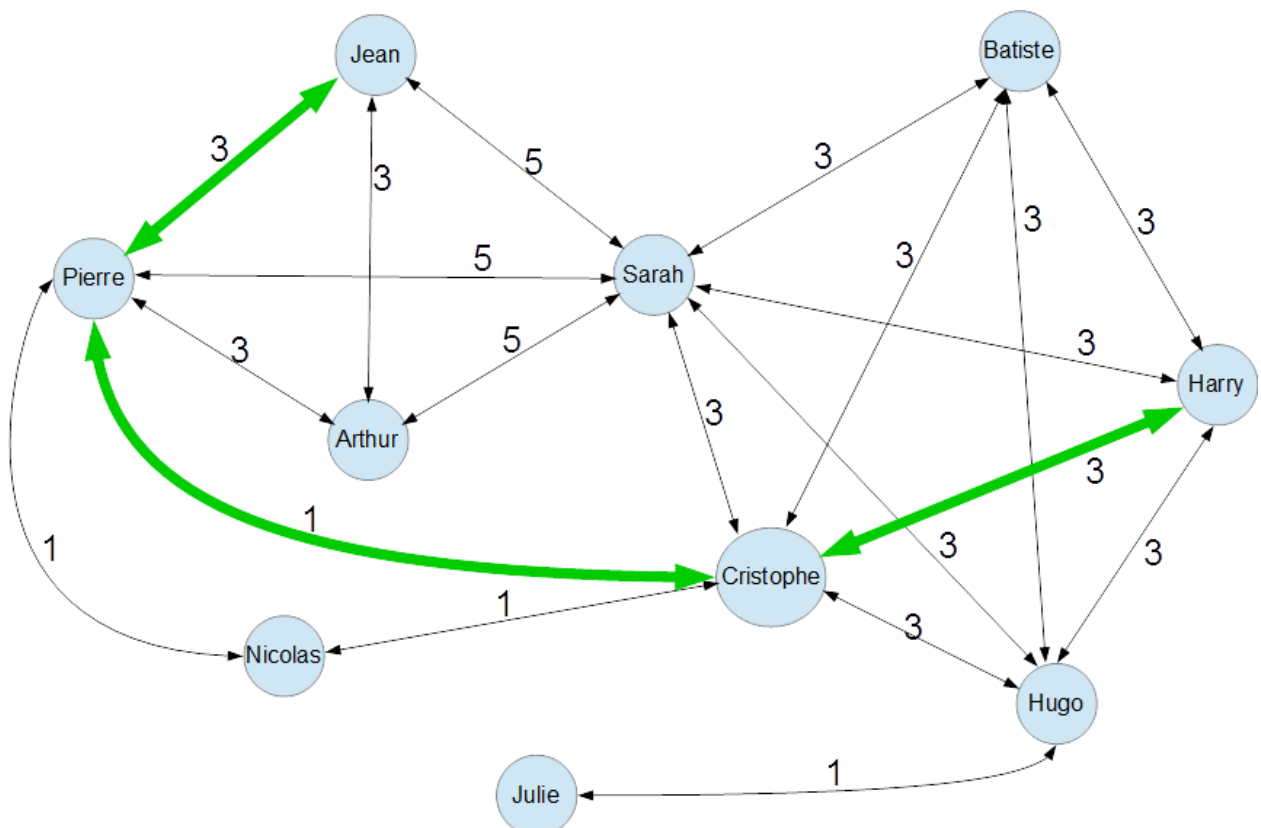
Illustration de cette stratégie dans NetLink:



Résultats:



Nous vérifions bien la validité des résultats sur le graphe initial:



Nous constatons ici que si le chemin proposé ici passe par un peu plus d'intermédiaire que dans le cas précédent mais le coût total du chemin proposé est moindre et donc les relations dans ce chemin sont en moyenne plus fortes.

5. Conclusion:

Pour conclure, nous pouvons voir ici tout l'intérêt et la puissance des concepts du Web-Sémantique tel que l'inférence dans une application concrète. En effet, ces notions permettent de mettre en œuvre des procédés adaptatifs et dynamiques tout en bénéficiant de la base de connaissances formidable qu'est le Web.

Plus précisément ici, nous pouvons nous rendre compte dans la mise au point de cette application que le projet NetLink a recours dans chacune de ses étapes de conceptions aux outils du Web-Sémantique pour surmonter nombre de difficultés et bénéficier de tous ses avantages.