

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

APPARIEMENT AUTOMATIQUE BASÉ SUR LA MODÉLISATION ET LA SIMULATION À BASE  
D'AGENTS : APPLICATION AU E-RECRUTEMENT

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

ULRICH ARNAUD GBADOE

MAI 2022

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.04-2020). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

## REMERCIEMENTS

De prime abord, j'aimerais dire un sincère merci à Dieu pour m'avoir permis de mener ce travail de recherche à bien.

Je tiens, ensuite à exprimer toute ma reconnaissance et ma gratitude à mon directeur de mémoire Monsieur Hakim Lounis, Professeur à l'Université du Québec à Montréal, pour sa patience, sa disponibilité et surtout ses judicieux conseils. Son exigence m'a grandement stimulé et m'a permis de me surpasser.

Je remercie aussi MITACS pour sa bourse de formation à la recherche qui m'a été octroyée au cours de ma maîtrise. En effet, ce programme m'a donné l'accès à une formation de pointe et m'a permis d'améliorer mes connaissances en programmation et en communication grâce à l'expertise des encadreurs.

Merci également à tous les professeurs de l'UQÀM avec qui j'ai suivi des cours tout au long de la maîtrise.

Ce mémoire n'aurait certainement pas vu le jour sans le soutien et l'amour inconditionnels de ma famille. Je pense ici à ma mère, mon père, mes frères et sœurs et à toutes les familles MAGLO, DANSOU, KPESOU.

Je ne saurais terminer sans remercier Ariane KOUAME, Stéphanie TSUMBU, Roméo ANAWI, Kerant MUDJUND, Carla TEKOU, Arsène SEGNEJ, Mme et M. MENSAH, Marie-ange OSSENI, Nasser ADE-DAYO ADEKOUJJO et toutes les personnes qui à différentes étapes de mon parcours ont participé d'une manière ou d'une autre au bon déroulement de ce mémoire de maîtrise.

## DÉDICACE

À tous ceux qui se sentent fiers de ce travail

## AVANT-PROPOS

## TABLE DES MATIÈRES

AVANT-PROPOS.....	iv
LISTE DES FIGURES.....	vii
LISTE DES TABLEAUX .....	ix
LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES.....	x
RÉSUMÉ.....	xi
ABSTRACT .....	xii
INTRODUCTION .....	1
0.1 Introduction générale .....	1
0.2 Problématique et Motivation .....	2
0.3 Objectifs et Contributions .....	5
0.4 Organisation du mémoire .....	7
CHAPITRE 1 ÉTAT DE L'ART.....	9
1.1 Introduction .....	9
1.2 Système d'appariement basé sur l'apprentissage automatique .....	10
1.3 Les Systèmes de Recommandation.....	13
1.4 Les ontologies du domaine RH.....	14
1.5 Modèles de recherche analytique .....	16
1.6 Conclusion.....	17
CHAPITRE 2 APPARIEMENT DE SCHÉMAS .....	19
2.1 Introduction .....	19
2.2 L'appariement des CV aux offres d'emploi comme un appariement de schémas .....	19
2.2.1 Le déroulement de l'appariement.....	29
2.3 Caractéristiques des approches existantes.....	30
2.4 Systèmes Complexes Adaptatifs .....	34
2.4.1 Système adaptatif vs déterministe .....	35
2.4.2 Système complexe vs compliqué.....	35
2.5 Conclusion.....	39
CHAPITRE 3 APPARIEMENT DE SCHÉMAS COMME UN SYSTÈME COMPLEXE ADAPTATIF .....	41
3.1 Introduction .....	41

3.2	Modélisation et simulation à base d'agents .....	41
3.2.1	La Simulation.....	45
3.2.2	La Modélisation.....	47
3.3	Modélisation et Simulation à Base d'Agents pour l'appariement de schémas (SMAS).....	51
3.4	Approche pour la prise de décision .....	52
3.5	Énoncé du modèle conceptuel.....	54
3.5.1	Agent élément de schémas.....	55
3.5.2	Stochasticité et adaptation .....	57
3.5.3	Appariement consensuel .....	59
3.5.4	Auto-organisation et émergence.....	60
3.5.5	Simulations multiples et analyse statistique .....	60
3.6	Transcription du modèle .....	62
3.7	Conclusion .....	62
CHAPITRE 4 MODÉLISATION ET CONCEPTION DU PROTOTYPE DE SIMULATION .....		63
4.1	Introduction .....	63
4.2	C'est quoi un « Agent » ? .....	63
4.2.1	Agent intelligent.....	66
4.3	Prototype et Expérimentation .....	70
4.3.1	Prototype .....	71
4.3.2	Analyses et Travaux .....	92
4.4	Conclusion .....	102
CHAPITRE 5 EXPÉRIMENTATIONS ET DISCUSSIONS DES RÉSULTATS DU PROTOTYPE .....		103
5.1	Introduction .....	103
5.2	Les expérimentations.....	103
5.2.1	Configuration expérimentale.....	105
5.2.2	Métriques d'évaluation.....	105
5.2.3	Résultats des expérimentations.....	108
5.3	Évaluation et étude des résultats .....	120
5.4	Conclusion .....	123
CHAPITRE 6 CONCLUSION GÉNÉRALE .....		124
6.1	Survol de la recherche.....	124
6.2	Apports de la recherche .....	125
6.3	Améliorations futures .....	126
BIBLIOGRAPHIE.....		128

## LISTE DES FIGURES

Figure 2.1 Exemple de schémas pour l'appariement (Bernstein <i>et al.</i> , 2011) .....	20
Figure 2.2 Processus de l'appariement de schémas (Do et Rahm, 2007) .....	22
Figure 2.3 Processus simplifié de l'appariement de schémas (Rahm et Peukert, 2019) .....	24
Figure 2.4 Tableau comparatif des outils d'appariement (Bernstein <i>et al.</i> , 2011) .....	27
Figure 2.5 Processus général de l'appariement de schémas (Bellahsene <i>et al.</i> , 2011) .....	29
Figure 2.6 Processus d'appariement de schémas (Martono et Azhari, 2017) .....	32
Figure 2.7 Comparaison entre système simple, compliqué et complexe .....	37
Figure 2.8 Systèmes Complexes Adaptatifs (Andrus, 2005).....	38
Figure 3.1 Objectifs d'une simulation (Treuil <i>et al.</i> , 2008).....	46
Figure 3.2 Représentation simplifiée d'un modèle à base d'agent (Treuil, Drogoul et al. 2008) .....	51
Figure 3.3 Illustration de l'exemple de référence pour la correspondance .....	54
Figure 4.1 Représentation de l'interaction entre l'agent et son environnement.....	65
Figure 4.2 Diagramme de classes pour notre prototype.....	73
Figure 4.3 Diagramme d'activités du comportement de l'agent <i>Agent_principal</i> .....	75
Figure 4.4 Diagramme de transition entre états de l'agent <i>Agent_principal</i> .....	75
Figure 4.5 Interface de la plateforme Repast Symphony.....	78
Figure 4.6 Disposition des agents dans le panneau de visualisation de notre prototype .....	79
Figure 4.7 Interface utilisateur affichant la vue détaillée de notre prototype .....	80
Figure 4.8 Interface utilisateur affichant la vue simplifiée de notre prototype.....	81
Figure 4.9 Panneau de visualisation au cours d'une simulation .....	82
Figure 4.10 Pseudo code pour le prétraitement .....	87
Figure 4.11 Pseudocode pour le chargement de nos données.....	87
Figure 4.12 Pseudocode pour le test de notre modèle.....	88
Figure 4.13 Pseudocode montrant la sortie de notre curriculum après traitement.....	88



Figure 4.14 Exemple d'une offre d'emploi.....	89
Figure 4.15 Panneau de visualisation à l'itération 1034 .....	95
Figure 4.16 Panneau de visualisation à l'itération 1737 .....	96
Figure 4.17 Panneau de visualisation à l'itération 4317 .....	96
Figure 4.18 Panneau de visualisation à l'itération 4745 .....	97
Figure 4.19 Panneau de visualisation à l'itération 6001 .....	97
Figure 4.20 Résultat final de la simulation .....	98
Figure 4.21 Interface d'interrogation de l'état interne des agents lors d'une simulation.....	99
Figure 4.22 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 1737 .....	100
Figure 4.23 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 4317 .....	100
Figure 4.24 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 5704 .....	100
Figure 4.25 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 6001 .....	101
Figure 5.1 Interface utilisateur montrant le résultat final du premier scénario.....	108
Figure 5.2 Panneau de visualisation au terme de la simulation du premier scénario .....	109
Figure 5.3 Interface utilisateur montrant le résultat final du deuxième scénario.....	113
Figure 5.4 Panneau de visualisation au terme de la simulation du deuxième scénario .....	114

## LISTE DES TABLEAUX

Tableau 4.1 Exemple de profil d'un candidat.....	85
Tableau 5.1 Tableau simplifié des résultats du scénario 1.....	109
Tableau 5.2 Tableau de contingence des simulations multiples pour le premier scénario.....	111
Tableau 5.3 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 1.....	112
Tableau 5.4 Tableau simplifié des résultats du scénario 2.....	114
Tableau 5.5 Tableau de contingence des simulations multiples pour le deuxième scénario.....	115
Tableau 5.6 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 2.....	116
Tableau 5.7 Tableau de contingence des simulations multiples pour le troisième scénario.....	117
Tableau 5.8 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 3.....	118
Tableau 5.9 Tableau de contingence des simulations multiples pour le quatrième scénario.....	119
Tableau 5.10 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 4.....	120
Tableau 5.11 Tableau récapitulatif des performances de l'appariement par rapport au nombre de simulations.....	121
Tableau 5.12 Tableau récapitulatif de l'impact du nombre de simulations individuelles sur les performances de l'appariement.....	121
Tableau 5.13 Tableau récapitulatif de l'impact de la durée des simulations individuelles sur les performances de l'appariement.....	122

## **LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES**

CV : Curriculum Vitae

OE : Offre d'emploi

SCA : Système Complexe Adaptatif

RH : Ressource Humaine

JD : Job Description

NLP : Natural Language Processing

NER : Named Entity Recognition

TALN : Traitement Automatique du Langage Naturel

## RÉSUMÉ

La croissance très importante du réseau Internet a fondamentalement changé la disponibilité en ligne des données et la quantité d'information électronique échangée. Cela a révolutionné l'accès à l'information et la gestion des savoirs dans les grands organismes. Parmi toutes les fonctions de gestion de ressources humaines (RH), il y a la tâche difficile d'embaucher l'aspirant qualifié ayant l'ensemble requis de compétences, correspondant aux spécifications du poste annoncé, et permettant ainsi d'atteindre les objectifs de l'organisation. Une stratégie de recrutement est un facteur primordial pour chaque organisation dans l'embauche d'employés qualifiés qui pourraient être plus efficaces dans la réalisation des objectifs de l'emploi. Ainsi, le recrutement électronique (e-recrutement) est un exemple assez récent d'application typique d'une telle approche de gestion de la connaissance et des données, y compris à travers le Web. Cependant, cette révolution du marché du travail, basée sur l'utilisation des technologies de l'information est encore en cours, et une évolution des outils et méthodes consacrés à la recherche et à la gestion des CV et des offres de travail, est toujours d'actualité. Ces techniques restent donc sujettes à des améliorations et c'est ce à quoi va s'atteler ce projet. De façon simple, le e-recrutement, vu comme un moyen pratique de gestion des ressources humaines via le Web, tend à faciliter et donc à automatiser le rapprochement entre les CV et les offres d'emploi publiés. Le principal problème observé est que ces ressources ne sont pas adéquatement exploitées : les outils de gestion et les procédés disponibles sont vraiment limités quand il s'agit de traiter ou d'interpréter sémantiquement le contenu d'une grande quantité de documents. On note également que ces ressources sont purement syntaxiques. Pour notre part, nous pensons que l'utilisation de l'IA pour la recherche d'emploi réduirait, entre autres, le temps de recherche et les coûts pour l'entreprise et le candidat. Elle sert aussi à produire une cartographie des talents en aidant les RH à acquérir les meilleurs talents, en ciblant leurs compétences pour les placer aux bons postes. C'est cette étape de rapprochement entre talents/compétences et postes à pourvoir que nous ciblons dans notre projet, et nous parlerons donc d'appariement pour l'e-recrutement.

Ainsi, de nombreux travaux ont œuvré pour résoudre cette problématique, et sans en produire une liste exhaustive vu les contraintes de cette proposition, au meilleur de notre connaissance, quasi toutes les solutions qui ont été proposées dans la littérature, appartiennent au paradigme réductionniste. Dans notre projet, nous défendons l'hypothèse que la problématique de l'appariement automatique mériterait qu'elle soit abordée dans son entièreté et vue comme un système complexe comprenant plusieurs composantes dont la simulation des interactions non-linéaires permettrait l'émergence du résultat final. Nous proposons donc un changement de paradigme en ayant recours à la théorie des systèmes complexes adaptatifs (SCA), provenant de la pensée systémique, pour proposer des réponses aux défis auxquels les approches réductionnistes classiques d'appariement automatique, font toujours face (e.g. complexité, incertitude).

Mots clés : E-recrutement, appariement automatique, systèmes complexes adaptatifs, modélisation et simulation à base d'agents, intelligence artificielle

## ABSTRACT

The very significant growth of the Internet has fundamentally changed the online availability of data and the amount of electronic information exchanged. This has revolutionized information access and knowledge management in large organizations. Among all human resource (HR) management functions, there is the difficult task of hiring the qualified candidate with the required skill set, matching the specifications of the advertised position, and thus achieving the organization's objectives. A recruitment strategy is a primary factor for every organization in hiring qualified employees who could be more effective in achieving the job objectives. Thus, electronic recruitment (e-recruitment) is a fairly recent example of a typical application of such an approach to knowledge and data management, including through the Web. However, this revolution in the labor market, based on the use of information technologies, is still ongoing, and an evolution of tools and methods dedicated to the search and management of CVs and job offers is still in progress. These techniques are still subject to improvement and that is what this project will address. In a simple way, e-recruitment, considered as a practical application of human resources management through the Web, tends to automate the matching of CVs and job offers published. The main problem is that these resources are often poorly exploited: the available management techniques and tools are purely syntactic and remain limited in the face of the large mass of documents to be processed and the need for a more semantic interpretation of their contents. For our part, we believe that the use of AI for job search would reduce, among other things, the search time and costs for the company and the candidate. It also serves to produce a talent map by helping HR acquire the best talent, targeting their skills to place them in the right positions. It is this step of matching talent/skills to job openings that we are targeting in our project, and we will therefore talk about matching for e-recruitment.

Thus, many works have worked to solve this problem, and without producing an exhaustive list given the constraints of this proposal, to the best of our knowledge, almost all the solutions that have been proposed in the literature belong to the reductionist paradigm. In our project, we defend the hypothesis that the problem of automatic matching deserves to be approached in its entirety and seen as a complex system comprising several components whose simulation of the non-linear interactions of these components would allow the emergence of the final result. We therefore propose a paradigm shift by using the theory of complex adaptive systems (CAS), stemming from systems thinking, to propose answers to the challenges that classical reductionist approaches to automatic matching still face (e.g. complexity, uncertainty).

Keywords : E-recruitment, automatic matching, complex adaptive systems, agent-based modeling and simulation, artificial intelligence

## INTRODUCTION

### 0.1 Introduction générale

Ce mémoire sera appliqué à l'étude de l'appariement des Curriculum Vitae (CV) aux offres d'emplois (OE) sous la forme d'un Système Complexe Adaptatif (SCA). D'après (van Gigch et Le Moigne, 1990) :

un système complexe est, par définition, un système que l'on tient pour irréductible à un modèle fini, aussi compliqué, stochastique, sophistiqué que soit ce modèle, quelque soit sa taille, le nombre de ses composants, l'intensité de leurs interactions. (van Gigch et Le Moigne, 1990)

De plus, il existe généralement dans ce système de nombreuses boucles de rétroaction : les comportements collectifs et les propriétés qui émergent au niveau macroscopique peuvent modifier les comportements des entités au niveau microscopique (Lesne, 2009). Le fonctionnement du système est une globalité irréductible, qu'il est impossible de représenter complètement avec un ensemble d'équations solvables et prédictives. Pour comprendre son fonctionnement, il est nécessaire de tout prendre en compte, ce qui amène à réaliser une simulation.

L'usage de la simulation informatique multi-agent se développe depuis une trentaine d'années:

la complexité du monde réel est reproduite dans des expérimentations *in silico*, où l'on observe les dynamiques individuelles et collectives d'entités informatiques, des « abstractions manipulables » du monde réel, conçues par des chercheurs de toutes disciplines. (Treuil *et al.*, 2008).

Cette simulation informatique peut être réalisée en utilisant un Système Multi-Agent (SMA), qui peut être défini comme une organisation d'agents (El Fallah-Seghrouchni et Briot, 2009). Chaque agent de ce système est :

une entité réelle ou virtuelle, dont le comportement est autonome, évoluant dans un environnement, qu'il est capable de percevoir, sur lequel il est capable d'agir et d'interagir avec les autres agents. (Costa et Demazeau, 1996).

Un SMA permet ainsi la représentation précise de l'ensemble des entités qui composent le système complexe. Il permet de spécifier l'environnement dans lequel elles évoluent, ainsi que les règles d'interactions entre ces entités. Par interaction, on entend la mise en relation dynamique de deux ou plusieurs entités par le biais d'un ensemble d'actions réciproques (El Fallah-Seghrouchni et Briot, 2009).

Nous avons choisi d'utiliser cette technologie des SMA pour étudier les sciences humaines et sociales, et plus particulièrement, dans la discipline des sciences économiques pour modéliser le marché du travail. Le marché du travail peut être vu comme un système complexe, dans le sens où il met en jeu un grand nombre d'interactions locales, entre des "travailleurs" et des "employeurs". À cette complexité, qui provient du grand nombre d'interactions, s'ajoute un degré de complexité supplémentaire inhérente aux sciences humaines et sociales, car il s'agit de modéliser les comportements humains (Kant, 2012).

## 0.2 Problématique et Motivation

La croissance fulgurante et très importante du réseau Internet a essentiellement changé la disponibilité des données en ligne, la qualité et la quantité d'information électronique échangée. Cela a transformé radicalement l'accès à l'information et la gestion des connaissances dans les grands organismes. Cette révolution, a par conséquent eu un impact significatif sur les méthodes utilisées par ces organisations pour trouver de nouveaux employés.

Parmi toutes les fonctions de gestion de ressources humaines (RH), il existe la tâche difficile d'embaucher le candidat qualifié ayant l'ensemble de compétences requis, correspondant aux spécifications du poste annoncé, et permettant ainsi d'atteindre les objectifs de l'organisation. Là où, par le passé, les employeurs disposaient d'un plus petit nombre de

candidats à embaucher, le recrutement en ligne a élargi les règles du jeu, permettant aux candidats du monde entier de rechercher des opportunités. L'embauche de futurs employés qualifiés qui devraient être plus efficaces dans la réalisation des buts et objectifs de l'emploi, nécessitera une stratégie de recrutement qui s'avère donc être un facteur primordial pour chaque organisation. Cette fonction majeure de l'organisation doit analyser des données pour le processus de prise de décision (Mansfield, 2016).

Ainsi, le recrutement électronique (e-recrutement) est un exemple assez récent d'application typique d'une telle approche de gestion de la connaissance et des données, y compris à travers le Web. Internet a clairement mis en évidence que les méthodes de recrutement classiques, qui utilisent les annonces d'emploi dans les journaux et les magasins, ou qui sollicitent les agences de recrutement, sont pour la plupart trop lentes, chères et insuffisantes dans leurs capacités à fournir des candidats de qualité dans des délais raisonnables (Kaur, 2015).

L'e-recrutement est le moyen d'embauche de personnel par la voie de ressources électroniques, en particulier Internet. Le terme "*e-recrutement*" a été utilisé pour la première fois dans les articles au milieu des années 1980. L'e-recrutement peut être scindé en deux catégories d'usages: les sites Web d'entreprise pour le recrutement qui sont propres aux compagnies qui désirent recruter et les sites d'emplois commerciaux (tels que monster.com) pour afficher ou publier des offres d'emploi annoncées par d'autres compagnies. Les sites Web d'entreprise appartiennent aux compagnies ayant un lien pour les offres d'emploi ou les options de carrière où les candidats peuvent se connecter pour les postes vacants. Si la compagnie annonce ses postes vacants sur d'autres sites Web spécialisés dans le recrutement, tels que : naukri.com, timesjob.com, monster.com, etc., alors elles adopteraient des sites d'emploi commerciaux pour le recrutement (Xie, 2022).

Les compagnies et les agents de recrutement ont déplacé une grande partie de leur processus de recrutement en ligne afin d'améliorer la vitesse à laquelle les candidats peuvent être jumelés avec des postes vacants. Les employeurs utilisent plusieurs technologies aussi



innovantes les unes que les autres afin de pouvoir combler les postes en un temps relativement court. Utiliser un système d'e-recrutement peut potentiellement faire gagner du temps à l'employeur, du fait que plusieurs personnes en RH évaluent indépendamment les candidats (Xie, 2022).

L'e-recrutement implique donc l'utilisation de ressources électroniques pour faciliter le processus d'embauche. En utilisant Internet pour se connecter à des candidats potentiels du monde entier, les entreprises ont pu réduire le temps et les coûts associés au recrutement de talents. Bien que cette nouvelle méthode d'embauche présente de nombreux avantages pour les organismes employeurs, il existe encore des défis lorsqu'il s'agit de trouver les talents qui conviennent le mieux à une organisation. Cependant, cette révolution du marché du travail, basée sur l'utilisation des technologies de l'information est toujours en cours, et une évolution des outils et méthodes consacrés à la recherche et à la gestion des CV et des offres de travail, est toujours d'actualité. Ces méthodes restent donc ouvertes à des améliorations et c'est ce à quoi va s'atteler ce mémoire.

De façon concrète, on a remarqué que les ressources existantes pour l'automatisation du processus du recrutement en ligne sont souvent mal exploitées. La majorité est en grande partie, basée sur les règles syntaxiques ce qui limite considérablement leur efficacité une fois qu'elles ont une grande quantité de documents à traiter. Ainsi, pour les professionnels des RH, mettre en œuvre un logiciel, y compris d'Intelligence Artificielle (IA), est un apport pour éliminer les tâches banales et analyser des masses de données et donc servir de solutionneur de problèmes. Pour notre part, nous pensons que l'utilisation de l'IA pour la recherche d'emplois réduirait, entre autres, le temps de recherche et les coûts pour l'entreprise et le candidat. Elle sert aussi à produire une cartographie des talents en aidant les RH à acquérir les meilleurs talents, en ciblant leurs compétences pour les placer aux bons postes. C'est cette étape de rapprochement entre talents/compétences et postes à pourvoir que nous ciblons dans notre projet, et nous parlerons donc d'appariement pour l'e-recrutement.

Dans la section suivante, nous allons présenter l'approche qui est au cœur de notre approche théorique et nous allons expliquer pourquoi son application serait pertinente afin de répondre à nos questions de recherche.

### 0.3 Objectifs et Contributions

Pour ce travail de recherche il s'agira pour nous dans un premier temps, d'explorer et d'analyser les techniques et les outils de gestion déjà existants dans le domaine d'appariement pour l'e-recrutement. En effet, l'objectif de notre projet consiste dans un premier temps à explorer, analyser, comprendre et comparer les nombreux travaux qui ont œuvré à résoudre cette problématique au cours de ces dernières décennies mais qui pour la plupart appartiennent au paradigme réductionniste, un paradigme qui se heurte au constat que l'appariement automatique entre des candidats (leurs CV) et les offres d'emploi, est une tâche complexe à plus d'un titre, en raison au moins de :

- l'hétérogénéité et l'ambiguïté intrinsèque aux éléments à appairer,
- le caractère incertain des résultats de l'appariement,
- le défi que peut poser l'optimisation de l'appariement comme l'explosion combinatoire, etc.

Ensuite, il s'agira d'expliquer la pertinence de l'hypothèse que nous défendons sur la problématique de l'appariement automatique des CV aux OE, qui mériterait d'être abordée dans son entièreté et vue comme un système complexe comprenant plusieurs composantes, dont la simulation des interactions non-linéaires permettrait l'émergence du résultat final.

Nous proposons donc un changement de paradigme en ayant recours à la théorie des systèmes complexes adaptatifs (SCA), issues de la pensée systémique, pour proposer des réponses aux défis auxquels les approches réductionnistes classiques d'appariement automatique, font toujours face (e.g. complexité, incertitude). Notre hypothèse principale est

donc de revoir le processus de l'appariement comme un SCA et de le modéliser en utilisant l'approche de la modélisation et de la simulation à base d'agents (Assoudi, 2017), (Assoudi et Lounis, 2014).

De façon plus concrète, nous essayons d'apporter une réponse à la question de savoir s'il serait possible d'aborder la problématique du e-recrutement en général, et de l'appariement automatique de CV à des offres d'emploi en particulier selon une approche systémique, de manière à appréhender sa complexité inhérente et donc à mieux apporter une réponse aux défis existants, notamment la complexité et l'incertitude?

L'approche de notre recherche, se centre sur la théorie de la complexité et plus particulièrement sur le paradigme des systèmes complexes adaptatifs. Notre hypothèse de départ est que le système d'appariement de CV aux offres d'emploi peut être modélisé comme un système complexe adaptatif. Il s'agira donc, dans un second temps de notre projet, d'explorer la problématique d'appariement en e-recrutement sous l'angle d'un SCA, tout en y intégrant des sous domaines de l'IA, incluant l'apprentissage automatique, la prise de décision sous incertitude (théorie bayésienne), le traitement automatique du langage, etc.

Ensuite, nous proposons une approche holistique différente des approches réductionnistes existantes, pour le processus d'e-recrutement afin de pouvoir participer à une meilleure gestion de la complexité et de l'incertitude inhérentes à ce dernier :

- Proposer une formalisation d'un modèle conceptuel à base d'agents pour le processus d'appariement (conceptuellement indépendant de l'implémentation et aussi de l'engin de simulation);
- Transformer ce modèle conceptuel en un programme informatique (i.e. prototype) s'exécutant sur une plateforme de simulation.

Par ailleurs, nous proposons aussi une méthode pour la sélection des relations de correspondance, basée sur l'exécution de simulations multiples et sur une analyse statistique (quantification de l'incertitude). Enfin nous évaluons, la viabilité de l'approche proposée et ce, en analysant ses performances, sur un ensemble de scénarios/données, aux résultats attendus.

Ainsi, nos travaux ont deux principales finalités : la première est de fournir, sous le prisme d'un système complexe adaptatif, une formalisation du modèle conceptuel du processus d'appariement, ce, en utilisant l'approche de la modélisation et simulation à base d'agents. La deuxième finalité sera de transposer ce modèle, en un programme informatique s'exécutant sur une plateforme de simulation.

#### 0.4 Organisation du mémoire

Notre travail de recherche est reflété en grande partie par l'organisation en chapitres du mémoire. En effet, ce-dernier est structuré en cinq chapitres.

Le mémoire commence donc au chapitre 1 par dresser un tour d'horizon de l'état de l'art sur la problématique de l'appariement pour l'e-recrutement, son évolution, les techniques utilisées et les domaines d'application. Nous décortiquons principalement, sans produire une liste exhaustive, quasiment toutes les solutions qui ont été proposées dans la littérature en relation avec cette problématique et enfin nous présentons quelques plate-formes existantes d'e-recrutement pour l'automatisation du processus de recrutement sur le Web.

Subséquent, le mémoire présente aux chapitres 2 et 3, l'approche suivie pour vérifier notre hypothèse. Ils sont consacrés à la présentation et l'explication de l'approche théorique et de la méthodologie de notre travail de recherche. Plus particulièrement, nous montrons comment le paradigme des systèmes complexes adaptatifs peut apporter des réponses à nos questions de recherche et comment par l'exploitation de l'approche de modélisation et de la

simulation à base d'agents nous comptons y arriver. Aussi, proposons-nous, dans le chapitre 3, une formalisation de notre modèle conceptuel à base agents du mécanisme d'appariement.

Suite à cela, le mémoire discute au chapitre 4 de la conception de notre prototype et de nos expérimentations pour le projet. Nous présentons l'implémentation du modèle conceptuel pour l'appariement automatique des CV aux offres d'emploi en utilisant le paradigme des agents. Nous décrivons en détail tous les agents utilisés dans notre modèle.

Dans le chapitre 5, nous présentons les résultats et nous procédons à la validation et à l'évaluation, d'une manière empirique, de la pertinence de l'approche proposée et ce en examinant les résultats obtenus et ceux attendus par l'utilisateur.

Le mémoire se conclut finalement par un retour sur les hypothèses de recherche, une synthèse de nos apports de solution, une analyse critique de notre approche, et une proposition d'éventuels travaux en vue d'apporter des améliorations.

# CHAPITRE 1

## ÉTAT DE L'ART

### 1.1 Introduction

La problématique de la recherche d'emploi a toujours été et continue d'être un sujet brûlant au sein de notre société. Pour cela, plusieurs chercheurs se sont évertués à apporter des solutions de par leurs travaux et études. C'est ainsi que, la théorie de la recherche d'emploi (job search theory) s'est muée en une référence depuis les années 1960-70. En effet, elle était essentielle pour étudier les interactions entre les agents sur le marché du travail et analyser le taux de chômage. Cette théorie, proposée au départ par (Stigler, 1962), s'appuie sur le fait que le mécanisme d'appariement entre un chercheur d'emploi et une offre d'emploi prend du temps et qu'il a un coût, ce qui cause donc des frictions sur le marché du travail. La théorie de la recherche d'emploi a été formalisée tout d'abord par des modèles de recherche analytiques qui se placent dans un contexte d'équilibre du marché et modélisent le comportement d'un agent représentatif (Stigler, 1962). Aussi, on a une autre approche pour modéliser le marché du travail qui a été conçue à partir des années 1970 avec les recherches de (Bergmann, 1974) et (Industriforskningens utredningsinstitut (Sweden), 1980). Il s'agit des modèles ACE (pour Agent-Based Computational Economics) qui sont développés comme de programmes informatiques qui simulent les interactions entre un grand nombre d'agents hétérogènes, employeurs et travailleurs, sur une plateforme multi-agent. On voit alors que le marché du travail est perçu, avec cette nouvelle approche, comme un système dynamique complexe. Le premier modèle ACE, qui a mis en œuvre la théorie de la recherche avec les deux côtés du marché du travail, est le modèle ARTEMIS développé au début des années 1980 (Ballot, 1981) ; (Ballot, 2002).

Pourquoi existe-t-il tant de personnes au chômage ou à la recherche d'emploi alors qu'au même moment, un grand nombre de postes ne sont toujours pas pourvus? Telle est la question à laquelle les modèles d'appariement ont essayé d'apporter une réponse. La théorie du *matching* ou encore « *mécanismes d'appariement* » a été mise en exergue lors de la gratification des trois auteurs P. Diamond, D. Mortensen et C. Pissarides au cours de la remise du prix d'économie en l'honneur d'Alfred Nobel en 2010 (Pellet, 2013). Ils se sont basés sur des travaux d'informations imparfaites pour expliquer le phénomène empirique de la coexistence du chômage et d'emplois vacants.

Ainsi, conformément aux progrès des développements technologiques, l'Intelligence Artificielle (IA) s'est propagée à divers secteurs d'activité. De nombreuses études ont été donc faites, par la suite, sur le processus d'appariement dans le domaine du e-recrutement en travaillant avec des données issues des réseaux sociaux. Pour ce faire, nous étudions ces systèmes afin de proposer une solution améliorée. Par ailleurs, nous concevons le recrutement par l'IA comme toute procédure qui utilise l'intelligence artificielle dans le but d'aider les entreprises lors de la sélection et du recrutement des candidats tandis que l'IA peut être vue comme « la capacité d'un système à interpréter correctement des données externes, à apprendre de ces données et à utiliser ces apprentissages pour atteindre des objectifs et des tâches spécifiques grâce à une adaptation flexible » (Kaplan et Haenlein, 2019). Ici, nous voulons nous référer à un concept large de l'IA qui englobe des approches complexes d'apprentissage automatique comme les réseaux de neurones profonds, mais inclut aussi des algorithmes simples reposants sur des analyses de régression ainsi que d'autres types d'algorithmes, tels que le traitement du langage naturel ou la reconnaissance vocale.

Dans la première partie de ce chapitre, nous passons en revue les travaux connexes dans les domaines de l'appariement Emploi-CV et de l'apprentissage avec des données bruitées. La seconde partie sera consacrée aux systèmes de recommandation. Nous allons exposer, les apports des ontologies dans le processus de sélection de candidats. Nous terminerons, par une synthèse des approches existantes et par une présentation de notre modèle basé sur l'appariement des schémas comme un système complexe adaptatif. Dans la dernière partie, nous essayerons d'expliquer en quoi les connaissances actuelles du domaine ne permettent pas de répondre efficacement à notre problématique.

## 1.2 Système d'appariement basé sur l'apprentissage automatique

À l'instar de l'application des techniques d'apprentissage automatique dans le divertissement, et d'autres domaines, l'application et le développement des techniques d'apprentissage automatique dans les systèmes de ressources humaines restent un peu à la traîne (Iwu *et al.*, 2016). Comme chaque entreprise a ses propres exigences particulières en matière de recrutement, il est nécessaire que les entreprises construisent leurs propres algorithmes informatiques pour sélectionner les CV qui répondent à leurs propres exigences. Par conséquent, les universitaires et le personnel des entreprises accordent de plus en plus d'attention à la recherche et à l'application de l'algorithme du système de ressources humaines conformément à leurs propres exigences (Li et Parlikad, 2016).

On a pu remarquer que l'utilisation des sites de réseaux sociaux comme outil de collecte d'informations sur les employés potentiels a proliféré depuis quelques années. Une étude nous montre que les sites de réseautage social figurent parmi les sites les plus visités sur Internet, juste derrière les principaux moteurs de recherche (Ronn, 2007). Ces sites de réseautage facilitent aux utilisateurs la présentation de leurs qualifications et leurs photos d'une manière plus dynamique que le format usuel d'un CV. Divers chercheurs ont constaté que, parmi les sites de réseaux sociaux populaires, celui qui est le plus utilisé à des fins de recrutement est LinkedIn (Ollington et al., 2013 ; Kluemper, 2013 ; Karl et Peluchette, 2013). En outre, étant donné que les recruteurs ont accès à de grandes quantités de candidats potentiels en peu de temps, il devenait primordial de repousser les limites de la recherche et du développement en apprentissage automatique en vue d'optimiser les chances d'une embauche réussie. C'est ce que LinkedIn, à travers son produit LinkedIn Recruiter a réussi à mettre au point. En effet, LinkedIn Recruiter est alimenté par une série d'algorithmes de recherche et de recommandation qui exploitent des architectures d'apprentissage automatique de pointe. Concrètement, après avoir intégré, le défi le plus difficile à relever dans l'expérience LinkedIn Recruiter était l'intégration des préférences des différentes entités dans le processus de recrutement, tels que les recruteurs, les contrats, les entreprises et les candidats. Pour relever donc ce défi, LinkedIn s'est basé sur une méthode statistique appelée Generalized Linear Mixed (GLMix) qui utilise l'inférence pour améliorer les problèmes de prédictions. Dans le fond, LinkedIn Recruiter utilise une architecture qui combine des fonctionnalités d'apprentissage du classement, des fonctionnalités d'interaction d'arborescence et des scores de modèle du Gradient Boosted Decision Tree (GBDT).

Au début du vingt-et-unième siècle, des chercheurs ont étudié le premier système CASPER au monde pour fournir une présélection personnalisée du recrutement des ressources humaines aux utilisateurs de l'entreprise en fonction de l'algorithme d'extraction et de la technologie de présélection des données de journal obtenues du serveur. (Wuest *et al.*, 2016) ont développé des algorithmes d'exigences des entreprises de recherche et de recrutement de personnel pour trouver les unités et les employés appropriés pour répondre aux exigences des demandeurs d'emploi et des entreprises (Wuest *et al.*, 2016). (Barreto *et al.*, 2017) ont utilisé les informations de base dans le CV du personnel pour correspondre aux exigences des postes de recrutement afin d'améliorer la satisfaction et l'efficacité du personnel de recrutement (Barreto *et al.*, 2017). (Longo *et al.*, 2017) ont créé des modèles et des algorithmes pour le dépistage des problèmes dans le système des ressources humaines grâce à l'apprentissage automatique, et ont analysé les données de l'historique de recherche d'emploi des demandeurs d'emploi pour spéculer



sur leur orientation de recherche d'emploi et leur degré d'adéquation avec les entreprises (Longo *et al.*, 2017). Mrugalska et Wyrwicka (2017) ont appliqué l'algorithme SimRank et le processus de hiérarchie analytique pour créer un modèle de sélection intelligent pour l'adéquation de l'emploi du personnel qui sera diplômé de l'université. Les méthodes complètes de spéculation d'exemples, de classification bayésienne et de récupération de texte ont été utilisées pour filtrer le CV des demandeurs d'emploi afin de répondre à leurs besoins particuliers, et les très bons résultats ont été obtenus (Mrugalska et Wyrwicka, 2017). Tokody (2018) a utilisé les mérites du filtrage de contenu et du filtrage collaboratif ensemble, en espérant améliorer la fonction de recommandation et de filtrage et l'efficacité du travail (Tokody, 2018). Mongia *et al.* (2020) ont proposé un modèle plus profond de facteurs potentiels basé sur l'apprentissage profond. La technologie de ce modèle est manifestement supérieure à celle du filtrage collaboratif actuel (Mongia *et al.*, 2020). Abualigah *et al.* (2019) ont proposé une nouvelle méthode de sélection des caractéristiques basée sur l'algorithme d'optimisation par essais de particules pour obtenir un sous-ensemble de caractéristiques contenant plus d'informations dans un espace de faible dimension. Ce sous-ensemble a été utilisé pour améliorer les performances de l'algorithme de text-clustering et diminuer le temps de calcul (Abualigah, 2019). Hammou *et al.* (2019) ont proposé un nouveau schéma de recommandation distribué pour le big data. Il a été conçu sur la base d'Apache Spark pour traiter les données à grande échelle, améliorer la qualité de la prédiction et résoudre le problème de la rareté des données (Hammou *et al.*, 2019). Avec l'amélioration et la popularisation rapides de la technologie connexe de l'apprentissage automatique, de plus en plus de chercheurs ont appliqué certains algorithmes d'apprentissage automatique au recrutement des ressources humaines dans les entreprises, notamment des modèles arithmétiques tels que le random forest, le réseau de neurones<sup>1</sup>, l'implicit semantic model, etc.

Tao *et al.* (2018) ont construit un algorithme de filtrage intelligent hybride basé sur le multi-agent par le biais de recherches indépendantes et ont proposé un modèle d'algorithme basé sur la perception en fonction de la prise en compte du temps (Tao *et al.*, 2018). Xu *et al.* (2018) ont élaboré un modèle d'algorithme basé sur le filtrage collaboratif en concevant un modèle implicite pour filtrer les informations, ont analysé et interprété la définition du degré d'influence sur la base de la confiance, et ont conçu un modèle de filtrage et de filtrage de base (Xu *et al.*, 2018). Jiang *et al.* (2019) ont proposé une nouvelle

---

<sup>1</sup> De l'anglais « neural network »

méthode de hachage modal croisé basée sur un modèle de facteur potentiel discret pour la recherche de similarité modale croisée (Jiang et Li, 2019).

En résumé, le modèle d'algorithme basé sur l'apprentissage automatique a été grandement appliqué dans le système de ressources humaines de l'industrie de la fabrication intelligente après de nombreuses années de développement et de progrès. En faisant correspondre, en triant et en filtrant les caractéristiques de l'emploi de l'utilisateur et les exigences des employés de l'entreprise, il peut améliorer efficacement la poursuite des objectifs pour les utilisateurs et les entreprises.

### 1.3 Les Systèmes de Recommandation

Le système de correspondance des emplois est une sorte de système de recommandation. Le système de recommandation a été présenté pour la première fois par (Resnick et Varian, 1997) qui ont souligné que dans un système de recommandation typique, les gens fournissent des recommandations en entrée, que le système regroupe ensuite et dirige vers les destinataires appropriés. Par la suite, les systèmes de recommandation ont été largement acceptés dans diverses industries et domaines académiques et gagnent du terrain au fil des ans. En général, les systèmes de recommandation sont appliqués dans divers domaines (tels que les livres, les produits numériques, films, musiques, programmes télévisés et sites Web) et aident les utilisateurs à trouver du contenu, des produits ou des services en regroupant et en analysant les données, produits ou services en agrégeant et en analysant les suggestions et les comportements des autres utilisateurs (Breugh, 2009), (Das *et al.*, 2007). Dans un article d'enquête détaillé (Lu *et al.*, 2015), les chercheurs ont pu prendre connaissance de l'état des connaissances sur le système de recommandation, y compris les applications dans le monde réel, les méthodes de recommandation, et les plateformes d'application.

Dans le domaine de l'adéquation entre l'offre et la demande d'emploi, de nombreuses recherches ont été menées pour discuter de différents systèmes de recommandation liés au problème du recrutement (Al-Otaibi et Ykhlef, 2012). Parmi elles, (Malinowski *et al.*, 2006) ont discuté d'un système de recommandation de mise en correspondance bilatérale pour rapprocher les personnes et les emplois à l'aide d'un algorithme de maximisation de l'espérance (EM)<sup>2</sup>, tandis que (Golec et Kahya, 2007) ont défini un modèle flou pour l'évaluation et la sélection des employés en fonction de leurs compétences avec des règles

---

<sup>2</sup> De l'anglais « Expectation-Maximization algorithm »

floues<sup>3</sup>. (Paparrizos *et al.*, 2011) ont utilisé une table de décision/bayes naïves (DTNB)<sup>4</sup> comme classificateur hybride. Bien que ces systèmes aient utilisé de nombreux attributs manuels et diverses techniques de recherche d'informations, ils n'ont toujours pas réussi à suivre l'évolution rapide des capacités informatiques et de l'intelligence des machines. Comme les travaux de (Zhang *et al.*, 2015), les nôtres tentent également d'optimiser l'adéquation entre le travailleur intellectuel et le poste, en tenant compte de diverses caractéristiques des travailleurs intellectuels. Par rapport aux travaux de (Guo *et al.*, 2016), notre solution est entièrement basée sur les données, sans utiliser d'outil sémantique extérieur (NLTK et DBpedia) comme ils l'ont fait. En outre, les données de CV sont largement collectées dans différents domaines (Lin *et al.*, 2016).

#### 1.4 Les ontologies du domaine RH

L'ontologie est le modèle fondamental pour organiser l'information répondant aux exigences d'un système de recrutement (Chandrasekaran *et al.*, 1999). L'ontologie est utilisée pour traiter la signification des données, informations ou connaissances cibles dans le modèle du Web sémantique (Calaresu et Shiri, 2015). De plus, l'ontologie surmonte la compréhension la plus élémentaire du Web sémantique en fournissant des capacités de raisonnement standard, normalement basées sur la spécification de règles d'inférence. Dans le système de filtrage des CV, l'ontologie est utilisée pour représenter les informations extraites des CV des candidats et des descriptions du poste des recruteurs. Il existe plusieurs études pour résoudre le problème de la sélection de CV pour une description de poste.

Plusieurs auteurs ont proposé de nombreux systèmes automatisés avec différentes approches. Dans (Senthil Kumaran et Sankar, 2013), les auteurs ont utilisé la cartographie ontologique pour construire un système automatisé de sélection intelligente des candidats au recrutement. Leur recherche s'est concentrée sur la construction de documents d'ontologie pour le problème des candidats afin de recommander les CV les plus appropriés pour le poste en utilisant la mesure de similarité. Ils ont utilisé la liaison de concepts pour connecter des données/documents en identifiant des concepts communément partagés, où les modèles de données sont des ontologies. Grâce à cela, ils ont utilisé le mappage d'ontologies pour classer les interactions sémantiques entre les éléments liés de diverses ontologies. Ils ont construit leur ontologie pour les caractéristiques des candidats afin de faire correspondre les

---

<sup>3</sup> De l'anglais « Fuzzy rule »

<sup>4</sup> De l'anglais « decision table/naive bayes hybrid classifier »

informations entre CV et JD. Cependant, la capacité d'extraire des compétences du texte de cette étude est assez limitée. Il n'extrait pas exactement les informations nécessaires pour construire une ontologie dans la pratique.

Une autre approche consiste à créer un système de recrutement électronique intégré pour l'exploration automatisée de la personnalité et le classement des candidats. L'étude de (Faliagka *et al.*, 2012) a présenté un système de recrutement en ligne mettant en œuvre un classement automatisé des candidats basé sur un ensemble d'exigences crédibles qui facilitent l'intégration des entreprises à leur infrastructure actuelle de gestion des ressources humaines (RH). Ce système construit a mis en œuvre un classement automatisé des candidats basé sur des paramètres objectifs obtenus à partir du profil LinkedIn des candidats. Le classement du profil des candidats est calculé par critères de sélection individuels à l'aide du processus de hiérarchie analytique (AHP), tandis que leur importance relative (poids) est contrôlée par le recruteur. Bien que cette étude ait montré un système utile pour extraire la personnalité des candidats, elle ne peut pas extraire les compétences des informations de profil des candidats. Ainsi, il est très difficile de postuler en recrutement informatique, L'apprentissage par transfert est une technique d'apprentissage automatique utile pour traiter les problèmes de langage (Pillai et Sivathanu, 2020), (Nguyen *et al.*, 2020), (Mishra *et al.*, 2020), en particulier l'extraction des mots principaux dans les documents. L'étude de (Mishra *et al.*, 2020) a utilisé l'ontologie et l'incorporation de mots pour extraire les compétences des candidats. Ils ont proposé une théorie sur l'appariement du candidat avec la culture de l'organisation pour donner un meilleur candidat adapté à l'entreprise. Dans (Huynh *et al.*, 2021), une approche utilisant le Policy Gradient par apprentissage par renforcement a été utilisée pour simuler le contrôle d'une voiture autonome. Cependant, ces méthodes ne peuvent pas être utilisées pour filtrer automatiquement les CV et leurs résultats ne sont pas efficaces pour postuler à la construction d'un système de recrutement dans le monde réel.

La recherche dans (Baad, 2019) a généré automatiquement une taxonomie des compétences professionnelles pour les systèmes de recrutement en utilisant des modèles de réseaux neuronaux récurrents pour capturer le contexte des mots de compétence. Cette étude a également mis en œuvre certaines méthodes pour le système, telles que la Bidirectional Long Short-Term Memory (BiLSTM), la Long Short-Term Memory (LSTM) et le Conditional Random Field (CRF). En outre, quelques méthodes basées sur des règles ont été utilisées pour extraire les compétences apparaissant devant les phrases précédentes indiquant le début des compétences de l'OE. Néanmoins, étant donné que leur marqueur du Part Of

Speech (POS) a étiqueté ces compétences comme d'autres balises, cette approche manque de nombreuses compétences et ne peut pas détecter les compétences des JD non structurés. De plus, cette méthode n'est pas efficace pour extraire les compétences du CV car la structure d'un CV est différente de celle de l'OE.

### 1.5 Modèles de recherche analytique

Certaines études dans le domaine du recrutement électronique se situent au niveau de l'annotation sémantique de documents du fait de sa simplicité et de sa capacité à expliciter le contenu des documents. L'apport réside dans le fait que les données formalisées produites seront utilisées dans un rapprochement automatique entre ces documents.

De prime abord, l'adaptation du système multi-agents BONOM (Rahm, 2011a) pour le domaine de l'indexation sémantique, a été introduite pour la recherche d'un profil particulier dans un ensemble de CV. Les agents forment une organisation selon une hiérarchie de domaines. Cette hiérarchie sera le canal par lequel les requêtes des agents utilisateurs seront véhiculées vers les agents spécialisés. À chaque domaine spécifique, on attribue l'ontologie des métiers/emplois correspondants. Ensuite, à cause de sa simplicité, de son opportunité dans la modélisation du contenu sémantique et de la disponibilité des normes et des outils du Web sémantique, plusieurs travaux se sont basés sur cette approche pour l'annotation sémantique.

Dans un second temps, en se basant sur une ontologie du domaine approprié, le Projet COMMONCV (Harzallah *et al.*, 2002) intitulé «Recherche & Emploi» a développé un noyau ontologique dédié à la représentation des compétences sous-jacentes à un CV. Aussi, voit-on dans le projet OS-SKILL (Yahiaoui *et al.*, s. d.), une application Internet/Intranet qui apporte des réponses aux besoins spécifiques de chaque entreprise en ce qui concerne la gestion des emplois et des compétences. On notera, par ailleurs, qu'elle entièrement paramétrable. Cette application offre de nombreuses fonctionnalités qui prennent en compte le «sens métier»: cartographie des métiers. OS-SKILL est à la fois une méthode et un outil:

- Une méthode car étant capable de construire des référentiels des métiers et des compétences basée sur la norme ISO 704, ce qui guide étape par étape la construction des Ontologies.
- Un outil qui considère le sens du «métier» et les besoins liés à la capitalisation et à la valorisation des connaissances. Il repose sur les technologies XML et les services Web.

## 1.6 Conclusion

Ainsi, de nombreux travaux ont œuvré pour résoudre cette problématique, et sans en produire une liste exhaustive vu les contraintes de cette proposition, au meilleur de notre connaissance, quasiment toutes les solutions qui ont été proposées dans la littérature, appartiennent au paradigme réductionniste. Ce dernier ramène le processus d'appariement, à une fonction linéaire avec un ensemble d'intrants et d'extrants. Cette fonction peut se décomposer en une série de modules, dont chacun est responsable du déroulement d'une étape du processus (e.g. sélection et exécution des apparieurs<sup>5</sup>) et se constituant d'éléments plus au moins simples (ex. apparieurs, fonctions, paramètres). Les tentatives de résolution des difficultés rencontrées par l'appariement automatique (ex. incertitude), suivent le même schéma de pensée, à savoir la focalisation sur un aspect bien précis du fonctionnement du processus et ensuite la tentative d'optimisation de ce module ou de la composante responsable de cet aspect. Cette façon de faire se heurte au constat que l'appariement automatique est une tâche complexe à plus d'un titre, en raison au moins de:

- (i) l'hétérogénéité et l'ambiguïté intrinsèque aux éléments à apparié,
- (ii) le caractère imprévisible des résultats de l'appariement,
- (iii) le défi que peut engendrer l'optimisation de l'appariement (explosion combinatoire), etc.

Dans notre projet, nous défendons l'hypothèse que la problématique de l'appariement automatique mériterait qu'elle soit abordée dans son entièreté et vue comme un système complexe comprenant plusieurs composants, dont la simulation des interactions non-linéaires permet l'émergence du résultat final. Nous proposons une adoption d'un nouveau paradigme en ayant recours à la théorie des systèmes complexes adaptatifs (SCA), issues de la pensée systémique, pour proposer des réponses aux défis difficiles auxquels les approches réductionnistes classiques d'appariement automatique, font toujours face (par exemple complexité, incertitude). L'hypothèse centrale est donc de considérer le processus de l'appariement comme un SCA et de le modéliser en utilisant l'approche de la modélisation et de la simulation à base d'agents (Assoudi et Lounis, 2014), (Assoudi, 2017). Le but visé étant l'exploitation des propriétés intrinsèques aux SCA (i.e. modèles à base d'agents), notamment l'émergence, la stochasticité

---

<sup>5</sup> De l'anglais « matcher »

et l'auto-organisation, pour contribuer à apporter des réponses aux questions ouvertes, à savoir mieux gérer la complexité et l'incertitude. Dans ce modèle d'appariement à base d'agents, chaque élément à appairer (champs d'un CV, rubriques d'une offre d'emploi, etc.) est modélisé comme un agent autonome, faisant partie d'un groupe (groupe source ou cible) dont la simulation des comportements et de l'interaction avec son environnement, au niveau micro, fait émerger au niveau macro, un système auto organisé qui représente la solution globale à l'appariement (i.e. relations entre les éléments à appairer). En d'autres termes, la résolution du processus d'appariement passe par l'effort individuel que déploie chaque agent d'un groupe source, d'une manière autonome et locale tout au long de la simulation, pour trouver la meilleure relation de correspondance dans le groupe cible (meilleur appariement). (Assoudi, 2017)

## CHAPITRE 2

### APPARIEMENT DE SCHÉMAS

#### 2.1 Introduction

Ce chapitre sera structuré en deux parties. Dans la première partie, nous allons de prime abord, montrer pourquoi l'appariement des CV aux OE peut être vu comme un appariement de schémas, tout en exposant les différentes techniques, méthodes et approches existantes pour le défi que représente l'appariement automatique de schémas. Puis, nous allons analyser les principales difficultés, inhérentes à une automatisation totale de ce processus, comme l'incertitude. Par ailleurs, nous exposerons les principales solutions proposées pour tenter de résoudre ces problèmes, comme l'optimisation. Pour finir, au terme de ce chapitre, nous présenterons une analyse qui se veut être une synthèse mettant en phase les principales insuffisances que nous relevons par rapport aux approches existantes ainsi que les limites qui, selon notre point de vue, empêchent d'apporter des réponses holistiques aux questions qui demeurent encore ouvertes.

Dans la deuxième partie, nous présentons le cadre théorique de notre nouvelle approche d'appariement automatique de schémas à savoir la théorie de la complexité, et de manière plus spécifique le paradigme des systèmes complexes adaptatifs. Aussi, allons-nous tenter d'explorer les racines de ce paradigme et présenter ses principales caractéristiques. Pour terminer, nous allons nous atteler à mettre de l'avant l'opérationnalisation d'un système complexe adaptatif via la modélisation et la simulation à base d'agents.

#### 2.2 L'appariement des CV aux offres d'emploi comme un appariement de schémas

L'appariement de schémas, selon les auteurs (Bernstein *et al.*, 2011), est le problème de la génération de correspondances entre les éléments de deux schémas. Un schéma est une structure formelle qui représente un artefact conçu, tel qu'un schéma SQL, Schéma XML, diagramme entité-relation, description d'ontologie, définition d'interface ou définition de formulaire. Une correspondance, toujours d'après ces auteurs est une relation entre un ou plusieurs éléments d'un schéma et un ou plusieurs éléments de l'autre. Par exemple, les correspondances dans la figure 2.1, identifient les colonnes qui représentent les mêmes concepts dans les deux schémas relationnels. Souvent, la relation est un-à-un, mais parfois elle ne l'est pas, comme l'Auteur correspondant au Nom et au Prénom dans la même figure.



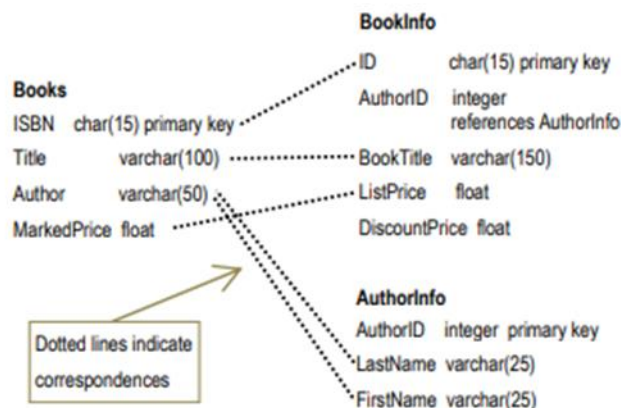


Figure 2.1 Exemple de schémas pour l'appariement (Bernstein *et al.*, 2011)

L'appariement de schémas vise à identifier les correspondances sémantiques entre les structures de métadonnées ou les modèles, tels que les schémas de base de données, les formats de message XML et les ontologies. La résolution de tels problèmes de correspondance est une tâche clé dans de nombreux domaines d'application, en particulier pour l'échange de données et pratiquement tous les types d'intégration de données (Rahm et Peukert, 2019). Ainsi, la correspondance de schémas est très importante pour les systèmes qui manipulent des données sur différentes sources de données. On a des exemples de domaines où ces applications sont utilisées, comme la bio-informatique, l'intégration de données sur le World Wide Web, le commerce électronique, l'entreposage de données et la collaboration scientifique. Par conséquent, l'appariement de schémas a attiré beaucoup l'attention de la communauté des chercheurs, au cours des dernières décennies et c'est encore à ce jour, un énorme domaine d'intérêt pour ces derniers (Yousfi *et al.*, 2020).

En ce qui concerne l'appariement des CV aux OE, nous pensons qu'il est une variante de l'appariement de schémas dans le sens où un curriculum ou une offre d'emploi est aussi une structure formelle qui représente donc une définition de compétences, de formations, d'expérience etc. Une correspondance entre ces deux structures, sera donc une relation entre un ou plusieurs éléments du curriculum et un ou plusieurs éléments de l'offre d'emploi. Puisque l'appariement de curriculum vitae aux offres d'emploi vise à identifier les correspondances sémantiques au même titre que l'appariement de schémas, on pense que la résolution de cet appariement devrait pouvoir se faire comme l'appariement de schémas.

Par ailleurs, divers systèmes d'appariement de schémas ont été développés pour résoudre le problème de manière semi-automatique. Bien qu'il y ait eu quelques évaluations, l'efficacité globale des systèmes de mise en correspondance automatique de schémas actuellement disponibles est largement incertaine. En effet, les évaluations ont été menées de diverses manières, ce qui rend difficile l'évaluation de l'efficacité de chaque système, et encore moins la comparaison de leur efficacité (Do *et al.*, 2002).

Les auteurs (Rahm et Bernstein, 2001a) définissent, au travers de leur étude sur les différentes approches pour la résolution de la problématique d'appariement de schémas, un schéma comme un ensemble d'éléments reliés par une structure de donnée. Le schéma est appelé à être représenté par une notation qui permet de modéliser d'une manière naturelle et logique la notion d'élément et de structure tel qu'un modèle orienté-objet, un modèle d'entité-relation, XML. Il doit être encore représenté sous la forme d'un modèle de graphes orientés. Par ailleurs, un élément aléatoire d'un schéma est relié à un autre élément d'un autre schéma, ceci est montré par une relation de correspondance nommée « Matching ». La relation entre les deux éléments peut être décrite par une expression de transformation « Mapping ». L'opération de l'appariement « Match » est définie comme une fonction qui prend en entrée les deux schémas et retourne en sortie les correspondances entre ces deux schémas (résultat de l'appariement). (Assoudi et Lounis, 2015)

Le processus d'appariement peut être visualisé sous la forme du déroulement d'un match, à savoir :

- la phase d'avant-match,
- le match et
- la phase d'après-match.

Pour mieux comprendre les processus d'un appariement, les auteurs (Do et Rahm, 2007) proposent une description de l'architecture de COMA++ qui se compose de cinq composants, le répertoire qui stocke de manière persistante les données liées aux correspondances, les pools de schéma et de mappage pour gérer les schémas et les mappages en mémoire, le personnalisateur de correspondance pour configurer les matchers, les stratégies de correspondance et le moteur d'exécution pour exécuter les opérations de correspondance. Tous les composants sont gérés et utilisés via un interface utilisateur graphique (GUI) intégrée.

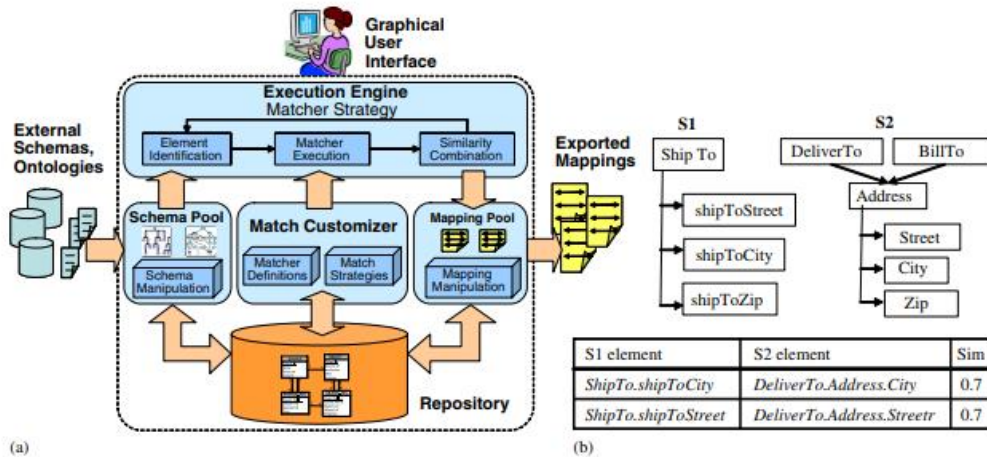


Figure 2.2 Processus de l'appariement de schémas (Do et Rahm, 2007)

Comme illustré sur la droite de la figure 2.2, les schémas, d'après les auteurs (Do et Rahm, 2007), sont uniformément représentés par des graphes acycliques orientés comme le format interne pour la correspondance. Le pool de schémas fournit différentes fonctions pour importer des données de schémas externes et pour les charger et les enregistrer depuis ou vers le dépôt. Ici, ils prennent en charge la définition de schéma XML (XSD), les données XML réduites (XDR), schémas relationnels via une connectivité de base de données ouverte (ODBC) et le langage d'ontologie Web (OWL).

À partir du pool de schémas, deux schémas arbitraires peuvent être sélectionnés pour démarrer une opération de correspondance. L'opération de l'appariement de manière générale est effectuée selon une stratégie de correspondance préalablement configurée. Comme indiqué à gauche dans figure 2.2, elle est basée, toujours d'après les auteurs (Do et Rahm, 2007), sur l'itération en trois étapes, à savoir:

- L'identification d'éléments pour déterminer les éléments de schéma pertinents pour l'appariement.
- L'exécution de l'appariement appliquant plusieurs correspondances pour calculer les similitudes d'éléments, tout en tenant compte du contexte.
- La dernière étape présentera les résultats d'un premier appariement qui pourra par la suite être considéré pour d'autres futures itérations.

Le résultat de correspondance obtenu, ou mapping, est un ensemble de correspondances, dont chacune des correspondances représente une seule paire d'éléments correspondants et un score de similarité indiquant la plausibilité de la relation de correspondance. La droite de la figure 2.2 montre dans la partie inférieure un tableau de résultat simple avec deux correspondances où les éléments sont décrits par leurs chemins.

La correspondance est le processus qui prend en entrée deux schémas, appelés source et cible, et produit un certain nombre de correspondances, entre les éléments de ces deux schémas (Rahm et Bernstein, 2001b). Le terme schéma est utilisé au sens large et inclut les schémas de bases de données (Madhavan *et al.*, 2001), les ontologies (Giunchiglia *et al.*, 2009), ou des modèles génériques (Atzeni et Torlone, 1996).

Une correspondance est définie comme un triplet  $\langle S_s, E_t, e \rangle$ , où  $S_s$  est un ensemble d'éléments de la source,  $E_t$  est un élément du schéma cible et  $(e)$  est une expression de correspondance qui spécifie une relation entre l'élément  $E_t$  et les éléments de  $S_s$ . Cependant, étant donné un schéma source et un schéma cible, un mapping est une relation, c'est-à-dire une contrainte, qui doit exister entre leurs instances respectives. Pour que les mapping soient générés, une exigence fondamentale est représentée par les correspondances entre les éléments des schémas. Ainsi, selon (Bohannon *et al.*, 2006), le concept de « Schema Matching » concerne la recherche des correspondances entre les éléments des schémas à appairer, alors que le concept de « Schema Mapping » concerne la transformation des correspondances entre les éléments des schémas à appairer.

De nombreux travaux ont voulu rendre le processus plus automatique, sans nécessiter d'intervention humaine.

L'hétérogénéité sémantique relatives aux différents schémas rend difficile, pour les approches automatiques, la détermination des correspondances, de sorte qu'il est nécessaire d'impliquer l'utilisateur. La complexité est particulièrement plus élevée pour les applications Big Data, qui impliquent fréquemment une correspondance de schémas à grande échelle ayant des milliers d'éléments, qui doivent être traités.

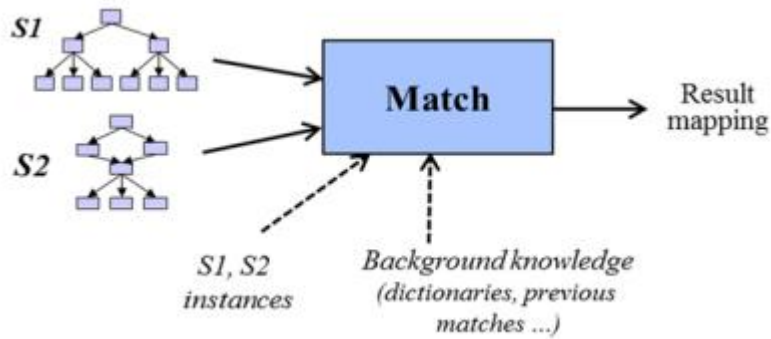


Figure 2.3 Processus simplifié de l'appariement de schémas (Rahm et Peukert, 2019)

D'après les auteurs (Rahm et Peukert, 2019), la plupart des approches pour l'appariement de schémas se concentre sur l'appariement bidirectionnel ou par paires, où deux schémas d'entrée sont appariés l'un à l'autre. Comme le montre la figure 2.3, la correspondance automatique peut également utiliser des données d'instances ( $S_1, S_2$  instances) pour les schémas d'entrée ou des connaissances de base telles que des dictionnaires ou des résultats de correspondance précédents qui simplifient l'identification des éléments correspondants dans les schémas d'entrée. Le résultat de la correspondance de schémas par paire est généralement une équivalence mappage contenant les correspondances sémantiques identifiées, c'est-à-dire des paires d'éléments de schéma sémantiquement équivalents (par exemple, des attributs de schéma ou des concepts d'ontologie).

Certaines approches de correspondance pour les ontologies tentent également de déterminer différents types de correspondances sémantiques, telles que les relations entre concepts (Grau *et al.*, 2013), (Arnold et Rahm, 2014). En raison de l'hétérogénéité sémantique généralement élevée des schémas, les algorithmes peuvent seulement déterminer des mapping approximatifs. Les mapping déterminés automatiquement peuvent ainsi nécessiter l'inspection et l'adaptation par des experts humains du domaine (suppression des correspondances erronées, ajout des correspondances manquantes) pour obtenir le mappage correct. Les résultats du mapping sont des entrées utiles pour d'autres tâches d'intégration de données telles que la fusion ou l'intégration des schémas respectifs, car un mappage indique les éléments qui ne doivent être représentés qu'une seule fois dans le résultat intégré. En fait, plusieurs de ces approches de fusion basées sur du mapping ont été proposées pour les schémas et les ontologies (Pottinger et Bernstein, 2003), (Raunich et Rahm, 2014). L'appariement de schémas a été un

domaine de recherche très actif au cours des dernières décennies, et de nombreuses techniques et prototypes d'appariement automatique ont été développés (Rahm et Bernstein, 2001a), (Grau *et al.*, 2013). Les correspondances sont généralement identifiées par une combinaison de techniques (également appelées *matchers*) pour déterminer la similitude des éléments. L'état de l'art de ces techniques, si nous nous référons au travaux des auteurs (Bernstein *et al.*, 2011), nous dévoile quelques unes, à savoir:

- L'appariement basé sur la structure des schémas : qui détermine la similitude des éléments sur la base de la similitude de leurs schémas voisins comme les ancêtres et/ou descendants.
- L'appariement basé sur les instances : qui détermine la similitude des éléments en fonction de leurs données d'instance associées, par exemple, les valeurs d'attributs ou les entités d'un concept.
- L'appariement basé sur la linguistique : qui se concentre sur la similitude des noms d'éléments basés sur des mesures de similarité de chaînes ou des informations de synonymie provenant de ressources de connaissances de base, telles que les dictionnaires.
- L'appariement basé sur une source auxiliaire d'information : qui détermine la similitude en exploitant des thésaurus, dictionnaires (e.g. acronymes, synonymes) externes pour trouver des correspondances.
- L'appariement basé sur les règles : qui se base principalement sur des règles d'appariement exprimées en logique de premier ordre pour trouver des correspondances.
- L'appariement hybride : combine plusieurs techniques.
- L'appariement basé sur les contraintes : qui se base sur l'exploitation de contraintes définies au niveau des éléments de schémas comme le type ou les contraintes d'intégrité, pour trouver des correspondances.

De nombreux travaux ont tenté d'améliorer l'efficacité et l'efficacite de l'appariement de schémas (et d'ontologies) par paires. Le principal problème de performance (efficacité) pour faire correspondre un grand nombre de schémas est le potentiel énorme espace de recherche si l'on doit comparer chaque élément du premier avec chaque élément du second schéma d'entrée. Pour les schémas avec des milliers ou des dizaines de milliers d'éléments, cela conduit à un espace de recherche de plusieurs millions de comparaisons. Cela entraînera non seulement un long temps d'exécution pour l'évaluation de plusieurs appariements (*matchers*), mais rendra également difficile la découverte de correspondances importantes pour autant de candidats possibles. Par conséquent, il est impératif que l'appariement à grande échelle réduise l'espace de recherche. De plus, on peut appliquer une correspondance parallèle pour améliorer

les temps d'exécution. On va noter que ces techniques (réduction de l'espace de recherche et appariement parallèle) sont également courantes pour l'appariement d'entités à grande échelle (Hartung *et al.*, 2013), (Qiu *et al.*, 2015).

L'appariement parallèle est relativement simple mais n'a été appliqué que dans quelques prototypes, pour la correspondance de schéma par exemple, dans Gomma (Kirsten *et al.*, 2011) et SPHeRe (Amin *et al.*, 2016). Ça peut être utilisé pour accélérer les matchs simples, par exemple, en partitionnant l'espace de recherche (ensemble d'éléments paires des schémas d'entrée) et évaluer différentes partitions en parallèle sur différents processeurs. De plus, différents apparieurs peuvent être évalués en parallèle les uns par rapport aux autres.

En outre, selon (Bernstein *et al.*, 2011), depuis 2001, plusieurs différentes techniques ont été développées, prenant en compte divers algorithmes qui exploitent de nouveaux types d'informations à savoir:

- L'appariement basé sur les graphes: qui s'appuie sur l'étude des relations entre les éléments des graphes (représentant les schémas). Selon (Melnik *et al.*, 2002), cette comparaison se base, par exemple, par calculs à virgule fixe sur la propagation des similarités en fonction des arcs des graphes comme la comparaison de la similarité entre les nœuds adjacents, ou encore par l'utilisation d'algorithmes de satisfaction de contraintes probabilistes (Doan *et al.*, 2002).
- L'appariement basé sur le lien de documents: où les concepts de deux ontologies sont considérés comme similaires si les entités faisant référence à ces concepts sont similaires (Lambrix et Tan, 2008).
- L'appariement basé sur le contenu des documents : où les instances d'un élément de schéma sont groupés dans un document qui est ensuite comparé à un autre document (représentant les instances d'un autre élément), en utilisant des mesures de similarité TF-IDF (utilisée dans le domaine de l'extraction de l'information) (Smith *et al.*, 2009), (Massmann et Rahm, 2008).
- L'appariement basé sur l'utilisation: à titre d'exemple, l'idée qui consiste à exploiter les informations extraites des journaux de requêtes pour trouver des correspondances entre les éléments des schémas à appairer (Elmeleegy *et al.*, 2008).

Selon les auteurs (Bernstein *et al.*, 2011), la plupart des techniques répertoriées dans différents travaux ont été implémentées dans un grand nombre d'outils de mise en correspondance de schémas et

d'ontologies (Euzenat et Shvaiko, 2007), (Rahm, 2011b), (Bellahsene *et al.*, 2011). La figure 2.4 montre un aperçu comparatif des outils sélectionnés :

		Cupid	COMA++	Falcon-			Agreement-	OII
Year of introduction		2001	2002/2005	AO	RIMON	ASMOV	Maker	Harmony
Input	<i>relational</i>	✓	✓	-	-	-	-	✓
schemas	<i>XML</i>	✓	✓	-	-	-	✓	✓
	<i>ontologies</i>	-	✓	✓	✓	✓	✓	✓
OAEI participation		-	✓	✓	✓	✓	✓	-
Comprehensive GUI		-	✓	(✓)	?	?	✓	✓
Matchers	<i>linguistic</i>	✓	✓	✓	✓	✓	✓	✓
	<i>structure</i>	✓	✓	✓	✓	✓	✓	✓
	<i>instance</i>	-	✓	-	✓	✓	✓	-
Use of external dictionaries		✓	✓	?	✓	✓	✓	✓
Schema partitioning		-	✓	✓	-	-	-	-
Parallel matching		-	-	-	-	-	-	-
Dynamic matcher selection		-	-	-	✓	-	-	-
Mapping reuse		-	✓	-	-	-	-	-

Figure 2.4 Tableau comparatif des outils d'appariement (Bernstein *et al.*, 2011)

- CUPID (Madhavan *et al.*, 2001) : est un outil d'appariement qui repose sur une approche hybride. C'est-à-dire, il combine à la fois une approche de similarité basée sur les éléments et une approche de la similarité basée sur les structures. Autrement dit, la similarité linguistique et une similarité structurelle forme une combinaison qu'est le degré de similarité.
- COMA++ (Aumueller *et al.*, 2005) (Do et Rahm, 2007) : est un outil de correspondance de schémas et d'ontologies développé à l'Université de Leipzig en Allemagne. Il étend le premier prototype COMA (Do et Rahm, 2002) en utilisant une approche composite pour combiner différents algorithmes de correspondance. COMA++ met en œuvre des améliorations significatives et offre une infrastructure complète pour résoudre de grands problèmes de correspondance dans le monde réel. Il favorise la création de différentes stratégies d'appariement et ce en permettant la combinaison d'un ensemble important de paramètres d'appariement et d'apparieurs. La combinaison de différents apparieurs, lui permet d'avoir de bonnes performances dans de nombreux domaines comme la mise en correspondance de schémas XML d'annuaires Web ou encore au niveau de méta-modèles dérivés d'UML. (Rahm, 2011a) (Rahm, 2011b).



- FALCON-AO (Hu *et al.*, 2008) : est un prototype d'appariement d'ontologies qui prend en charge une approche de partitionnement pour réduire l'espace de recherche et utilise des coefficients de similarité de schémas linguistiques et structurels pour contrôler la combinaison des résultats de l'apparieur. Par ailleurs, la correspondance basée sur les instances n'est pas encore fournie. (Rahm, 2011a) (Rahm, 2011b). Il a été développé à l'Université du Sud-Est à Nanjing, en Chine (Hu *et al.*, 2008).
- RIMOM (Li *et al.*, 2008): est l'un des premiers systèmes mettant en œuvre une sélection dynamique d'apparieurs. Les éléments du schéma et leurs instances sont d'abord mis en correspondance linguistiquement ensuite la correspondance structurelle n'est appliquée que si les schémas présentent une similarité structurelle suffisante. Il a été développé à l'Université Tsinghua de Pékin, Chine (Li *et al.*, 2008).
- ASMOV (Jean-Mary *et al.*, 2009): Automated Semantic Matching of Ontologies with Verification (ASMOV) est l'un des systèmes les plus performants lors des récents concours de correspondance OAEI<sup>6</sup>. Sa caractéristique la plus distinctive est un post-traitement approfondi des résultats combinés de l'appariement pour éliminer les incohérences potentielles parmi l'ensemble des correspondances candidates. (Rahm, 2011a) (Rahm, 2011b).
- AgreementMaker (Cruz *et al.*, 2009) : est un outil d'appariement d'ontologies développé à l'Université de l'Illinois à Chicago. Il fournit une interface graphique sophistiquée pour que l'utilisateur puisse contrôler l'exécution itérative des apparieurs.
- OII Harmony (Seligman *et al.*, 2010) : est le composant de jumelage au sein de l'Open Information Integration project sur le développement d'une infrastructure accessible au public, pour l'intégration de l'information. Il fournit de nombreuses fonctionnalités connues des prototypes d'appariements précédents ainsi qu'une interface graphique. La correspondance basée sur les instances n'est pas encore prise en charge. La combinaison des résultats de l'apparieur utilise une combinaison non linéaire de valeurs de similarité pour favoriser les apparieurs avec des valeurs de similarité plus élevées (Mork *et al.*, 2008). Selon (Smith *et al.*, 2009), Harmony est capable de faire correspondre des schémas plus grands avec environ 1 000 éléments chacun.

---

<sup>6</sup> OAEI : Ontology Alignment Evaluation Initiative

On constate qu'à haut niveau, les techniques d'appariement partagent toutes des similitudes importantes en ce qui concerne le fonctionnement du processus et ce malgré la multiplication des approches, des techniques et des outils d'appariement.

### 2.2.1 Le déroulement de l'appariement

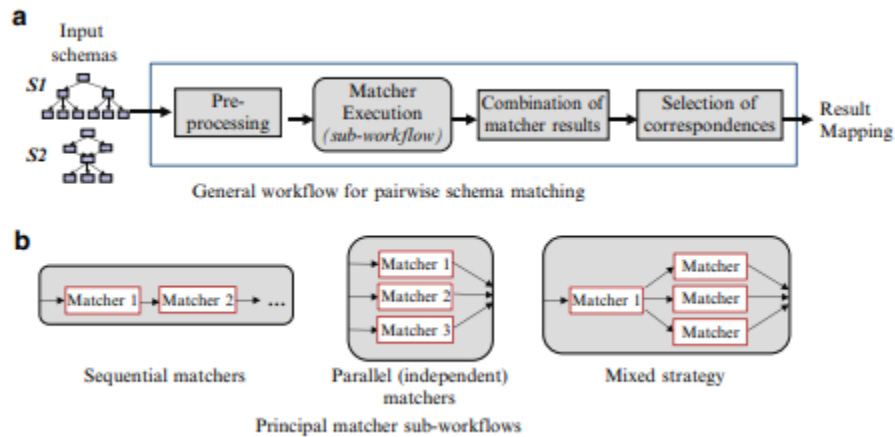


Figure 2.5 Processus général de l'appariement de schémas (Bellahsene *et al.*, 2011)

Selon les auteurs (Gal, 2006), l'appariement automatique de schémas peut être scindé en deux principales phases. Lors de la première phase, le score ou degré de similarité entre toutes les paires d'éléments des deux schémas est calculé et une matrice de similarité est créée. Au cours de la seconde phase, une seule correspondance est identifiée et sélectionnée comme la meilleure correspondance, à partir de la matrice de similarité produite lors de la première phase. Une fonction soumise à certaines contraintes est nécessaire pour assurer la sélection de la meilleure correspondance.

Le déroulement de l'appariement est comparable à une suite d'étapes ou flux de travail, débutant par l'importation des schémas en entrées, et se terminant par la production d'un alignement en sortie, en passant successivement par l'étape de prétraitement comme la transformation des schémas en graphe, de l'étape de l'exécution de l'apparieur (i.e. l'exécution d'un sous-processus impliquant l'utilisation de différentes techniques et stratégies), suivie de l'étape de la combinaison des résultats produits par les différents apparieurs et enfin de l'étape de la sélection de l'alignement à partir des résultats combinés (Rahm, 2011b).

D'après les auteurs (Bellahsene *et al.*, 2011), la figure 2.5a montre un flux de travail général pour la correspondance automatique de schémas par paires, tel qu'il est utilisé dans de nombreux systèmes de correspondance. Les schémas sont d'abord importés dans un format de traitement interne. Un prétraitement supplémentaire peut être appliqué, tel que l'analyse de fonctionnalités de schéma ou de tokens de nom d'indexation pour se préparer à un calcul plus rapide de similitudes de nom. La partie principale est un sous-flux de travail pour exécuter plusieurs appareurs, chacun déterminant un ensemble préliminaire de correspondances.

Après l'exécution du sous-workflow d'appareurs, il y a généralement plusieurs étapes de post-traitement, en particulier la combinaison des résultats individuels d'appareur et enfin la sélection des correspondances du résultat combiné. Comme indiqué sur la figure 2.5b, les appareurs individuels peuvent être exécutés soit séquentiellement, indépendamment (en parallèle), soit de manière mixte. Dans l'approche séquentielle, les appareurs ne sont pas exécutés indépendamment, mais les résultats des appareurs initiaux sont utilisés comme entrée par les appareurs suivants. Une stratégie commune, par exemple, utilisée dans Cupid (Madhavan *et al.*, 2001), consiste d'abord à exécuter un appareur linguistique pour comparer les noms des éléments du schémas, puis, à utiliser les similitudes obtenues comme entrées pour l'appariement basé sur la structure. Dans la stratégie des appareurs parallèles, les appareurs individuels sont autonomes et peuvent être exécutés indépendamment les uns des autres (Bellahsene *et al.*, 2011). Cela présuppose, une grande flexibilité pour sélectionner les appareurs pour l'exécution et la combinaison. Par ailleurs, ces appareurs peuvent également être physiquement exécutés en parallèle, par exemple, sur un matériel multi-serveur. D'autre part, l'autonomie des appareurs individuels peut créer des calculs redondants, par exemple, des similarités de noms à utiliser pour la correspondance structurelle. La stratégie mixte combine l'exécution séquentielle et parallèle des appareurs et est donc plus complexe (Bellahsene *et al.*, 2011).

### 2.3 Caractéristiques des approches existantes

Il s'agira principalement pour nous, dans cette partie de mettre en lumière, les différents éléments qui, selon nous, font qu'il y a un rapport indissociable entre les approches actuelles et les limitations auxquelles elles font face, comme la réduction de l'incertitude.

Le réductionnisme encore appelé l'anti-holisme, consiste à décomposer les choses afin d'étudier chaque élément indépendamment. Après avoir compris le fonctionnement des divers éléments, il s'agit de les

assembler pour comprendre le système de base étudié comme par exemple la sociologie à partir de la psychologie, l'écologie à partir de la biologie, la thermodynamique à partir de la physique statistique, etc. C'est-à-dire qu'ici toutes les parties sont inférieures à la somme du tout. Autrement dit, le réductionnisme prône la réduction de la complexité des systèmes ou des phénomènes à leurs éléments fondamentaux qui seraient alors plus simples à comprendre et à étudier (Fortin, 2005). Selon cette approche, pour comprendre l'homme, il faudrait suivre une démarche qui ressemblerait à :

- Étudier comment réagissent et se comportent les particules élémentaires constituant les atomes et leurs interactions : constituants et fonctionnement de toute la matière (physique nucléaire).
- Comprendre les différents constituants chimiques composés d'atomes : les molécules, enzymes, ADN, etc.
- Se pencher sur cellules qui composent notre corps, en vue d'étudier leur fonctionnement (biologie cellulaire).
- Savoir des divers organes et avoir une connaissance sur leurs fonctions car ils utilisent toutes les entités étudiées précédemment, spécifiquement le cerveau. On s'intéressera également aux os, à la peau, aux muscles, etc. (médecine, neurologie, podologie, etc.) (Fortin, 2005).

Cette approche réductionniste, malgré sa grande efficacité et ses preuves dans beaucoup de domaines, montre cependant des limites pour certains sujets. L'une de ses limites correspond à la complexité des équations à résoudre pour retrouver les caractéristiques macroscopiques de certains systèmes. Ainsi l'auteur, fort des succès impressionnants de la toute récente physique quantique, Dirac pouvait écrire en 1929 :

les lois physiques fondamentales nécessaires pour la théorie mathématique de la plus grande partie de la physique et de la totalité de la chimie sont ainsi complètement connues, et la difficulté est seulement que l'application de ces lois conduit à des équations beaucoup trop compliquées pour qu'on puisse les résoudre. (Dirac, 1929).

Par ailleurs, une abstraction de la réalité à une linéarisation de simples relations de causes et d'effets entre des constituants fondamentaux d'un système complexe, apparaît, pour l'explication de certains phénomènes ou la résolution de certains problèmes, hautement simplificatrice. On peut par exemple, dénoncer comme simpliste, la prétention de décrire la vie uniquement en disséquant les objets inanimés qui participent à son mouvement, sans nécessairement céder au vitalisme. Le gène, la molécule, l'atome ou l'ion appartiennent indiscutablement à la vie, participent de la vie, mais leur connaissance, aussi élaborée soit-

elle, n'est pas celle de la vie. La vie continue de bénéficier d'une sorte d'« extra-territorialité », comme dit François Dagognet.

La pensée réductionniste est à l'origine des caractéristiques qui sont, à notre avis, à la racine des causes qui empêchent l'appariement automatique de schémas de faire face complètement aux défis et difficultés.

On en veut pour preuve le processus d'appariement schématisé sur la figure 2.6, où on peut voir que toutes les approches actuelles suivent la pensée réductionniste. Elles ramènent le processus d'appariement de schémas, à une fonction linéaire avec un ensemble d'intrants qui seront par exemple les ensembles de données et les algorithmes utilisés et d'extrants qui représentent les résultats de l'alignement. Cette fonction peut se décomposer en une série de modules, dont chacun est responsable du déroulement d'une étape du processus, comme la sélection et l'exécution des apparieurs, et se constituant d'éléments plus ou moins simples (e.g. apparieurs, fonctions et paramètres).

Les essais de résolution des difficultés rencontrées par l'appariement automatique de schémas, comme l'incertitude, suivent le même schéma de pensée, à savoir la focalisation sur un aspect bien précis du fonctionnement du processus et ensuite la tentative d'optimisation de ce module ou de la composante responsable de cet aspect. (Assoudi, 2015)



Figure 2.6 Processus d'appariement de schémas (Martono et Azhari, 2017)

Au vu de tout ce qui précède, l'hypothèse que nous défendons dans ce mémoire est que nous pensons que la problématique de l'appariement automatique de schémas mériterait qu'elle soit abordée dans son entièreté et vue comme un système complexe comprenant plusieurs composantes. La simulation des interactions non-linéaires de ces composantes permettrait l'émergence du résultat final, ce qui représente

une propriété d'une approche holistique et systémique qui, par opposition à l'approche réductionniste, va s'intéresser au tout et à ses qualités émergentes.

Le "holisme" (qui vient du grec « holos » signifiant « entier ») (Wikipédia, 2022) est une théorie d'après laquelle, avant de connaître une chose, quelle qu'elle soit, il faudrait apprendre à connaître la totalité de son environnement. Autrement dit, c'est le « tout » qui définit les objets se situant à l'intérieur par les interactions que tous ces objets exercent entre eux. On peut citer une phrase pour résumer : « Le tout est plus que la somme de ses parties » (Principe d'émergence) selon Aristote (Delahaye, 2017). L'holisme est vraiment utilisé en sociologie ou pour des théories de l'évolution. En effet, les choses comme les hommes, les animaux, la matière, etc., dans leur ensemble représentent un grand intérêt pour un holiste qui va en faire son domaine d'étude, et ne va pas les diviser pour les analyser indépendamment de manière cartésienne. C'est ce qu'est une analyse systémique. La démarche cause/conséquence n'est pas adaptée pour ce genre de réflexion voilà pourquoi l'on parle d'approche non cartésienne systémique. Dans ce genre d'analyse on définit ce qu'on appelle « univers » en physique, tel que le « tout », qui traduit un système possédant une forme avec des limites à l'intérieur, desquelles une logique est présente et où les entités auxquelles on s'intéresse font des échanges. (« Holisme Vs Réductionnisme », 2006)

Les caractéristiques profondes et communes, à tous les systèmes actuels d'appariement, pouvant partiellement expliquer leur incapacité à surmonter la limitation de la complexité de l'appariement et ainsi de répondre à d'autres défis tel que l'incertitude, peuvent être déclinées comme suit: ces systèmes sont compliqués et non pas complexes, linéaires (analytiques, déterministes et prévisibles) et non pas non-linéaires, centralisés et non pas décentralisés (parallélisme et résolution par émergence), et enfin, configurables et non pas adaptables (auto-configuration et auto-optimisation).(Assoudi, 2017)

Somme toute, au vu des éléments cités plus haut, nous proposons donc un changement de paradigme en ayant recours à la théorie des Systèmes Complexes Adaptatifs (SCA), issue de la pensée systémique, pour proposer des réponses aux défis auxquels les approches réductionnistes classiques d'appariement automatique, font toujours face. L'hypothèse centrale est donc de voir le processus de l'appariement comme un SCA et de le modéliser en se servant de l'approche de la modélisation et de la simulation à base d'agents (Assoudi, 2017). Nous voulons, par l'exploitation des propriétés intrinsèques aux SCA comme les modèles à base d'agents (notamment l'émergence, la stochasticité et l'auto-organisation), apporter des réponses aux questions toujours ouvertes, à savoir mieux gérer l'incertitude et la complexité.

## 2.4 Systèmes Complexes Adaptatifs

Dès 1997, Kevin Dooley définissait le Système Complexe Adaptatif (SCA) comme un groupe d'agents semi-autonomes qui interagissent de manière interdépendante pour produire des modèles à l'échelle du système, de sorte que ces modèles influencent ensuite le comportement des agents. Dans les systèmes humains, à toutes les échelles, on voit des modèles qui émergent des interactions des agents dans ce système. Par exemple, les pensées, les expériences ou les perceptions interagissent pour créer des nouvelles modes de pensée. Les attitudes partagées à l'égard de l'alimentation, de l'exercice et de la conscience physique interagissent pour façonner les modèles de santé au sein de l'équipe ou de la communauté. Dans une organisation ou une entreprise, les individus jouent leurs rôles, leurs relations et leurs attentes, pour générer des modèles de concurrence ou d'innovation. Dans les organisations et les communautés, l'histoire, les traditions et les attentes influencent tous le comportement pour façonner les modèles dominants que nous considérons comme la culture de ce groupe. (Human Systems Dynamics Institute, 2016).

Les SCA sont des systèmes conscients, auto-organisés qui sont capables d'apprendre et de s'adapter aux changements de leur situation afin de survivre. Ils comprennent les systèmes biologiques, les écosystèmes, les systèmes sociaux, les sociétés humaines et d'insectes, les virus et le cerveau humain (McMillan, 2008).

Ils sont complexes parce qu'ils sont composés d'un grand nombre d'agents individuels, indépendants et interdépendants. Ceux-ci interagissent les uns avec les autres et avec leurs environnements créant des systèmes d'une grande complexité tel qu'il est impossible de connaître, de prédire ou de mesurer avec précision toutes les nombreuses interactions qui ont lieu sur une période de temps donnée. Ils sont non linéaires et auto-organisés car ils ne sont contrôlés d'aucune manière hiérarchique, mais s'autorégulent et répondent spontanément aux événements qui les entourent, selon un ensemble de principes sous-jacents. Une volée d'oiseaux, par exemple, s'organise lorsqu'elle prend son envol afin d'éviter un prédateur. Aucun oiseau ne donne l'instruction ni ne guide la direction ou la vitesse de réglage de la volée. Les oiseaux sont guidés par des lois ou des principes sous-jacents, qui ont évolué au fil des millénaires dans le cadre du processus de survie de l'espèce. (Carlisle et McMillan, 2017).

Aussi, le cerveau humain est un bon exemple de la façon dont un SCA est capable d'apprendre, de s'adapter, d'utiliser le feedback et de créer ses propres modèles mentaux. On peut penser à la façon dont ses milliards de neurones réagissent spontanément et répondent à tout ce qui se passe dans notre tête

ainsi qu'à tout ce qui se passe dans le monde extérieur. Il interagit avec tout ce qu'il rencontre afin qu'on développe de nouvelles compétences et techniques simples (apprentissage en boucle unique), et en même temps, que l'on révise nos modèles mentaux afin que de nouveaux comportements significatifs émergent (apprentissage en double boucle), (Carlisle et McMillan, 2017).

Les systèmes complexes adaptatifs trouvent une de leurs multiples racines dans la pensée systémique (théorie des systèmes). La systémique prône une approche holistique, contrairement à une approche réductionniste, pour la compréhension d'un système. En effet, pour la systémique, le tout est plus que la somme des parties. Cette notion de totalité implique comme démarche, pour la compréhension du fonctionnement d'un système, l'étude non seulement de l'ensemble des composants du système, mais aussi des interactions de ces composants entre eux (Bertalanffy *et al.*, 1968).

Il convient avant toute chose de mettre en lumière les principales différences dans le cadre de la comparaison de systèmes. En effet, il y aurait, d'après (Wendell, 2003), deux grandes distinctions à faire lorsqu'il s'agit de comparer les systèmes :

- Adaptatif vs déterministe.
- Complexe vs compliqué.

#### 2.4.1 Système adaptatif vs déterministe

Selon (Wendell, 2003), les systèmes adaptatifs sont constitués d'agents qui ont un lien avec leurs voisins et qui répondent aux changements avec un certain degré de liberté, comme répondre avec des règles simples. Un système avec une telle structure produit des réponses qui ne sont pas déterminées et qui peuvent être hautement non-linéaires. En revanche, ce qui caractérise les systèmes déterministes, c'est que ces systèmes sont déterministes et prévisibles dans la mesure où la relation entre les données en entrée et en sortie est linéaire. Dans sa forme la plus simple, cela signifie que de petites entrées créeront de petites sorties et que de grandes entrées sont nécessaires pour de grandes sorties.

#### 2.4.2 Système complexe vs compliqué

Un système complexe est un système qui est constitué d'un ensemble d'éléments homogènes ou hétérogènes, ayant une interaction entre eux de manière non linéaire (interactions rétroactives). Ceci met



ainsi en œuvre une dynamique permettant à l'ensemble du système de pouvoir exister comme un tout, qui n'est pas identique à la simple somme de ses composants. (Draâ, 2011)

On peut identifier alors deux niveaux :

- un niveau micro, qui s'apparente au niveau des composants, avec des propriétés locales à chacun d'eux ;
- un niveau macro, qui représente l'ensemble du système, avec de nouvelles propriétés, que l'on ne retrouve dans aucun des composants pris de manière individuelle.

On parle alors d'émergence de nouvelles propriétés. Les émergences induites par les interactions rétroactives sont dues au caractère complexe d'un système. Ces interactions, dans la dynamique du système, favorisent la mise en œuvre des mécanismes de son adaptation et de son évolution par rapport aux évolutions de son environnement. Il sera question alors de système complexe adaptatif.

Aussi, dans les systèmes complexes, les relations entre les agents (i.e., éléments du système) sont plus importantes que les agents eux-mêmes. Ensuite, les règles simples du système complexe produisent des réponses complexes et adaptatives. Enfin, dans les systèmes complexes, les agents ont la latitude de répondre selon les limites des règles. Un exemple de système complexe peut être la fourmilière d'une colonie de fourmis, dans laquelle les fourmis sont reliées entre elles, agissent selon des règles simples (avec une certaine latitude) et où l'existence des fourmis individuellement a moins d'importance par rapport au tout que représente la colonie (la somme de toutes les relations des fourmis de la colonie).

Par opposition, dans les systèmes compliqués, les éléments et leurs relations ont une importance équivalente. Par ailleurs, avec les systèmes compliqués, les résultats des algorithmes simples se trouvent être des réponses simples et prédictibles. Il est à noter aussi qu'avec les systèmes compliqués la réponse des composants est complètement déterminée. Par exemple, un Airbus serait plutôt compliqué. En effet, un système compliqué est composé d'un grand nombre de sous-systèmes, eux-mêmes composés de sous-systèmes et ainsi de suite. C'est un système où chacun des sous-systèmes est plus simple que l'ensemble auquel il participe (c'est le cas pour un avion, on termine avec les boulons et des composants élémentaires).

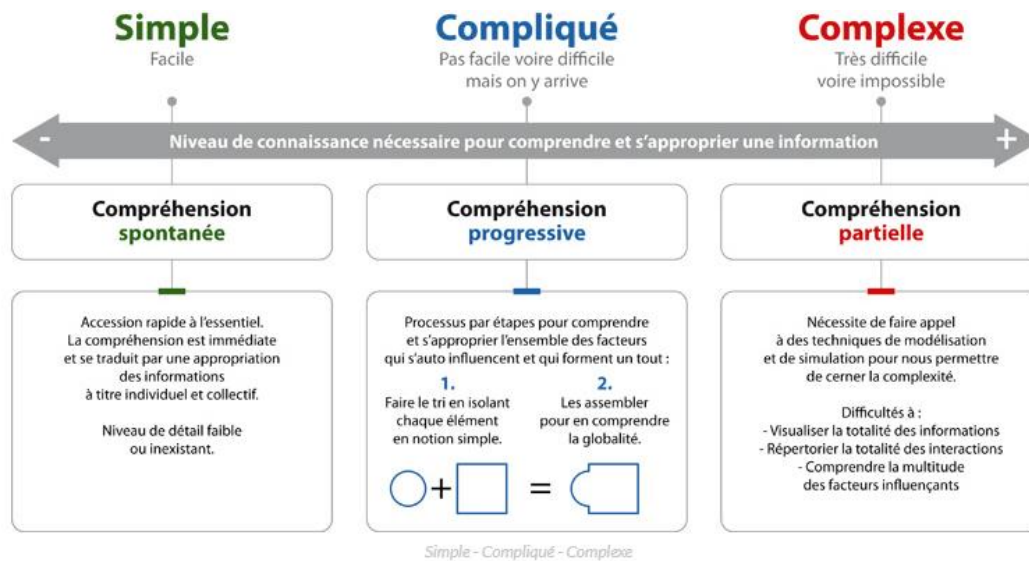


Figure 2.7 Comparaison entre système simple, compliqué et complexe <sup>7</sup>

Sur la figure 2.7, on se rend compte de manière plus succincte, qu'on utilise le terme "compliqué" quand nous sommes capables de comprendre l'incidence de chaque facteur sur lui-même et sur les autres. On utilise le mot "complexe" quand nous ne sommes pas capables de voir et de comprendre l'incidence d'un facteur sur lui-même ou sur les autres. Le terme "simple" est appliqué lorsque la compréhension de l'incidence est plus spontanée.

Les systèmes linéaires et déterminés compliqués donnent des résultats contrôlables et prévisibles. Les systèmes adaptatifs complexes peuvent produire des résultats innovants, créatifs et émergents.

La notion d'émergence est aussi importante dans le domaine de la complexité. Un phénomène est dit émergent si la mise en œuvre des outils analytiques traditionnels ne permet pas d'expliquer le comportement du système. Dans de telle situation, le tout se comporte d'une manière inexplicable par l'étude des parties (ou agents). Certaines propriétés et caractéristiques du système ne peuvent pas être expliquées par une combinaison de comportements de composants.

<sup>7</sup> <https://www.simplixi.fr/difference-complexe-et-complique/>

Il serait important de relever, à ce stade, qu'il n'y a pas une définition formelle pour définir un système complexe adaptatif, malgré la pléthorique des définitions intéressantes que l'on peut trouver dans la littérature à ce sujet. Les auteurs (Bourguine *et al.*, 2008), dans leur livre, abordent, à ce propos, la structure des systèmes complexes adaptatifs, comme : « Structurés sur plusieurs niveaux d'organisation, composés d'entités hétérogènes elles-mêmes complexes ». Sur la dynamique de ces systèmes, les auteurs stipulent que : « Les systèmes complexes, depuis les objets nanoscopiques de la physique et de la biologie jusqu'à l'écosphère, résultent de processus d'émergence et d'évolution : les interactions individuelles engendrent des comportements collectifs qui peuvent manifester des structures organisées. Ces structures émergentes influencent en retour les comportements individuels. Les causes sont multiples et la causalité fonctionne à la fois de façon ascendante et descendante entre les niveaux d'organisation».

Étant donné qu'il n'existe pas, dans la littérature, de définition formelle pour les systèmes complexes adaptatifs, on ne trouve pas une liste unifiée recensant toutes les caractéristiques de ces systèmes. Cela étant, il existe cependant, un quasi-consensus sur certaines caractéristiques, considérées communes aux systèmes complexes adaptatifs.

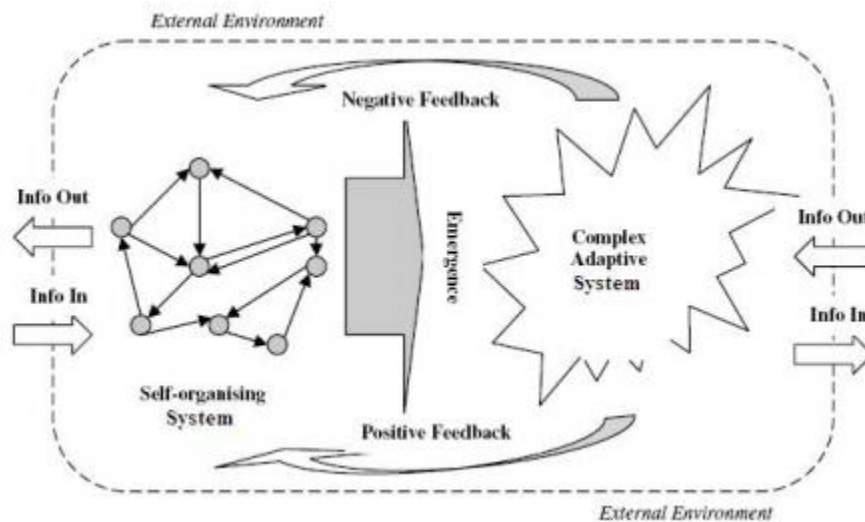


Figure 2.8 Systèmes Complexes Adaptatifs (Andrus, 2005)

La figure 2.8 (Andrus, 2005), met en relief la caractéristique de la rétroaction (négative ou positive) qui peut avoir une influence certaine sur le comportement global du système. En plus de cette caractéristique, on distingue certaines à savoir: la non-linéarité, l'adaptation, l'auto-organisation et l'émergence.

Dans (Rolland, 2006), on évoque d'autres caractéristiques telles que:

- le parallélisme, qui se traduit par le fait que les systèmes complexes adaptatifs se composent d'un grand nombre d'agents qui interagissent simultanément par l'envoi et la réception de signaux.
- l'action conditionnelle, qui relève du fait que les actions des agents dans un système complexe adaptatif dépendent généralement des signaux qu'ils reçoivent. Autrement dit, les agents ont une structure contenant une série de règles (SI signal x ALORS exécuter acte y). L'action peut elle-même déclencher un signal, qui peut être envoyé d'une manière ouverte dans l'environnement de l'agent, permettant des rétroactions assez complexes.
- la modularité, qui voudrait dire qu'au niveau de l'agent, des groupes de règles se combinent souvent pour agir en tant que "sous-programmes". Pour exemple, l'agent peut réagir à la situation actuelle en exécutant une séquence de règles. Ces "sous-programmes" agissent comme des blocs de construction qui peuvent être combinés pour gérer des situations nouvelles, plutôt que d'essayer d'anticiper chaque situation possible avec une règle distincte.
- L'adaptation et l'évolution, expliquées par le fait que les agents d'un système complexe adaptatif changent dans le temps. Ces changements sont généralement des adaptations qui améliorent la performance, plutôt que des variations aléatoires.

## 2.5 Conclusion

Dans la première partie de ce chapitre nous avons montré pourquoi l'appariement des CV aux OE peut être vu comme un appariement de schémas. Aussi, avons-nous exposé les différentes techniques, méthodes et approches existantes pour le défi que représente cette tâche qu'est l'appariement automatique de schémas. Ensuite, nous avons analysé les principales difficultés, inhérentes à l'automatisation totale de ce processus, telle que l'incertitude. Par ailleurs, nous proposons des solutions proposées qui selon nous, pourraient résoudre ces problèmes, comme l'optimisation. La première partie s'achève par une synthèse mettant en relief les principales critiques que nous formulons à l'encontre des approches existantes ainsi que les limites qui, selon nous, empêchent d'apporter des réponses holistiques aux questions qui demeurent encore ouvertes. Ensuite dans la seconde partie, nous présentons le cadre

théorique de notre nouvelle approche d'appariement automatique de schémas à savoir la théorie de la complexité, et de manière plus spécifique le paradigme des systèmes complexes adaptatifs. Aussi, avons-nous exploré les racines de ce paradigme et présenté ses caractéristiques. Dans le prochain chapitre, nous allons aborder l'appariement automatique des schémas comme un système complexe adaptatif. Il s'agira de présenter le cadre théorique ainsi que notre cheminement méthodologique pour traiter notre problématique.

## CHAPITRE 3

### APPARIEMENT DE SCHÉMAS COMME UN SYSTÈME COMPLEXE ADAPTATIF

#### 3.1 Introduction

Dans la première partie de ce chapitre, nous présentons de façon détaillée le cadre théorique dans lequel on se place pour aborder la problématique de l'appariement automatique, ainsi que notre cheminement méthodologique. Pour ce faire, dans un premier temps, on va par rapport à la théorie des systèmes complexes adaptatifs recentrer la problématique de l'appariement automatique de schémas, de sorte à clarifier et fixer certains concepts de notre problématique de recherche au cadre théorique et vice versa. Dans un second temps, on discute de notre cheminement méthodologique qui se base particulièrement sur l'approche de modélisation et de la simulation à base d'agents<sup>8</sup>.

Dans la deuxième partie, on propose une formalisation du modèle conceptuel de notre approche (qui s'inspire de l'appariement automatique des schémas) de simulation multi-agents pour l'appariement automatique de curriculum vitae aux offres d'emploi. Aussi, on discutera brièvement de notre volonté de transcrire notre modèle conceptuel en un programme informatique s'exécutant sur une plateforme de simulation.

Une synthèse des points cruciaux abordés tout au long de ce chapitre fera office de conclusion pour la fin du chapitre.

#### 3.2 Modélisation et simulation à base d'agents

La montée en puissance et la disponibilité croissante des ordinateurs personnels ont permis à des chercheurs de toutes disciplines de disposer de ressources de calcul jusque-là l'apanage des seuls calculateurs. L'émergence progressive de ce que l'on peut appeler, faute de mieux, l'« approche systèmes complexes » (Waldrop, 1993), (Dodder et Dare, 2000) dans les sciences de l'Homme, de la matière et de la nature ont contribué au succès des pratiques de modélisation et de simulation. Elles possèdent en commun le fait de représenter de façon explicite, des entités et leurs comportements, sous une forme

---

<sup>8</sup> De l'anglais « multi-agent simulation (MAS) »

informatique et d'étudier, grâce à la puissance qu'offrent les ordinateurs actuels, les structures émergentes qui naissent, *in silico*, de leurs interactions virtuelles.

Les techniques de modélisation et de simulation à base d'agents, donnent des outils d'exploration de la relation entre niveaux de description d'un système. Les techniques sont plus souples et versatiles que les rares méthodes mathématiques développées dans le passé (par exemple, en physique ou en économie). Conséquence de la propagation des « idées complexes », elles en sont aussi le vecteur, ce qui explique sans doute leur adoption croissante dans bon nombre de disciplines.

Avant d'aborder la modélisation proprement dit, il est important de nous attarder sur la raison pour laquelle nous pensons que l'appariement automatique des schémas, qui est l'approche sur laquelle se base notre solution, peut se conceptualiser comme un système complexe adaptatif. Autrement dit, l'appariement des CV aux OE peut se conceptualiser comme un système complexe adaptatif et ainsi tirer profit des caractéristiques fondamentales de cette théorie, ainsi que de son approche de modélisation. Le but d'apporter des réponses novatrices aux questions, comme l'incertitude, encore ouvertes dans ce domaine.

Notre vision de la problématique de l'appariement des CV aux OE selon la théorie des systèmes complexes adaptatifs, s'appuie sur trois points de vue à savoir:

- Point de vue d'un système: En effet, l'appariement automatique de schémas peut être vu comme un système car étant un ensemble cohérent, constitué de plusieurs composantes, liées entre elles, en l'occurrence les schémas, les apparieurs, etc. Aussi, si l'on se tient strictement à la définition du terme système, du latin « *systema* », qui désigne « un ensemble cohérent de parties étroitement liées », il est clair que l'appariement automatique de schémas est un système. Par ailleurs, si on veut se référer à la définition de la systémique selon (Bertalanffy *et al.*, 1968), l'appariement automatique de schémas est un système dont les différentes composantes, par exemple les éléments de schémas, ou encore les apparieurs, doivent rentrer en interaction au sein de ce système avec l'objectif de produire un résultat final qui sera l'alignement.
- Point de vue de la complexité: L'appariement automatique de schémas doit être pensé et modélisé comme un système complexe car d'après (Wendell, 2003), dans les systèmes complexes, les relations entre les agents qui représentent les éléments du système sont plus importantes que les agents eux-mêmes, contrairement aux systèmes compliqués où les éléments et leurs relations ont

une importance équivalente. Aussi, dans les systèmes complexes, des règles simples peuvent produire des réponses surprenantes et complexes à la fois, contrairement aux systèmes compliqués, où la résultante des algorithmes simples n'est autre que des réponses simples et prédictibles. En outre, dans les systèmes complexes, les agents ont la latitude de répondre selon les limites des règles, par opposition aux systèmes compliqués, où la réponse des composants est complètement déterminée.

- Point de vue de l'adaptabilité: Ce dernier point nous renvoie à la notion d'adaptation des systèmes complexes. Concernant l'appariement automatique de schémas, on avance que grâce à des règles simples, les agents, éléments de schémas, durant la simulation de notre modèle, peuvent montrer un haut niveau d'adaptation face à des situations complexes. Elle peut se manifester à plusieurs niveaux :
  - 1) une adaptation aux perturbations liées à l'augmentation de l'hétérogénéité lexicale qui est causée par l'ajout de plusieurs autres rubriques (auto-configuration), ou encore,
  - 2) une adaptation face aux modifications qui peuvent toucher l'assemblage du système d'appariement comme le déploiement de nouveaux composants, par exemple, l'ajout de nouveaux appariements ou le retrait d'appariements non performants (auto-configuration), et enfin,
  - 3) une adaptation permettant l'exploration dynamique de l'espace des combinaisons possibles durant le processus d'appariement, dans le but de maximiser efficacement les paramètres et stratégies pour le processus d'appariement, et ce, sans pour autant subir le problème de l'explosion combinatoire (auto-optimisation).

Notre vision pour l'appariement automatique de schémas prend en considération l'ensemble des facteurs et paramètres influençant le système de l'appariement de schémas dans son entièreté. On pense que la concrétisation d'une modélisation de l'appariement de schémas comme un système complexe adaptatif, rendrait possible la conception d'un système d'appariement automatique, regroupant simplicité et efficacité, ce qui serait une contribution importante au domaine du recrutement, entre autres.

On imagine que la conceptualisation de l'appariement de schémas sous la perspective des systèmes complexes adaptatifs c'est-à-dire, en exploitant leurs caractéristiques fondamentales, va permettre la conception d'un:



- système qui aura une compréhension facile, de par le fait que le système serait décomposé en plusieurs entités simples « agents » (i.e. deux groupes d'agents représentant les schémas source et cible), interagissant selon des règles simples.
- système efficace et efficient, capable d'une manière autonome de s'adapter et de s'auto-organiser, et ainsi de faire émerger la solution pour n'importe quel scénario d'appariement. Étant donné que ce résultat émergent qu'est l'alignement émane des interactions, des choix et des comportements individuels des agents, cette efficacité voulue, doit se traduire principalement par la réduction de l'incertitude sur l'alignement. L'efficience, quant à elle, doit se manifester par la réduction de l'effort qu'il faut à cette efficacité, notamment la réduction de l'effort utilisateur nécessaire à la configuration ou à l'optimisation du système d'appariement, et cela sans influencer négativement le temps de réponse qui devrait rester dans les limites du viable et de l'acceptable.

Étant donné que l'appariement automatique des schémas de façon générale peut être conceptualisé sous la perspective des systèmes complexes adaptatifs, l'appariement automatique plus spécifiquement des CV aux OE peut aussi être conceptualisé sous la même perspective.

Les systèmes complexes adaptatifs (SCA), par ailleurs, sont composés d'agents autonomes en interaction (Holland, 2006). Les agents des SCA, ont des comportements et des propriétés. Ils interagissent et s'influencent mutuellement, apprennent de leurs expériences et adaptent leurs comportements pour être mieux adaptés à leur(s) environnement(s). En modélisant ces agents individuellement, les effets résultants de la diversité qui existe entre les agents en ce qui concerne leurs attributs et leurs comportements peuvent être observés car ils donnent lieu au comportement dynamique du système dans son ensemble.

Par ailleurs, selon (Bonabeau, 2002) et (Macal et North, 2010), la modélisation à base d'agents fournit un mécanisme de modélisation des systèmes complexes adaptatifs. La modélisation à base d'agents a été utilisée avec succès pour modéliser des systèmes complexes adaptatifs dans de nombreux domaines, notamment l'archéologie, la biologie, l'écologie, les chaînes d'approvisionnement, l'analyse des marchés de consommation, la planification militaire et l'économie (Macal et North, 2007), (Macal et North, 2009).

On trouve dans la littérature des termes similaires pour désigner l'approche qui consiste à modéliser et à simuler un système complexe : modélisation et simulation à base d'agents<sup>9</sup> (ABMS), simulation multi-agents<sup>10</sup> (MAS), modélisation individus-centrée (IBM), modélisation à base d'agents (ABM) ou simulation à base d'agents (ABS) (Siegfried, 2014).

La modélisation et simulation à base d'agents est un nouveau paradigme de modélisation et un des plus excitants dans la modélisation depuis l'invention des bases de données relationnelles (North et Macal, 2007). Bien que ce nouveau paradigme de modélisation soit lié à plusieurs domaines de recherche, notamment la théorie de la complexité, la théorie des systèmes, la dynamique des systèmes, l'informatique, la science de la gestion et les sciences sociales, il est cependant particulièrement lié aux approches de modélisation et de simulation classiques. Il s'appuie sur ces domaines non seulement pour ses fondements théoriques, sa vision conceptuelle et sa philosophie, mais aussi pour ses techniques de modélisation (Macal et North, 2009).

La modélisation à base d'agents et une modélisation ascendante « bottom-up », le système modélisé étant décrit comme une collection d'agents et sa dynamique globale comme le produit de ces interactions (Bonabeau, 2002).

### 3.2.1 La Simulation

C'est avec l'apparition et la disponibilité de systèmes informatiques de plus en plus puissants que la notion de « simulation » est devenue partie intégrante de la panoplie de nombreuses disciplines scientifiques, à tel point que le terme même de simulation ne désigne plus maintenant que des simulations informatiques et ce malgré le fait que la notion de « simulation » ait été utilisée en sciences avant que les premiers ordinateurs n'apparaissent (Siegfried, 2014).

Si nous nous référons à la littérature contemporaine sur le sujet (voir par exemple (Simpson, 2006)), on remarque dans les nombreuses définitions proposées pour le terme de simulation, la présence d'au moins trois grandes acceptions dont la plus pertinente pour nous est celle de (Shannon, 1998) qui considère que la simulation est à un modèle dynamique ce que l'expérimentation est à un système réel. Dans cette

---

<sup>9</sup> De l'anglais « ABMS : Agent Based Modelisation and Simulation »

<sup>10</sup> De l'anglais « MAS : Multi-Agent Simulation »

perspective, où le modèle représente une certaine forme de compréhension d'un système de référence, la simulation est un ensemble de tests grâce auquel il est possible pour un scientifique de raffiner cette compréhension et d'en tirer des connaissances nouvelles. Nous l'avons donc retenue car elle seule, donne à la simulation l'image d'une activité spécifique avec ses règles, ses méthodes, et surtout un rôle dans l'exploration scientifique. Elle nous paraît de plus, clairement représentative des pratiques actuelles de simulation.

En voulant donner une définition de la simulation, il est nécessaire de définir dans un premier temps une expérimentation, qui est une activité par laquelle, en fonction d'objectifs précis, et à l'aide d'un dispositif expérimental, on perturbe selon un protocole déterminé un système réel pour en comprendre le fonctionnement. Aussi, il est important de préciser que seuls les modèles dynamiques sont susceptibles d'être soumis à ce processus. En effet, un modèle statique, par définition, ne peut pas être perturbé ni soumis à une quelconque évolution. Après ces mises au point, une simulation peut être considérée comme une activité par laquelle, en fonction d'objectifs précis, et à l'aide d'un dispositif expérimental informatique (appelé simulateur), on perturbe selon un protocole déterminé un modèle dynamique.

Les objectifs d'une simulation peuvent, selon (Simpson, 2006), être subdivisés en cinq catégories comme suit:

La simulation peut être utilisée à des fins de ...	
Validation, Évaluation, Vérification	La simulation a comme objectif de tester une hypothèse du modèle du système de référence, de le vérifier ou d'accréditer la théorie qui a servi à le construire.
Communication, Formation, Visualisation	La simulation a comme objectif de « montrer » et de partager le modèle de la dynamique du système de référence.
Compréhension, Exploration, Explication	La simulation sert à comprendre le fonctionnement du système de référence, en considérant le modèle comme une réplique miniature qui pourra être étudiée plus facilement.
Contrôle, Action, Pilotage	La simulation a comme objectif de servir de support à une prise de décision ou à un contrôle qui influera sur l'état (réel) du système de référence.
Prévision, Prédiction, Anticipation	La simulation sert à prévoir les évolutions possibles du système de référence en fonction d'évolutions ou de perturbations spécifiques.

Figure 3.1 Objectifs d'une simulation (Treuil *et al.*, 2008)

Ces différents usages et objectifs assignés à la simulation ne sont évidemment pas sans conséquences sur la façon dont sont constitués les modèles. Ils vont par exemple influencer sur le choix des hypothèses de simplification, sur le niveau de représentation du système de référence, sur le degré de fidélité que le modèle devra posséder à son égard, ou sur la façon dont il pourra être validé.

La simulation est considérée aujourd'hui, avec la théorie et l'expérimentation, comme le troisième pilier de la science. L'analyse de nombreux systèmes, processus et phénomènes est souvent faisable en développant et en exécutant des modèles de simulation (Siegfried, 2014), (Minar *et al.*, 1996).

En science et spécialement dans l'étude des systèmes complexes, les programmes informatiques jouent un rôle important comme équipement scientifique (Minar *et al.*, 1996). Les simulations informatiques-logiciels comme outils expérimentaux ont pris leurs places à côté des outils expérimentaux physiques. Les modèles informatiques offrent plusieurs avantages, par rapport aux méthodes expérimentales traditionnelles. Ils ont aussi plusieurs désavantages, en particulier la complexité de la tâche de programmation du logiciel.

En règle générale, pour faire exécuter des simulations, on doit avoir recours à des simulateurs. Un simulateur est un programme (ou plateforme) informatique capable d'interpréter des modèles dynamiques. Il est utilisé pour produire les perturbations désirées sur ces modèles (Treuil *et al.*, 2008).

Pour résumer donc, selon un protocole et avec un objectif précis, la simulation est l'activité au cours de laquelle, on utilise un simulateur afin de faire évoluer les entrées d'un modèle dynamique, l'exécuter, et en recueillir les sorties. Tout modèle écrit en respectant le méta-modèle associé au simulateur et comportant au moins un paramètre d'entrée peut se soumettre à ce processus de simulation. (Treuil *et al.*, 2008) (Assoudi, 2017).

### 3.2.2 La Modélisation

La modélisation, selon (Thomson *et al.*, 2010) est l'activité qui permet de construire des modèles. Avec l'expérimentation, elles constituent les deux principales composantes, de la démarche scientifique. Les chercheurs passent en effet une bonne partie de leur temps à construire, essayer, comparer et réviser des modèles, et un nombre important de publications sont dédiées à l'application, à l'interprétation et à la présentation de modèles. (Treuil *et al.*, 2008).

Il n'y a pas de définitions conventionnelles, universelles d'un modèle scientifique, car en dépit du fait que tous les scientifiques en manient une vision intuitive, il en existe un nombre considérable d'interprétations selon la discipline, les objectifs et le point de vue épistémologique invoqué (Sarkar et Pfeifer, 2006).

Puisque le cadre de notre travail ne concerne pas les maquettes, les modèles réduits ou les modèles animaux, qui sont des dispositifs du monde réel conçus pour être soumis à expérimentation, on va plus se concentrer sur les modèles conçus pour être simulés sur ordinateur. Nous proposons, après avoir fait cette distinction, une définition consensuelle, vis-à-vis de la littérature existante, de la notion de modèle scientifique. Cette dernière, qui est une construction abstraite permet de comprendre le fonctionnement d'un système de référence en apportant une réponse à une question qui le concerne. Autrement dit, c'est une représentation simplifiée abstraite d'un système de référence. (Sarkar et Pfeifer, 2006).

Un modèle n'est donc pas le système qu'il décrit, mais une simplification délibérée de ce dernier, plus facile à manipuler que le système lui-même. Elle est fondée sur des hypothèses qui ressortent, soit du domaine scientifique concerné, soit d'un choix assumé de limiter le niveau de complexité à prendre en compte. Cette simplification est exprimée dans un langage de modélisation dont le choix dépend d'une part des besoins du domaine scientifique pris en compte, d'autre part de considérations sur les propriétés souhaitées pour le modèle (généralisation, démonstration analytique, traduction en programme informatique, communication à une audience spécifique, etc.), (Treuil *et al.*, 2008).

Par ailleurs, selon l'auteur (Glennan, 2005), le défi est de comprendre la relation entre théories et modèles comme une relation entre le général et le particulier: là où une théorie a pour ambition de décrire le fonctionnement et l'évolution d'un ensemble de systèmes de référence, le modèle est une instanciation de cette théorie pour un système en particulier. (Treuil *et al.*, 2008)

Un modèle à base d'agents, selon les auteurs (Macal et North, 2010), présente principalement trois éléments:

- Les agents : un ensemble d'agents exhibant des attributs et des comportements.
- Les relations : un ensemble de relations entre les agents basés sur des méthodes d'interactions. La topologie sous-jacente définit comment et avec qui l'agent interagit.
- L'environnement : les agents interagissent dans le contexte d'un environnement.

(Russell et Norvig, 2010) classent la structure d'un agent en cinq types de base, par ordre de généralité ascendante :

- Agent réflexif simple : c'est le type d'agent le plus simple. Il se base sur des règles simples (i.e. Si condition Alors action). Il n'a pas de mémoire et ne réagit donc pas, par rapport à ses états ou expériences passées. (Assoudi, 2017)
- Agent réflexif avec état interne : ce type d'agent ressemble au précédent dans la mesure où les deux se basent sur des règles simples type (i.e. Si condition Alors action). En revanche, ce type a besoin de maintenir un modèle décrivant son état actuel. Ce modèle va être à la base de la prise de décision. (Assoudi, 2017)
- Agent basé sur les buts: ce type se base sur les buts (objectifs) pour prendre des décisions sur ses actions. En effet, il ne s'appuie pas sur des règles simples, contrairement aux deux types précédents, pour décider de l'action à prendre, mais engage des tâches d'exploration et de planification pour la recherche et la sélection d'actions lui permettant d'atteindre ses buts. (Assoudi, 2017)
- Agent basé sur l'utilité : ce type est considéré comme étant le type le plus évolué dans la mesure où il raisonne sur les décisions concernant ses actions par rapport à leur utilité. En effet, l'objectif pour ce type d'agent est d'explorer et de planifier ses actions, tout comme le fait le type précédent, mais en choisissant l'action qui va mener l'agent à un état avec la meilleure utilité (i.e. mesure de performance permettant le choix précis entre les états, selon la satisfaction de l'agent). (Assoudi, 2017)
- Agent apprenant : ce type est capable d'opérer, avec des connaissances limitées, dans des environnements inconnus et devenir ensuite, au fil de ses expériences et de son apprentissage, capable de prendre des décisions lui permettant au final d'atteindre ses objectifs. Il est à noter que l'apprentissage peut être considéré comme une faculté supplémentaire que l'agent peut avoir, quel que soit son type. (Assoudi, 2017)

D'après les auteurs (Assoudi et Lounis, 2014) (Chaib-draa et Dignum, 2002), les agents communiquent selon deux modes de communication :

- Communication directe: ce type de communication correspond à un échange de messages, généralement structuré selon un protocole ou langage de communication entre agents. Ces

langages permettent de représenter des actions ou des actes communicatifs comme informer, proposer, accepter, etc. Parmi ces langages, on trouve le langage FIPA-ACL « FIPA-Agent Communication Language » ou encore le langage KQML « Knowledge Query and Manipulation Language ».

- Communication indirecte: ce type de communication correspond à un échange de messages fait par l'intermédiaire de l'environnement, plus précisément en générant un évènement à la suite d'actions qui effectuent des changements observables dans l'environnement. On aura comme exemple, un agent qui veut une ressource et qui envoie un message de son besoin aux autres agents.

Par ailleurs, dans le contexte d'un environnement émergent et auto-organisé, la communication indirecte qui se matérialise avec la modification des agents et de leur environnement s'appelle la stigmergie. Cette notion est une notion très importante pour les systèmes complexes adaptatifs, d'où la pertinence de préciser sa définition (« Stigmergie », 2020):

la stigmergie est un mécanisme de coordination indirecte entre les agents. Le principe est que la trace laissée dans l'environnement par l'action initiale stimule une action suivante, par le même agent ou un agent différent. De cette façon, les actions successives ont tendance à se renforcer conduisant ainsi à l'émergence spontanée d'activités cohérentes, apparemment systématiques. (« Stigmergie », 2020).

La validation des modèles de simulation à base d'agents est un autre point qui prend beaucoup d'importance dans la littérature. Ainsi, les auteurs (Remondino et Correndo, 2006), distinguent trois types de validation:

- Validation empirique : qui est une validation se basant sur la comparaison des résultats obtenus par le modèle, par rapport à ce que l'on peut observer sur le vrai système.
- Validation prédictive : elle cherche à prouver que les résultats obtenus par le modèle peuvent avoir une validité dans des situations qui ne sont pas encore observables directement dans la réalité. Ce type de validation est souvent utilisé dans l'analyse « what-if ».
- Validation structurelle : ce sera le type de validation que l'on adoptera pour la validation de notre modèle de simulation à base d'agents pour l'appariement automatique de schémas. Cette

validation cherche à savoir comment les résultats sont obtenus. Elle permet de s'assurer que le modèle calque bien la réalité. (Assoudi, 2017)

### 3.3 Modélisation et Simulation à Base d'Agents pour l'appariement de schémas (SMAS)

Tout modèle à base d'agents est un système composé d'entités multiples ou agents qui évoluent dans un environnement. Cet environnement, est conçu comme une entité particulière, dans lequel ils sont localisés. Ces agents sont dotés d'attributs, de comportements, de capacités de communication et de perception. L'ensemble des valeurs des attributs d'une entité à un instant donné constitue l'état de cette entité, et la réunion de l'ensemble des états des entités forme l'état du système. Les capacités de perception des entités leur permettent de consulter un sous-ensemble de façon localisée dans l'environnement (Treuil *et al.*, 2008).

Ci-dessous, les auteurs nous proposent une représentation simplifiée d'un modèle à base d'agents, qui a un certain vocabulaire mais n'impose cependant aucune contrainte particulière sur la structure des entités et des comportements, pas plus qu'elle ne fournit d'indications sur une « agentification » particulière.

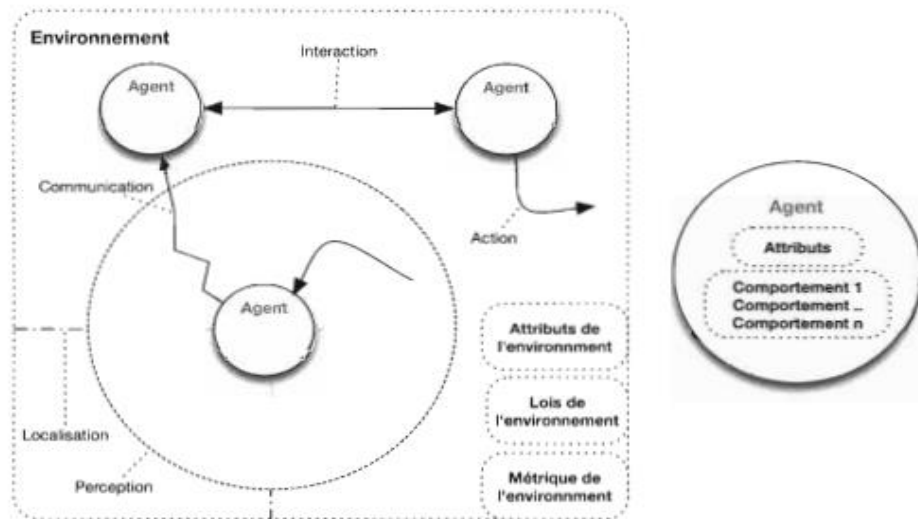


Figure 3.2 Représentation simplifiée d'un modèle à base d'agent (Treuil, Drogoul et al. 2008)

On retrouve de nombreux méta-modèles « agents » dans la littérature, comme: communication entre agents (FIPA 2007 (s. d.)), mécanismes d'exécution des comportements (Wooldridge *et al.*, 2000), formation des groupes d'agents (Ferber *et al.*, 2004), interaction avec l'environnement (Marilleau *et al.*,



2006), etc. Ils restent tous cependant, à de rares exceptions près, indépendants du domaine auquel ils s'appliquent et des outils logiciels qui les implémentent. (Treuil *et al.*, 2008)

L'hypothèse au cœur de ce mémoire est de considérer le processus de correspondance CV aux OE comme un système complexe adaptatif et de le modéliser en utilisant l'approche de la modélisation et de la simulation à base d'agents. Le but recherché est l'exploitation des propriétés intrinsèques aux modèles à base d'agents, notamment l'émergence, la stochasticité et l'auto-organisation, pour contribuer à trouver des réponses aux questions ouvertes, à savoir mieux faire face à la complexité et l'incertitude. (Assoudi, 2017)

Dans notre modèle à base d'agents, chaque élément à appairer (champs d'un CV, rubriques d'une offre d'emploi, etc.) est modélisé comme un agent autonome, faisant partie d'un groupe (groupe source ou cible) dont l'interaction avec son environnement et dont la simulation des comportements, au niveau micro, fait ressortir au niveau macro, un système auto organisé, qui représente les interactions entre les éléments à appairer, ce qui constitue la solution globale à l'appariement.

Autrement dit, la résolution du processus d'appariement passe par l'effort individuel que génère chaque agent d'un groupe (groupe source ou cible), d'une manière autonome et locale durant toute la durée de la simulation, pour trouver la meilleure relation de correspondance dans le groupe opposé (meilleur appariement).

Dans la prochaine section, nous étudions les approches qui permettent une bonne prise de décision en ce qui concerne la sélection des candidats. Nous formulons dans la section suivante, notre vision de la conceptualisation de l'appariement automatique de schémas comme un système complexe adaptatif. Cette conceptualisation consiste en une formalisation d'un modèle de simulation multi-agents, pour l'appariement automatique de schémas, qui soit lié à l'implémentation.

### 3.4 Approche pour la prise de décision

D'après (Cappelli, 2019), les employeurs dépensent une fortune pour le recrutement au sein de leurs entreprises. On en veut pour preuve une moyenne d'environ 4 129\$ par emploi aux États-Unis selon les estimations de la SHRM (Society for Human Resource Management). Un montant qui se veut presque peu élevé si l'on veut le comparer au montant alloué pour l'embauche visant à combler des postes de direction

comme le révèle le chiffre de 66 millions de postes aux États-Unis par année. Toujours d'après les estimations de la SHRM, la grande partie du budget de 20 milliards de dollars que les compagnies allouent au département de ressources humaines est en grande proportion dépensée pour le recrutement.

Afin de pouvoir déterminer, ce sur quoi les recruteurs se basent pour choisir les meilleurs candidats pour des offres d'emploi, les auteurs (Pernilla and Joergen 2013) ont mené une étude approfondie, en temps réel, de la sélection de professionnels de l'informatique (développeurs de systèmes, développeurs d'affaires et consultants en stratégie) dans deux entreprises informatiques de taille moyenne.

Le processus de recrutement commençait dans les deux organisations et de façon générale, par l'examen des candidatures par des spécialistes RH qui pré-sélectionnaient des candidats prometteurs. Les candidats retenus ont ensuite été interviewés une, deux ou trois fois par des sélectionneurs (spécialistes RH, managers et collègues). Ensuite des tests de compétence ont été fait, pour au final aboutir à une décision humaine.(Pernilla and Joergen 2013)

Notre travail, a pour but alors, d'apporter une amélioration considérable au processus en vue d'alléger le travail des spécialistes RH et par conséquent de réduire le faramineux budget que les entreprises lui consacrent. Nous ne voulons pas prétendre remplacer le jugement de l'homme par le prototype expérimental, mais nous voulons renforcer son jugement par des résultats concrets et logiques, ce qui fera un parfait alliage pour une prise de décision optimale. De façon plus précise, notre outil va pré-sélectionner les meilleurs candidats que les spécialistes devront par la suite utiliser pour la suite du processus notamment, pour les interviews et les tests. Pour ce faire, pour un ou plusieurs curriculum donné, l'outil devra déterminer la meilleure correspondance en termes de compétence, expérience et formation, par rapport à une ou plusieurs offres. Inversement, pour une ou plusieurs offres d'emploi, l'outil doit déterminer par rapport aux informations requises, le ou les meilleurs candidats possibles en se basant aussi sur chacune de leurs rubriques. Le but poursuivi est d'avoir, selon la requête du spécialiste, pour une offre d'emploi, un classement des meilleurs candidats, ou pour un candidat, un classement des meilleures offres auxquelles il pourra être éligible.

Ainsi, pour notre scénario d'appariement, on disposera de deux schémas d'entrée à appairer (les CV qui représenteront les schémas sources et les offres d'emplois qui seront les schémas cibles), qui devront produire les alignements attendus c'est-à-dire, l'ensemble des correspondances correctes souhaitées par les spécialistes RH.

L'exemple de référence illustré sur la figure 3.3, nous permet de mieux visualiser les concepts fondamentaux rattachés à la formulation de notre modèle conceptuel.

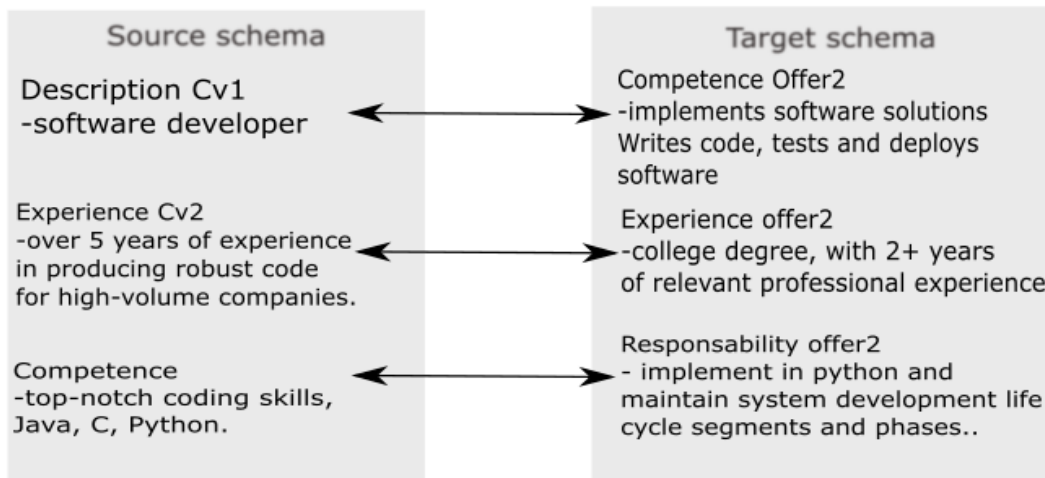


Figure 3.3 Illustration de l'exemple de référence pour la correspondance

### 3.5 Énoncé du modèle conceptuel

Notre approche d'appariement des CV aux OE, compte tenu du changement de paradigme de la conception d'un système d'appariement, se retrouve dans une autre catégorie par rapport aux approches existantes. Ainsi, la différence notable entre elle et les autres approches est que les approches existantes partagent toutes les mêmes traits fondamentaux issus de la pensée réductionniste.

Notre approche pour le modèle conceptuel, par contre, se place dans le rang des approches provenant de la pensée holistique ou systémique, et se voit donc grandement influencée par les fondements théoriques découlant de la théorie des systèmes complexes adaptatifs.

La modélisation et simulation à base d'agents a été pour nous un choix qui s'imposait de toute évidence. En effet, (Macal et North, 2010) avaient mis en relief le lien entre les systèmes complexes adaptatifs et la modélisation à base d'agents en ces termes: « Agent-based modelling is a way to model the dynamics of complex systems and complex adaptive systems.»

L'idée au cœur de notre modèle de simulation multi-agents repose principalement sur l'appariement automatique de schémas. En effet, cette idée voudrait que l'on cesse de voir les schémas (dans notre cas, les CV aux OE) à appairer comme des intrants statiques au système d'appariement, mais plutôt, une fois importés dans le système, comme un ensemble d'agents représentant les éléments de schémas, divisé en deux groupes et où chaque groupe représente un schéma donné (schéma source ou schéma cible). À ce moment, ces éléments de schémas deviennent des agents autonomes qui ont la capacité de se fixer un objectif, dans le cadre d'une simulation, qui est de trouver la meilleure correspondance dans le groupe opposé. Aussi, la simulation de leurs comportements et de leurs interactions avec leur environnement, au niveau micro, fait ressortir au niveau macro, un réseau auto-organisé qui constitue la solution globale de l'appariement. Autrement dit, la résolution du processus d'appariement commence par l'effort individuel que déploie chaque agent d'un groupe, d'une manière autonome et locale, tout au long de la simulation, pour trouver la meilleure relation de correspondance dans le groupe opposé qui sera donc considéré comme le meilleur appariement.(Assoudi 2017)

Nous aurons cinq concepts de base autour desquels s'articulera notre modèle conceptuel de simulation multi-agents pour l'appariement de schémas, à savoir:

- l'agent élément de schéma,
- la stochasticité et l'adaptation,
- l'appariement consensuel,
- l'auto-organisation et l'émergence de l'alignement, et enfin
- les simulations multiples et l'analyse statistique (quantification de l'incertitude).

### 3.5.1 Agent élément de schémas

Une des parties importantes dans notre travail de conceptualisation de l'appariement automatique des CV aux OE en un système complexe adaptatif, est sans aucun doute, l'agentification.

Cette agentification doit passer par une compréhension profonde et globale du modèle derrière les systèmes d'appariement classiques, derrière la dynamique générale gouvernant ce processus, et plus

exactement derrière la logique du contrôle centralisé des différentes parties du processus. Comme par exemple, la sélection et la combinaison d'appariements, le calcul de similarités et la construction des matrices de similarités, sélection des meilleures correspondances, etc. Cette compréhension sera après transposée vers un nouveau modèle décentralisé, qui a pour but de déléguer le contrôle des différentes parties de l'appariement à des entités autonomes avec une dynamique d'évolution et de coévolution dans le temps, favorisant ainsi l'atteinte de leur objectif premier, à savoir réussir à trouver le meilleur appariement dans le groupe opposé. (Assoudi and Lounis 2014)

Pour cela, on peut dire que le composant au cœur de notre modèle à base d'agents est l'agent élément de schéma. Ce dernier, au cours de chaque itération de la simulation, passe par trois étapes principales :

- Perception de l'environnement : l'agent élément de schéma a la possibilité de percevoir son environnement de différentes manières. Sa perception de l'environnement ne se fait pas, par ailleurs, à l'aide d'un capteur proprement dit. D'une part, il est capable d'interroger son environnement pour connaître les agents disponibles pour appariement dans le groupe opposé. Par la suite, il débutera sa recherche pour trouver le meilleur appariement et ce, en calculant le degré de similarité entre lui et les autres agents disponibles pour l'appariement dans le groupe opposé. D'autre part, des messages de l'environnement, peuvent être perçus par l'agent, sous la forme d'événements, lui notifiant qu'un agent du groupe opposé l'a choisi pour un appariement potentiel, ce que l'on appelle « appariement candidat », c'est-à-dire comme le meilleur agent, pour l'instant, pour former une potentielle relation de correspondance. (Assoudi 2017)
- Raisonnement sur l'action: au cours de cette phase, l'agent élément de schéma, avec la connaissance des résultats obtenus lors de la phase de perception, raisonne sur l'action à prendre. La caractéristique principale de la version de notre prototype est qu'elle est basée sur un agent de type réflexif dont la prise de décision est basée sur des règles simples ainsi que sur la réaction à des événements. Le raisonnement donc se traduit donc par cette prise de décision qui va conditionner le choix de l'action et ensuite la transition entre les différents états internes de l'agent. (Assoudi 2017)
- Action sur son environnement : l'agent, ici, peut agir sur son environnement de différentes manières. Aussitôt, l'agent du groupe opposé trouvé comme le meilleur appariement pour lui, il transmet une demande à l'environnement après avoir désigné cet agent comme un appariement candidat. Cette demande a pour but de créer un arc unidirectionnel pointant vers cet agent

(relation de correspondance candidate). Si l'autre agent du groupe opposé, à son tour, désigne l'agent comme un appariement candidat c'est-à-dire s'ils se désignent mutuellement et en même temps comme appariement candidat, alors, les deux agents éléments de schéma demandent à l'environnement de créer des arcs bidirectionnels pour signaler la création d'un appariement que l'on appelle « appariement consensuel », et ainsi passer à un état de stabilité. (Assoudi 2017)

Il serait important de préciser que notre agent, a une connaissance restreinte limitée malgré l'interaction constante avec son environnement car il ne connaît pas les choix et les préférences internes des autres agents.

Par ailleurs, pour la conception de notre modèle, on ne s'attardera pas sur la reprise de tous les aspects qui ont été démontrés dans les approches existantes, comme par exemple, la structure des schémas ni sur l'exploitation des données des instances de schémas et de leur structure, comme cela a été fait dans plusieurs travaux. Notre choix est motivé par le fait que nous voulons d'abord nous focaliser et donner la priorité à la validation de notre hypothèse centrale : l'appariement de curriculum vitae aux offres d'emploi comme système complexe adaptatifs, et ensuite apporter des améliorations futures, une fois la validation de l'hypothèse établie par l'intégration de ces idées.

### 3.5.2 Stochasticité et adaptation

L'intégration de la stochasticité comme caractéristique essentielle dans notre modèle a principalement pour but de résoudre le déterminisme et le caractère prévisible comme on le voit avec les solutions actuelles. En effet, c'est cet aspect contraignant qui se traduit par le fait que les mêmes entrants conduisent aux mêmes sortants, ce qui nous motive à changer de paradigme. Par conséquent, la stochasticité par transmission de variations dans l'environnement, va pouvoir forcer les agents à l'adaptation et va ainsi stimuler l'émergence de solutions surprenantes mais stables.(Assoudi 2017)

Le principe de « l'ordre à partir du bruit » qui a été formulé par le cybernéticien (Foerster, 2007) en 1960, nous explique ce paradoxe. En effet, ce principe stipule que l'auto-organisation est facilitée par des perturbations aléatoires comme du bruit, ce qui permet au système d'explorer un grand nombre d'états dans son espace d'états. Autrement dit, un certain niveau de bruit dans un système auto-organisateur, lui évite de se figer et ainsi devenir inadaptable. Diverses théories ont exposé des principes similaires, tel que « l'ordre via les fluctuations ». Elles veulent toutes, exprimer que l'auto-organisation a besoin d'entropie

et de diversité pour permettre la continuité de l'adaptation et de l'évolution du système dans le temps. Par conséquent, il serait légitime de se demander comment les systèmes existants, c'est-à-dire les systèmes d'appariement déterministes, peuvent prétendre à la faculté d'adaptation, à la nouveauté, au changement des scénarios d'appariement, au changement des configurations, au déploiement de nouvelles composantes, etc., sans utiliser le principe de l'ordre à partir du bruit ni un outil comme la stochasticité. D'après la théorie de la cybernétique, cela ne serait pas possible, car pour qu'un système puisse supporter des perturbations, la nouveauté et s'y adapter, une part d'indétermination et de bruit est nécessaire.

Par conséquent, notre modèle conceptuel va intégrer à différents moments, différents niveaux de bruit et d'aléatoire, ce qui lui permettra d'évoluer, de s'adapter et finalement de faire émerger une auto-organisation. Une garantie d'un degré de liberté, fournie par l'introduction du bruit et de l'aléatoire, permet à notre modèle d'explorer une panoplie de chemins et de possibilités lui faisant, pour finir, atteindre son objectif.

Plus exactement, les agents, dans notre modèle, font appel à l'environnement, durant chaque itération de la simulation, pour la sélection des mesures de similarité comme la mesure de *Levenshtein*, *Jaro*, *JaroWinkler*, des fonctions d'agrégation comme la fonction *Weighted*, *Max*, *Average*, ou encore des différentes valeurs requises, à savoir les seuils et les poids pour réaliser les différentes parties de l'appariement.

Au cours de chaque itération, toutes les fois que l'environnement est sollicité par un agent, la composante ou la valeur demandée est retournée aux agents, par l'environnement d'une manière totalement aléatoire. On veut dire, de façon plus concise, que pour le calcul de similarité par exemple, l'agent ne sait pas au préalable quelle mesure de similarité il va utiliser durant l'itération, afin de trouver l'agent le plus similaire dans le groupe opposé. Cet aspect aléatoire est appliqué aussi pour tous les autres paramètres, comme les fonctions d'agrégation entre autres.

En d'autres termes, l'agent aura l'occasion de tester de nouvelles combinaisons à chaque itération, et cela sans se confronter au problème de l'explosion combinatoire que peut causer la tentative d'une exploration exhaustive de l'espace de recherche des combinaisons possibles. On est convaincu donc, que l'utilisation des notions de simulation et de stochasticité dans notre modèle, est la solution pour régler le problème de l'explosion combinatoire, dans le contexte de l'appariement de schémas, qui a été discuté dans

plusieurs recherches (Rahm et Bernstein, 2001b), et qui continue d'être un vrai obstacle pour les approches existantes.

### 3.5.3 Appariement consensuel

L'étape de la sélection des appariements prometteurs ou encore de la sélection des correspondances est une étape très importante en ce qui concerne le processus d'appariement générique. Ainsi, les différentes stratégies, qui peuvent être appliquées, peuvent l'être selon différentes directions, telles que: unidirectionnelle, bidirectionnelle, ou encore selon l'algorithme du mariage stable (issu de la théorie des jeux) (Gale et Shapley, 1962). L'algorithme du mariage stable a par ailleurs, été associé dans la littérature à deux étapes de l'appariement à savoir la combinaison des similarités ainsi que l'étape de la sélection des correspondances. Ainsi, les auteurs (Rahm, 2011b) définissent le problème du mariage stable comme une solution raisonnable pour la sélection des correspondances (dont le degré de similarité excède typiquement un certain seuil), dans le sens où une correspondance est sélectionnée seulement si l'un des deux éléments de schémas constituant la paire est le plus similaire à l'autre élément de la paire et vice versa (réciprocité). Les auteurs (Bellahsene *et al.*, 2011), dans ce même canevas, citent l'algorithme du mariage stable comme faisant partie des techniques permettant la déduction des correspondances. (Marie et Gal, 2008) le cite aussi comme une option pour renforcer la contrainte de cardinalité 1:1.

L'algorithme du mariage stable a contribué à répondre efficacement aux problèmes nécessitant un appariement stable, comme l'appariement hôpitaux-patients ou encore l'appariement facultés de médecine-étudiants, et même, qu'il a aussi souvent été utilisé par les approches classiques d'appariement. On pensait qu'il nous aurait aidé dans la conceptualisation de notre modèle et ainsi permettre d'évaluer à quel point on pouvait à notre tour en tirer avantage, d'autant plus qu'avec l'agentification de notre modèle, cet algorithme semblait, au premier abord, être adéquat pour notre modèle.

Nous nous sommes ravisés très vite car, après consultation des évaluations disponibles dans l'état de l'art, on s'est rendu compte que cet algorithme affichait une centralisation, un déterminisme et une rigidité qui n'allait pas du tout dans le sens des caractéristiques de liberté de choix, d'aléatoire et de décentralisation qui singularisent notre modèle. En dépit de la nature, centralisé et déterministe, de l'algorithme du mariage stable, il favorise toujours la stabilité au détriment du degré de satisfaction des entités à appairer, autrement dit, les entités doivent se contenter, des fois, de choix qui n'étaient pas leurs premiers choix.



Ce point va à l'encontre de but que notre modèle poursuit avec les agents éléments de schémas, qui est de déterminer le meilleur appariement dans le groupe opposé.

Pour cela, on a inséré dans notre modèle une nouvelle approche, l'appariement consensuel, pour la sélection des correspondances, qui par analogie se trouve basé sur le consensus. En effet, l'appariement consensuel traduit le fait qu'aucune décision n'est prise sur la sélection d'une correspondance avant que les deux parties, chacun de son côté et d'une manière unilatérale, ne se désignent mutuellement comme meilleur appariement dans le groupe opposé (appariement candidat). Par ailleurs, ce dernier, grâce aux propriétés d'évolution et de co-évolution de notre modèle, donne aux agents la possibilité (grâce à l'évolution dans le temps) de converger vers une solution localement optimale pour les deux agents, et ce indépendamment des trajectoires totalement différentes que les deux agents faisant partie du consensus peuvent emprunter pour aboutir à ce consensus.

#### 3.5.4 Auto-organisation et émergence

Dans notre modèle, le mécanisme d'auto-organisation est le mécanisme qui opère au cœur pour permettre l'émergence d'une solution à l'appariement de schémas. En effet, dans le contexte des systèmes complexes, le mécanisme d'auto-organisation est celui par lequel l'émergence d'une solution globale au niveau macroscopique se produit à partir des interactions au niveau local, c'est-à-dire au niveau microscopique des agents et ce, sans aucun contrôle central. Autrement dit, dans notre modèle, la résolution du problème de l'appariement passe par l'émergence d'une auto-organisation globale à partir des interactions locales, gouvernées à partir de règles simples des agents. D'une part, cette émergence d'une structure auto-organisée est aussi le produit d'une recherche stochastique dynamique de chaque agent dans son espace de recherche. Cette recherche stochastique fait que tout le système évolue vers un état d'optimum global pour le problème traité. En outre, cette notion d'émergence laisse voir, d'autre part, que la formation de cette nouvelle structure à partir des conditions initiales n'était pas prédictible.

#### 3.5.5 Simulations multiples et analyse statistique

Notre modèle, avec tous les concepts fondamentaux, a été conceptualisé avec comme perspective de pouvoir participer à une réduction de l'incertitude en juste une seule itération. Cependant lors de nos premières itérations, nous avons relevé deux principales remarques qui nous ont poussé à considérer l'analyse statistique sur des simulations multiples encore appelées méta-simulation.

Pour la première remarque, on s'est rendu compte qu'on avait à certains moments la formation d'appariements consensuels incorrects en dépit des nombreuses mesures prises pour empêcher ces erreurs. L'une de ces mesures est que dans notre modèle, lors de chaque itération de la simulation, chaque agent exécute toutes les étapes d'un processus d'appariement générique. C'est-à-dire que l'agent durant sa durée de vie qui correspond à la durée de la simulation peut exécuter des dizaines, si ce n'est, des centaines de fois, un processus d'appariement complet. Ceci, dans le but de trouver un appariement consensuel qui n'est autre qu'une correspondance stable entre lui et un agent du groupe opposé. L'appariement consensuel nous laissait penser que le mauvais choix d'un agent lors d'une simulation, dans la plupart des cas, ne peut pas constituer un danger pour la formation d'une correspondance incorrecte. On s'est vite ravisé car il n'existe pas de modèle d'appariement automatique pouvant éviter totalement des erreurs d'appariement. Et donc, il est évident, vu l'incertitude inhérente à l'appariement automatique, que l'on ne peut pas avoir la prétention de dire que notre modèle pourrait être capable de résoudre totalement cette incertitude.

La deuxième remarque vient de la variabilité du nombre d'itérations nécessaire afin d'obtenir toutes les bonnes correspondances possibles. En effet, quelque fois, la simulation peut atteindre le nombre maximal des itérations sans que toutes les correspondances possibles aient été trouvées. Il est donc important de signaler que pour des raisons de performance, on limite la durée maximale de la simulation à l'aide d'un paramètre, comme par exemple *run length* à 5000 itérations.

Au vu de ce qui précède, une quantification et une meilleure réduction de l'incertitude s'impose. Pour cela, on s'est inspiré d'une partie de la méthode Simulation Monte-Carlo par rapport à l'application de l'analyse statistique sur des simulations stochastiques, que l'auteur (Laurencelle 2001)<sup>11</sup> décrit en ces termes:

« La simulation stochastique qui est une application de type Monte Carlo, consiste à évaluer le modèle en le faisant vivre ou fonctionner d'une façon réaliste et comportant un ou plusieurs aspects aléatoires ».

Ainsi, contrairement à cette méthode, nous ne cherchons pas nécessairement à trouver, à partir des résultats des simulations, une distribution sur un paramètre donné ni à évaluer notre modèle.

Par contre, une analyse statistique nous servira à rendre plus stable le résultat des différentes simulations constituant la méta-simulation, et ainsi à mesurer ou quantifier l'incertitude concernant ces résultats, en calculant la fréquence d'apparition d'une correspondance au niveau de l'ensemble des simulations.

---

<sup>11</sup> (Hasard, nombres aléatoires et méthode Monte Carlo - Louis Laurencelle - Google Livres, s. d.)

L'exécution de simulations multiples ou méta-simulation peut nous aider à aller au-delà de la quantification de l'incertitude concernant les correspondances pour nous permettre d'étudier et d'analyser la performance des différentes composantes du système, comme la performance des différents appariements. Cependant, on doit noter aussi que l'exécution de simulations multiples n'est pas sans conséquence au niveau de la performance, en ce sens qu'elle peut être assez coûteuse au niveau du temps de réponse. Toutefois, ce temps peut être, réduit avec l'exécution parallèle des simulations sur plusieurs machines ou sur des micro-processeurs multi-cœur.

Au regard de tout ce qui précède, nous pensons que l'introduction de l'analyse statistique sur le résultat de plusieurs simulations s'imposait d'elle-même. On trouve donc qu'il est irréfutable que ce concept s'intègre parfaitement dans la logique de notre modèle de simulation multi-agents pour l'appariement de curriculum vitae aux offres d'emplois.

### 3.6 Transcription du modèle

La transcription du modèle consiste en une transformation du modèle conceptuel en un programme informatique qui devrait s'exécuter sur une plateforme de simulation. Ainsi, nous proposons dans ce mémoire, un prototype basé sur des agents de type réactif, pour l'appariement. Chaque élément à appairer (champs d'un CV, rubriques d'une offre d'emploi, etc.) est modélisé comme un agent autonome, appartenant à un groupe (groupe source ou cible) dont la simulation des comportements et de l'interaction avec son environnement, au niveau micro, fait ressortir au niveau macro, un système auto organisé, qui représente la solution globale à l'appariement (i.e, relations entre les éléments à appairer).

### 3.7 Conclusion

Diverses approches ont été proposées pour traiter la problématique de l'appariement automatique des CV aux OE. Par rapport à cela nous avons proposé un changement de paradigme en vue d'aborder la problématique comme un système complexe adaptatif. Dans ce chapitre, nous avons détaillé l'approche proposée ainsi que notre cheminement. En outre, nous proposons une formalisation du modèle conceptuel de notre approche de simulation multi-agents pour l'appariement automatique de curriculum vitae aux offres d'emploi. Le chapitre suivant détaillera les étapes suivies pour la transcription de notre modèle conceptuel en un prototype de simulation d'agent. Nous présenterons par ailleurs le traitement de nos données.

## CHAPITRE 4

### MODÉLISATION ET CONCEPTION DU PROTOTYPE DE SIMULATION

#### 4.1 Introduction

Nous voulons à travers ce chapitre présenter les étapes de la transcription de notre modèle conceptuel, modélisation et simulation à base d'agents pour l'appariement automatique des schémas (SMAS) en un prototype de simulation d'agents réflexifs pour l'appariement des CV aux OE.

Nous allons pour ce faire, de prime abord, présenter en détail l'implémentation du prototype pour l'appariement des CV aux OE tout en mettant en relief l'architecture interne de l'agent réflexif pour l'appariement.

Ensuite, nous décrivons la préparation de nos données et présentons la démarche poursuivie.

Pour terminer, nous faisons une analyse et établissons une représentation des interactions des éléments du système d'appariement au cours du processus de simulation. Cela nous permettra de mesurer la performance des différents apparieurs pour les différents scénarios d'appariement.

#### 4.2 C'est quoi un « Agent » ?

Nous voulons par cette section, décrire ce qu'est un agent et étudier ses comportements. Pour ce faire, il convient de rappeler le modèle opérationnel, qui selon (Treuil *et al.*, 2008) est un modèle dynamique présenté dans un langage qui respecte le méta-modèle associé à un simulateur, et qui peut être directement interprété ou exécuté par ce simulateur. Ainsi, ce modèle vise à traduire notre modèle conceptuel vers un programme informatique de simulation multi-agents pour l'appariement de schémas.

Par ailleurs, ce modèle opérationnel ou modèle exécutable, doit décrire les spécifications nécessaires à son implémentation en un programme informatique qui peut être exécuté par la plateforme de simulation. Ces spécifications seront par la suite transformées en un programme informatique de simulation multi-agents, en s'inspirant du paradigme de la programmation orientée-agents. Nous trouvons au vu de tout cela qu'il serait important de présenter les spécifications des composantes principales de ce modèle, notamment l'agent réflexif.

D'après (Shoham, 1993), l'agent peut être considéré comme un objet dans le paradigme de la programmation orientée-objets, ayant un comportement autonome, capable de raisonner sur son état, c'est-à-dire les croyances qu'il a sur le monde, dans l'objectif de décider des actions à prendre pour arriver à un but donné.

Selon S. Russell et P. Norvig :

On appelle agent toute entité qui peut être considérée comme percevant son environnement grâce à des capteurs et qui agit sur cet environnement via des effecteurs. (Russell et Norvig, 2010).

Aussi, M. Wooldridge disait :

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. (Wooldridge *et al.*, 2000).

Par ailleurs, J. Ferber donne une définition très détaillée prenant en compte toutes les composantes d'un agent:

On appelle agent une entité physique ou virtuelle a) qui est capable d'agir dans son environnement, b) qui peut communiquer directement avec d'autres agents, c) qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser), d) qui possède des ressources propres, e) qui est capable de percevoir (mais de manière limitée) son environnement, f) qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune), g) qui possède des compétences et des offres de services, h) qui peut éventuellement se reproduire, i) dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit. (Ferber *et al.*, 2004).

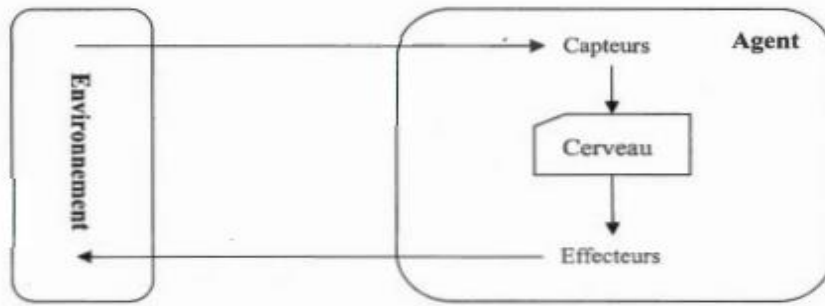


Figure 4.1 Représentation de l'interaction entre l'agent et son environnement<sup>12</sup>

Une définition plus simple, si on se réfère à la figure 4.1, d'un agent pourrait être: un agent est une entité située dans un environnement, qui est capable de percevoir cet environnement grâce à des capteurs, qui peut agir sur cet environnement grâce à des effecteurs, et qui a un ou des objectifs.(Grouls, 2013)

Il est intéressant de remarquer qu'un agent n'est pas un objet. On résumera l'exposé de M. Wooldridge qui revient à présenter trois différences fondamentales (Wooldridge *et al.*, 2000).

- La première différence est qu'une entité extérieure est capable d'utiliser directement l'état interne d'un objet (avec des attributs publics par exemple), alors qu'on ne peut pas manipuler l'état interne d'un agent ni l'obliger à effectuer une fonction. (Grouls, 2013).
- La deuxième différence notable est le caractère autonome du comportement d'un agent, contrairement à un objet qui reçoit des ordres. (Grouls, 2013).
- La troisième différence concerne l'exécution des comportements. Dans le cas d'un agent, l'exécution ne dépend que de lui. (Grouls, 2013) Alors que dans le cas d'un objet, ce choix est lié au système (plateforme logicielle par exemple) auquel il fait partie intégrante et qui planifie ses temps et durées d'exécution.

<sup>12</sup> (« Multi-agent system », 2022), (*Agents et systèmes multi-agents : vers une synthèse de ces concepts*, s. d.)

#### 4.2.1 Agent intelligent

Il serait pertinent d'établir en terme d'intelligence, la différence entre un agent intelligent et un agent simple. Pour M. Wooldridge, un agent est pris comme intelligent quand il est capable d'effectuer des actions flexibles de manière autonome, par rapport à ses objectifs (Jennings 1995, Wooldridge 1999).

Par flexible, M. Wooldridge fait référence à la réactivité qui sous-entend qu'un agent intelligent doit pouvoir réagir en un temps raisonnable aux changements de l'environnement. Il y'a aussi la pro-action par laquelle l'agent doit pouvoir prendre l'initiative d'un comportement tendant vers un but précis, et la sociabilité par laquelle l'agent intelligent doit être capable d'interagir avec d'autres agents intelligents.

Ainsi, on peut noter comme résultante, une caractéristique très importante et propre à un agent intelligent, qui est l'autonomie.

Par ailleurs, la modélisation des simulations multi-agents se base d'une manière générale sur la modélisation des agents dont les interactions « locales », tout au long de la simulation, conduisent à faire émerger une solution « globale ».

Pour récapituler, la modélisation multi-agents se définit, dans une certaine mesure, en termes de comportements individuels comme des agents participant à la simulation. Ayant comme composante centrale, l'agent élément de schéma, notre modèle suit donc bien cette règle et ne fait pas exception.

L'agent, en plus de la phase d'initialisation de la simulation, possède trois autres phases, à savoir:

- La phase de la perception de son environnement, ensuite, et en se basant sur l'ensemble des percepts, il passe à:
- La phase de la prise de décision concernant les actions à prendre, et enfin et après avoir sélectionné l'action à prendre, l'agent complète ce cycle par:
- La phase de l'exécution des actions pendant laquelle l'agent va poser des gestes concrets, modifiant ainsi son état ou l'état de son environnement.

Nous allons pouvoir décrire plus en profondeur les différentes phases de notre agent dans la section suivante.

#### 4.2.1.1 La phase de perception

Notre agent *Agent\_principal*, durant la phase de perception, perçoit son environnement non pas par des capteurs proprement dit mais par l'interrogation de son environnement. En effet, il procède à des calculs de similarité ou encore en capturant certains évènements. Un ensemble de percepts en ressortira comme résultat, ce qui permettra à l'agent, de reconnaître non seulement les agents de l'autre groupe, prêts pour l'appariement, mais aussi le degré de similarité textuelle ou encore de similarité lexicale, qui existe au niveau de l'attribut *name* ainsi qu'au niveau de l'attribut *comment*, entre cet agent et les autres agents du groupe opposé.

Pour chaque agent du groupe opposé, l'agent commence le calcul de similarité lexicale pour les deux attributs *name* et *comment*, en sélectionnant aléatoirement une mesure de similarité qui se trouve aussi générée aléatoirement par l'environnement à partir d'un bassin de mesures. Ladite mesure, calcule et retourne le score de similarité qui est ensuite comparé à un seuil aléatoire qui est une valeur aléatoire à l'intérieur d'un intervalle, retournée par l'environnement. Selon l'étape de perception, si le score dépasse le seuil aléatoire, et est éventuellement supérieur au meilleur score obtenu lors des itérations antérieures, alors le score obtenu est considéré comme meilleur score pour l'attribut *name* ou pour l'attribut *comment*.

Une autre action de perception se traduit par le fait que l'agent capture des évènements, des interrogations ou des calculs de similarités provenant de l'environnement qui lui permettent aussi de mieux percevoir le monde autour. Afin de pouvoir capturer des évènements bien précis, comme l'évènement qui se déclenche lorsqu'il est choisi par un autre agent pour un appariement candidat, l'agent effectue un tri des évènements autour de lui.

#### 4.2.1.2 La phase de décision

S'appuyant sur les résultats de la phase de perception, l'agent, au cours de la phase de décision, raisonne, délibère et décide de l'action à prendre. On peut regrouper les grandes décisions impliquant le choix d'actions possibles à prendre par l'agent, lors de plusieurs itérations de la même simulation, comme suit:



- la décision concernant la convergence des similarités *name* et *comment* et la sélection d'un appariement candidat,
- la décision concernant la réinitialisation des croyances concernant l'appariement candidat, et enfin,
- la décision concernant l'appariement consensuel.

La première décision concerne le fait que pendant le même cycle perception-décision-action, et avant que l'agent décide de sélectionner un agent du groupe opposé comme un appariement candidat, il y a une convergence, pointant vers le même agent du groupe opposé, entre l'agent sélectionné comme ayant la meilleure similarité au niveau de l'attribut *name* qui est le nom de l'élément de schémas (curriculum vitae ou offre d'emplois) et celui sélectionné comme ayant la meilleure similarité au niveau de l'attribut *comment* qui représente la description de l'élément de schéma. En ce moment, et dans le cas où la convergence est possible, l'agent émet une demande d'une fonction d'agrégation (*Max*, *Average*, *Weighted*) qui est tirée aléatoirement de l'environnement. L'agent passe, suite à ce tirage, au calcul d'agrégation des deux scores pour les deux éléments (source et cible) de schémas pour l'attribut *name* et l'attribut *comment* de chacun. Si le nouveau score agrégé est supérieur à un seuil aléatoire et éventuellement supérieur au meilleur score agrégé déjà obtenu lors des itérations antérieures, alors l'agent du groupe opposé est désigné comme appariement candidat, ce qui met fin à la première phase.

Si après un certain temps (représenté par un nombre d'itérations), dont la valeur est fixée par un nombre aléatoire dans un intervalle, le délai d'attente d'un consensus est écoulé, alors l'agent prend une deuxième décision qui est de réinitialiser ses croyances concernant l'appariement candidat en vue de revenir en mode exploration.

La formation d'un appariement consensuel est conditionnée par une désignation mutuelle des deux agents comme appariement candidat. La dernière décision vient essentiellement en réponse à l'évènement qui vient notifier à l'agent qu'il a été choisi comme appariement candidat. L'agent doit alors vérifier si les conditions sont réunies pour former un appariement consensuel.

Pendant la phase de décision, les agents font des choix, posent des actions qui laissent voir leur capacité d'apprentissage et d'adaptation. Mais il est important de relever que de par la nature de notre agent

réflexif, celui-ci ne peut réaliser des tâches de raisonnement ou de planification. Sa capacité d'adaptation se traduit par un cycle de rétroactions où il peut revoir et revenir sur ses décisions-actions défailtantes, prises lors des itérations passées (comme réinitialiser ses croyances concernant l'appariement candidat après l'écoulement du temps d'attente alloué pour un appariement consensuel). Ce qui, par conséquent, lui permettra d'explorer de nouvelles possibilités, comme l'utilisation de nouvelles mesures de similarité, de nouvelles fonctions d'agrégation, etc.

#### 4.2.1.3 La phase d'action

Au cours de la phase d'action, l'agent exécute les actions sélectionnées lors de la phase de décision. C'est d'ailleurs avec cette phase que l'itération courante de la simulation se termine. Ainsi, durant cette phase, notre agent accomplit les actions fixées lors de la phase de décision, à savoir sélectionner un appariement candidat, supprimer ses croyances concernant un appariement candidat dans le cas où le délai d'attente est écoulé, ou encore former un appariement consensuel. Pour pouvoir réaliser ces actions, l'agent, à l'aide de ses effecteurs, change son état interne et demande à l'environnement de changer le réseau des appariements.

De prime abord, lorsque la décision concernant le choix d'un appariement candidat est prise par l'agent, ce-dernier suit les étapes suivantes:

- il met à jour l'objet ayant les informations relatives à l'appariement candidat se trouvant dans la classe *CandidateMatchingAttribute*, après,
- il fait passer son état interne de l'état *UNDEFINED\_MATCH\_STATE* à l'état *CANDIDATE\_MATCH\_STATE*, et enfin,
- il passe à la création d'un arc dirigé, dans le réseau des appariements, vers l'agent qu'il a choisi comme appariement candidat.

Aussitôt qu'il y'a un changement d'état lié à l'appariement candidat, un signal est propagé par l'environnement. Ce signal aura pour but d'indiquer au reste des agents, y compris à l'agent concerné par le choix, que cet agent est prêt et disponible pour un appariement consensuel.

Au second abord, une fois que le délai d'attente d'un consensus est écoulé, l'agent prend une décision et suit les étapes suivantes:

- l'agent débute en supprimant ses croyances concernant l'appariement candidat, et ce en réinitialisant l'objet contenant les informations relatives à l'appariement candidat, ensuite,
- il fait passer encore son état interne de l'état *CANDIDATE\_MATCH\_STATE* à l'état *UNDEFINED\_MATCH\_STATE*, et enfin,
- il enchaîne par la suppression de l'arc dirigé, qui avait été créé initialement dans le réseau des appariements et qui pointait vers l'agent qu'il avait choisi comme appariement candidat.

L'agent finira, une fois sa décision concernant la formation d'un appariement consensuel prise, par exécuter les actions suivantes:

- il va mettre à jour l'objet contenant les informations concernant l'appariement consensuel qui se trouve dans la classe *ConsensualMatchingAttribute*, ensuite,
- il va faire passer son état interne de l'état *CANDIDATE MATCH STATE* à l'état *CONSENSUAL MATCH STATE*. Dès qu'il atteint ce nouvel état, l'agent se trouvera alors dans un état irréversible qui sonne la fin du cycle perception-décision-action, et qui est la fin de la phase d'exploration de son environnement à la recherche d'un appariement. Cependant, l'agent est maintenu en vie jusqu'à la fin de la simulation, et enfin,
- il crée un arc dirigé, dans le réseau des appariements, vers l'agent avec lequel il forme un appariement consensuel.

#### 4.3 Prototype et Expérimentation

Après avoir discuté dans la section précédente de ce qu'est un agent, ainsi que de ses différentes étapes de transition, il s'agira pour nous dans cette section, de présenter le prototype et de discuter de nos premières expérimentations avec ce prototype. Il s'agira avant tout, de présenter en amont, le prototype proprement dit qui implémente notre modèle opérationnel.

Dans notre contexte, nos expérimentations seront issues d'une simulation qui fait référence au produit de notre prototype, simulation multi-agents pour l'appariement de schémas, ayant pour objectif de fournir la solution à un scénario d'appariement donné par la simulation des interactions entre les éléments de schémas représentés comme des agents. Concrètement, il s'agit de fournir la solution au scénario d'appariement des CV aux OE.

Selon (Choi et Kang, 2013) :

« if our brain is not powerful enough to simulate a given complex system, we rely on computers to perform a computer simulation ».

Pour cela, quel que soit le sens du terme “simulation”, dans le contexte de notre prototype, le recours à la puissance des ordinateurs est non négociable.

Nous avons à cœur de démontrer que notre modèle à base d'agents pour l'appariement, dans lequel chaque élément à appairer représenterait les champs d'un CV d'une part et les rubriques d'une offre d'emploi d'autre part, peut être modélisé comme un agent autonome, appartenant à un groupe (groupe source ou groupe cible) dont la simulation des comportements et l'interaction avec son environnement, au niveau micro, fait émerger au niveau macro, un système auto organisé. Ce système auto organisé représenterait la solution globale à l'appariement. La vérification et la validation de notre modèle conceptuel et de notre modèle opérationnel, passent par le développement d'un prototype. Ce dernier, devient, la pierre angulaire, de la vérification de la transcription du modèle conceptuel vers le modèle opérationnel, et de la validation de l'hypothèse centrale concernant l'utilisation de la simulation multi-agents pour l'appariement de schémas.

#### 4.3.1 Prototype

Pour notre modèle opérationnel de simulation multi-agents pour l'appariement des CV aux OE, l'implémentation a été faite sur la plateforme Repast Symphony dans sa dernière version 2.8, publiée le 22 Oct 2020 et initialement développée par les auteurs (Macal et North, 2007).

En plus de la puissance et la flexibilité de cette plateforme, nous l'avons choisi aussi à cause du fait que le langage Java est le langage principal de programmation et aussi du fait que le code source de cette plateforme est ouvert.

Nous avons ensuite choisi le cadriciel DkPro (version 2.1.0) (Bar *et al.*, 2013) qui implémente une grande variété de mesures de similarité, à savoir des mesures structurelles, stylistiques, textuelles, sémantiques ou encore phonétiques dans un même et unique référentiel et dont le langage est Java pour nos calculs de similarité. DkPro est un cadriciel qui englobe en outre, SecondString (Cohen *et al.*, 2003) et SimMetrics (Chapman, 2005) qui sont des bibliothèques qui fournissent d'autres mesures de similarités additionnelles, faisant de lui, un outil puissant et bien adapté à notre besoin.

Enfin, nous avons choisi le langage de programmation R (version R 4.1.1) (Ripley, 2001) pour la gestion de l'analyse statistique. Son intégration dans l'outil Repast Symphony (Macal et North, 2010) fait de lui un choix plus qu'évident pour nous.

#### 4.3.1.1 Modélisation

D'après (Guedes et Vicari, 2009), la modélisation des systèmes multi-agents a été rendue possible avec le langage UML (Unified Modeling Language) grâce aux nouvelles notations concernant les agents de la version UML 2.0 (Parunak et Bauer, s. d.) (Huget, 2004). C'est d'ailleurs pour cette raison que nous avons choisi ce langage UML pour la modélisation et la conception de notre système multi-agents. L'avantage de cette notation sera donc d'être totalement basée sur un langage de modélisation largement accepté dans l'ingénierie logicielle, ce qui rend d'après (Huget, 2004), plus aisé le passage de l'ingénierie logicielle vers l'ingénierie des systèmes multi-agents. En effet, la modélisation orientée agents, du point de vue des fondements de la notation ou encore de l'approche, devient, avec le langage UML une spécialisation simple de la modélisation orientée objets. Nous tirons avantage des diagrammes offerts par le langage UML pour modéliser aussi bien les aspects statiques que les aspects dynamiques de notre système multi-agents.

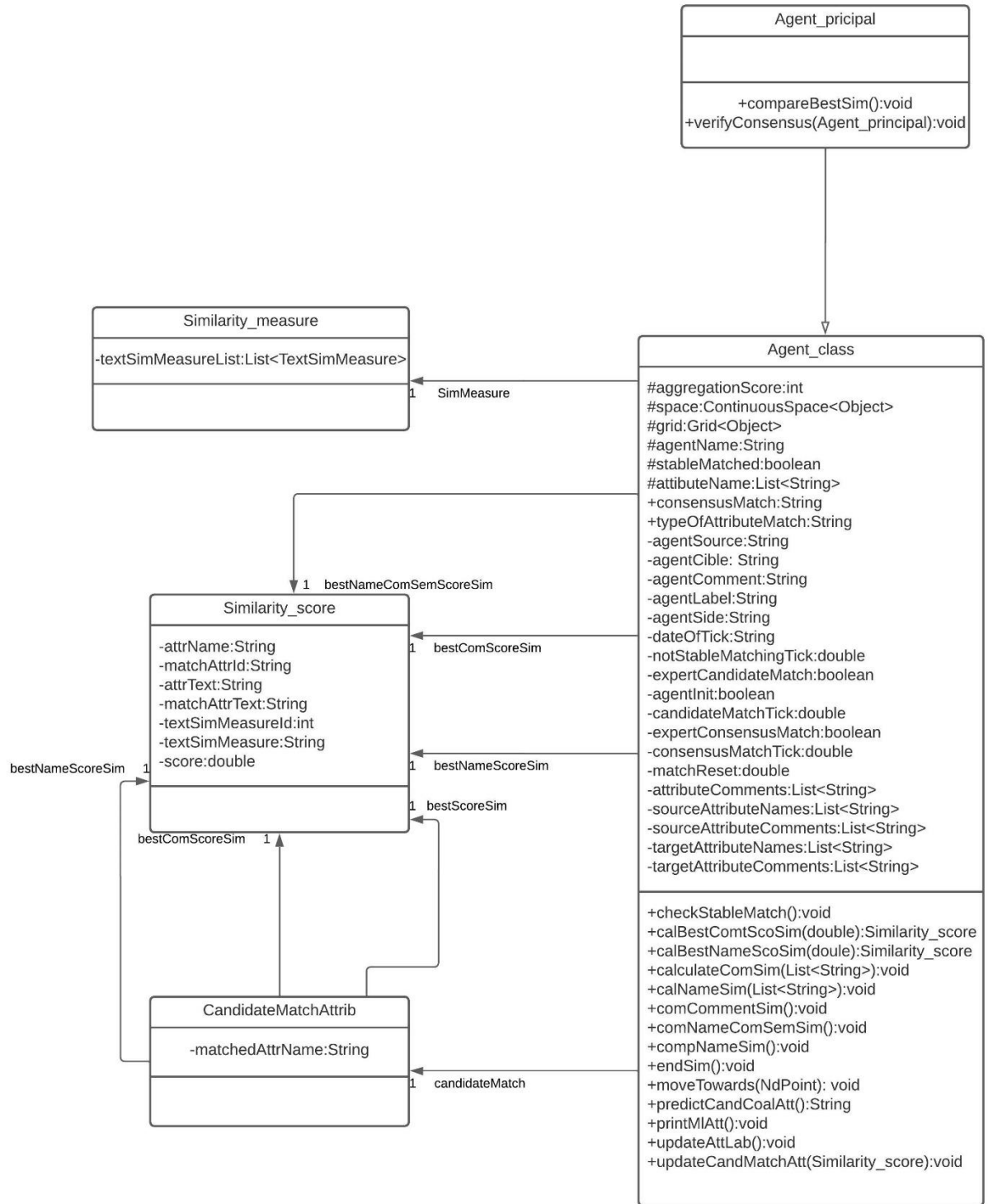


Figure 4.2 Diagramme de classes pour notre prototype

Concernant l'aspect statique, le diagramme de classes, de modéliser l'architecture interne de l'agent (i.e, ses attributs et ses comportements). Quant à la modélisation de l'aspect dynamique, nous nous sommes appuyés sur le diagramme d'activités pour modéliser le comportement global de l'agent et sur le diagramme d'états-transitions pour modéliser la transition des états internes de l'agent.

La classe *Agent\_class* est une classe abstraite montrant le concept d'agent élément de schéma et implémente les propriétés, les règles, ainsi que le comportement de l'agent en ne tenant pas compte du type de l'agent, ce qui permettrait la réutilisation et le partage des caractéristiques communes entre les différents types. Ainsi, durant la simulation, les classes qui sont directement référencées sont les classes qui implémentent la classe abstraite en l'occurrence, la classe *Agent\_principal*, qui représente donc une classe enfant implémentant les méthodes abstraites de la classe abstraite *Agent\_class*.

La classe *Agent\_principal* représente une spécialisation implémentant les comportements d'un agent réflexif. Pour pouvoir stocker et représenter toutes les informations issues de la sélection d'un appariement candidat, la classe *CandidateMatchingAttribute* est référencée par la classe *Agent\_principal*. Plus spécifiquement, cette classe permet de stocker, d'un côté, les informations concernant l'autre agent (i.e. l'index et le nom de l'autre agent) et de l'autre côté, grâce à la classe *Similarity\_Score*, les informations concernant les scores obtenus (le score obtenu pour le nom, celui obtenu pour la description, ainsi que l'agrégation des deux).

Les mesures de similarités provenant du référentiel DKPro, sont générées aléatoirement par l'environnement et permettent aux agents de faire leurs calculs de similarité. Lors de la phase de perception, elles sont stockées dans la classe *Similarity\_Measure*. Aussi, l'interface *TextSimilarityMeasure* du référentiel DKPro permet la sélection aléatoire, par l'agent, des mesures de similarité, ce qui montre l'aspect dynamique de la modélisation du comportement de l'agent réflexif.

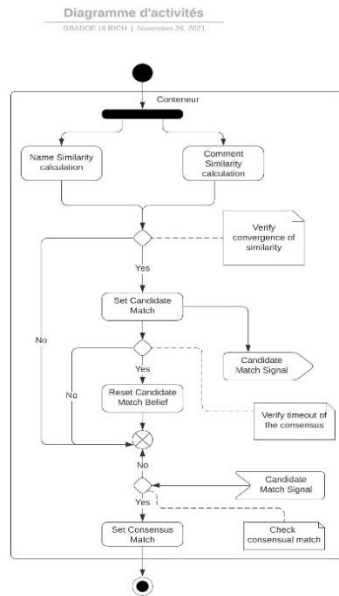


Figure 4.3 Diagramme d'activités du comportement de l'agent *Agent\_principal*

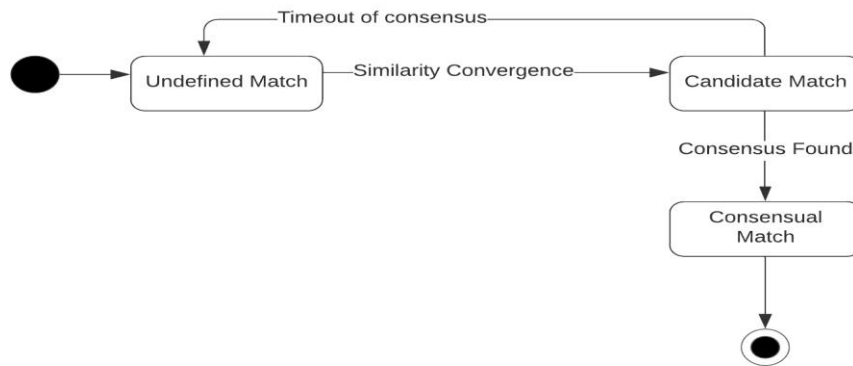


Figure 4.4 Diagramme de transition entre états de l'agent *Agent\_principal*



Le diagramme d'activités de la figure 4.3 met en évidence le comportement de l'agent, dont le but est de trouver un appariement consensuel, durant une itération. On peut remarquer, comme le montre ce diagramme, que l'approche de la sélection de l'appariement consensuel est une approche naïve, consistant en l'attente d'un consensus qui doit coïncider, pour les deux agents (ce qui peut impliquer une durée plus longue pour la simulation).

La figure 4.4 est un diagramme d'états-transitions, à base de machines à états finis et permet de constater les transitions des états internes, lors de la phase d'action du cycle perception-décision-action de l'agent *Agent\_principal*.

#### 4.3.1.2 Implémentation

La présente section traduit les théories précédemment énoncées en des exigences plus spécifiques pour l'application de l'appariement conceptualisée.

Pour l'implémentation de notre prototype, nous avons décidé d'utiliser l'outil Repast Symphony.

En effet, selon les auteurs (North *et al.*, 2013), l'outil REcursive Porous Agent Simulation Toolkit (Repast) a été développé à l'origine par Sallach, Collier et d'autres (North *et al.*, 2013) à l'Université de Chicago en 2000, et a ensuite été étendu par le laboratoire national d'Argonne en tant qu'infrastructure logicielle réutilisable pouvant prendre en charge une « découverte rapide des sciences sociales », basée sur une vaste expérimentation informatique (Sallach et Macal, 2001). Les versions successives de Repast ont étendu le système pour gérer le développement d'applications de simulation d'agents à grande échelle. Repast est géré par l'organisation bénévole à but non lucratif Repast Organization for Architecture and Design (ROAD). ROAD est dirigée par un conseil d'administration qui comprend des membres d'un éventail d'organisations gouvernementales, universitaires et industrielles. Le Laboratoire National d'Argonne est particulièrement actif dans le maintien et le développement de Repast. De nombreux autres groupes contribuent au développement de Repast. Par exemple, l'Université du Michigan s'est récemment associée à Argonne pour ajouter plus de 20 modèles de démonstration à Repast Symphony avec le support du Google Summer of Code 2011. Le système Repast, y compris le code source, est disponible directement

sur le Web (Repast – The Repast Suite)<sup>13</sup>. Toutes les versions de Repast utilisent une licence de style "nouveau BSD" et incluent des bibliothèques tierces avec des licences compatibles.

Par ailleurs, pour notre travail, il est aussi un outil de choix pour nos expérimentations car, il offre une interface très riche pour la visualisation, la collecte de données, l'exécution des simulations multiples, etc. Nous nous sommes, par la suite, focalisé sur le développement en maintenant certaines parties manuelles, telles que l'automatisation de l'importation des schémas ou encore l'automatisation de l'exécution des scripts statistiques. Repast Symphony est un environnement largement utilisé, gratuit et open source pour la modélisation à base d'agents des systèmes complexes adaptatifs. C'est un environnement de deuxième génération qui s'appuie sur la bibliothèque Repast 3 décrite dans (Macal et North, 2007).

Repast 3 est une famille de trois bibliothèques de modélisation à base d'agents, libres et open source (North et al. 2006). Les trois bibliothèques sont Repast J basé sur Java, Repast .NET basé sur C# et Repast Py basé sur NQP (Not Quite Python). Repast 3 utilise une « nouvelle licence de type BSD » (Berkeley Software Distribution) et inclut des bibliothèques tierces avec des licences. Plusieurs conceptions ont été réalisées lors de la mise en œuvre des bibliothèques. La conception orientée objet est essentielle pour permettre à la fois à la bibliothèque et aux modèles utilisateurs d'être flexibles. L'utilisation de modèles de conception est utile pour améliorer la qualité de la mise en œuvre. Les environnements doivent permettre une programmation transparente dans différentes langues pour aider simultanément à réduire les barrières à l'entrée pour les nouveaux utilisateurs et augmenter la flexibilité pour les experts.

---

<sup>13</sup> <https://repast.github.io/index.html>

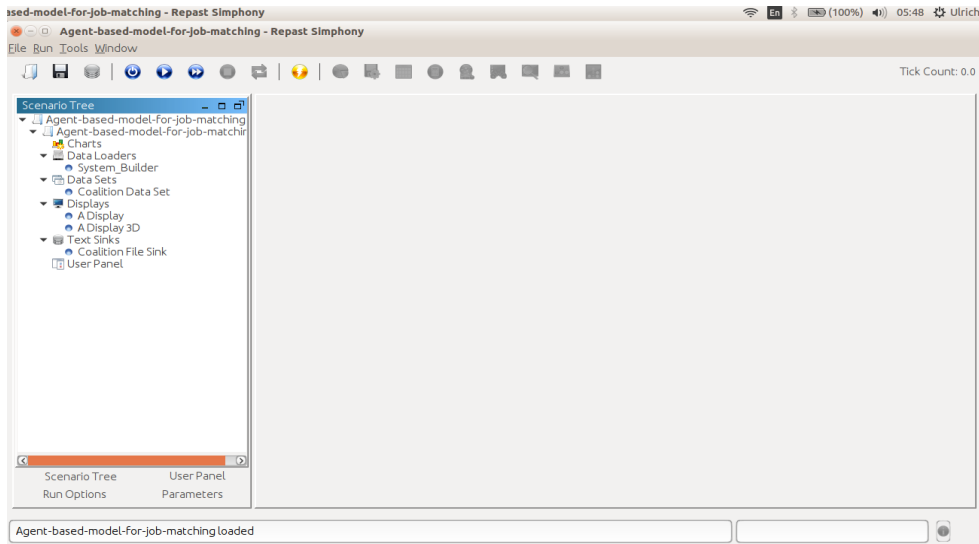


Figure 4.5 Interface de la plateforme Repast Simphony

Sur la figure 4.5, on voit l'interface de notre outil de simulation Repast Simphony, une fois la simulation débutée, on pourra voir les agents essayer de communiquer.

Pendant l'exécution de la simulation, l'utilisateur peut suivre son évolution sur le panneau de visualisation de l'outil Repast Simphony. Au début de la simulation, à l'itération 0, comme l'illustre la figure 4.6, le panneau de visualisation nous permet de voir les agents des deux schémas dispersés aléatoirement dans l'espace de l'environnement. Notons que contrairement à d'autres modèles de simulation, l'emplacement des agents n'a pas d'importance pour notre modèle.

Les agents affichent, comme on peut le remarquer aussi sur la figure 4.6, une sorte de libellé qui correspond à l'attribut *name*, un attribut qui fait référence au titre de la section du curriculum vitae ou de l'offre d'emploi. Par ailleurs, un jeu de couleurs nous permet de catégoriser les agents de chaque schéma. Ainsi, les éléments du schéma source auront la couleur rouge, la couleur bleue sera pour les éléments du schéma cible.

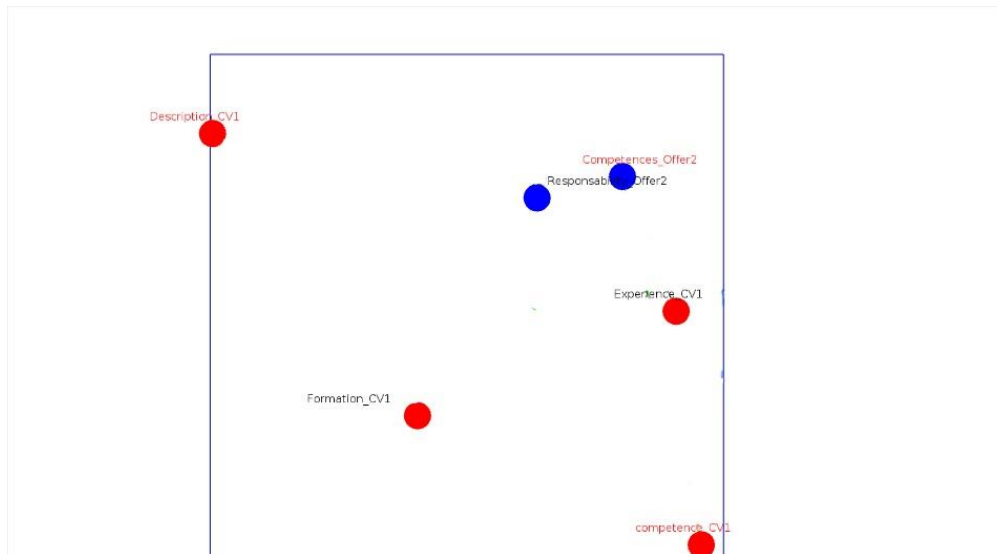


Figure 4.6 Disposition des agents dans le panneau de visualisation de notre prototype

Étant donné que notre but est de créer un système dont les utilisateurs seront des personnes des Ressources Humaines (RH) et donc qui n'auront pas forcément l'informatique comme domaine de prédilection, notre prototype sera muni de deux autres interfaces plus simples et plus conviviales. Ces interfaces serviront, au terme d'une simulation, à afficher uniquement les données dont un gestionnaire RH a besoin pour la prise de décision au cours du processus d'embauche. Ainsi, la première interface affiche la vue détaillée mais en prenant soin de n'afficher strictement que les informations qui seront utiles au professionnel RH. En effet, la première interface affichera les différentes rubriques des CV qui se seraient appariées avec les rubriques des offres d'emploi ainsi que leur taux de correspondance. La deuxième interface quant à elle va reprendre le même principe en classant les meilleurs CV pour les différentes offres d'emploi, après avoir préalablement calculé leurs scores finaux d'appariement par rapport aux taux de correspondance obtenu pour chaque rubrique.

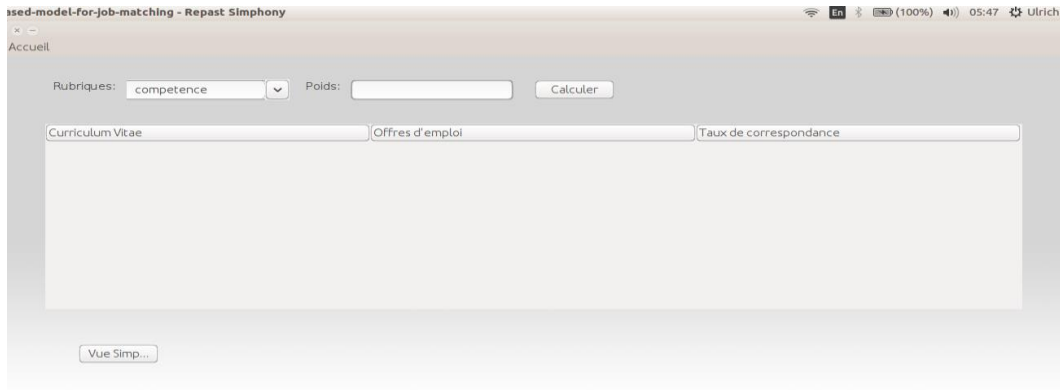


Figure 4.7 Interface utilisateur affichant la vue détaillée de notre prototype

La figure 4.7 ci-dessus nous montre:

- Un tableau qui servira à stocker les rubriques des différents éléments (CV et OE) qui se sont appariées avec le taux de correspondance de chacun.
- La section en haut du tableau, qui permet au recruteur à l'aide de la liste déroulante de majorer une rubrique qu'il jugera plus importante dans le cas où il voudra, par exemple, départager deux profils intéressants pour une offre d'emploi.

En outre, on a un bouton en dessous du tableau qui pourra permettre au gestionnaire de passer de la vue détaillée à la vue simplifiée (Figure 4.8).

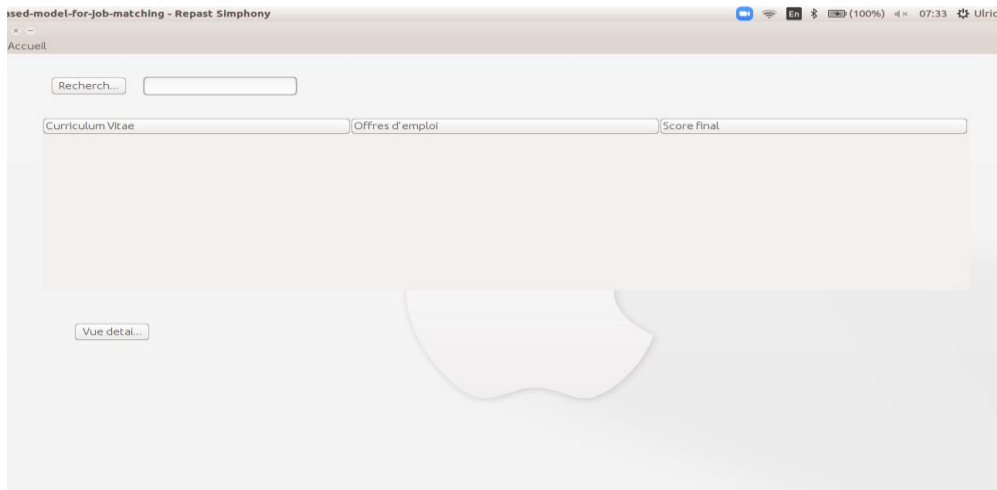


Figure 4.8 Interface utilisateur affichant la vue simplifiée de notre prototype

Au niveau de la vue simplifiée (figure 4.8), on a :

- Un champ de recherche qui permet de rechercher suivant une offre d'emploi, les meilleurs candidats correspondants et vice versa.
- La colonne « taux de correspondance » est remplacée par « score final » qui permet de voir ou d'estimer les pourcentages d'appariement des candidats aux différentes offres d'emploi et vice versa.

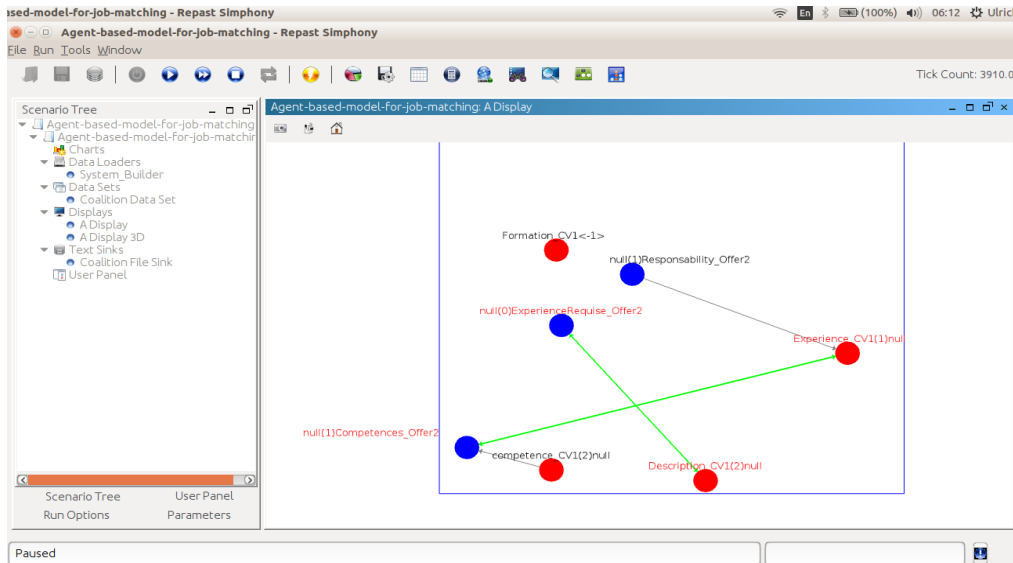


Figure 4.9 Panneau de visualisation au cours d'une simulation

Durant la simulation proprement dite, comme on peut le voir sur la figure 4.9, des arcs dirigés avec respectivement la couleur grise pour un appariement candidat et la couleur verte pour un appariement consensuel, sont créés entre les agents. Une concaténation du nom de l'agent, du score agrégé se met à jour au niveau du libellé de l'agent, lorsqu'un consensus est trouvé. Ces informations nous permettent de savoir notamment la trajectoire ayant menée l'agent à trouver justement cet appariement consensuel.

Mis à part le contrôle de la durée de la simulation géré par le paramètre qui définit le nombre d'itérations, nous avons choisi volontairement de ne pas en avoir d'autres, d'une part parce qu'on veut relever la nature autonome et adaptative de notre approche et d'autre part parce que toutes les valeurs des paramètres importants sont directement générées de manière aléatoire par l'environnement, au cours de la simulation.

#### 4.3.1.3 Prétraitement des données

Nous présentons dans ce travail, notre méthode d'appariement des CV aux OE, qui repose sur la simulation multi-agents et sur l'exploration de différentes caractéristiques à partir du contenu de ces agents.

Le principal défi ici est de savoir comment extraire des modèles d'offres d'emplois et de CV dont les différentes caractéristiques ou sections vont constituer nos agents. En effet, pour calculer la similitude

entre un emploi et un CV, notre système d'appariement a besoin de modèles numériques structurés, de chaque document.

Pour obtenir les données structurées, certains gestionnaires des ressources humaines demandent aux demandeurs d'emploi de saisir leurs profils dans des formulaires champ par champ, et le recruteur saisit ses descriptions de poste de la même manière. Cependant, la plupart des utilisateurs sont réticents à suivre le processus qu'ils trouvent fastidieux. Les demandeurs d'emploi préfèrent télécharger leur curriculum vitae directement et les recruteurs préfèrent publier l'intégralité des descriptions de poste sur des sites Web. Il est souvent observé que les CV sont remplis avec excès d'informations, souvent sans rapport avec ce que l'évaluateur cherche. Par conséquent, le processus d'évaluation des CV en vrac devient souvent fastidieux.

Au vu de cette situation, nous voulons donc extraire les informations structurées d'une source de données non structurée, comme les CV aux OE. Ainsi, pour pouvoir préparer l'ensemble de données que nous utiliserons pour nos expérimentations, nous nous sommes basés sur une combinaison de techniques de l'apprentissage machine et du Traitement Automatique du Langage Naturel (TALN).

#### ➤ Les Curriculum Vitae

Concrètement, nous voulons pouvoir utiliser n'importe quel curriculum vitae et n'importe quelle offre d'emploi non structuré en les transformant en des données structurées avec les informations pertinentes dont nous aurons besoin. Autrement dit nous nous intéressons dans un premier temps au domaine du traitement du langage naturel et de la recherche d'informations appelé reconnaissance d'entité nommée et comment nous pouvons l'appliquer pour générer automatiquement des résumés de curriculum vitae. L'idée est d'extraire uniquement les entités principales telles que le nom, la formation, les compétences, etc.

Étant donné que nous traitons des centaines de documents textes, les techniques de NLP seront utiles pour résoudre les différents problèmes à savoir :

- Extraction de données utiles dans du texte libre: procéder de façon manuelle peut être facile et rapide avec peu de documents. Par contre, s'évertuer à le faire avec une centaine de documents est plus fastidieux et requiert plus d'expertise.



- Synonymie : on a remarqué qu'une compétence peut avoir plusieurs dénominations dans la même langue (des synonymes), ce qui sous-entend que l'ajout des synonymes aux données extraites peut grandement nous aider à trouver plus de candidats dans notre base de données de profils candidats.
- Traduction : il est important de souligner qu'une offre d'emploi est, la plupart du temps, dans une seule langue. Force est de constater que la pléiade de profils que nous manipulons est réparti entre la langue anglaise et française. En effet, plusieurs candidats sont bilingues et pourraient être compétents et intéressés par une offre d'emploi dans une autre langue que celle de leur profil. Il est donc indispensable pour nous de pouvoir traduire les données extraites de chaque offre d'emploi dans la langue seconde sinon, on laisserait de côté des millions de profils de candidats dont le contenu textuel est dans la langue seconde. Nous allons utiliser les techniques de traduction automatique pour résoudre ce problème.

Pour commencer, nous allons créer un modèle utilisant la bibliothèque Spacy en Python qui extraira les principaux points d'un curriculum vitae. Nous allons former le modèle sur près de 150 CV. Une fois le modèle prêt, nous extrairons le texte d'un nouveau CV et le transmettrons au modèle pour obtenir le résumé. La collecte de données d'entraînement est une étape cruciale lors de la création de tout modèle d'apprentissage automatique.

Préalablement à la présentation de notre approche, nous décrivons les différents types de données dont nous disposons pour ce projet.

Nos données sont constituées de milliers de profils de candidats rendus anonymes qui proviennent de réseaux sociaux publics. Elles sont pour la plupart soit en format JSON soit en PDF. Les champs contenus dans un profil nous permettent de déterminer le parcours scolaire et professionnel d'une personne, ses compétences, son expérience, sa description, etc.

<b>ID</b>	<b>Summary</b>	<b>Skills</b>	<b>Educa-tion</b>	<b>Experience</b>
3	Senior Web Developer specializing in front end development. Experienced with all stages of the development cycle for dynamic web projects. Well-versed in numerous programming languages including HTML5, PHP OOP, JavaScript, CSS, MySQL. Strong background in project management and customer relations.	-Project management- Strong decision maker-Complex problem solver Creative design- Innovative- Service-focused	Bachelor of Science: Computer Information Systems – 2014 Columbia University, NY	Web_Developer - 09/2015 to 05/2019 at Luna Web Design, New York: -Cooperate with designers to create clean interfaces and simple, intuitive interactions and experiences.-Develop project concepts and maintain optimal workflow.-Work with senior developer to manage large, complex design projects for corporate clients.- Complete detailed programming and development tasks for front end public and internal websites as well as challenging back-end server code. -Carry out quality assurance tests to discover errors and optimize usability.

Tableau 4.1 Exemple de profil d'un candidat

La première ligne du tableau 4.1 décrit les principales catégories d'information présentes dans les profils. Le champ *ID* désigne un identifiant numérique unique du profil. Le champ *SUMMARY* varie d'un profil à l'autre mais pour la plupart, ce champ présente un bref résumé de la personnalité du candidat et fait état de certaines informations additionnelles, comme la motivation du candidat ainsi que ses perspectives d'avenir. Le champ *SKILLS* nous montre les différentes compétences dont dispose le profil, à titre illustratif,

on n'en a retenu que quelques-uns. Le champ *EDUCATION* qui fait référence au cursus scolaire du candidat ainsi qu'aux diplômes obtenus. Le champ *EXPERIENCES* nous montre, les différentes expériences du profil avec la période correspondante. Dans un souci de confidentialité, nous avons anonymisé les noms des compagnies en remplaçant certaines lettres par des astérisques.

Dans ce projet, nous avons utilisé 150 CV pour former notre modèle, étant donné qu'on cherche de prime abord la validation de notre hypothèse expérimentale. Notre ensemble de données a été collecté sur Internet et sera disponible en tant que fichier rattaché au code source de notre prototype expérimental.

Notons aussi au passage, bien que nous avons la chance d'avoir une importante base de profils candidats pour nos expériences, plusieurs profils sont « vides ou pauvres en contenu ». Nous en voulons pour preuve, environ 21% de CV sans compétences renseignées. Cependant, nous avons une quantité de profils candidats non négligeable afin de mener à bien nos expérimentations avec nos méthodes d'appariement.

Nous passons à présent à la phase de prétraitement qui a été produite sur Google Colab ou Colaboratory qui est un service cloud offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche en apprentissage automatique. Nous avons choisi cette plateforme car elle permet d'entraîner des modèles de Machine Learning directement dans le cloud. Ainsi, sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur, nous pouvons l'utiliser ce qui nous offre une plus grande flexibilité.

Il est important de rappeler que pour notre prétraitement, nous ne supprimerons aucune information jugée pas assez pertinente ni des profils candidats ni des offres d'emplois. En effet, nous voulons donner la chance aux agents lors de la simulation, d'apprendre afin de prendre eux-mêmes des décisions. Nous souhaitons juste mieux structurer les données brutes afin que tous les profils candidats aient la même structure, c'est-à-dire le même format ce qui permettrait d'avoir une entrée uniforme au cours de la simulation.

Nous avons donc eu à importer ces bibliothèques, comme on peut le voir sur la figure 4.10:

```
[ ] import spacy
import pickle
import random
```

Figure 4.10 Pseudo code pour le prétraitement

Nous allons charger ensuite les données d'entraînement. Les données représentent le contenu du curriculum vitae extrait d'un fichier PDF. Ensuite, on a un dictionnaire composé d'une étiquette, de l'index de début et de fin de la valeur dans le CV. Comme étiquette nous avons par exemple “*Expérience Professionnelle*”, qui représente l’expérience tirée des entreprises dans lesquelles a travaillé le candidat et qui se veut être une étiquette personnalisée ayant plusieurs valeurs dans le CV.

```
▶ train_data = pickle.load(open(donnees_entrainement.pkl', 'rb'))
train_data[0]
```

Figure 4.11 Pseudocode pour le chargement de nos données

Nous voulons pouvoir traiter les CV dans deux langues, à savoir le français et l'anglais. Pour se faire, nous avons d'abord chargé un modèle SpaCy en français, puis nous avons introduit le modèle SpaCy en anglais (disponible tous les deux sur le site: <https://spacy.io/models/>). Ensuite, nous écrivons une fonction qui prendra les données d'apprentissage en entrée. Dans la fonction, nous allons d'abord ajouter une reconnaissance d'entités nommées en dernière position dans le pipeline de traitement. Ensuite, nous ajouterons nos étiquettes personnalisées dans le pipeline.

Étant donné que la reconnaissance d'entité nommée n'utilise aucune fonctionnalité définie par le tagueur et l'analyseur, et ainsi de suite, cela signifie que nous pouvons les échanger ou supprimer des composants individuels du pipeline sans affecter les autres. Nous allons maintenant préparer nos données pour l'entraînement en désactivant tous les composants du pipeline sauf “NER”, celui de la reconnaissance des entités nommées que nous allons seulement entraîner.

L'entraînement a été fait pendant 20 itérations. Afin d'éviter le surapprentissage du modèle, nous mélangeons les données d'apprentissage à chaque itération, pour garantir que le modèle ne fait aucune généralisation basée sur les données des exemples. Afin d'améliorer les résultats d'apprentissage, nous avons défini un taux "droupout" qui aura pour but de rendre difficile, la mémorisation des données d'entraînement, pour le modèle. Cela rend plus difficile pour le modèle de mémoriser les données d'entraînement. L'étape suivante sera d'entraîner le modèle sur nos données et de le tester avec une des données de test, tout en utilisant un format bien déterminé par rapport aux informations qui nous intéressent.

Pour valider, notre modèle, nous allons maintenant le tester sur un CV qui se trouve dans nos données de test. Après extraction du texte de notre CV, nous le soumettons à notre modèle pour obtenir le format voulu:

```
import sys, fitz
fname = '/content/drive/MyDrive/Resume-and-CV-Summarization-and-Parsing-with-Spacy-in-Python-master/Resume-and-CV-Summarization-and-Parsing-with-Spacy-in-Python-
doc = fitz.open(fname)
text = ""
for page in doc:
    text = text + str(page.getText())

tx = " ".join(text.split('\n'))
print(tx)
```

Figure 4.12 Pseudocode pour le test de notre modèle

Ci-dessous, on voit un exemple de CV dans le format voulu:

```
[15] doc = model(tx)
for ent in doc.ents:
    print(f'{ent.label_.upper():{30}}- {ent.text}')
```

NAME	- Alice Clark
LOCATION	- Delhi
COMPANIES WORKED AT	- Microsoft
DESIGNATION	- Software Engineer
COMPANIES WORKED AT	- Microsoft
DESIGNATION	- Microsoft Rewards Live dashboards:
COMPANIES WORKED AT	- Microsoft
COMPANIES WORKED AT	- Microsoft
COMPANIES WORKED AT	- Microsoft
COLLEGE NAME	- Indian Institute of Technology - Mumbai
GRADUATION YEAR	- 2001
SKILLS	- Machine Learning, Natural Language Processing, and Big Data Handling

Figure 4.13 Pseudocode montrant la sortie de notre curriculum après traitement

➤ Les Offres d'emploi

Par ailleurs, on possède aussi des offres d'emploi elles aussi, récoltées sur les plateformes sociales professionnelles publiques. Nos offres sont en format PDF et JSON. Tout comme dans les CV, plusieurs champs nous permettront d'apparier nos offres d'emploi avec des CV. Ainsi, afin d'avoir une meilleure connaissance, on distingue dans chaque offre, des champs : description du poste, son emplacement géographique, la mission du futur détenteur du poste, les exigences en termes d'éducation, d'expérience professionnelle et de compétences requises pour le poste. Nous allons travailler sur une base de données d'environ plus de 100 offres d'emplois canadiennes.

Id	Titre	Description	Place
9	Maintenance Millwright	-Industries Limited has enjoyed steady growth over the years providing metal finishing solutions. To continue our growth we are currently seeking a dynamic, well-organized Maintenance Technician/Millwright to join our team. SUMMARY: The primary duties of the Maintenance Technician/Millwright is to maintain and repair plant equipment. MAJOR RESPONSIBILITIES/ DUTIES: Surveillance of plant equipment and shop upkeep: Organize and prioritize workload-Ensure minimal down-time of equipment-Complex problem solver Creative design Use hand and power tools and welding equipment. Health & Safety• Ensure Compliance REQUIREMENTS / QUALIFICATIONS: • Educational background in mechanical engineering, coupled with a Millwright and/or Electrician license (or equivalent) is considered an asset • Minimum of five (5) years experience working in a related position • Excellent interpersonal, written and verbal communication skills • Solid trouble-shooting and problem solving skills • Ability to prioritize, plan and schedule work effectively with a production team in a fast paced and values driven environment -Strong organization and time management skills to ensure deadlines are met.	Cambridge, ON

Figure 4.14 Exemple d'une offre d'emploi

Tout comme dans le cas des CV, on distingue sur la figure 4.14, le champ *ID* qui correspond à un identifiant numérique unique de l'offre. Le champ *COMPANY NAME* correspond au nom de la compagnie. Le champ *DESCRIPTION* indique les informations sur l'entreprise et le poste. On y trouve notamment, les compétences et l'expérience exigées mais non balisées. Le champ *PLACE* correspond à l'emplacement géographique de l'offre d'emploi. La plus grande difficulté se trouve bien évidemment dans l'extraction automatique des informations dans le champ *DESCRIPTION*. En effet, elle requiert une combinaison de plusieurs techniques de traitement automatique de la langue. Pour ce faire, on s'est inspiré du travail de (Dieng, 2016).

Pour l'extraction des caractéristiques utiles des offres d'emploi, on s'appuiera sur deux ontologies *CNP-NOC* et *ESCO*.

L'ontologie *CNP-NOC* regroupe la classification nationale des professions canadiennes. Elle représente une ressource gouvernementale canadienne qui répertorie tous les métiers qu'on rencontre dans l'économie canadienne. Comme tout document fédéral officiel, il est bilingue, ce qui nous donne un avantage. Comme l'ontologie *ESCO*, *CNP-NOC* nous sera très utile également, dans l'extraction des items utiles contenus dans les offres d'emplois. (Dieng, 2016)

En outre, *ESCO* (Vrang, Papantoniou et al. 2014), est elle aussi une ontologie du domaine des ressources humaines développée par une agence de l'union européenne. Il s'agit contrairement à *CNP-NOC*, d'une classification des compétences, qualifications et occupations européennes. Cette ontologie a l'avantage d'être multilingue, mais dans le cadre de notre travail de recherche, nous nous limitons aux dénominations dans les langues anglaise et française. Par ailleurs, elle offre des dénominations alternatives pour chaque compétence, qualification et occupation. De plus, il existe des relations entre les occupations et les compétences de sorte que dans *ESCO* on peut retrouver les compétences de chaque métier ou groupe de métiers. (Dieng, 2016)

Nous avons la technique de *NER* (*Named Entity Recognition*) dans le domaine du traitement automatique des langues (*TALN*). *NER* (également connu sous le nom d'identification d'entité, entité Chunking et extraction d'entité) est une sous-tâche d'extraction d'information qui vise à localiser et classer les entités nommées dans le texte en catégories prédéfinies, telles que les noms des personnes, organisations, lieux, expressions de temps, quantités, valeurs monétaires, pourcentages, etc.

Nous procédons à l'extraction automatique de certaines données dans les offres d'emploi. Dans le but d'effectuer une évaluation des extracteurs automatiques, nous avons pensé à utiliser une référence afin de procéder à la vérification de la performance de nos algorithmes. Pour la mise en place de cette ressource qui fera office de référence, nous avons effectué un tirage au sort de 20 OE parmi les centaines d'OE dont nous disposons. Pour ces 20 offres, nous procédons à l'extraction manuelle du titre, du nombre d'années d'expérience et des compétences requises.

Les caractéristiques des 80 offres d'emploi restantes, qui nous intéressent principalement et que nous voulons extraire sont: le métier contenu dans le titre de l'offre d'emploi, les compétences requises contenues dans la description de l'offre d'emploi et enfin, l'expérience professionnelle exigée par l'offre d'emploi.

- Le titre: le titre d'une offre d'emploi est souvent indiqué dans le champ portant le même nom. Cependant, on remarque que dans certains cas, des données additionnelles sont ajoutées par l'employeur pour mieux détailler ladite offre. Ces informations varient et peuvent représenter le statut de l'emploi (temps plein, temps partiel, temporaire ou permanent), le secteur d'activité de l'entreprise (construction, aéronautique, etc.), la localisation du poste (ville, province et pays) ou toute autre information que l'employeur juge pertinente. Cependant, dans notre cas, on entend par le titre d'une offre d'emploi, le métier. C'est la raison pour laquelle, on n'annote ici que le métier ou l'occupation contenus dans le titre de l'offre d'emploi. L'extraction du métier se fait en gardant certains éléments comme le secteur d'activité car le but de la simulation est de pouvoir donner la chance aux agents de disposer de toutes informations pertinentes pour la prise de décision.

Par contre, on procède à la suppression du bruit en éliminant des mots. Nous avons dû constituer manuellement un anti dictionnaire, c'est-à-dire une liste de termes qui sont toujours omis.

Exemples de bruit : temps complet, 12\$/h, poste basé à Montréal, etc.

Cet anti dictionnaire sera utilisé plus tard en guise d'informations supplémentaires lors de la simulation.



- *Expérience* : l'expérience représente, le nombre d'années requises pour le poste. Dans la plupart des cas, elle est transcrite par une phrase explicite via le champ *description* qui nous indique l'expérience.

Dans le souci de bien réaliser notre appariement, nous devons pouvoir réussir l'extraction de *l'expérience* avec un taux de succès acceptable. Pour ce faire, proposons une méthode pour extraire efficacement *l'expérience* exigée par une offre d'emploi que voici :

- On élabore de prime abord, une liste de tous les termes qui font directement référence à une *expérience*: *experience, expérience, years, années, related, etc.*
  - On réalise la construction d'une expression régulière avec cette liste.
  - À l'aide de notre expression régulière, on procède à la recherche de l'expérience dans le champ *description* de nos OE.
- *Compétences* : de façon générale, les compétences requises pour le poste sont répertoriées dans le champ *description* de l'offre d'emploi. Par rapport à l'extraction du métier, l'extraction des compétences dans les OE est une tâche un peu plus difficile. La difficulté réside dans le fait d'extraire les *compétences* minimales dans du texte libre. Notre procédé afin de pour pouvoir extraire de ce texte seulement les groupes de mots qui sont des *compétences*, se décline en ces phases :
    - On procède, dans un premier temps, à la construction d'un arbre de *compétences* à l'aide des ontologies avec la liste de *compétences* les plus fréquentes dans les profils.
    - On recherche, dans un second temps, dans cet arbre, les suites de mots du champ *description* des offres dans l'arbre des *compétences*.

#### 4.3.2 Analyses et Travaux

Après la partie du prétraitement de nos données, vient la question de la pertinence de l'utilisation de la simulation dans le contexte de la problématique de l'appariement. Les travaux de (Assoudi 2017) ainsi que

nos premières expérimentations nous ont fait voir à quel point la simulation était un outil puissant pour répondre naturellement, à des questions faisant référence à l'analyse d'hypothèses. Autrement dit, la simulation s'est avérée efficace dans la gestion des questions qui s'intéressent à ce qui pourrait se produire, dans le cadre d'un scénario d'appariement donné, avec une combinaison donnée, d'appariements, de fonctions d'agrégation, de paramètres de configuration, plutôt qu'avec une autre. La problématique de l'explosion combinatoire à laquelle se confrontait les approches classiques se trouve résolue par l'approche de simulation.

Avant de commencer, nos expérimentations, il convient de définir un scénario d'appariement ensuite d'expérimenter ce scénario d'appariement avec un outil connu de la littérature, afin de pouvoir en évaluer la pertinence et d'avoir des résultats que l'on pourrait prendre comme point de référence pour les expérimentations futures, avec notre prototype.

Une fois la question de la pertinence résolue, nous passons à l'étape de l'expérimentation du prototype, durant laquelle on procède à l'évaluation préliminaire des performances par rapport aux résultats attendus par l'utilisateur et par rapport aux résultats obtenus comme point de référence lors de la première étape.

Afin de pouvoir procéder à la validation de notre prototype, il va falloir choisir l'exemple de référence comme scénario d'appariement qui doit être à la fois facile à décrire, pour mieux expliquer le déroulement des expérimentations, et avoir un niveau de difficulté élevé pour mieux situer les performances obtenues lors des expérimentations.

Notre exemple de référence sera un scénario composé de 1 CV et de 2 offres d'emplois, chacun avec deux rubriques, ce qui est relativement assez simple car ayant peu d'éléments. Le but poursuivi étant de valider le prototype, nous aurons donc à comparer afin de pouvoir déterminer la difficulté que peut poser ledit scénario d'appariement à un outil d'appariement automatique de curriculum vitae aux offres d'emploi, existant.

Avant d'aller plus loin, il serait pertinent de donner les résultats attendus par l'utilisateur pour l'exemple de référence, choisi comme scénario d'appariement pour l'expérimentation de notre prototype. Ainsi, les correspondances à trouver sont comme suit:

*Competence\_CV* doit s'apparier avec *Responsability\_Offer*

*Experience\_CV* doit s'apparier avec *ExperienceRequise\_Offer*

*Formation\_CV* doit s'apparier avec *Competence\_Offer*

Les agents avec la terminaison "*\_CV*" sont relatifs aux CV et les agents avec la terminaison "*\_Offer*" aux offres d'emploi.

Pour les CV, l'agent *Competence\_CV* nous renseigne sur les compétences des candidats, l'agent *Experience\_CV* nous renseigne sur les expériences professionnelles des candidats, l'agent *Formation\_CV*, sur le parcours scolaire des candidats.

En ce qui concerne les offres d'emplois, l'agent *Competence\_Offer* nous renseigne sur les compétences requises et les exigences pour le poste afin de pouvoir remplir efficacement les tâches liées au poste. L'agent *Responsability\_Offer* lui, fait référence aux tâches dont les candidats sélectionnés seront responsables et pour finir, l'agent *ExperienceRequise\_Offer* fait état du nombre d'années d'expériences pertinentes requis afin de pouvoir être un candidat potentiel à l'offre.

#### 4.3.2.1 Simulation

Pour notre simulation, on aura deux schémas d'entrée:

- les CV seront les schémas sources. Ils seront composés de trois éléments qui seront nos agents: *Competence\_CV*, *Experience\_CV* et *Formation\_CV*
- les offres d'emplois seront les schémas cibles. Ils seront composés aussi de trois agents à savoir: *Responsabilite\_Offer*, *ExperienceRequise\_Offer* et *Competence\_Offer*.

Au cours d'une simulation nous effectuons des pauses à de différents moments afin de montrer clairement le déroulement du modèle opérationnel de notre simulation multi-agents pour l'appariement des CV aux offres d'emplois. Nous allons nous appuyer sur la richesse de la plateforme Repast Symphony en termes d'outils pour capturer l'état de notre simulation lors des différentes pauses. Ainsi, une série de captures d'écran ont été prises à des moments différents de la simulation, nous renseignant ainsi sur l'évolution et

la coévolution (agent-agent et agent-environnement). La figure 4.15 nous montre, par exemple, la situation à l'itération 1034 de la simulation.

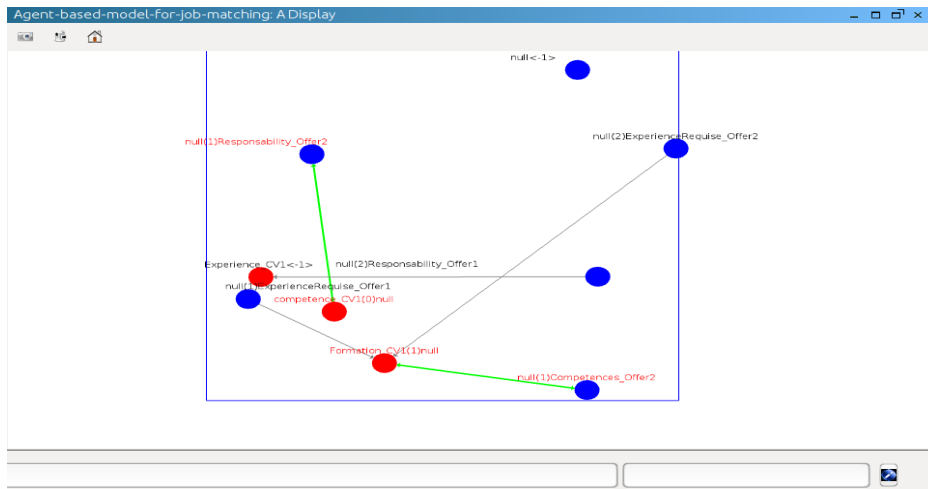


Figure 4.15 Panneau de visualisation à l'itération 1034

Les changements observés à l'itération 1034 de la simulation sont:

Les agents *Formation\_Cv1* et *Competence\_Offer2* ont déjà formé un appariement consensuel stable. De même que pour les agents *Competence\_Cv1* et *Responsability\_Offer2*. On note aussi, les choix erronés de l'agent *ExperienceRequise\_Offer2* qui a choisi comme appariement candidat *Formation\_Cv1* malgré que ce-dernier ai déjà opéré un appariement consensuel stable. Les agents *ExperienceRequise\_Offer1*, *Responsaility\_Offer1* ont aussi fait des choix erronés en choisissant respectivement comme appariement candidat les agents *Formation\_Cv1* et *Experience\_Cv1*.

À l'itération 1737 nous remarquons, par rapport à l'itération 1034 un changement tel que :

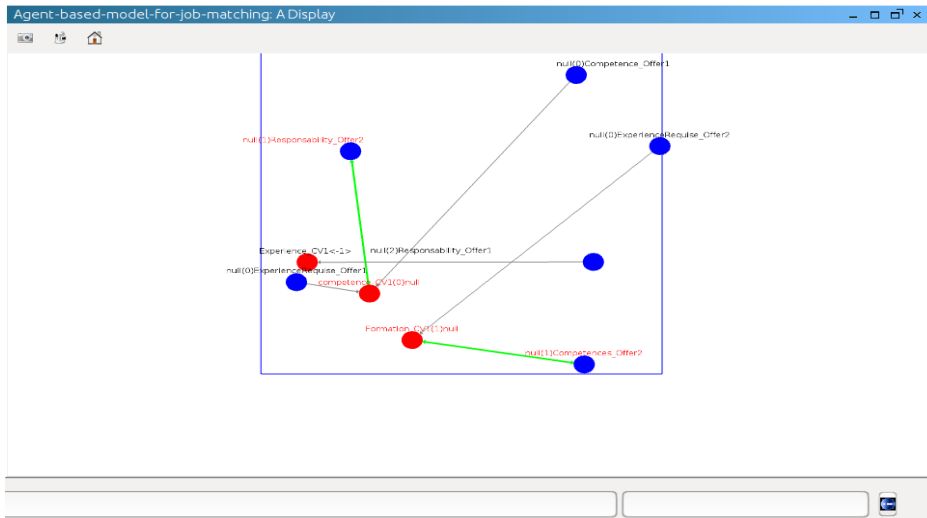


Figure 4.16 Panneau de visualisation à l'itération 1737

L'agent *Competence\_Offer1* qui n'avait pas encore opéré un choix, a choisi l'agent *Competence\_Cv1* comme appariement candidat. On remarque aussi qu'après une certaine itération, l'agent *ExperienceRequise\_Offer1* efface ses croyances concernant son premier choix d'appariement candidat pour choisir l'agent *Competence\_Cv1*, cette fois-ci.

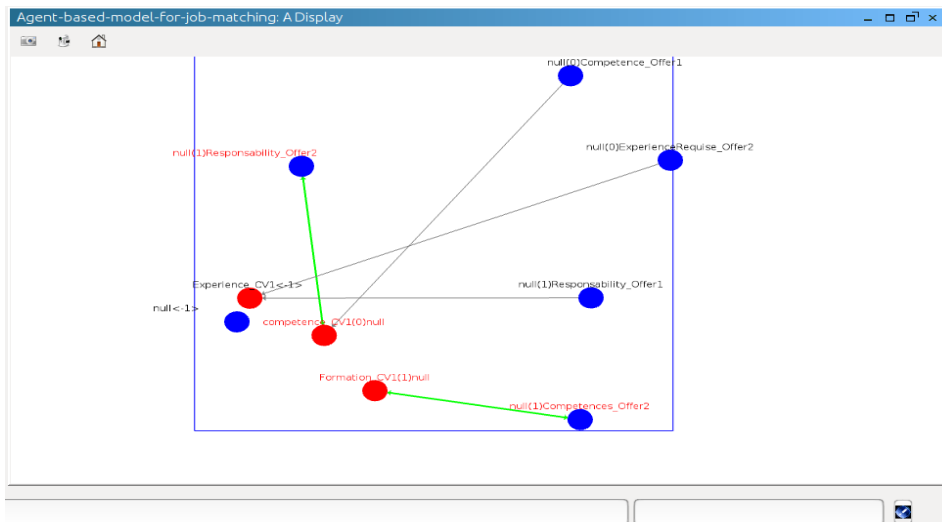


Figure 4.17 Panneau de visualisation à l'itération 4317

À l'itération 4317, nous remarquons que l'agent *ExperienceRequise\_Offer2* efface ses croyances concernant son premier appariement, pour finalement choisir *Experience\_Cv1* comme appariement candidat.

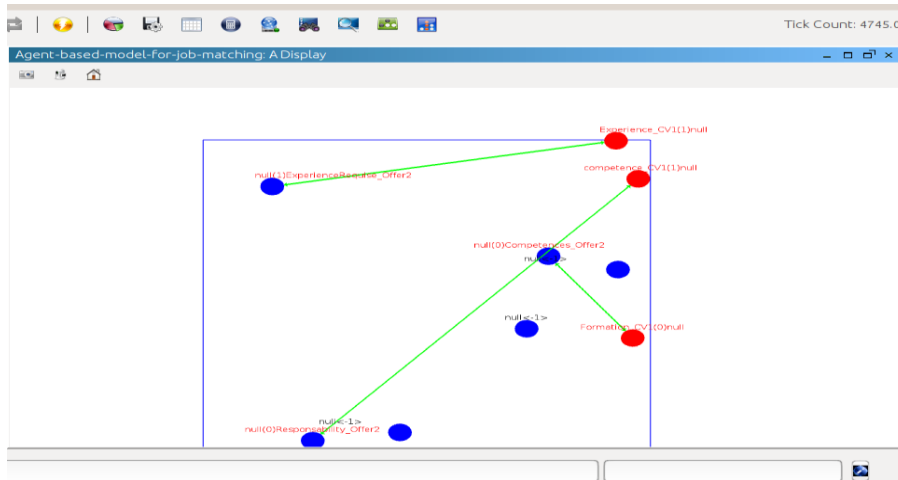


Figure 4.18 Panneau de visualisation à l'itération 4745

À l'itération 4745, les agents *Experience\_Cv1* et *Experience\_Offer2* se sont finalement choisis mutuellement pour pouvoir former un appariement consensuel stable. Ce choix constitue le dernier appariement consensuel, car on n'en aura plus d'autres jusqu'à la fin de la simulation, comme on peut le voir sur la figure suivante :

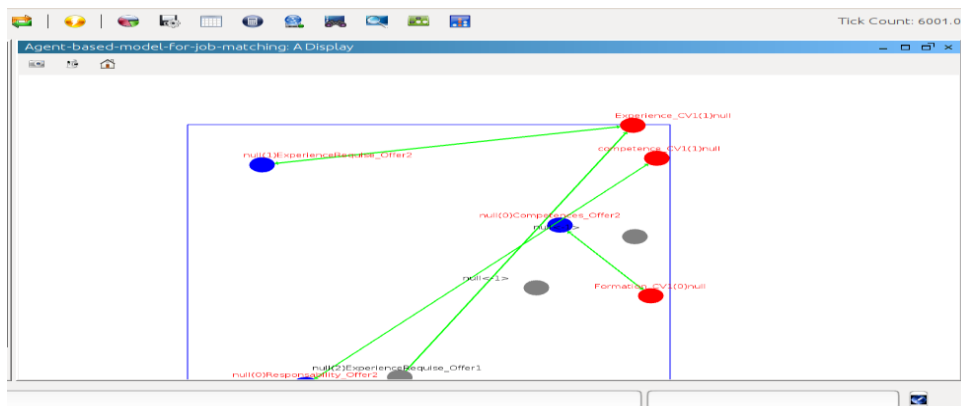
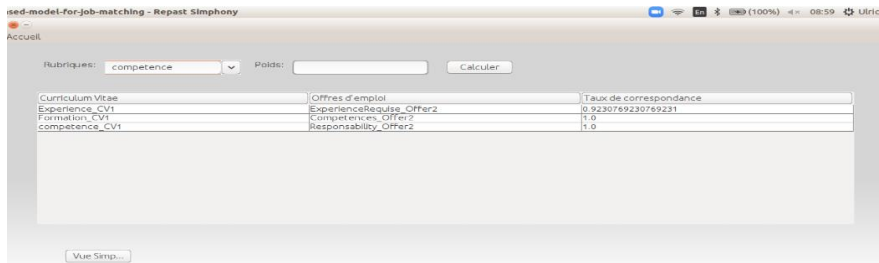


Figure 4.19 Panneau de visualisation à l'itération 6001

Nous devons noter que le nombre d'itération fixé à 6001 était nécessaire afin de trouver le résultat, car les agents devraient opérer plusieurs combinaisons pour trouver le chemin idéal. Cependant, ces 6001 itérations ont duré un peu moins de 5 secondes. Ci-dessous, nous avons la vue détaillée du résultat de l'appariement tel que présenté à un gestionnaire en Ressources Humaines.



Curriculum Vitae	Offres d'emploi	Taux de correspondance
Experience_CV1	ExperienceRequisse_Offer2	0.9230769230769231
Formation_CV1	Competences_Offer2	1.0
competence_CV1	Responsability_Offer2	1.0

Figure 4.20 Résultat final de la simulation

#### 4.3.2.2 Analyse des résultats de la simulation

Les agents sont parvenus à ce résultat final sans que nous n'ayons eu besoin de faire une quelconque optimisation, comme le réglage de certains paramètres importants, comme le seuil ou les poids. En outre, nous n'avons pas eu besoin non plus de choisir les fonctions d'agrégations, ni les mesures de similarité, que l'agent utilise afin d'avoir de meilleurs résultats. Ce point très important vient mettre en lumière le mécanisme de coévolution dont nous parlons dans les chapitres précédents. Ce mécanisme de coévolution, en effet, se traduit par le fait que compte tenu de la situation à laquelle il fait face, l'agent a pu faire sa propre configuration et sa propre optimisation, ce qui a conduit à l'auto-organisation et donc à aboutir au résultat final.

Par ailleurs, ce point si important vient appuyer notre hypothèse faite au départ selon laquelle il est possible de résoudre la problématique de l'appariement des CV aux offres d'emploi en se basant sur l'appariement des schémas. Contrairement aux différentes informations recueillies dans la littérature sur le défi que représente la configuration et l'optimisation, l'appariement automatique de schémas s'en sort plutôt bien. On vient de prouver expérimentalement, qu'on peut obtenir des résultats corrects, et ce, sans aucune préparation ou optimisation.

Repast Symphony nous permet d'avoir un panneau SQL. Dans notre cas, ce sera l'interface JoSQL grâce à la librairie JoSQL 86 (Bentley, s.d.), qui aura pour but de nous permettre d'interroger l'état interne des agents pour la découverte de certains attributs pertinents.

La figure 4.21 illustre l'interface JoSQL pour l'interrogation de l'état des agents, à un moment donné de la simulation.

Side	Name	Comment	NameSimMeas...	NameS...	CmtSimMe...	CmtSim...	AggrSc...	AggrFunction	Candid...	Consen...
SOURCE	competence_CV1		JaroSecondStri...		1	JaroSecond...	0,81944	1	MAX	Java, An...
SOURCE	Experience_CV1								MAX	UNMAT...
SOURCE	Formation_CV1		JaroSecondStri...		1	JaroSecond...	0,95455	0,97727	AVERAGE	Havard, ...
TARGET		Competence_Offer1							WEIGHTED	Havard, ...
TARGET		Responsability_Offer1	JaroSecondStri...	0,36338	JaroSecond...		0,625	0,44187	WEIGHTED	Oracle
TARGET		ExperienceRequise_Offer1	JaroSecondStri...	0,36944	JaroSecond...		0,625	0,49722	AVERAGE	Havard
TARGET		Competences_Offer2	JaroSecondStri...	0,62407	JaroSecond...		0,95455	0,78931	AVERAGE	Havard
TARGET		Responsability_Offer2	JaroSecondStri...	0,50196	JaroSecond...		0,53125	0,51666	AVERAGE	Java
TARGET		ExperienceRequise_Offer2	JaroSecondStri...	0,41349	JaroSecond...		0,53125	0,44882	WEIGHTED	Havard

Figure 4.21 Interface d'interrogation de l'état interne des agents lors d'une simulation

En analysant l'état interne à chaque pause de notre simulation, on a une meilleure idée sur l'évolution de la simulation. Ainsi, en regardant de plus près les différentes captures d'écran de l'état interne lors des différentes pauses de la simulation on se rend compte que :

Pour chaque agent, les mesures de similarités, les fonctions d'agrégation changent et ce lors de chaque pause. Les agents dès lors, empruntent des chemins différents pour pouvoir atteindre le consensus en parallèle et d'une façon autonome. La sélection aléatoire des mesures de similarités et des fonctions d'agrégation se produit non seulement pour les agents recherchant un appariement candidat, mais aussi pour ceux ayant déjà formé un appariement consensuel. Ainsi malgré qu'ils utilisent différentes voies, ils arrivent tout de même à former les correspondances attendues par l'utilisateur. Ci-dessous, à l'aide de nos différentes captures d'écran on peut clairement observer cet aspect totalement aléatoire dans le choix des mesures de similarités et des fonctions d'agrégation.

Les captures d'écran ci-dessous illustrent l'état interne, grâce au panneau JoSQL, des agents lors de chacune de nos pauses au cours de la précédente simulation.



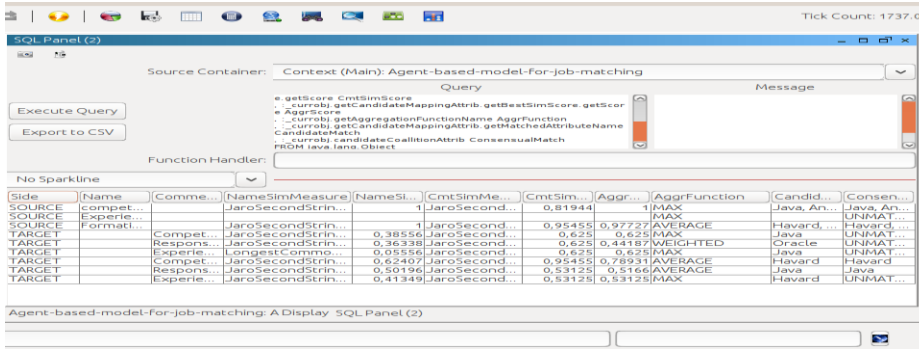


Figure 4.22 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 1737

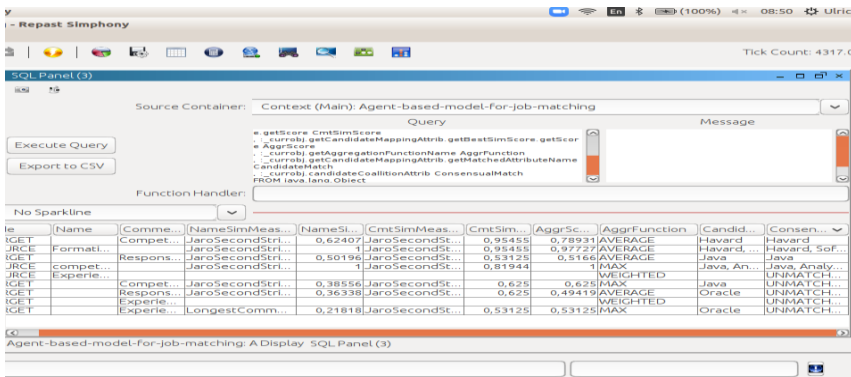


Figure 4.23 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 4317

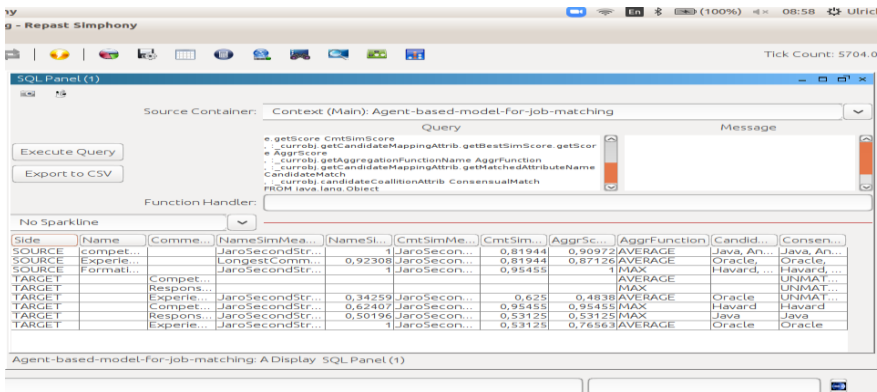


Figure 4.24 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 5704

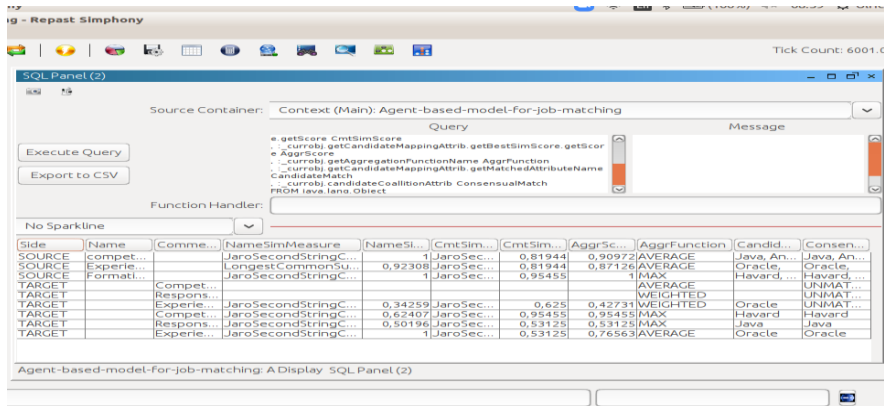


Figure 4.25 Interface JoSQL pour l'interrogation de l'état interne des agents à l'itération 6001

On peut voir au niveau des colonnes *NameSimilarityMeasure* et *AggregationFunction*, que les mesures de similarités et les fonctions d'agrégation changent, à chaque itération. L'agent essaie ainsi de déterminer par lui-même, les meilleurs paramètres, et teste toutes les mesures au sein du pool de similarités pour au final choisir les meilleures similitudes.

Aussi, l'analyse des premières expérimentations effectuées sur ce prototype, révèle que le prototype non seulement arrive à trouver les résultats attendus par l'utilisateur, mais il arrive aussi à produire ces résultats en affichant les concepts tels que l'autonomie, la décentralisation, la stochasticité, etc. Par rapport à ces expérimentations réalisées et en attendant des simulations plus poussées sur notre prototype, nous pouvons commencer par confirmer, que l'idée qui consiste à dire que l'appariement des CV aux OE, peut être résolu et modélisé via une simulation multi-agents. C'est donc une idée novatrice et viable.

#### 4.3.2.3 Simulations multiples

Bien que notre prototype arrive à trouver toutes les correspondances attendues par l'utilisateur, celui-ci n'arrive pas à avoir le même résultat à chaque exécution. Malgré la place importante de l'aléatoire dans notre modèle, et vu sa nature non-déterministe les premiers résultats sont vraiment encourageants et le pourcentage de bonnes correspondances frôle les 100%. On s'aperçoit qu'avec les mêmes scénarios d'appariement, on arrive à avoir les correspondances attendues et ce avec différentes trajectoires

empruntées. Par contre, dans certains scénarios, le prototype peine à trouver la totalité des correspondances attendues, ce qui met en relief le caractère non-prévisible du prototype.

Nous nous sommes tournés vers les simulations multiples et l'analyse statistique car nous pensons qu'ils pourraient nous aider à effectuer un pas de plus vers la réduction de l'incertitude, en utilisant la « loi des grands nombres ». Cette réduction n'est malheureusement pas possible à travers des exécutions individuelles et isolées de la simulation. Cependant, nos expérimentations faites sur des simulations individuelles, nous ont permis de découvrir les spécificités, telles que la stochasticité, l'adaptation pour le calcul de similarité, l'auto-organisation, ainsi que l'émergence de l'alignement qui sont liées à notre modèle conceptuel, et nous avons eu à les évoquer notamment dans les chapitres précédents.

Pour les simulations multiples, nous aurons toujours recours à notre outil de base Repast Symphony qui nous offre une pléiade d'outils et de fonctionnalités. Non seulement, Repast Symphony nous permet de réaliser des simulations individuelles, comme c'était le cas dans la section précédente, mais il nous permet aussi de réaliser des exécutions en lot. Cependant, bien que la visualisation du déroulement de cette dernière ne soit pas possible on peut toujours avoir tous les détails sous la forme d'un fichier texte qu'il aura généré.

#### 4.4 Conclusion

Dans ce chapitre, nous avons présenté en détail l'implémentation du prototype pour l'appariement des CV aux OE. Nous avons aussi mis un accent sur l'architecture interne de l'agent réflexif pour l'appariement. Pour terminer nous avons analysé les interactions des éléments du système d'appariement au cours du processus de simulation, ce qui permet de mesurer la performance des différents apparieurs pour les différents scénarios d'appariement. Dans le prochain chapitre, il sera question principalement de présenter nos expérimentations et discuter de la validation des résultats issus de nos expérimentations.

## CHAPITRE 5

### EXPÉRIMENTATIONS ET DISCUSSIONS DES RÉSULTATS DU PROTOTYPE

#### 5.1 Introduction

Dans cette section, nous présentons les résultats de notre processus d'appariement décrit dans notre exemple de référence. Nous allons, en outre, discuter de la validation, tant pour les simulations uniques que pour les simulations multiples.

La validation va s'appuyer principalement sur les résultats logiques que notre outil aura retenus. Autrement dit, pour chaque offre d'emploi, nous avons déterminé le nombre de candidats communs entre ceux retenus par notre outil et ceux qu'un humain aurait sélectionné d'une façon logique. Nous avons aussi calculé le temps utilisé par notre outil pour sélectionner des candidats.

#### 5.2 Les expérimentations

Avant de commencer, il convient d'expliquer comment nous avons défini le processus.

Notre prototype, avant de donner une réponse sur la base d'une requête, va entamer ce que l'on appelle l'évaluation de la proximité sémantique entre chacun des attributs (ici *name* et *comment*) des rubriques de l'offre d'emploi et les attributs des rubriques correspondants aux CV des candidats. Cette évaluation de la correspondance des différentes rubriques vise à déterminer, par exemple, les points communs entre les compétences techniques requises pour le poste versus les compétences techniques acquises des candidats, les formations requises versus les formations suivies par les candidats, les expériences requises versus les expériences acquises, etc. Ainsi, pour comparer deux candidats, notre prototype devra regarder l'attribut de spécialisation requis pour le poste et ceux des acquis des candidats. En fonction des mesures de similarités dans le pool de similarité il peut voir, lequel des candidats est sémantiquement plus proche de l'attribut des spécialisations requises pour le poste en question.

La somme des scores attribués à chaque attribut d'élément évalué donnera finalement le score total des acquis d'un candidat par rapport aux requis du poste pour lequel la requête fut émise, et vice versa.

Aussi, il est important de noter que nous avons pris le soin de laisser finalement la décision finale revenir à l'humain car il revient au professionnel RH de moduler l'importance de chacun des attributs dans un CV ou dans une offre d'emploi pour le calcul du score total. Pour ce faire, via notre interface utilisateur, le recruteur aura la possibilité de configurer le poids relatif qu'il désire que chaque attribut d'un CV ait. Ce poids sera le facteur par lequel notre prototype multipliera le score de chaque attribut des éléments comparés, leur donnant ainsi différentes importances dans la sommation du score total.

Puis, en dernière instance, le prototype ordonnera, dans un tableau, les candidats en fonction du score total de chacun. Il sélectionnera les meilleurs par rapport aux offres d'emploi disponibles.

Pour tester notre prototype, nous avons réalisé plusieurs simulations en incrémentant à chaque fois le nombre de CV et d'offres d'emplois. Ainsi, chaque nouvelle simulation sera un scénario de différentes tailles avec une hétérogénéité lexicale plus élevée que le précédent ce qui fait que nos expériences ont été menées de façon graduelle.

Notre premier scénario est un scénario de petite taille qui comporte en entrée 4 curriculum vitae et deux offres d'emploi. Chaque curriculum vitae a 2 rubriques à savoir: "*Competence*", "*Eperience*" qui constitueront les agents sources et chaque offre d'emploi a également 2 rubriques à savoir: "*Competence\_offer*", "*Responsability\_offer*" qui seront les agents cibles. Il est important de noter que chaque agent a deux attributs "*name*" et "*comment*" et que l'hétérogénéité lexicale ici peut être considéré comme faible.

Dans le deuxième scénario on a décidé de monter légèrement en taille, ainsi on aura en entrée 10 curriculum vitae pour 5 offres d'emploi. On maintiendra les deux rubriques initiales pour les CV à savoir: "*Competence*", "*Eperience*", qui seront les agents sources. Ensuite, on gardera aussi les deux rubriques initiales des offres d'emploi à savoir: "*Competence\_offer*", "*Responsability\_offer*" qui constitueront les agents cibles. On doit noter ici que le niveau d'hétérogénéité lexicale peut être considéré comme moyen.

Pour notre troisième scénario, on va encore accroître la taille de nos données en entrée. Autrement dit, cette fois-ci on aura 30 CV et 15 OE. Le nombre de rubriques des curriculum vitae passe à 3 qui sont: "*Competence*", "*Eperience*" et "*Formation*". Le nombre de rubriques des offres d'emploi passera aussi à 3 rubriques à savoir: "*Competence\_offer*", "*Responsability\_offer*" et

“*ExperienceRequired\_offer*”. Comme dans les cas précédents, chaque agent conserve ses deux attributs “*name*” et “*comment*”. Dans ce scénario on a ajouté plusieurs offres d'emploi similaires et on s'attend donc que notre prototype puisse pouvoir gérer ce cas de figure correctement.

Le quatrième scénario aura 50 CV en entrée pour 30 OE. Le nombre de rubriques des curriculum vitae passe à 4 qui sont: “*Competence*”, “*Eperience*” “*Formation*” et “*Description*”. Le nombre de rubriques des offres d'emploi, lui, est maintenu à 3. Aussi, dans ce scénario, on a des candidats qui évoluent dans le même domaine mais avec de légères spécificités ainsi que des offres d'emploi des entreprises qui opèrent dans un même secteur d'activité mais dont les besoins sont différents.

Pour le cheminement de nos expérimentations, on s'est inspiré des travaux de (Assoudi 2017). Ainsi, nous aurons une série de 4 méta-simulations et chaque méta-simulation aura 5 simulations individuelles. En raison du grand nombre d'agents en entrée, on veut s'assurer que les résultats trouvés par les agents ne sont pas trouvés juste pour une simulation mais le sont pour chacune des simulations. Autrement dit, on veut vérifier la répétition des résultats dans le temps. Une analyse statistique sera faite sur le résultat final de l'appariement d'un scénario pour déterminer la fréquence d'apparition d'une correspondance sur les 5 simulations.

#### 5.2.1 Configuration expérimentale

Nos expérimentations ont été faites sur un ordinateur portable ayant un micro-processeur multi-cœur plus exactement 5 cœurs avec 64 Go de mémoire.

#### 5.2.2 Métriques d'évaluation

D'après les auteurs (Bellahsene *et al.*, 2011), l'évaluation de l'ensemble des correspondances trouvées par les systèmes d'appariement automatique de schémas se fait généralement en se basant sur les métriques d'évaluation de performance comme la précision, le rappel et la F-mesure Ils permettent disent-ils d'évaluer la performance par rapport aux correspondances attendues. Pour ce faire, comme dans le domaine de l'extraction de l'information ou avec l'évaluation des classificateurs, on aura besoin de quatre nombres importants à savoir :

- Le nombre de toutes les correspondances correctes reconnues comme correctes par notre outil, noté : VP (Vrais Positifs)

- Le nombre de toutes les correspondances incorrectes reconnues comme correctes par notre outil, noté : FP (Faux Positifs)
- Le nombre de toutes les correspondances correctes reconnues ou considérées comme incorrectes prenant en compte les correspondances non trouvées par notre outil. Il est noté : FN (Faux Négatifs)
- Le nombre de toutes les correspondances incorrectes reconnues comme incorrectes par notre outil. Il est noté VN (Vrais Négatifs).

Nous avons remarqué dans la littérature que les mesures de performance les plus connues dans le domaine d'appariement de schémas sont : la précision, le rappel et la F-mesure.

Ces mesures sont notamment utilisées dans de récents travaux, comme ceux ci-dessous :

- Les travaux de (Zhang, Shin et al. 2021) qui utilisent un modèle basé sur les techniques du traitement de langage naturel pour obtenir des correspondances sémantiques entre les schémas sources et les schémas cibles.
- Ceux des auteurs (Shraga, Gal et al. 2020) qui proposent une nouvelle étape de post-traitement pour l'appariement de schémas qui vise à améliorer le résultat final d'appariement sans intervention humaine.
- Les travaux de (Nguyen, Weidlich et al. 2019) qui développent un modèle probabiliste pour identifier les correspondances les plus incertaines afin de guider le travail de validation de l'expert.

Cette liste n'étant bien évidemment pas exhaustive, notre but est de montrer la pertinence de ces mesures et de prouver qu'elles sont toujours d'actualités, afin de l'adopter pour l'évaluation de notre prototype.

#### 5.2.2.1 La précision

La précision est la proportion des items pertinents parmi l'ensemble des items proposés (Wikipédia, 2021). Autrement dit, c'est une mesure qui détermine par calcul la proportion des correspondances correctes trouvées par rapport à l'ensemble des correspondances trouvées par notre outil. Par rapport aux nombres

étudiés plus hauts, on peut dire que c'est le rapport des Vrais Positifs sur la somme des Vrais Positifs et des Faux Positifs.

$$\text{Équation 5.1 Formule de la précision } \mathbf{Précision} = \frac{VP}{VP+FP}$$

#### 5.2.2.2 Le rappel

Le rappel quant à lui est la proportion des items pertinents proposés parmi l'ensemble des items pertinents (Wikipédia, 2021). En d'autres mots, cette mesure calcule la proportion des correspondances correctes trouvées par le prototype par rapport à la somme des correspondances correctes trouvées et des correspondances correctes non trouvées par le prototype. De façon plus concise, c'est le rapport des Vrais Positifs sur la somme des Vrais Positifs et des Faux Négatifs.

$$\text{Équation 5.2 Formule du Rappel } \mathbf{Rappel} = \frac{VP}{VP+FN}$$

#### 5.2.2.3 La F-mesure

La F-mesure est définie comme la moyenne harmonique de la précision et du rappel (Wikipédia, 2021). C'est le rapport du produit de la précision et du rappel sur la somme de la précision et du rappel, le tout multiplié par 2. Elle est également nommée F-score.

Une F-mesure dont la valeur serait proche de l'indique généralement une très bonne performance d'appariement.

$$\text{Équation 5.3 Formule de la F-mesure } \mathbf{F - mesure} = 2 \times \frac{\text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}}$$

Pour notre expérimentation, nous allons déterminer les mesures d'évaluation que nous venons de voir plus haut à savoir : la précision, le rappel et la F-mesure afin d'évaluer la performance de chaque simulation individuelle et de pouvoir comparer les correspondances obtenues par rapport à celles attendues par l'utilisateur.

Le même procédé est appliqué aux simulations multiples où en amont, nous calculons la fréquence d'apparition d'une correspondance sur les 15 simulations qui les composent. Ensuite, nous pouvons enchaîner avec le calcul des mesures d'évaluations pour évaluer la performance de chaque méta-simulation et la comparaison des correspondances obtenues à celles attendues par l'utilisateur. Il est important de rappeler que cette méthode d'évaluation est inspirée des travaux de (Assoudi 2017).

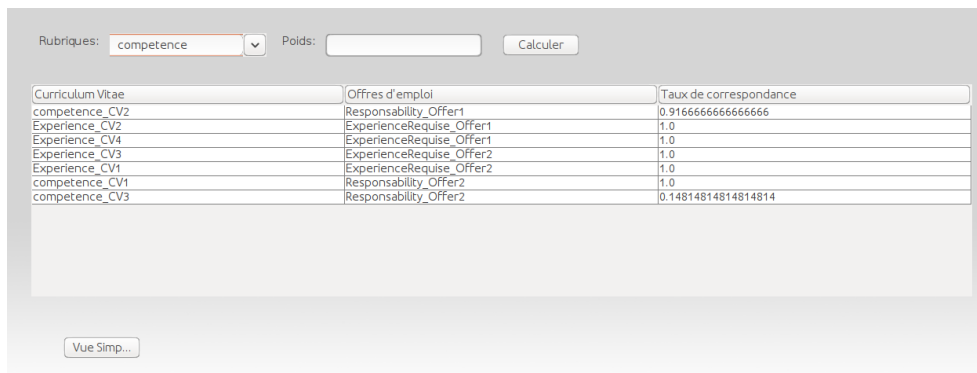


### 5.2.3 Résultats des expérimentations

Puisque la simulation multiple ne nous permet pas de voir le résultat de la simulation sur le panneau de simulation, nous avons stocké les données dans un fichier texte de sortie. Pour ce faire la présentation de nos résultats se fera par scénario afin d'évaluer les paramètres, les comportements de nos agents et de notre prototype en général.

#### ❖ Premier Scénario

Avant de passer à la simulation multiple pour l'analyse statistique, on a voulu étudier le fonctionnement d'une simulation. Ainsi, notre premier scénario prend en compte 4 curriculum vitae et deux offres d'emplois. La simulation nous a permis de voir que le prototype a réalisé tous les appariements que feraient un humain ou un professionnel des Ressources Humaines. Ainsi les agents ont réussi à former des appariements consensuels auxquels on s'attendait à l'exception de l'agent *Compétence\_Cv4* qui n'a pas réussi à trouver d'appariement comme on peut s'en apercevoir sur la figure 5.1 et la figure 5.2



Curriculum Vitae	Offres d'emploi	Taux de correspondance
competence_CV2	Responsability_Offer1	0.9166666666666666
Experience_CV2	ExperienceRequisse_Offer1	1.0
Experience_CV4	ExperienceRequisse_Offer1	1.0
Experience_CV3	ExperienceRequisse_Offer2	1.0
Experience_CV1	ExperienceRequisse_Offer2	1.0
competence_CV1	Responsability_Offer2	1.0
competence_CV3	Responsability_Offer2	0.14814814814814814

Figure 5.1 Interface utilisateur montrant le résultat final du premier scénario

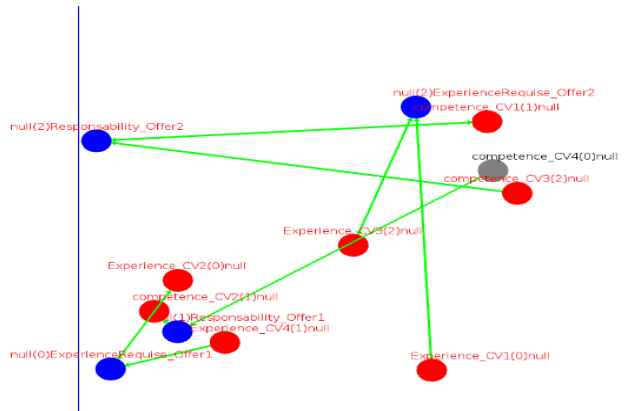


Figure 5.2 Panneau de visualisation au terme de la simulation du premier scénario

En regardant de plus près la figure 5.1 on se rend compte que le nombre d'appariements affichés est équivalent à la somme des éléments des deux schémas. Autrement dit, un curriculum est choisi comme le plus pertinent pour une offre d'emploi lorsqu'au moins un ou tous ses agents (ici nommés rubriques) s'apparient à au moins un ou à tous les agents (rubriques) de l'offre d'emploi en question. Le degré de pertinence ou le score final d'un appariement CV-OE est calculé en divisant la somme des taux de correspondances des différentes rubriques par la somme des rubriques.

Ainsi, on peut résumer le résultat dans ce tableau :

Curriculum Vitae	Offres d'emploi	Degré de pertinence
CV_2	OffreEmploi_1	0.95
CV_4	OffreEmpoi_1	0.50
CV_3	OffreEmploi_2	0.74
CV_1	OffreEmploi_2	1

Tableau 5.1 Tableau simplifié des résultats du scénario 1

Dans ce scénario, on a remarqué que deux profils (CV) candidats ont été jugés pertinents par l'outil pour chacune des deux offres d'emploi mais avec des scores différents. Le but du prototype qui était de

sélectionner les candidats pertinents afin de pouvoir laisser la décision de sélection finale à l'utilisateur est parfaitement bien rempli. Dans notre cas précis, le gestionnaire verra que le candidat numéro 1 est mieux classé par l'outil et donc plus pertinent que le candidat numéro 3 pour l'offre d'emploi numéro 2. De même, pour l'offre d'emploi numéro 1, le candidat numéro 2 est jugé plus pertinent, car ayant un meilleur score, par l'outil que le candidat numéro 4. Il aura alors la possibilité de prendre sa décision finale, lors d'une éventuelle entrevue.

Passons maintenant en revue les résultats compilés de l'appariement des agents des 4 méta-simulations dans le tableau ci-dessous :

Paramètres Méta-Sim	Nombre Sim-d'itérations	Corresp. attendues	Bonnes Corresp. Trouvées(VP)	Fausse Corresp. Trouvées(FP)	Corresp. Non trouvées (FN)	Rappel	Précision	F- mesure
<b>1</b>	1230	8	5	0	3	62%	100%	76%
<b>2</b>	2450	8	6	0	2	75%	100%	86%
<b>3</b>	3895	8	6	1	1	75%	86%	80%
<b>4</b>	4142	8	7	0	1	87%	100%	93%
<b>5</b>	6001	8	8	0	0	100%	100%	100%
<b>Méta-sim</b>	3543.6	8	6.4	0.2	1.4	79%	97%	87%
<b>1</b>	904	8	4	1	3	50%	80%	61%
<b>2</b>	597	8	4	2	2	50%	66%	56%
<b>3</b>	333	8	4	3	1	50%	57%	53%
<b>4</b>	5080	8	7	1	0	87%	87%	87%
<b>5</b>	7344	8	8	0	0	100%	100%	100%
<b>Méta-sim 2</b>	2851.6	8	5.4	1.4	1.2	70%	80%	74%
<b>1</b>	6543	8	8	0	0	100%	100%	100%
<b>2</b>	5983	8	7	1	0	87%	87%	87%
<b>3</b>	3567	8	6	0	2	75%	100%	86%
<b>4</b>	4395	8	7	0	1	87%	100%	93%
<b>5</b>	2759	8	5	0	3	62%	100%	76%
<b>Méta-sim 3</b>	4649.4	8	6.6	0.2	1.2	82%	97%	88%

1	7495	8	8	0	0	100%	100%	100%
2	4759	8	8	0	0	100%	100%	100%
3	3659	8	6	1	1	75%	86%	80%
4	5606	8	8	0	0	100%	100%	100%
5	2924	8	7	0	1	87%	100%	93%
<b>Méta-sim 4</b>	4888.6	8	7.4	0.2	0.4	92%	97%	94%

Tableau 5.2 Tableau de contingence des simulations multiples pour le premier scénario

Étant donné que le but de ce mémoire est d'évaluer la capacité de notre outil à trouver les CV pertinents aux offres d'emploi nous nous sommes aussi penchés sur l'analyse statistique de l'appariement des agents (ici rubriques) de ces CV aux OE disponibles pour la simulation. Il est important de rappeler notre notion de pertinence sur laquelle on s'est basé pour la modélisation du prototype.

- Pour qu'un CV soit considéré comme pertinent pour une offre d'emploi il faut qu'au moins une rubrique de ce CV s'apparie avec au moins une rubrique de cette offre d'emploi.
- Le degré de pertinence est calculé par rapport au score similarité de chaque appariement. Autrement dit, le degré de pertinence d'un  $CV_j$  ayant  $i$  rubriques est le rapport entre la somme de tous les scores de similarités issus des appariements de chaque rubrique du  $CV_j$  sur la somme de toutes les rubriques du  $CV_j$ . Si  $x_i$  est le score d'appariement de la rubrique  $i$  du  $CV_j$  et que  $r_j$  est le nombre de rubriques du  $CV_j$  alors on peut déduire la formule ci-dessous :

Équation 5.4 Formule du degré de pertinence **degré de pertinence** =  $\frac{\sum_{i=1}^n x_i}{r_j}$

L'analyse du tableau 5.2 nous montre qu'à part la méta-simulation 2, notre outil a réussi à trouver presque toutes les bonnes correspondances. On en veut pour preuve les valeurs assez élevées de nos mesures de performance comme le Rappel, la Précision et la F-mesure. Nous voulons aussi mettre en lumière un point essentiel, la colonne des données des méta-simulations qui est ici une moyenne des données issues de nos cinq simulations individuelles constituant la méta-simulation. Somme toute, avec ce scénario, on peut

dire que notre outil arrive à très bien performé car étant capable de trouver plus de 95% des résultats attendus par l'utilisateur.

Paramètres Sim- Méta- Sim	Nombre Moyen d'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F-mesure
Méta-sim 1	3543.6	4	3	0.2	0.8	75%	93%	83%
Méta-sim 2	2851.6	4	2.2	0.8	1	55%	73%	62%
Méta-sim 3	4649.4	4	3.6	0	0.4	90%	100%	94%
Méta-sim 4	4888.6	4	4	0	0	100%	100%	100%

Tableau 5.3 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 1

Le tableau 5.3 représente l'analyse statistique faite sur l'appariement des CV aux OE. Ici on veut savoir, pour chaque méta-simulation, le nombre de candidats pertinents qui a été trouvé par rapport au nombre de candidats pertinents attendus. Il s'agit encore une fois d'une agrégation des résultats issus des 5 simulations individuelle. Nous voulons donc mesurer le taux d'erreur et évaluer ainsi les résultats avec les mêmes mesures de performance utilisée plus haut à savoir : la Précision, le Rappel et la F-mesure.

En examinant les résultats de ce tableau, on se rend compte que notre outil performe bien par rapport à la durée des simulations. Il est important de ne pas perdre de vue que nous voulons évaluer sa capacité à trouver rapidement des candidats qui ont un lien avec une offre d'emploi donnée et donc qui pourrait être potentiellement éligible pour le poste rattaché à l'offre. Aussi, nous remarquons que la performance des agents individuels peut avoir une influence ou un impact sur la performance global pour le choix du meilleur candidat. Ainsi, si nous avons un CV qui possède une seule rubrique et que la correspondance de cette dernière est considérée comme un faux positif cela va directement agir sur la décision de l'outil de considérer ce CV comme pertinent (à tort) pour l'offre d'emploi en question puisqu'ayant au moins une rubrique s'appariant à au moins une rubrique de l'offre d'emploi. De plus, le choix de cette rubrique

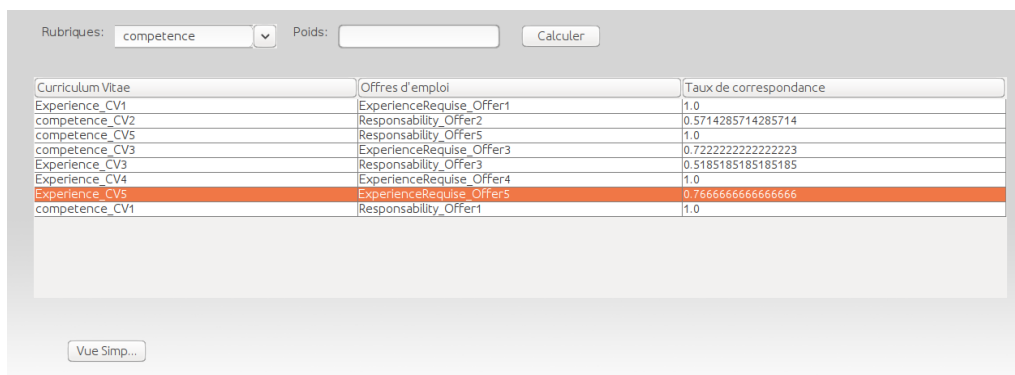
comme faux négatif peut aussi conduire l'outil à une erreur car il jugerait un candidat non pertinent pour une offre d'emploi alors qu'il l'est, en réalité.

Ceci nous amène à une autre remarque par rapport au fait que dans certains cas, l'appariement d'une rubrique peut être considéré comme un faux négatif sans que cela n'influence la décision globale de l'outil de considérer quand même le candidat comme pertinent. Rappelons-le, il suffit qu'au moins une seule rubrique s'apparie à au moins une rubrique de l'offre d'emploi pour que le CV du candidat soit choisi comme pertinent. Le but poursuivi est de n'exclure aucun candidat qui aura dans son CV une rubrique pertinente pour pouvoir permettre au gestionnaire des ressources humaines d'avoir la décision finale dans le processus d'embauche.

#### ❖ Deuxième scénario

Notre deuxième scénario aura une plus grande taille. Nous augmentons le nombre de CV à 10 avec chacun deux rubriques et le nombre d'offres d'emploi à 5 avec pour chacune deux rubriques aussi.

Dans un premier temps et tout comme dans le premier scénario, nous avons effectué une seule simulation individuelle et nous remarquons, que l'outil est parvenu à trouver les correspondances auxquelles on s'attendait.



Curriculum Vitae	Offres d'emploi	Taux de correspondance
Experience_CV1	ExperienceRequise_Offer1	1.0
competence_CV2	Responsability_Offer2	0.5714285714285714
competence_CV5	Responsability_Offer5	1.0
competence_CV3	ExperienceRequise_Offer3	0.7222222222222223
Experience_CV3	Responsability_Offer3	0.5185185185185185
Experience_CV4	ExperienceRequise_Offer4	1.0
Experience_CV5	ExperienceRequise_Offer5	0.7666666666666666
competence_CV1	Responsability_Offer1	1.0

Figure 5.3 Interface utilisateur montrant le résultat final du deuxième scénario

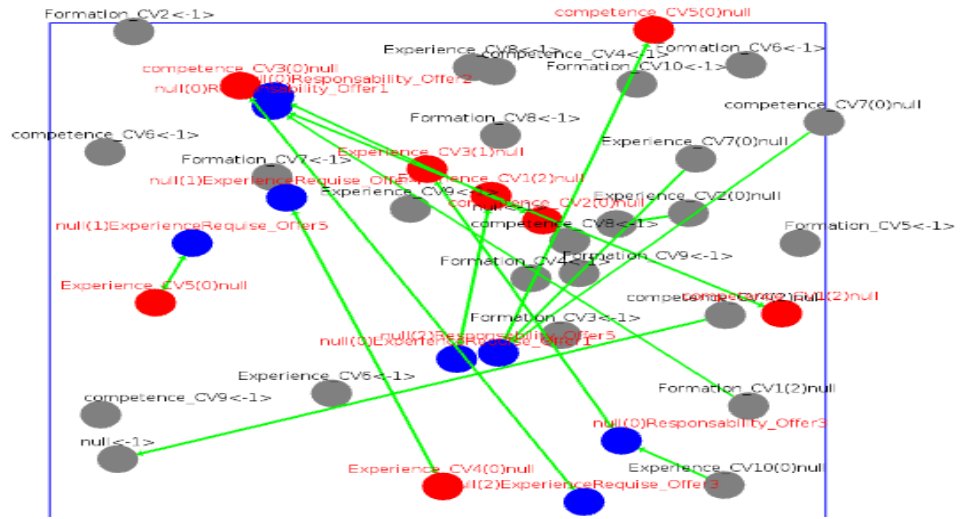


Figure 5.4 Panneau de visualisation au terme de la simulation du deuxième scénario

Les résultats sont donc résumés dans le tableau simplifié ci-dessous :

Curriculum Vitae	Offres d'emploi	Degré de pertinence
CV_1	OffreEmploi_1	0.95
CV_2	OffreEmpoi_2	0.50
CV_3	OffreEmploi_3	0.65
CV_4	OffreEmploi_4	0.5
CV_5	OffreEmploi_5	0.85

Tableau 5.4 Tableau simplifié des résultats du scénario 2

Pour voir la fréquence des résultats, nous passons à la simulation multiple et nous compilons dans un tableau, l'ensemble des résultats tout en calculant les mesures de performance.

Paramètres- Simd	Nombre d'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F- mesure
<b>1</b>	6840	20	19	0	1	95%	100%	97%
<b>2</b>	5094	20	18	0	2	90%	100%	94%
<b>3</b>	7474	20	20	0	0	100%	100%	100%
<b>4</b>	8595	20	20	0	0	100%	100%	100%
<b>5</b>	6440	20	19	1	0	95%	95%	95%
<b>Méta-sim 1</b>	6888.6	20	19.2	0.2	0.6	96%	99%	97%
<b>1</b>	7533	20	20	0	0	100%	100%	100%
<b>2</b>	6543	20	19	1	0	95%	95%	95%
<b>3</b>	7103	20	20	0	0	100%	100%	100%
<b>4</b>	5094	20	17	3	0	85%	85%	85%
<b>5</b>	7500	20	19	0	1	95%	100%	97%
<b>Méta-sim 2</b>	6754.6	20	19	0.8	0.2	95%	96%	95%
<b>1</b>	7986	20	17	0	3	85%	100%	91%
<b>2</b>	8322	20	20	0	0	100%	100%	100%
<b>3</b>	6456	20	19	0	1	95%	100%	97%
<b>4</b>	4586	20	16	4	0	80%	80%	80%
<b>5</b>	9988	20	20	0	0	100%	100%	100%
<b>Méta-sim 3</b>	7467.6	20	18.4	0.8	0.8	92%	96%	93.6%
<b>1</b>	9790	20	20	0	0	100%	100%	100%
<b>2</b>	7893	20	19	0	1	95%	100%	97%
<b>3</b>	8443	20	20	0	0	100%	100%	100%
<b>4</b>	9446	20	20	0	0	100%	100%	100%
<b>5</b>	6959	20	18	0	2	90%	100%	94%
<b>Méta-sim 4</b>	8506.2	20	19.4	0	0.6	97%	100%	98%

Tableau 5.5 Tableau de contingence des simulations multiples pour le deuxième scénario



L'analyse des résultats compilés des méta-simulations issus du tableau 5.5 nous montre qu'encore une fois malgré la légère augmentation de la taille des éléments et donc des agents, notre prototype arrive bien à trouver, dans la majorité du temps, toutes les correspondances auxquelles l'utilisateur s'attendait. On doit noter que le résultat n'est pas parfait car il y'a quelques correspondances incorrectes comme on peut le voir au niveau de la méta-simulation 1, 2 et 3. Nous avons aussi remarqué que plus la taille augmente plus le nombre d'itérations nécessaire à l'obtention de bons résultats augmente. Contrairement au premier scénario, notre outil pour pouvoir trouver toutes les bonnes correspondances a eu besoin que le nombre de simulations soit supérieur à 5000.

Paramètres -Sim	Nombre Moyen d'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F- mesure
<b>Méta-sim 1</b>	6888.6	10	7.6	1	1.4	76%	88%	81%
<b>Méta-sim 2</b>	6754.6	10	7.2	1.8	1	72%	80%	75%
<b>Méta-sim 3</b>	7467.6	10	8	1.2	0.8	80%	86%	82%
<b>Méta-sim 4</b>	8506.2	10	9.5	0	0.5	95%	100%	97%

Tableau 5.6 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 2

Le tableau 5.6 nous montre les résultats pour les correspondances candidats-offres et on remarque que les résultats obtenus sont relativement bons en ce sens que malgré quelques mauvaises correspondances on obtient des scores de précision au-dessus de 80% en moyenne. On constate aussi que la moyenne de la F-mesure hormis la deuxième méta-simulation est aussi supérieure à 80% ce qui constitue un bon taux. Il est important de noter aussi que notre outil parvient à ces résultats avec moins de 9000 itérations ce qui correspond en réalité à environ 50-60 secondes. Notre prototype est donc capable de bien performer dans un temps relativement court.

#### ❖ Troisième scénario

Notre troisième scénario prend en compte 30 CV avec 3 rubriques chacun et 15 offres aussi avec 3 rubriques chacune. La taille de ce scénario est considérée comme grande. Nous avons compilé les résultats de ce scénario dans le tableau ci-dessous :

Paramètres	Nombre Simd'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F- mesure
<b>1</b>	10001	90	82	0	8	91%	100%	95%
<b>2</b>	11474	90	88	2	0	97%	97%	97%
<b>3</b>	10474	90	84	0	6	93%	100%	96%
<b>4</b>	9473	90	80	5	5	88%	94%	89%
<b>5</b>	8344	90	78	5	7	86%	94%	89%
<b>Méta-sim 1</b>	9953.2	90	82.4	2.4	5.2	91%	97%	93%
<b>1</b>	8533	90	78	2	10	86%	97%	91%
<b>2</b>	9463	90	88	0	2	97%	100%	98%
<b>3</b>	10575	90	85	0	5	94%	100%	96%
<b>4</b>	11584	90	90	0	0	100%	100%	100%
<b>5</b>	11043	90	87	3	0	97%	97%	97%
<b>Méta-sim 2</b>	10 239.6	90	85.6	1	3.4	94%	98%	96%
<b>1</b>	11595	90	90	0	0	100%	100%	100%
<b>2</b>	10475	90	78	8	4	86%	91%	88%
<b>3</b>	11175	90	90	0	0	100%	100%	100%
<b>4</b>	8564	90	74	6	10	82%	92%	86%
<b>5</b>	9475	90	79	4	7	87%	95%	90%
<b>Méta-sim 3</b>	10 256.8	90	82.2	3.6	4.2	91%	95%	92%
<b>1</b>	8885	90	70	2	18	78%	97%	86%
<b>2</b>	9565	90	78	0	12	86%	100%	92%
<b>3</b>	11113	90	90	0	0	100%	100%	100%
<b>4</b>	10005	90	86	4	0	95%	95%	95%
<b>5</b>	9898	90	80	5	5	88%	94%	90%
<b>Méta-sim 4</b>	9893.2	90	80.8	2.2	7	89%	97%	92%

Tableau 5.7 Tableau de contingence des simulations multiples pour le troisième scénario

En analysant le tableau 5.7, on remarque que malgré la grande taille de l'échantillon de nos schémas, le prototype réussi à trouver quasiment toutes les correspondances attendues. Par ailleurs, on remarque

aussi que contrairement aux deux précédents scénarios on a beaucoup plus de correspondances qui n'ont pas été trouvées. Ceci peut être dû au fait que nous avons voulu maintenir le nombre d'itération en-dessous de 12000 pour voir si la durée de la simulation pouvait influencer la performance. On voit donc que oui, malheureusement, la durée maximale à porter préjudice à la bonne performance de notre outil en ce qui concerne la sélection de toutes les correspondances attendues par l'utilisateur.

On peut conclure que pour ce scénario, notre outil est parvenu à trouver environ plus de 95% des correspondances attendues par l'utilisateur et ce pour chaque méta-simulation. Par contre, on remarque que l'hétérogénéité du lexique relativement élevé, combinée à la taille de nos données en entrée, entraîne un temps d'itération plus long afin de pouvoir trouver toutes les correspondances attendues au niveau de chaque simulation.

Paramètres Sim- Méta-Sim	Nombre Moyen d'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F- mesure
Méta-simulation 1	9953.2	30	24.7	3.3	2	82%	88%	84%
Méta-simulation 2	10 239.6	30	27.3	1.7	0	94%	94%	94%
Méta-simulation 3	10 256.8	30	25.7	3.3	1	85%	88%	86%
Méta-simulation 4	9893.2	30	27	3	0	90%	90%	90%

Tableau 5.8 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 3

❖ Quatrième scénario

Pour notre quatrième et dernier scénario, on a voulu augmenter le nombre des CV à 50 avec 4 rubriques chacun et le nombre des offres d'emploi à 30 avec 3 rubriques chacune. Dans ce scénario, on considère la taille de nos schémas très grande et l'hétérogénéité du lexique aussi relativement très grande, car nous voulons voir si avec un nombre aussi élevé, notre outil performera toujours bien ou pas. Ci-dessous, nous avons le tableau 5.8 qui récapitule les résultats des 4 différentes méta-simulations.

Paramètres- Sim	Nombre d'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F- mesure
<b>1</b>	15 893	200	198	0	2	99%	100%	99%
<b>2</b>	13 749	200	193	0	7	96%	100%	98%
<b>3</b>	14 459	200	190	2	8	95%	99%	97%
<b>4</b>	13 885	200	191	0	9	95%	100%	97%
<b>5</b>	11 474	200	183	8	9	91%	96%	93%
<b>Méta-sim 1</b>	13 892	200	191	2	7	95%	99%	96%
<b>1</b>	11 994	200	181	5	14	90%	97%	93%
<b>2</b>	16 449	200	199	0	1	99%	100%	99%
<b>3</b>	15 344	200	190	0	10	95%	100%	97%
<b>4</b>	13 449	200	181	4	15	90%	98%	93%
<b>5</b>	12 649	200	179	2	19	89%	98%	93%
<b>Méta-sim 2</b>	13 977	200	186	2.2	11.8	92%	98%	95%
<b>1</b>	18 478	200	200	0	0	100%	100%	100%
<b>2</b>	16 232	200	197	0	3	98%	100%	98%
<b>3</b>	12 294	200	180	3	17	90%	98%	93%
<b>4</b>	17 383	200	198	0	2	99%	100%	99%
<b>5</b>	19 754	200	200	0	0	100%	100%	100%
<b>Méta-sim 3</b>	16 828.2	200	195	0.6	4.4	97%	99%	98%
<b>1</b>	16 334	200	192	6	2	96%	97%	97%
<b>2</b>	12 373	200	179	0	21	89%	100%	94%
<b>3</b>	15 843	200	180	7	13	90%	96%	92%
<b>4</b>	17 393	200	195	0	5	97%	100%	98%
<b>5</b>	20 001	200	200	0	0	100%	100%	100%
<b>Méta-sim 4</b>	16 388.8	200	189.2	2.6	8.2	94%	98%	96%

Tableau 5.9 Tableau de contingence des simulations multiples pour le quatrième scénario  
En étudiant le tableau 5.9, d'emblée, on voit encore une fois que notre outil performe vraiment bien. Pour un scénario aussi complexe nous avons eu de bons résultats. On en veut pour preuve les valeurs des

mesures comme la *Précision* qui frôle à chaque fois les 99% ainsi que la F-mesure qui est pour chaque méta-simulation supérieure à 90%. Cependant, ces excellents résultats sont contrastés par le nombre d'itérations relativement très élevé dont a besoin notre prototype avant de pouvoir trouver toutes les correspondances attendues par l'utilisateur. On remarque qu'à partir de 18000 itérations, le prototype est capable de pouvoir trouver quasiment toutes les correspondances attendues.

Paramètres Sim-	Nombre Moyen d'itérations	Corresp. attendues	Bonnes Corresp. trouvées	Fausse Corresp. trouvées	Corresp. Non trouvées	Rappel	Précision	F- mesure
<b>Méta-sim 1</b>	13 892	50	28.8	7.2	14	58%	80%	67%
<b>Méta-sim 2</b>	13 977	50	32	7.7	10.3	64%	81%	71%
<b>Méta-sim 3</b>	16 828.2	50	48	0	2	96%	100%	98%
<b>Méta-sim 4</b>	16 388.8	50	49.5	0	0.5	99%	100%	99%

Tableau 5.10 Tableau récapitulatif des correspondances CV-offres d'emploi lors des méta-simulations du scénario 4

### 5.3 Évaluation et étude des résultats

Nous voulons à travers cette partie passer en revue les résultats issus de nos expérimentations. En clair, il s'agit de discuter de la performance de notre prototype et de sa capacité à réduire l'incertitude. Nous voulons aussi évaluer dans quelle mesure, il permet de réduire l'effort humain en ce qui concerne son optimisation et sa configuration. Il convient aussi d'étudier ou d'évaluer le temps de réponse nécessaire pour atteindre les correspondances attendues pour un scénario.

De prime abord, afin d'avoir un aperçu général et de mesurer la performance des méta-simulations, nous avons compilé, dans un tableau, les résultats concernant les correspondances CV-OE attendues et celles trouvées au niveau des méta-simulations et ce pour tous les scénarios. En effet, si nous jetons un petit coup d'œil au tableau 5.11, pour chaque différent scénario, nous nous rendons compte que malgré les différents niveaux d'ambiguïté ou d'hétérogénéité lexicale, notre prototype parvient à trouver dans la plupart des cas, plus de la moitié des correspondances pour chaque scénario. La moyenne des pourcentages de chaque scénario nous montre que notre outil performe bien et est capable de trouver correctement plus de 80% de bonnes correspondances attendues par l'utilisateur.

Scénarios	Num méta-sim	Corresp attendues	Corresp trouvées	Corresp trouvées en %
• Scénario1	• 1	• 4	• 3	• 75%
• Scénario1	• 2	• 4	• 2.2	• 55%
• Scénario1	• 3	• 4	• 3.6	• 90%
• Scénario1	• 4	• 4	• 4	• 100
• Scénario2	• 1	• 10	• 7.6	• 76%
• Scénario2	• 2	• 10	• 7.2	• 72%
• Scénario2	• 3	• 10	• 8	• 80%
• Scénario2	• 4	• 10	• 9.5	• 95%
• Scénario3	• 1	• 30	• 24.7	• 82%
• Scénario3	• 2	• 30	• 27.3	• 91%
• Scénario3	• 3	• 30	• 25.7	• 85%
• Scénario3	• 4	• 30	• 27	• 90%
• Scénario4	• 1	• 50	• 28.8	• 57%
• Scénario4	• 2	• 50	• 32	• 64%
• Scénario4	• 3	• 50	• 48	• 96%
• Scénario4	• 4	• 50	• 49.5	• 99%

Tableau 5.11 Tableau récapitulatif des performances de l'appariement par rapport au nombre de simulations

Cependant, on remarque que les résultats des premières itérations sont moins bons par rapport aux dernières dans la plupart des scénarios ce qui nous a poussé à vouloir vérifier si le nombre de simulations individuelles composant une méta-simulation pouvait avoir une influence sur la performance de notre outil. Pour ce faire, nous avons conduit une nouvelle simulation avec les scénarios 3 et 4, mais en prenant le soin de réduire le nombre de simulations individuelles de 4 à 2.

Scénarios	Num méta-sim	Corresp attendues	Corresp trouvées	Corresp trouvées en %
• Scénario3	• 1	• 30	• 19	• 63%
• Scénario3	• 2	• 30	• 24	• 80%
• Scénario4	• 1	• 50	• 20	• 40%
• Scénario4	• 2	• 50	• 25	• 50%

Tableau 5.12 Tableau récapitulatif de l'impact du nombre de simulations individuelles sur les performances de l'appariement

Nous constatons que la performance pour le scénario 3 a baissé ainsi que celle du scénario 4. Autrement dit, notre prototype a moins performé et n'a pas pu trouver toutes les correspondances attendues surtout

au niveau du scénario 4 où il n'a trouvé que la moitié des correspondances. Nous avons la preuve que le nombre de simulations individuelles est un paramètre important, surtout pour des scénarios ayant des schémas de grande taille ainsi qu'un niveau d'hétérogénéité lexicale élevé (comme le scénario 4). Il joue un rôle clé pour permettre au prototype de faire une meilleure réduction de l'incertitude et par conséquent d'avoir une meilleure performance.

Par ailleurs, nous avons remarqué aussi que les résultats varient en fonction du nombre d'itérations composant une simulation individuelle. Afin de vérifier, notre point, on a aussi mené des expériences sur tous les scénarios avec une méta-simulation ayant 3 simulations individuelles. En effet, on a repris la précédente expérience avec cette fois-ci tous les scénarios et en baissant de façon graduelle, le nombre maximal de 13500 itérations à 5000, et de 5000 à 700 itérations.

Scénarios	Nombre Itérations	Corresp attendues	Corresp trouvées	Corresp trouvées en %
• Scénario1	• 13500	• 4	• 4	• 100%
• Scénario1	• 5000	• 4	• 4	• 100%
• Scénario1	• 700	• 4	• 3	• 75%
• Scénario2	• 13500	• 10	• 10	• 100%
• Scénario2	• 5000	• 10	• 8.5	• 95%
• Scénario2	• 700	• 10	• 3	• 30%
• Scénario3	• 13500	• 30	• 27.5	• 91%
• Scénario3	• 5000	• 30	• 14	• 46%
• Scénario3	• 700	• 30	• 5	• 16%
• Scénario4	• 13500	• 50	• 45.8	• 91%
• Scénario4	• 5000	• 50	• 23.3	• 46%
• Scénario4	• 700	• 50	• 7.4	• 14%

Tableau 5.13 Tableau récapitulatif de l'impact de la durée des simulations individuelles sur les performances de l'appariement

On remarque que pour le premier scénario, la performance baisse lors de la troisième simulation mais reste tout de même bonne avec 75% de bonnes correspondances trouvées. Par contre, on remarque pour les autres scénarios une dégradation flagrante de la performance de l'outil, à mesure que le nombre de simulation baisse. Nous remarquons que pour la durée de vie de la simulation de 700 itérations, la performance a surtout baissé pour les scénarios 3 et 4. Ces scénarios ont tous deux des schémas avec de grandes tailles. Ceci peut s'expliquer par le fait que notre modèle est capable de faire émerger la solution optimale par des mécanismes en rapport avec une dimension temporelle comme l'adaptation, l'évolution et la coévolution. Ce rapport avec le temps implique que la durée d'exécution des simulations est un

facteur primordial pour la manifestation d'une évolution du système vers l'émergence d'une solution optimale pour le problème de correspondance.

Au second abord, nous avons voulu nous pencher sur le fait qu'effectivement, notre outil n'a réellement besoin d'aucun effort utilisateur pour l'obtention des bonnes correspondances, comme ce fut le cas avec nos précédentes expérimentations. Étant donné que notre système d'appariement automatique a été conçu comme un système adaptatif et donc capable de s'adapter automatiquement à tous les changements et à tous les scénarios, il est important de relever le fait qu'il a su bien s'adapter à nos changements de scénarios et ce d'une manière autonome pour faire ressortir les correspondances attendues. Ceci étant, nous pouvons donc confirmer que notre outil d'appariement automatique de CV aux offres d'emploi, n'a besoin d'aucun effort de la part de l'utilisateur pour pouvoir trouver correctement les correspondances attendues.

Au vu de tout ce qui précède, on peut dire que notre prototype a pu répondre à nos différentes interrogations. Nous pouvons donc dire, que pour un délai relativement tolérable (environ 5 à 8 minutes, dans les scénarios les plus complexes), notre outil arrive à de très bons résultats.

#### 5.4 Conclusion

Dans ce chapitre, nous avons à travers les expériences qui ont été faites, apporté une contribution avec notre approche au domaine de la gestion des ressources humaines notamment au processus d'embauche des candidats. Nous avons pour ce faire, confirmé la viabilité de notre approche qui est la modélisation de l'appariement candidats-offres sous le prisme des systèmes complexes adaptatifs plus exactement en une simulation multi-agents. Par ailleurs, on a pu remarquer que notre approche reste très performante et reste au-dessus des autres performances des systèmes existants surtout quand on a des données de grande taille. Il est important, de plus, de souligner que plus qu'un simple outil, notre prototype s'inscrit dans un changement de paradigme pour l'appariement des CV aux OE. En effet, d'après nos recherches et au meilleur de nos connaissances, la correspondance entre les candidats et les offres d'emploi n'a jamais été abordée en se basant sur la pensée systémique de manière à déterminer toute la complexité qui lui est propre.



## **CHAPITRE 6**

### **CONCLUSION GÉNÉRALE**

Nous voulons dans ce chapitre, récapituler les éléments importants de notre recherche sur la correspondance ou l'appariement candidats-offres d'emploi ou CV-OE. En effet, nous voulons passer en revue les points pertinents et présenter notre contribution dans ce travail. Pour finir, nous présentons aussi les limites ou contraintes rencontrées lors de cette recherche, ainsi que les travaux futurs que nous jugeons importants pour l'évolution de notre prototype.

#### 6.1 Survol de la recherche

Dans le cadre de cette recherche nous avons dressé le cadre théorique de notre approche, les objectifs que nous visons et les motivations qui nous permettent de pouvoir penser qu'un changement de paradigme était nécessaire dans la problématique de la correspondance entre un candidat et une offre d'emploi. Nous avons vu que l'e-recrutement est un domaine en plein essor, et qu'il fallait apporter des réponses innovantes aux problèmes d'incertitude et de complexité auxquels les approches actuelles font face.

Pour ce faire, dans la première partie, nous avons tout d'abord établi l'état de l'art de tout ce qui se faisait quant à la correspondance entre les candidats et les offres d'emplois. Généralement, les travaux se concentrent sur les techniques de la recherche d'information ou sur les ontologies ou encore sur la proposition de nouvelles techniques d'apprentissage automatique. Autrement dit, ils exhibent les mêmes caractéristiques fondamentales comme la linéarité, le déterminisme et la centralisation qui font d'eux des systèmes compliqués et non complexes, incapables de donner des solutions à la fois simples et innovantes.

Nous avons, dans les chapitres 2 et 3, présenté le concept de l'appariement automatique de schémas dont s'inspire notre approche. Ensuite, nous avons exposé proprement dit, l'approche suivie découlant de la pensée systémique, plus concrètement du domaine des systèmes complexes adaptatifs. Cette approche tire ses origines de l'approche de simulation multi-agents basé sur un modèle non-déterministe, non

prévisible, aléatoire et sans contrôle centrale. Le but étant que chaque simulation puisse produire à partir des mêmes entrées, des résultats différents. Aussi, la multi-simulation permet une meilleure qualité des résultats et donc la réduction de l'incertitude. Par ailleurs, notre modèle conceptuel se veut être un appariement de schémas sous le prisme d'une simulation d'interactions des éléments de deux groupes d'agents (les agents des CV d'une part et les agents des OE d'autre part) dans laquelle, chaque groupe représente un des deux schémas à apparier et chaque agent représente un élément du schéma.

Dans la quatrième partie, c'est-à-dire au chapitre 4, nous avons procédé à la conception de notre modèle opérationnel et au développement de notre prototype s'exécutant sur la plateforme de simulation Repast Symphony. Nous avons aussi mis en place, une interface utilisateur présentant les résultats de façon beaucoup plus conviviale et beaucoup plus simple pour un utilisateur moins averti. Nous avons également en amont, effectué la préparation des données (CV et OE), ayant recours aux techniques de traitement du langage naturel et aux techniques d'extraction de texte. Le processus de préparation de données nous a servi à passer d'un ensemble de données non structuré et brut, vers un ensemble plus structuré dont le prototype a besoin.

Enfin, dans la cinquième partie, nous avons réalisé une série de tests, dans un premier temps par des simulations individuelles et dans un second temps par des métas simulations. Le but poursuivi était de déterminer la viabilité de notre approche en démontrant son efficacité, la facilité de sa prise en main (elle doit pouvoir nécessiter peu d'effort chez l'utilisateur pour atteindre ses meilleurs résultats), le tout en donnant des résultats dans un délai raisonnable. Nous avons obtenu des résultats satisfaisants qui démontrent la fiabilité de notre approche.

## 6.2 Apports de la recherche

L'objectif principal de notre travail est de proposer un prototype basé sur une nouvelle approche pour l'appariement des CV aux OE. Cette nouvelle approche s'inscrit dans un changement de paradigme, autrement dit, nous voulons que la correspondance CV-offre d'emploi soit abordée en adoptant la pensée systémique. Nous proposons, clairement, donc que cette correspondance ou cet appariement soit considéré(e) comme un système complexe adaptatif et soit modélisé(e) comme une simulation multi-agent. Cette étude nous a permis de mieux découvrir l'appariement des schémas, les systèmes multi-agents afin de pouvoir utiliser leurs caractéristiques intrinsèques pour produire et améliorer les résultats attendus de notre prototype.

Nous avons scindé la problématique en quatre différentes parties :

- Analyser et comparer les différentes approches existantes pour l'appariement des CV aux OE comme les ontologies, les systèmes de recommandation, les techniques d'apprentissage automatique etc.
- Procéder au traitement des données brutes recueillies sur internet, en faisant l'extraction des rubriques pertinentes pour traiter notre problématique. Concevoir ou développer notre prototype basé sur la plateforme de simulation Repast Symphony.
- Effectuer des expérimentations en procédant à des simulations simples et à des méta-simulations, dans le but de déterminer l'efficacité et l'efficacité de notre prototype.
- Discuter et analyser les résultats obtenus, et en tirer des conclusions.

### 6.3 Améliorations futures

À présent que la viabilité de notre approche a été démontré, il est important de nous pencher sur les parties ou les points qui méritent d'être amélioré(e)s à l'avenir, ce qui contribuerait par la même occasion à l'amélioration de notre modèle de simulation multi-agents pour l'appariement des CV aux OE. Pour ce faire, nous avons scindé les travaux futurs en différentes parties :

- Amélioration de l'interface utilisateur, ce qui permettrait au gestionnaire de directement ajouter les CV ainsi que les OE et de les soumettre, par un bouton, à la plateforme de simulation Repast Symphony ;
- Amélioration du temps de réponse. Nous avons remarqué lors de nos expériences que plus la taille de nos données d'entrée augmentait, plus le temps de réponse et donc de simulation augmentait. Afin de pouvoir avoir un temps de réponse plus court, pour un usage professionnel, nous pensons qu'il serait pertinent d'avoir un matériel d'exécution avec une puissance de calcul et donc une mémoire vive (RAM), supérieures.
- Expérimentation d'un nouveau type d'agent (Agent rationnel). En effet un agent rationnel est un agent qui peut avoir un comportement basé sur le raisonnement et la délibération par rapport au choix des actions. Nous pensons que faire évoluer l'agent réflexif vers un agent rationnel peut être perçue comme une solution pour répondre aux défis que posent les limites d'un agent purement réflexif.

Somme toute, notre changement de paradigme pour la résolution de la problématique de correspondance candidat-offre d'emploi s'est révélé être un succès. Ainsi, le passage vers une approche qui se base sur le domaine des systèmes complexes adaptatifs issue de la pensée systémique apporte une nouvelle perspective jusqu'ici et au meilleur de nos connaissances jamais explorée dans la gestion des ressources humaines notamment pour le recrutement électronique. Ce changement apporte donc un vent de renouveau à un domaine en plein essor et représente, par conséquent, une avancée notable pour l'optimisation de l'e-recrutement.

## BIBLIOGRAPHIE

- Abualigah, L. M. Q. (2019). *Feature selection and enhanced krill herd algorithm for text document clustering*. Springer.
- Al-Otaibi, S. T. et Ykhlef, M. (2012). A survey of job recommender systems. *International Journal of Physical Sciences*, 7(29), 5127-5142.
- Andrus, D. C. (2005). Toward a complex adaptive intelligence community: The wiki and the blog. *Studies in intelligence*, 49(3), 63-70.
- Arnold, P. et Rahm, E. (2014). Extracting semantic concept relations from wikipedia. Dans *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)* (p. 1-11).
- Assoudi, H. (2017). *L'appariement de schémas comme un système complexe adaptatif : une nouvelle approche basée sur la modélisation et la simulation à base d'agents*, 248.
- Assoudi, H. et Lounis, H. (2014). Towards an Agent-Based Simulation Model for Schema Matching. Dans *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems* (p. 197-198). <https://doi.org/10.1109/SASO.2014.42>
- Assoudi, H. et Lounis, H. (2015). Schema Matching as complex adaptive system. Dans *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)* (p. 1-7). <https://doi.org/10.1109/SITA.2015.7358442>
- Atzeni, P. et Torlone, R. (1996). Management of multiple models in an extensible database design tool. Dans *International Conference on Extending Database Technology* (p. 77-95). Springer.
- Aumueller, D., Do, H.-H., Massmann, S. et Rahm, E. (2005). Schema and ontology matching with COMA++. Dans *In Sigmod Conference* (p. 906-908). ACM Press.
- Baad, D. (2019). *Automatic job skill taxonomy generation for recruitment systems*.
- Ballot, G. (1981). *Marché du travail et dynamique de la répartition des revenus salariaux* [éditeur inconnu].
- Ballot, G. (2002). Modeling the labor market as an evolving institution: model ARTEMIS. *Journal of Economic Behavior & Organization*, 49(1), 51-77. [https://doi.org/10.1016/S0167-2681\(02\)00058-6](https://doi.org/10.1016/S0167-2681(02)00058-6)
- Bar, D., Zesch, T. et Gurevych, I. (2013). *DKPro Similarity: An Open Source Framework for Text Similarity*, 6.
- Barreto, L., Amaral, A. et Pereira, T. (2017). Industry 4.0 implications in logistics: an overview. *Procedia manufacturing*, 13, 1245-1252.
- Bellahsene, Z., Bonifati, A. et Rahm, E. (dir.). (2011). *Schema Matching and Mapping*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-16518-4>

- Bergmann, B. R. (1974). A Microsimulation of the Macroeconomy with Explicitly Represented Money Flows. Dans *NBER Chapters* (p. 475-489). National Bureau of Economic Research, Inc. <https://ideas.repec.org/h/nbr/nberch/10173.html>
- Bernstein, P. A., Madhavan, J. et Rahm, E. (2011). Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11), 695-701. <https://doi.org/10.14778/3402707.3402710>
- Bertalanffy, A. R., Boulding, K. E., Ashby, W. R., Mead, M. et Bateson, G. (1968). *L. von Bertalanffy, General System Theory*. New York: George Braziller.
- Bohannon, P., Elnahrawy, E., Fan, W. et Flaster, M. (2006). Putting context into schema matching. Dans *Proceedings of the 32nd international conference on Very large data bases* (p. 307-318). Citeseer.
- Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3), 7280-7287.
- Bourgine, P., Chavalarias, D. et Cohen-Boulakia, C. (2008). *Déterminismes et complexités: du physique à l'éthique*. La Découverte.
- Breaugh, J. A. (2009). The use of biodata for employee selection: Past research and future directions. *Human Resource Management Review*, 19(3), 219-231.
- Calaresu, M. et Shiri, A. (2015). Understanding semantic web: A conceptual model. *Library Review*.
- Cappelli, P. (2019, 1<sup>er</sup> mai). Your Approach to Hiring Is All Wrong. *Harvard Business Review*. <https://hbr.org/2019/05/your-approach-to-hiring-is-all-wrong>
- Carlisle, Y. M. et McMillan, E. (2017). Complex adaptive systems and strategy as learning. Dans *Global innovation and entrepreneurship* (p. 43-60). Springer.
- Chaib-draa, B. et Dignum, F. (2002). Trends in agent communication language. *Computational intelligence*, 18(2), 89-101.
- Chandrasekaran, B., Josephson, J. R. et Benjamins, V. R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems and their applications*, 14(1), 20-26.
- Cohen, W. W., Ravikumar, P. et Fienberg, S. E. (2003). *A Comparison of String Metrics for Matching Names and Records*, 6.
- Costa, A. da R. et Demazeau, Y. (1996). Toward a formal model of multi-agent systems with dynamic organizations. Dans *Proceedings of the International Conference on Multi-Agent Systems*, MIT Press, Kyoto, Japan (vol. 431).
- Cruz, I. F., Antonelli, F. P. et Stroe, C. (2009). AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2), 1586-1589.
- Das, A. S., Datar, M., Garg, A. et Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. Dans *Proceedings of the 16th international conference on World Wide Web* (p. 271-280).

- Delahaye, J.-P. (2017). *Le tout est-il plus que la somme de ses parties ?* Purlascience.fr. Pour la Science. <https://www.purlascience.fr/sd/mathematiques/https://www.purlascience.fr/sd/mathematiques/le-tout-est-il-plus-que-la-somme-de-ses-parties-9716.php>
- Dieng, M. A. (2016). *Développement d'un système d'appariement pour l'e-recrutement*.
- Dirac, P. A. M. (1929). Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 123(792), 714-733.
- Do, H.-H., Melnik, S. et Rahm, E. (2002). Comparison of schema matching evaluations. Dans *Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World* (p. 221-237). Springer.
- Do, H.-H. et Rahm, E. (2002). COMA—a system for flexible combination of schema matching approaches. Dans *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases* (p. 610-621). Elsevier.
- Do, H.-H. et Rahm, E. (2007). Matching large schemas: Approaches and evaluation. *Information Systems*, 32, 857-885. <https://doi.org/10.1016/j.is.2006.09.002>
- Doan, A., Madhavan, J., Domingos, P. et Halevy, A. (2002). Learning to map between ontologies on the semantic web. Dans *Proceedings of the 11th international conference on World Wide Web* (p. 662-673).
- Dodder, R. et Dare, R. (2000). *Complex Adaptive Systems and Complexity Theory: Inter-related Knowledge Domains*, 14.
- El Fallah-Seghrouchni, A. et Briot, J.-P. (2009). *Technologies des systemes multi-agents et applications industrielles*. Hermès Lavoisier.
- Elmeleegy, H., Ouzzani, M. et Elmagarmid, A. (2008). Usage-based schema matching. Dans *2008 IEEE 24th International Conference on Data Engineering* (p. 20-29). IEEE.
- Euzenat, J. et Shvaiko, P. (2007). *Ontology matching* (vol. 18). Springer.
- Faliagka, E., Tsakalidis, A. et Tzimas, G. (2012). An integrated e-recruitment system for automated personality mining and applicant ranking. *Internet research*.
- Ferber, J., Gutknecht, O. et Michel, F. (2004). From Agents to Organizations: An Organizational View of Multi-agent Systems. Dans P. Giorgini, J. P. Müller et J. Odell (dir.), *Agent-Oriented Software Engineering IV* (p. 214-230). Springer. [https://doi.org/10.1007/978-3-540-24620-6\\_15](https://doi.org/10.1007/978-3-540-24620-6_15)
- Gal, A. (2006). Managing uncertainty in schema matching with top-k schema mappings. Dans *Journal on Data Semantics VI* (p. 90-114). Springer.
- Giunchiglia, F., Dutta, B. et Maltese, V. (2009). Faceted lightweight ontologies. Dans *Conceptual modeling: foundations and applications* (p. 36-51). Springer.

- Glennan, S. (2005). Modeling mechanisms. *Studies in history and philosophy of science part C: studies in history and philosophy of biological and biomedical sciences*, 36(2), 443-464.
- Golec, A. et Kahya, E. (2007). A fuzzy model for competency-based employee evaluation and selection. *Computers & Industrial Engineering*, 52(1), 143-161.
- Grau, B. C., Dragisic, Z., Eckert, K., Euzenat, J., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A. O. et Lambrix, P. (2013). Results of the ontology alignment evaluation initiative 2013. Dans *OM: Ontology Matching* (p. 61-100). No commercial editor.
- Grouls, A. (2013). *Agents et systèmes multi-agents: vers une synthèse de ces concepts/mémoire présenté comme exigence partielle de la maîtrise en informatique par Alexandre Grouls;[directeur de recherche, Roger Nkambou]*.
- Guedes, G. T. A. et Vicari, R. M. (2009). Applying AUML and UML 2 in the Multi-agent Systems Project. Dans C. A. Heuser et G. Pernul (dir.), *Advances in Conceptual Modeling - Challenging Perspectives* (p. 106-115). Springer. [https://doi.org/10.1007/978-3-642-04947-7\\_14](https://doi.org/10.1007/978-3-642-04947-7_14)
- Guo, S., Alamudun, F. et Hammond, T. (2016). Résumatcher: A personalized résumé-job matching system. *Expert Systems with Applications*, 60, 169-182.
- Hammou, B. A., Lahcen, A. A. et Mouline, S. (2019). An effective distributed predictive model with Matrix factorization and random forest for Big Data recommendation systems. *Expert Systems with Applications*, 137, 253-265.
- Hartung, M., Kolb, L., Groß, A. et Rahm, E. (2013). Optimizing similarity computations for ontology matching-experiences from gomega. Dans *International Conference on Data Integration in the Life Sciences* (p. 81-89). Springer.
- Harzallah, M., Leclère, M. et Trichet, F. (2002). CommOnCV: modelling the competencies underlying a curriculum vitae. Dans *Proceedings of the 14th international conference on Software engineering and knowledge engineering - SEKE '02* (p. 65). ACM Press. <https://doi.org/10.1145/568760.568773>
- Hasard, nombres aléatoires et méthode Monte Carlo - Louis Laurencelle - Google Livres.* (s. d.). [https://books.google.ca/books?hl=fr&lr=&id=dcuc9soyfrMC&oi=fnd&pg=PP3&dq=Laurencelle+2001+simulation+stochastic&ots=re\\_ompd8Bo&sig=ILpnwwVG\\_\\_JSjVEly-gYXYwTvl#v=onepage&q&f=false](https://books.google.ca/books?hl=fr&lr=&id=dcuc9soyfrMC&oi=fnd&pg=PP3&dq=Laurencelle+2001+simulation+stochastic&ots=re_ompd8Bo&sig=ILpnwwVG__JSjVEly-gYXYwTvl#v=onepage&q&f=false)
- Holland, J. H. (2006). Studying complex adaptive systems. *Journal of systems science and complexity*, 19(1), 1-8.
- Hu, W., Qu, Y. et Cheng, G. (2008). Matching large ontologies: A divide-and-conquer approach. *Data & Knowledge Engineering*, 67(1), 140-160.
- Huget, M.-P. (2004). Agent UML notation for multiagent system design. *IEEE Internet Computing*, 8(4), 63-71. <https://doi.org/10.1109/MIC.2004.6>



- Huynh, A. T., Nguyen, B. T., Nguyen, H.-T., Vu, S. et Nguyen, H. D. (2021). A Method of Deep Reinforcement Learning for Simulation of Autonomous Vehicle Control. Dans *ENASE* (p. 372-379).
- Industriens utredningsinstitut (Sweden). (1980). *Micro simulation: models, methods and applications : proceedings of a symposium on micro Simulation methods in Stockholm, September 19-22, 1977*. Almqvist and Wiksell International.
- Iwu, C. G., Kapondoro, L., Twum-Darko, M. et Lose, T. (2016). *Strategic human resource metrics: A perspective of the general systems theory*.
- Jean-Mary, Y. R., Shironoshita, E. P. et Kabuka, M. R. (2009). Ontology matching with semantic verification. *Journal of Web Semantics*, 7(3), 235-251.
- Jiang, Q.-Y. et Li, W.-J. (2019). Discrete latent factor model for cross-modal hashing. *IEEE Transactions on Image Processing*, 28(7), 3490-3501.
- Kant, I. (2012). *Fundamental principles of the metaphysics of morals*. Courier Corporation.
- Kaplan, A. et Haenlein, M. (2019). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, 62(1), 15-25. <https://doi.org/10.1016/j.bushor.2018.08.004>
- Kaur, P. (2015). *E-recruitment: A conceptual study*, 5.
- Kirsten, T., Gross, A., Hartung, M. et Rahm, E. (2011). GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *Journal of biomedical semantics*, 2(1), 1-24.
- Lambrix, P. et Tan, H. (2008). Ontology alignment and merging. Dans *Anatomy Ontologies for Bioinformatics* (p. 133-149). Springer.
- Lesne, A. (2009). Systems biology: The multiscale organization of living systems. *Medecine Sciences: M/S*, 25(6-7), 585-587.
- Li, H. et Parlikad, A. K. (2016). Social internet of industrial things for industrial and manufacturing assets. *IFAC-PapersOnLine*, 49(28), 208-213.
- Li, J., Tang, J., Li, Y. et Luo, Q. (2008). Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and data Engineering*, 21(8), 1218-1232.
- Lin, Y., Lei, H., Addo, P. C. et Li, X. (2016). Machine learned resume-job matching solution. *arXiv preprint arXiv:1607.07657*.
- Longo, F., Nicoletti, L. et Padovano, A. (2017). Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Computers & industrial engineering*, 113, 144-159.
- Lu, J., Wu, D., Mao, M., Wang, W. et Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, 12-32.

- Macal, C. M. et North, M. J. (2007). Agent-based modeling and simulation: Desktop ABMS. Dans *2007 Winter Simulation Conference* (p. 95-106). IEEE.
- Macal, C. M. et North, M. J. (2009). Agent-based modeling and simulation. Dans *Proceedings of the 2009 winter simulation conference (WSC)* (p. 86-98). IEEE.
- Macal, C. M. et North, M. J. (2010). *Tutorial on agent-based modelling and simulation. J Simul 4 (3): 151–162.*
- Madhavan, J., Bernstein, P. A. et Rahm, E. (2001). Generic schema matching with cupid. Dans *vldb* (vol. 1, p. 49-58). Citeseer.
- Malinowski, J., Keim, T., Wendt, O. et Weitzel, T. (2006). Matching people and jobs: A bilateral recommendation approach. Dans *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)* (vol. 6, p. 137c-137c). IEEE.
- Mansfield, W. (2016). Marion, 2016, Staffing Decisions. *Artificial Intelligence and Human Resources.*
- Marie, A. et Gal, A. (2008). A.: Boosting Schema Matchers (vol. 5331, p. 283-300).  
[https://doi.org/10.1007/978-3-540-88871-0\\_20](https://doi.org/10.1007/978-3-540-88871-0_20)
- Marilleau, N., Philippe, L. et Université de Franche-Comté. UFR des sciences et techniques. (2006). Méthodologie, formalismes et outils de modélisation-simulation pour l'étude des systèmes complexes: application à la mobilité géographique.
- Martono, G. H. et Azhari, S. N. (2017). Review implementation of linguistic approach in schema matching. *International Journal of Advances in Intelligent Informatics*, 3(1), 1-9.
- Massmann, S. et Rahm, E. (2008). Evaluating Instance-based Matching of Web Directories. Dans *WebDB*.
- McMillan, E. (2008). *Complexity, management and the dynamics of change: Challenges for practice.* Routledge.
- Melnik, S., Garcia-Molina, H. et Rahm, E. (2002). Similarity flooding: A versatile graph matching algorithm and its application to schema matching. Dans *Proceedings 18th international conference on data engineering* (p. 117-128). IEEE.
- Minar, N., Burkhart, R., Langton, C. et Askenazi, M. (1996). *The swarm simulation system: A toolkit for building multi-agent simulations.*
- Mishra, R., Rodriguez, R. et Portillo, V. (2020). An AI based talent acquisition and benchmarking for job. *arXiv:2009.09088 [cs]*. <http://arxiv.org/abs/2009.09088>
- Mongia, A., Jhamb, N., Chouzenoux, E. et Majumdar, A. (2020). Deep latent factor model for collaborative filtering. *Signal Processing*, 169, 107366.
- Mork, P., Seligman, L., Rosenthal, A., Korb, J. et Wolf, C. (2008). The harmony integration workbench. Dans *Journal on Data Semantics XI* (p. 65-93). Springer.

- Mrugalska, B. et Wyrwicka, M. K. (2017). Towards lean production in industry 4.0. *Procedia engineering*, 182, 466-473.
- Multi-agent system. (2022, 10 avril). Dans *Wikipedia*. Récupéré le 21 avril 2022 de [https://en.wikipedia.org/w/index.php?title=Multi-agent\\_system&oldid=1081949818](https://en.wikipedia.org/w/index.php?title=Multi-agent_system&oldid=1081949818)
- Nguyen, H. D., Huynh, T., Hoang, S. N., Pham, V. T. et Zelinka, I. (2020). Language-oriented Sentiment Analysis based on the Grammar Structure and Improved Self-attention Network. Dans *ENASE* (p. 339-346).
- North, M. J., Collier, N. T., Ozik, J., Tatara, E. R., Macal, C. M., Bragen, M. et Sydelko, P. (2013). Complex adaptive systems modeling with Repast Symphony. *Complex Adaptive Systems Modeling*, 1(1), 3. <https://doi.org/10.1186/2194-3206-1-3>
- Paparrizos, I., Cambazoglu, B. B. et Gionis, A. (2011). Machine learned job recommendation. Dans *Proceedings of the fifth ACM Conference on Recommender Systems* (p. 325-328).
- Parunak, H. V. D. et Bauer, B. (s. d.). *jodell@compuserve.com www.jamesodell.com*, 15.
- Pellet, S. (2013). Les mécanismes d'appariement sur le marché du travail. *Regards croisés sur l'économie*, 13(1), 25-30.
- Pillai, R. et Sivathanu, B. (2020). Adoption of artificial intelligence (AI) for talent acquisition in IT/ITeS organizations. *Benchmarking: An International Journal*.
- Pottinger, R. A. et Bernstein, P. A. (2003). Merging models based on given correspondences. Dans *Proceedings 2003 VLDB Conference* (p. 862-873). Elsevier.
- Qiu, D., Barbosa, L., Dong, X. L., Shen, Y. et Srivastava, D. (2015). Dexter: large-scale discovery and extraction of product specifications on the web. *Proceedings of the VLDB Endowment*, 8(13), 2194-2205.
- Rahm, E. (2011a). *Schema Matching and Mapping*. [https://www.academia.edu/20750861/Schema\\_Matching\\_and\\_Mapping](https://www.academia.edu/20750861/Schema_Matching_and_Mapping)
- Rahm, E. (2011b). Towards large-scale schema and ontology matching. Dans *Schema matching and mapping* (p. 3-27). Springer.
- Rahm, E. et Bernstein, P. A. (2001a). A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4), 334-350.
- Rahm, E. et Bernstein, P. A. (2001b). On matching schemas automatically. *VLDB journal*, 10(4), 334-350.
- Rahm, E. et Peukert, E. (2019). *Large-Scale Schema Matching*.
- Raunich, S. et Rahm, E. (2014). Target-driven merging of taxonomies with ATOM. *Information Systems*, 42, 1-14.
- Remondino, M. et Correndo, G. (2006). MABS Validation Through Repeated Execution and Data Mining Analysis. *International Journal of Simulation: Systems, Science and Technology*, 7.

- Resnick, P. et Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56-58.
- Ripley, B. D. (2001). The R Project in Statistical Computing. *MSOR Connections*, 1(1), 23-25.  
<https://doi.org/10.11120/msor.2001.01010023>
- Russell, S. et Norvig, P. (2010). *Intelligence artificielle: Avec plus de 500 exercices*. Pearson Education France.
- Sallach, D. L. et Macal, C. M. (2001). Introduction: The Simulation of Social Agents. *Social Science Computer Review*, 19(3), 245-248. <https://doi.org/10.1177/089443930101900301>
- Sarkar, S. et Pfeifer, J. (2006). *The philosophy of science*. Psychology Press.
- Seligman, L., Mork, P., Halevy, A., Smith, K., Carey, M. J., Chen, K., Wolf, C., Madhavan, J., Kannan, A. et Burdick, D. (2010). Openii: an open source information integration toolkit. Dans *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (p. 1057-1060).
- Senthil Kumaran, V. et Sankar, A. (2013). Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT). *International Journal of Metadata, Semantics and Ontologies*, 8(1), 56-64.
- Shannon, R. E. (1998). Introduction to the art and science of simulation. Dans *1998 winter simulation conference. proceedings (cat. no. 98ch36274)* (vol. 1, p. 7-14). IEEE.
- Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1), 51-92.  
[https://doi.org/10.1016/0004-3702\(93\)90034-9](https://doi.org/10.1016/0004-3702(93)90034-9)
- Siegfried, R. (2014). *Modeling and simulation of complex systems: A framework for efficient agent-based modeling and simulation*. Springer.
- Simpson, J. (2006). *Simulations are not models*.
- Smith, K., Morse, M., Mork, P., Li, M., Rosenthal, A., Allen, D., Seligman, L. et Wolf, C. (2009). The role of schema matching in large enterprises. *arXiv preprint arXiv:0909.1771*.
- Stigler, G. J. (1962). Information in the Labor Market. *Journal of Political Economy*, 70(5, Part 2), 94-105.  
<https://doi.org/10.1086/258727>
- Stigmergie. (2020, 3 mai). Dans *Wikipédia*. Récupéré le 21 avril 2022 de  
<https://fr.wikipedia.org/w/index.php?title=Stigmergie&oldid=170399010>
- Tao, F., Qi, Q., Liu, A. et Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48, 157-169.
- Thomson, B. K., Kelvin, B. et Thomson, W. (2010). *Baltimore lectures on molecular dynamics and the wave theory of light*. Cambridge University Press.
- Tokody, D. (2018). Digitising the European industry-holonic systems approach. *Procedia Manufacturing*, 22, 1015-1022.

- Treuil, J.-P., Drogoul, A. et Zucker, J.-D. (2008). *Modélisation et simulation à base d'agents: exemples commentés, outils informatiques et questions théoriques*. Dunod.
- van Gigch, J. P. et Le Moigne, J.-L. (1990). The design of an organization information system: intelligent artifacts for complex organizations. *Information & management*, 19(5), 325-331.
- Waldrop, M. M. (1993). *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster.
- Wendell, J. (2003). *Complex Adaptive Systems. Beyond Intractability*. Eds. Guy Burgess and Heidi Burgess. Boulder, USA: University of Colorado, Conflict Research Consortium.
- Wooldridge, M., Jennings, N. R. et Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3(3), 285-312.  
<https://doi.org/10.1023/A:1010071910869>
- Wuest, T., Weimer, D., Irgens, C. et Thoben, K.-D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1), 23-45.
- Xie, Q. (2022). Machine learning in human resource system of intelligent manufacturing industry. *Enterprise Information Systems*, 16(2), 264-284.  
<https://doi.org/10.1080/17517575.2019.1710862>
- Xu, L. D., Xu, E. L. et Li, L. (2018). Industry 4.0: state of the art and future trends. *International journal of production research*, 56(8), 2941-2962.
- Yahiaoui, L., Boufaïda, Z. et Prié, Y. (s. d.). *Automatisation du e-recrutement dans le cadre du Web sémantique*, 11.
- Yousfi, A., El Yazidi, M. H. et Zellou, A. (2020). xmatcher: Matching extensible markup language schemas using semantic-based techniques. *International Journal of Advanced Computer Science and Applications*, 11(8), 655-665.
- Zhang, L., Fei, W. et Wang, L. (2015). Pj matching model of knowledge workers. *Procedia computer science*, 60, 1128-1137.
- (2006, 1<sup>er</sup> février). Holisme Vs Reductionnisme. *La Science Pour Tous*. <https://lasciencepourtous.cafe-sciences.org/articles/holismevsreductionnisme/>
- (S. d.). *Agents et systèmes multi-agents : vers une synthèse de ces concepts*, 105.
- (S. d.). <http://www.fipa.org/specs/fipa00001/index.html>