



**HAL**  
open science

## Agents compétence autonomes pour l'amélioration de la collaboration entre acteurs humain

Olivier Chator, Jean-Marc Salotti

► **To cite this version:**

Olivier Chator, Jean-Marc Salotti. Agents compétence autonomes pour l'amélioration de la collaboration entre acteurs humain. 1er Colloque International "Connaissances et Informations en Action" (CIA): Transfert et organisation des connaissances en contexte, May 2014, Bordeaux, France. pp.28-38. hal-01163237

**HAL Id: hal-01163237**

**<https://hal.science/hal-01163237>**

Submitted on 12 Jun 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## **CHATOR OLIVIER**

Doctorant

Laboratoire : IMS, UMR CNRS 5218, ENSC, IPB

Université de Bordeaux

## **SALOTTI JEAN-MARC**

Professeur

Laboratoire : IMS, UMR CNRS 5218, ENSC, IPB

Université de Bordeaux

# **Agents compétence autonomes pour l'amélioration de la collaboration entre acteurs humain**

## **INTRODUCTION**

En tant qu'institution publique, le "Conseil Général de la Gironde" (CG33) étend son action sur les 1.5 million d'habitants du département. Ces actions relèvent de domaines variés : l'école, les transports, le tourisme, la solidarité, etc. L'une de ses missions importantes est la définition de politiques publiques et la promotion de pratiques autour des thématiques de Développement Durable (DD). Concrètement, il peut s'agir de la transformation de quartiers en éco-quartiers [1]. Les retours d'expérience terrain montrent que ce type de projet est très complexe. Il nécessite la collaboration de nombreux acteurs publics et privés, sous la direction d'un Maître d'œuvre (MO) tel qu'un cabinet d'architectes. L'un des points clé de ce constat est que chaque acteur impliqué dans un projet n'a qu'une vision partielle des capacités (compétences) des autres. Cette connaissance limitée est souvent problématique et a un impact sur l'efficacité de la collaboration inter-acteurs (perte de temps à rechercher qui détient la compétence, implication d'un autre acteur dans le projet alors que la compétence est déjà détenue en interne mais qu'on ne le sait pas, etc.). Dans tous les cas, le MO doit prendre des décisions en tenant compte de nombreux paramètres, souvent antagonistes entre eux. Exemple : la minimisation des coûts et l'obtention de labels énergétique pour les Bâtiments Basse Consommation (BBC). Dans le but d'accroître la réciprocité des échanges et la collaboration des acteurs impliqués dans ses projets de DD, le CG33 a décidé l'élaboration d'un système informatique de partage des compétences : la Bourse aux Compétences du Développement Durable (BCDD). En généralisant à tout domaine d'activité professionnel, ce sont finalement des compétences, et non spécifiquement de tel ou tel acteur humain, dont on a besoin pour mener à bien des projets.

Il devient alors évident que les compétences ont un rôle central, sont en évolution constantes au fil du temps (ne serait-ce que par les techniques employées). Les acteurs humains ne font que les concrétiser dans les projets. Une approche traditionnelle de construction du système informatique de partage aurait pu être d'utiliser une base de données « statique » intégrant des liens directs entre compétences et acteurs. Un projet de recherche a été mené pour trouver une meilleure solution, tenant compte du dynamisme des compétences. La proposition est ainsi d'implémenter un Système Multi-Agent (SMA) où les compétences sont des agents autonomes, dynamiques et apprenants [2, 3, 4]. La modélisation est décrite à la section 2. Les premiers résultats issus de l'expérimentation seront fournis à la section 3. Une conclusion sera proposée à la section 4.

# MODELISATION

## Problématique

Quelque soit le projet (construire un bâtiment répondant aux normes BBC ou identifier 15 joueurs dans une équipe de rugby), le MO doit définir un objectif, identifier les compétences pour le réaliser et trouver une liste potentielle d'acteurs qui vont concrétiser ces compétences. Une approche simpliste consiste à simplement trier cette liste d'acteurs en fonction de leur niveau d'expérience pour chaque compétence. Mais d'autres critères que l'expertise doivent être pris en compte. Exemple : pour un projet de construction, ce peut être le coût, la qualité et la capacité à savoir poser tel ou tel matériau plutôt qu'un autre. Pour une équipe de rugby, pour un poste donné, ce sera le nombre d'essais déjà marqués, le nombre de pénalités réussies et le nombre de cartons jaunes (à minimiser). Le système doit suggérer une liste ordonnée selon des valeurs pondérées de critères divers et variés.

Dans un système de partage de compétences efficace, on cherche à assurer une réciprocité des échanges et à apporter un bénéfice à chaque acteur. Il convient alors de sélectionner différents acteurs au cours des projets dans le but de leur permettre de monter en compétence. Tout le problème consiste alors à définir la bonne stratégie pour atteindre l'objectif global : accroître et faciliter la collaboration des acteurs humains.

## Définir une compétence

En anglais, le mot « skill » désigne la capacité à exploiter des connaissances et savoir-faire dans le but de résoudre des problèmes. Il diffère du mot « competency » qui désigne généralement un ensemble de comportements ou d'actions qu'il est nécessaire de mettre en œuvre dans un contexte particulier [5]. Pour simplifier, il est proposé ici de considérer qu'une compétence  $C$  est définie comme étant une somme de « compétences Élémentaires »  $C_{et}$  avec (avec  $C_{et} \in CEt$ ). Tout le problème consiste à déterminer dans quelle mesure (selon quelle pondération) chaque  $C_{et}$  participe à la définition de  $C$ . En fonction des projets, une compétence peut : être créée à n'importe quel moment, posséder une définition qui évolue dynamiquement (liste de  $C_{et}$ ), être divisée en d'autres compétences, éventuellement disparaître.

Il est alors pertinent de définir  $C$  sous la forme d'un agent autonome, dynamique, apprenant, et ayant son cycle de vie propre dans un SMA. Dans les sciences cognitives, la notion d' « *embodiment of mind* » est souvent requise pour que l'on puisse parler d'agent [6, 7]. Si une compétence  $C$  est « incarnée » (concrétisée) par un acteur humain, elle n'a pas de motivations à proprement parler. On peut toutefois les définir artificiellement.

## Proposition théorique

Définition : Toute compétence est traduite de manière unique au sein du SMA par un Agent Compétence (AC). Chaque AC est un agent cognitif, non-conversationnel et non dialogique [2, 4] qui :

1. Se définit principalement par 3 composantes clé :
  - Des mécanismes de perception et d'échange avec les utilisateurs humains du système ou les autres agents du SMA
  - Des attributs internes : une liste de compétences élémentaires, une date de création, un « âge »
  - Des mécanismes d'action traduits dans des comportements permettant l'atteinte de buts

2. Possède à minima 3 buts (bien que définis artificiellement) :
  - Constituer sa propre définition
  - Proposer une liste d'acteurs humains capable d'exploiter cette compétence pour concrétiser un projet
  - Etablir des liens (relations) avec d'autres agents
3. Possède des ressources : Tout AC est incarné par un(des) acteur(s) humain(s) et/ou fait appel à d'autres ACs.
4. Possède un « cycle de vie » décliné en 3 « âges » et qui sont : enfance, adolescence et maturité. Chaque âge traduit un niveau d'autonomie différent.
5. Possède un « modèle comportemental » dédié et évolutif par apprentissage.
6. S'insère dans un SMA de type « évènementiel » plutôt que « temporel ». Dans un SMA classique, les agents humains évoluent dans un univers où le temps est simulé comme dans la réalité. Dans notre cas, le principe de fonctionnement est différent. Ce sont les interrogations formulées par les utilisateurs, les mises à jour du système ou bien l'évolution des situations environnementales qui font évoluer l'univers de l'agent et par voie de conséquence les agents eux-mêmes.

On notera qu'un AC n'échange jamais directement avec les acteurs humains mais via un agent « technique » en respect des normes FIPA [9]. On admettra également que les 2 âges particuliers qui pourraient correspondre à la création de l'agent et à sa suppression (notions de naissance et de mort), sont exclus du cycle de vie, leur durée étant quasiment nulle et considérée comme non significative. Lors du passage d'un âge à l'autre, on dira que l'AC « grandit » (ou évolue). Les transitions entre les âges sont assurées de manière automatique par des règles comportementales que l'on qualifiera de « règles de croissance ». Ces dernières se basent sur l'analyse de conditions environnementales propre à l'AC. Par exemple, une des règles de croissance peut déclencher le passage automatique d'un âge à l'autre si l'on dispose de  $n$  évaluations de projets dans lesquels l'AC a été impliqué. Chaque AC étant autonome, ces règles peuvent bien entendu varier et être adaptées pour chacun d'eux. Dans tous les cas, elles feront partie de l'ensemble des règles, dites « éducatives de base » que l'on fournira à l'AC au moment de sa création dans le système.

### **Concernant l'apprentissage**

Les mécanismes d'apprentissage d'un agent compétence AC concernent 2 points principaux. Le premier concerne le calcul des pondérations des  $C_{et}$  qui évoluent au fil du temps pour la sélection d'une liste d'acteurs capables de concrétiser AC. Le second concerne l'analyse de similitudes environnementales dans le but de mettre en œuvre un établissement autonome de liens entre AC et les autres agents compétences. Dans la littérature, nous trouvons de nombreux mécanismes utilisés pour l'apprentissage dans les SMA. On citera, par exemple : Processus Décisionnel de Markov [10] avec apprentissage par renforcement [11], théorie des jeux (jeux matriciels [12] et jeux stochastiques [13]), réseaux bayésiens [14], méthodes de raisonnement à partir de cas [16]... Nous avons toutefois choisi ici de proposer notre propre vision de l'apprentissage compte tenu du rôle central accordé aux validations humaines à l'âge d'adolescence, avant que l'AC ne devienne complètement autonome une fois mature. On notera également que lors de l'établissement de liens inter-agents, l'AC fait évoluer dynamiquement ses règles comportementales. Il y a donc non seulement apprentissage mais modification dynamique du modèle comportemental dans le système que nous proposons.

## Apprentissage pour la sélection des acteurs

Le comportement de sélection des acteurs est déclenché suite à réception d'une requête utilisateur, i.e. de type évènementiel. Cette requête possède deux paramètres principaux que sont la compétence souhaitée et la liste (valeurs numériques) des niveaux d'importance accordés à chaque critère d'efficacité (i.e. compétence élémentaire) qui la définit. Pour l'illustration, considérons que chaque poste d'une équipe de rugby soit une compétence. Dans ce contexte, imaginons qu'un entraîneur adresse la demande qui suit à notre système :

1. Je souhaite trouver quelqu'un qui puisse jouer au poste de « demi de mêlée »
2. Je précise le niveau d'importance (valeur sur 4, égale à 1 par défaut) que j'attache à chaque critère d'efficacité:
  - Points marqués au pied : 3
  - Touches trouvées : 1
  - Mêlées perdues : 4
  - Matches déjà joués : 2

Tout l'enjeu consiste à fournir à l'utilisateur une liste d'acteurs humains pouvant concrétiser la compétence demandée. La constitution de cette liste repose sur des calculs, liés aux mécanismes d'apprentissage, et basés sur les valeurs des poids des critères. L'apprentissage se déroule en 3 étapes :

**Etape 1** : Pour toute nouvelle requête  $t$ , l'utilisateur précise les poids  $W_{1(t)} \dots W_{n(t)}$  qu'il souhaite voir associés à chacune des compétences élémentaires  $C_1 \dots C_n$ .

**Etape 2** : Le système recalcule les poids associés à chaque critère via l'équation (1). Il se base pour se faire sur les valeurs saisies à l'étape 1 et sur la moyenne des poids de chaque critère dans les précédentes itérations qui ont été mémorisées et validées :

$$\overline{W}_{k(t+1)} = \frac{(t \times \overline{W}_{k(t)}) + W_{k(t+1)}}{t + 1} \quad (1)$$

**Etape 3** : Le système construit une liste d'acteurs à renvoyer à l'utilisateur en utilisant :

- La valeur recalculée des poids des critères à l'étape 2
- La valeur correspondante à l'évaluation de chaque critère, suite à réalisation de la compétence par un acteur donné, dans les projets passés. Il s'agit, à posteriori et en d'autres termes, d'évaluer la concrétisation de la compétence par un acteur sur la base de ses compétences élémentaires.

Considérons que  $f$  soit le nombre de fois où un acteur a concrètement mis en œuvre une compétence dans des projets par le passé, et que  $E_{k(a,j)}$  soit la valeur d'évaluation du critère  $k$  pour le projet  $j$ . La valeur moyenne des évaluations du critère  $k$ , mémorisée au fil des différents projets passés, nous est alors donnée par l'équation (2):

$$\overline{E}_{k(a,t)} = \frac{\sum_{j=1}^f E_{k(a,j)}}{f} \quad (2)$$

Finalement, pour une compétence donnée et au fil du temps, nous définissons « l'évaluation globale d'un acteur » à travers l'équation (3) :

$$Eval(a) = \sum_{k=1}^t \overline{W}_{k(t)} \times \overline{E}_{k(a,t)} \quad (3)$$

Le sens du tri de la liste final des acteurs qui est retournée à l'utilisateur repose sur ces valeurs calculées. La méthode proposée ici normalise les notations, i.e. qu'un acteur ayant systématiquement des évaluations « faibles » n'est pas pénalisé. De même, et dans tous les cas, l'utilisateur pourra toujours choisir de confier la réalisation de la compétence à un utilisateur qui ne se situe pas en tête de liste. En lien avec le cycle de vie de AC, aux âges d'enfance et d'adolescence, une demande de validation du choix de l'acteur est faite par l'utilisateur. Le stockage de la décision humaine induit que les poids calculés à l'équation (2) sont mémorisés et considérés comme pertinents. Il y a apprentissage avec adaptation des valeurs des poids au fil du temps. On notera également que lorsque l'agent compétence est « mature », cette validation est considérée comme implicite. La méthode d'apprentissage choisie est aussi bien représentative de l'univers professionnel modélisé dans le SMA.

### **Apprentissage pour l'établissement des liens inter-agents**

Un utilisateur peut rechercher les compétences potentiellement liées à celle qu'il souhaite mettre en œuvre. C'est souvent le cas lorsqu'on ne connaît pas la liste exhaustive des actions à réaliser pour mener à bien un projet complet et que ledit projet n'est pas décrit. Dans notre proposition, le comportement d'établissement de liens inter-agents peut être déclenché de 2 manières :

1. Suite à réception d'une requête utilisateur, i.e. de type évènementiel.
2. Suite à déclenchement d'une règle comportemental (cf. plus haut), i.e. de manière automatique & autonome.

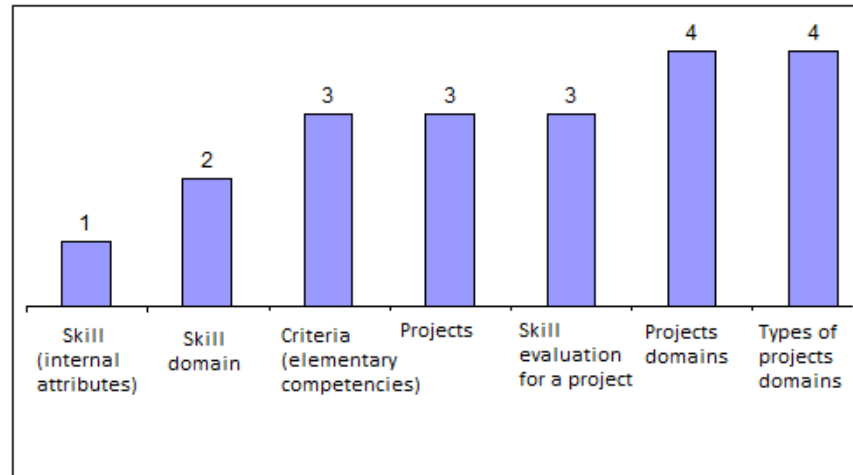
Illustrons le premier cas par un exemple. Un utilisateur, qui travaille dans une mairie, souhaite mener une campagne de sensibilisation de la population sur le sujet des éoliennes. Ce sujet n'a pas, par le passé, déjà fait l'objet d'un projet. En dehors de compétences purement techniques sur les éoliennes, notre utilisateur sait qu'il a besoin d'une compétence en organisation d'évènements. Il va chercher à identifier celles qui lui manquent pour organiser son projet. Il adresse une requête à l'AC « organisation d'évènement » qui lui renverra la liste des compétences qui lui sont liés, comme par exemple « réaliser campagne de communication auprès des citoyens » ou bien « réserver la salle polyvalente ».

Le second cas, quand à lui, est déclenché de manière périodique et autonome par l'exécution d'une règle comportementale. Chaque AC, au minimum à l'âge d'adolescence, tente d'établir des relations avec les autres au sein du SMA. L'identification des liens met en œuvre des mécanismes d'apprentissage en 3 étapes.

#### **Etape 1 : Recherche des similitudes.**

Pour un agent compétence S, l'idée est de calculer pour tous les autres ACs, un indicateur nommé « *coefficient de proximité* » à S et exprimé en pourcentage. Ce coefficient est calculé suite à la recherche de « *similarités* » à S, i.e. d'éléments que les autres agents ont en commun avec S. Ce pourront être, par exemple : un projet commun, une compétence élémentaire partagée, des mots clés similaires, etc. Pour ce faire, S

analyse dynamiquement et récursivement l'arbre de la base de données (i.e. l'environnement). La « profondeur » d'une similarité correspond au nombre de nœuds de type « table » qu'il a fallu parcourir pour atteindre la donnée similaire. Cette profondeur est un bon indicateur du degré de similarité. Une analogie peut être faite avec les hyperliens sur Internet. Le niveau de similarité entre les pages peut être estimé par rapport au nombre de liens qu'il a fallu suivre entre les pages. On fournit une illustration graphique des similarités des profondeurs associées sur la Figure 1.



**Figure 1** Etablissement de liens inter-agents : profondeur & similarités

Intuitivement, ce graphique tend bien à montrer des niveaux d'importance allant du plus faible (à gauche) au plus fort (à droite). Effectivement, plus la profondeur est faible, plus les similitudes peuvent être considérées comme « significatives ».

**Etape 2** : Calcul du coefficient de proximité avec les hypothèses suivantes :

- **D** est la profondeur maximale de l'arbre, i.e. que toute compétence peut être atteinte à partir d'une autre si l'on suit au maximum 4 liens.
- **d=1** comme étant la profondeur de la table courante.
- L'analyse récursive débute à partir de la table des agents, pour la compétence S.

Algorithmiquement, 6 sous-étapes sont enchaînées :

1. Pour chaque valeur de champ de l'enregistrement dans la table courante, une requête SQL est générée dynamiquement pour retrouver les autres compétences partageant cette valeur.
2. Si une compétence  $S_i$  est trouvée, la première fois, son coefficient de proximité  $P_{S_i} = 0$
3. Considérant **n** comme étant le nombre de champs où des valeurs communes ont été trouvées,  $P_{S_i}$  est mis à jour selon l'équation (4) :

$$P_{S_i} = P_{S_i} + (n \times (D - d)) \quad (4)$$

4. Les clés étrangères et les tables de jointures de la table courante sont utilisées pour passer au prochain nœud de l'arbre des tables, i.e. à la table suivante. **d** est incrémentée : **d=d+1**
5. Si **d** est inférieur ou égale à **D**, boucler sur le point 3.
6. Normaliser tous les  $P_{S_i}$  avec l'équation (5) :

$$P_{Si} = \frac{P_{Si}}{P_{Si \max}} \times 100 \quad (5)$$

### **Etape 3** : Renvoi du résultat vers l'utilisateur & demande de validation

Une valeur de seuil est appliquée sur les  $P_{Si}$  (e.g. 65% arbitrairement et par défaut) puis, la liste des compétences trouvées est renvoyée à l'utilisateur pour validation. Cette validation est demandée de 2 manières différentes à l'âge de l'adolescence:

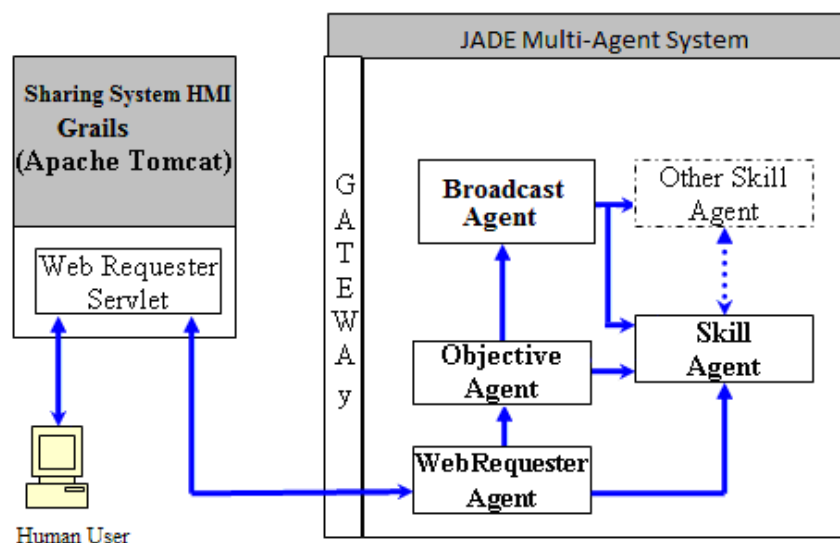
1. Directement à l'écran si c'est l'utilisateur qui effectue la recherche de lien de manière active
2. Par mail à l'administrateur de la Bourse aux Compétences si la recherche de liens est déclenchée par l'agent de manière autonome.

Dans les 2 cas, la validation est associée à la création dynamique d'une règle comportementale, ce qui implique une évolution du comportement même de l'agent. Lorsque l'AC est « *mature* », il est entièrement autonome. Il crée alors ses propres liens entre les ACs et les règles comportementales associées sans demander de validation à quiconque. Nous sommes bien ici dans un système apprenant, qu'il y ait passage ou non par une validation humaine intermédiaire. L'une des pistes de travail à explorer est l'optimisation du calcul de la valeur de seuil du coefficient de proximité. Actuellement, l'utilisateur peut le définir, s'il le souhaite, et transmettre sa valeur au système parmi les paramètres qui lui sont envoyés. Dans le cas contraire, c'est la valeur par défaut (positionnée arbitrairement à 65%) qui s'applique.

## **IMPLEMENTATION ET RESULTATS**

### **Implémentation**

Le modèle proposé dans cet article a été concrètement réalisé en utilisant le SMA JADE [8] et ses outils standards. Côté IHM utilisateur, les développements de la BCDD au CG33 ont été réalisés via l'utilisation des technologies GRAILS. Voir Figure 2.



**Figure 2** Architecture de notre Bourse aux Compétences

Tout AC possède une table mémoire dédiée en base de données où il stocke les paramètres entrants et les décisions qu'il prend. Les règles comportementales de tout agent compétence AC sont stockées au format XML et possèdent une grammaire spécifique (arborescence, attributs, tags). Voir tableau N°1.



Balise XML	Attribut	Obligatoire	Commentaire
rules	description	X	Balise principale
	sensor	X	Dans l'implémentation, désigne le nom de la classe java pilotant (pool de thread) tous les capteurs de l'AC
ruleGroup	description		Texte décrivant le groupe de règles
	weight		Poids par rapport aux autres groupes de règles
rule	description	X	Texte décrivant la règle
	weight		Poids par rapport aux autres règles dans le groupe
	mandatory		1 si la règle est obligatoire, 0 sinon
when	description		Texte décrivant la condition de déclenchement
	sensor	X	Capteur de l'AC à utiliser (e.g. nom de la classe java dans notre implémentation) pour vérifier que la condition de déclenchement est remplie
	params		Liste des paramètres du capteur au format nom=valeur. Le séparateur de liste est le caractère
	result		Nom de la variable, débutant par le caractère \$, contenant le résultat de l'exécution du capteur
	operator		Opérateur logique dans l'expression de la condition
	table		Dans l'implémentation, désigne le nom de la table du modèle de données où la condition doit être vérifiée
	field		Dans l'implémentation, désigne le nom du champ de la table du modèle de données où la condition doit être vérifiée
	value		Dans l'implémentation, désigne la valeur du champ de la table du modèle de données où la condition doit être vérifiée. Cette valeur est exprimée sous la forme d'une expression régulière (standard java)
otherwise			Clause utilisée si la condition exprimée dans <when> n'est pas vérifiée.
do	description		Texte décrivant le comportement déclenché
	effector	X	Effecteur de l'AC à utiliser (e.g. nom de la classe java dans notre implémentation) pour réaliser le comportement voulu
	params		Liste des paramètres de l'effecteur au format nom=valeur. Le séparateur de liste est le caractère
	result		Nom de la variable, débutant par le caractère \$, contenant le résultat de l'exécution de l'effecteur

**Tableau 1 : Grammaire XML du modèle comportemental**

### Simulateur de sélection d'acteurs

En Gironde, 61 autorités locales se sont regroupées au sein d'un « réseau du développement durable ». Depuis quelques mois, via l'outil BCDD, ils partagent des expériences et des compétences. Un simulateur automatisé concernant l'identification des listes d'acteur a été aussi été mis en œuvre. Son écran utilisateur principal de paramétrage est fourni Figure 3.

Paramétrage du simulateur	Valeurs des paramètres
Compétence(s)	Choisir de manière aléatoire parmi toutes les compétence(s) <input checked="" type="radio"/> Oui <input type="radio"/> Non
Nombre global d'itérations de simulation à générer	Répartir les simulations aléatoirement entre les types d'itération <input checked="" type="radio"/> Oui <input type="radio"/> Non 1
Types d'itération à générer	<input checked="" type="checkbox"/> (200) Agents Compétences : Trouver meilleur Acteur

Simuler !

**Figure 3 Ecran de paramétrage du simulateur**

La simulation est basée sur l'appel d'un agent compétence unique dans le but d'observer ses évolutions comportementales au fil du temps. Pour chaque tir de simulation, les poids des différentes compétences élémentaires de l'AC (4 au total) sont positionnés de manière aléatoire pour refléter des variations de niveau d'importance. Lorsque l'on récupère la liste d'acteurs potentiels pour l'AC, une validation aléatoire est également générée dans le but de refléter les variations de décisions humaines. Le simulateur a été utilisé de nombreuses fois avec des scénarios à 100, 200 puis 300 tirs.

### Résultats

On notera tout d'abord qu'une expérimentation grandeur nature est en cours au CG33 depuis fin 2013 pour la BCDD, avec un SMA contenant 110 agents compétence. Nous ne disposons pas à ce jour d'assez de données significatives pour les présenter. Cependant, les premiers résultats observables semblent encourageants. Les ACs (à travers les calculs de pondération de leurs  $C_{et}$ ) évoluent et répondent bien aux sollicitations des utilisateurs membres du réseau des Agenda 21. L'utilisation du simulateur, quand à elle, nous conduit à vérifier les hypothèses formulées pour la méthode de calcul des pondérations des  $C_{et}$  et leur convergence au fil du temps. Voir la Figure 4.

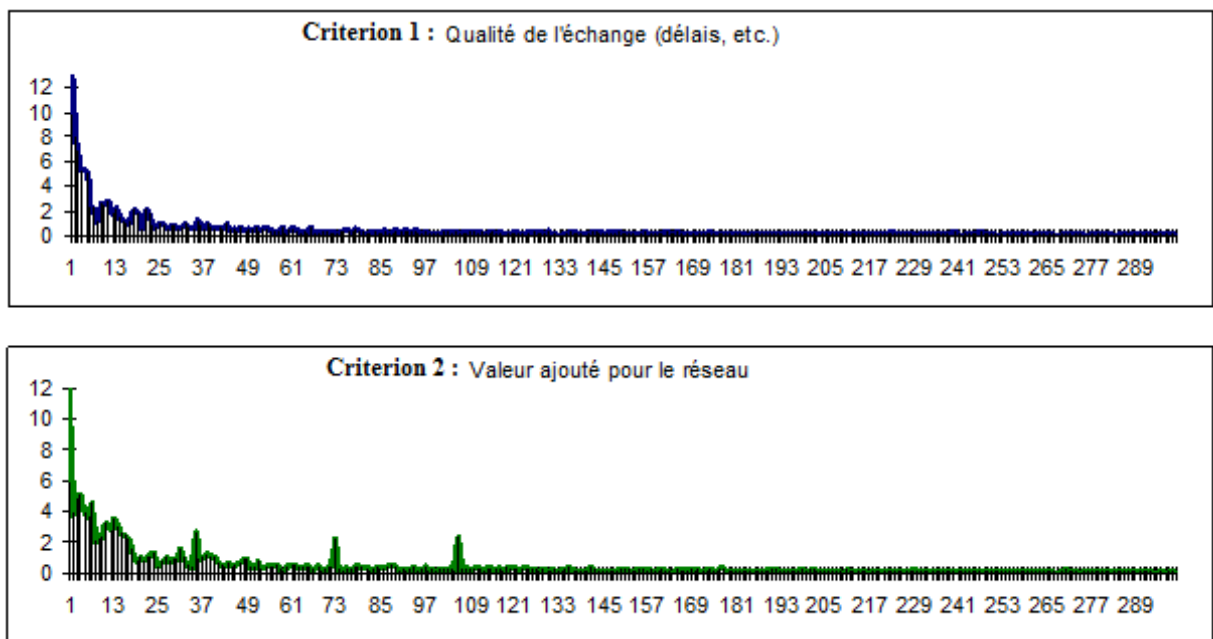


Figure 4 Convergence des valeurs des pondérations des  $C_{et}$

Nous pouvons également vérifier l'efficacité du mécanisme d'apprentissage pour les calculs de pondération des  $C_{et}$  via l'atteinte d'une certaine stabilité, observable aux alentours des 120 requêtes et validation prononcées par les utilisateurs. Voir la Figure 5.

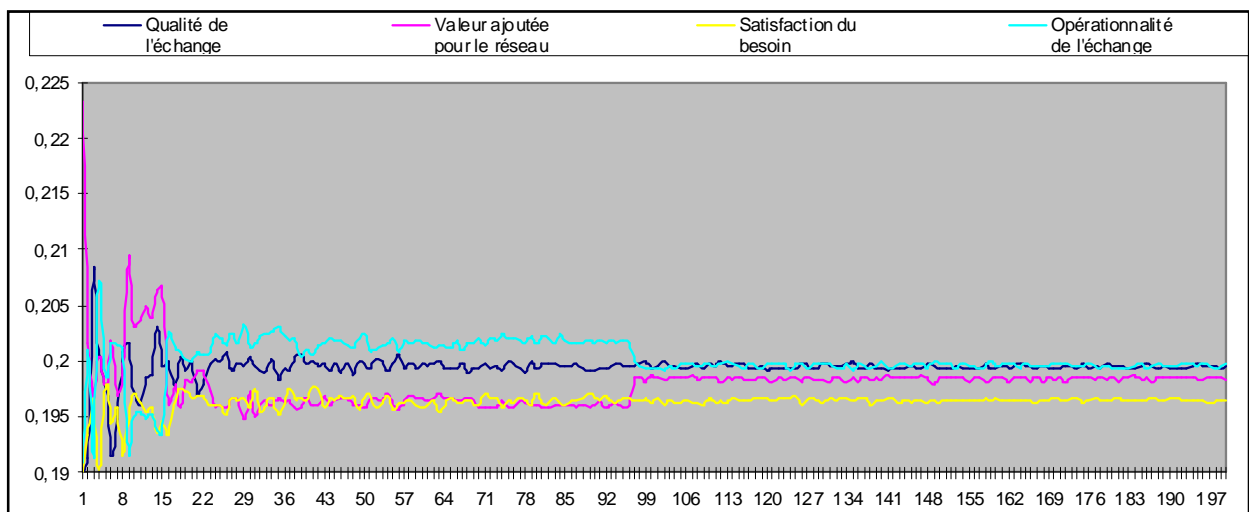


Figure 5 Stabilisation des valeurs des pondérations des  $C_{et}$

La Figure 5 nous permet également de constater que les poids de certains critères sont plus élevés que d'autres au fil des simulations. D'un point de vue théorique, il n'est en effet pas insensé de dire qu'une  $C_{et}$  prend une part plus importante qu'une autre dans la définition globale de la compétence modélisée par l'AC. Si ce constat est vrai en environnement simulé, il est également valable dans le concret où l'on verra, au fil du temps, se dégager les  $C_{et}$  les plus importantes dans la définition de chaque compétence. Finalement, pour démontrer une application possible à tout domaine professionnel en dehors des projets de DD, le système a été testé dans le cadre de la sélection de joueurs dans une équipe de rugby.

## CONCLUSION

Généralement, pour des agents dans un SMA, les compétences ne sont vues que comme des capacités mises en œuvre à travers des règles comportementales qui en définissent les actions. Ici, les compétences sont des agents apprenants. Leur motivation est de s'auto-définir (liste de  $C_{et}$ ) et d'établir des liens entre elles. Une des conséquences est que la subjectivité de choix d'un acteur (tendance à vouloir élire toujours le même) disparaît, les compétences étant placées au cœur d'un système générique à tout domaine fonctionnel. Les notions de réciprocité, de coopération humaine, de motivation personnelle, d'autonomie d'action et de prises de décision, ainsi que de montée en compétence des acteurs humains sont mises en avant. Allons plus loin : considérant qu'une compétence  $C$  puisse être transversale (de part les  $C_{et}$  qui la compose) à des domaines fonctionnels, des institutions publiques ou des entreprises quel que soit le secteur d'activité, quid d'une réflexion à une nouvelle organisation du travail et un nouveau modèle social des relations au travail ?

### **Références bibliographiques : (15 maximum)**

[1] Bardou, M. 2011. De la stratégie à l'évaluation : des clés pour réussir un Agenda 21 local ("From the strategy to the assessment: the keys to succeed in the elaboration of a local Agenda 21"). Collection "Références" du Service de l'Économie, de l'Évaluation et de l'Intégration du Commissariat Général au Développement Durable, Paris, France

[2] Ferber, J. 1999. Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison Wesley, London UK

[3] Jennings, N.R., Wooldridge M., and Sycara K. 1998. A roadmap of agent research and development. Int Journal of Autonomous Agents and Multi-Agent Systems. Kluwer Academic Publishers, Boston, USA

[4] Weiss G. 2013. Multiagent Systems, a Modern Approach to Distributed Artificial Intelligence. MIT Press, USA

[5] Mulder, M., Weigel, T. and K. Collins 2007. The concept of competence in the development of vocational education and training in selected EU member states: a critical analysis. Journal of Vocational Education and Training , vol. 59, 1, pp. 67-88, London, UK

[6] Pfeifer R. and Bongard J.C. 2007. How the body shapes the way we think: a new view of intelligence. The MIT Press, USA

[7] Varela F., Thompson E. and Rosch E. 1991. The Embodied Mind: Cognitive Science and Human Experience. MIT Press, USA

[8] Bellifemine F., Poggi A., and Rimassa G. 1999. "JADE – A FIPA-compliant agent framework", 4th International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-99), London, UK

- [9] FIPA, “Foundation for Intelligent Physical Agents - Abstract Architecture Specification (Standard Version)”, publication des Technical Committee de la FIPA, Geneva, Switzerland, 2002
- [10] Putterman M. L. 1994. “Markov Decision Processes. Discrete stochastic dynamic programming”, Wiley-Interscience, New York, USA
- [11] Mitchell M. 1997, “Machine Learning”, chap. 13 “Reinforcement Learning”, p. 367-390, USA
- [12] Von Neumann O. and Morgenstern O. 1944, “Theory of Games and Economic Behaviour”, Princeton University Press, USA
- [13] Shapley L. 1953, “Stochastic games”, Proceedings of National Academy of Science, 39:1095–1100, USA
- [14] Boutilier C., Dean T. and Hanks S. 1999, “Decision theoretic planning: Structural assumptions and computational leverage”, Journal of Artificial Intelligence Research, USA
- [15] Plaza E. and Ontañón S. 2002, “Cooperative multi-agent learning in Adaptive Agents and Multi-Agent Systems”, volume 2636 of Lecture Notes in Computer Science, pages 1-17, USA