

Université, Recherche et Développement (URED) Numéro spécial Actes du CNRIA'2012



**URED**

**UNIVERSITÉ, RECHERCHE ET DÉVELOPPEMENT**

Revue Pluridisciplinaire de  
l'Université Gaston Berger de Saint-Louis, Sénégal

**Numéro Spécial 2012**

Actes du CNRIA'2012  
Colloque International sur la Recherche  
en Informatique et ses applications

4<sup>e</sup> édition  
Du 25 au 27 avril 2012 à Thiès et Bambey, Sénégal

© Presses Universitaires de Saint-Louis  
ISSN : 0850-2161

SÉRIE SCIENCES EXACTES



*UNIVERSITÉ, RECHERCHE ET DÉVELOPPEMENT  
(URED)*

Revue pluridisciplinaire  
de l'Université Gaston Berger de Saint-Louis  
Sénégal

Série des sciences exactes

NUMÉRO SPÉCIAL

Actes du CNRIA' 2012

Colloque National sur la Recherche  
en Informatique et ses Applications

4<sup>e</sup> édition

Du 25 au 27 avril 2012 à Thiès et Bambey, Sénégal

© Presses Universitaires de Saint-Louis, 2012

**ISSN : 0850-2161**

**UNIVERSITÉ, RECHERCHE ET DÉVELOPPEMENT (URED)**  
**Revue pluridisciplinaire de l'Université Gaston Berger de Saint-Louis du Sénégal**

B.P. : 234 Saint-Louis - Tél.(221) 339611906/9611768 - Fax (221) 33339611884

Email : ured@ugb.sn

La revue *URED* paraît deux fois par an (janvier et juin)

**PRÉSIDENT D'HONNEUR** Me Abdoulaye WADE, Président de la République du Sénégal

**DIRECTEUR DE PUBLICATION** : Pr Mary-Teuw Niane, Recteur de l'Université Gaston Berger de Saint-Louis

**COMITE SCIENTIFIQUE**

Abdoullah CISSE (Sénégal)	Mawéja MBAYA (Sénégal)
Ibrahim Khalil CISSE (Sénégal)	Gora MBODJ (Sénégal)
Yves CHARBIT (France)	Ahmadou Lamine NDIAYE (Sénégal)
Souleymane Bachir DIAGNE (États-Unis)	Mary Teuw NIANE (Sénégal)
Adama DIAW (Sénégal)	Modeste NZI (Côte d'Ivoire)
Jean-Jacques GERVAIS (Canada)	Youssef OUKNINE (Maroc)
Abdoulaye Elimane KANE (Sénégal)	Abdou Salam SALL (Sénégal)
Babacar KANTE (Sénégal)	Mamadou Moustapha SALL (Burkina Faso)
Issiaka Prosper LALEYE (Sénégal)	Ndiawar SARR (Sénégal)
Dominique LAURENT (France)	Ben Sikini TOGUEBAYE (Sénégal)
Gane Samb LO (Sénégal)	Hamidou TOURE (Burkina Faso)
Claude LOBRY (France)	Samba TRAORE (Sénégal)
Edris MAKWARD (Sénégal)	Thierry VERHELST (Belgique)

**COMITE DE RÉDACTION**

Rédacteur en chef	: Mamadou CAMARA
Membres d'honneur	: Issiaka-Prospér LALEYE & Baydallaye KANE
Secrétaire de Rédaction	: Daouda MAR
Finances	: Adama BARRY
Relations extérieures	: Gora MBODJ
Composition, Impression et Diffusion	: Malamine DIOUF, Amsata NDIAYE, Mamadou Youri SALL, Mamadou SY
Secrétaire chargée de l'administration	: Awa Tall GUEYE

**Couverture** : Estampillage et photo (médaillon) de la  
Bibliothèque de l'Université Gaston Berger de Saint-Louis du Sénégal.

**ABONNEMENTS**

Sénégal et Afrique	: Abonnement annuel 5000 F. CFA pour les deux séries Abonnement annuel 3000 F. CFA pour une série
Autres pays	: Abonnement Annuel 25 € pour les deux séries Abonnement Annuel 15 € pour une série

**PRIX D'UN NUMÉRO**

Sénégal et Afrique	: 2000 F. CFA + 1000 F. CFA de frais d'envoi
Autres pays	: 18 € (frais d'envoi compris)

**ABONNEMENT DE SOUTIEN**

40 €

Les règlements (virements et mandats) devront être libellés à l'ordre de Monsieur l'Agent Comptable de l'Université Gaston Berger de Saint-Louis pour le compte de la revue URED : compte n° 9531 790 080 /64 B.I.C.I.S. Saint-Louis, Sénégal.

NB : Les Opinions émises dans les articles n'engagent que leurs auteurs.

Authors are solely responsible for the views expressed in their articles.

© Presses Universitaires de Saint-Louis

ISSN : 0850-2161

4<sup>ème</sup>

EDITION

# CNRIA '2012



Du 25 au 27 Avril à Thiès et Bambey  
SÉNÉGAL

**PROCEEDINGS (ACTES)**

**COLLOQUE NATIONAL SUR LA RECHERCHE  
EN INFORMATIQUE ET SES APPLICATIONS**



## **Préface**

Le CNRIA (Colloque National sur la Recherche en Informatique et ses Applications) a connu, depuis sa création, trois (03) éditions qui se sont tenues respectivement en 2005, 2010 et 2011. Ces différentes éditions ont compté en moyenne une vingtaine de communications dans le domaine des réseaux, des bases de données, des systèmes distribués, de la modélisation, etc.

Pour la présente édition, dix-neuf articles ont été retenus par le comité scientifique. Deux conférences invitées sont aussi prévues.

La première édition du CNRIA s'est tenue en avril 2005 à Saint-Louis dans le cadre de la composante 2 du Projet d'Appui à l'Enseignement Supérieur Professionnalisé au Sénégal (PROSENSUP). Il s'agissait pour nous (i) de favoriser la création d'un pôle de recherche sur les technologies de l'information et de la communication (TIC); (ii) de renforcer et de créer des partenariats entre chercheurs sénégalais en informatique; et (iii) de valoriser la recherche nationale dans le domaine des NTIC.

Mais force est de reconnaître que le colloque qui se voulait national a pris dès sa première édition une dimension internationale avec la participation de collègues venus de la Mauritanie, du Cameroun, de la France et du Québec. Ce qui est normal dans le contexte actuel de la recherche.

5 ans après cette première édition, l'organisation de la seconde édition en avril 2010 à Saint-Louis a permis de relancer le colloque et la troisième édition qui s'est tenu à Dakar en avril 2011 a été un réel succès.

Il faut dire que le CNRIA est devenu aujourd'hui un évènement scientifique important pour les chercheurs informaticiens, et pas seulement en tant que manifestation scientifique mais surtout en tant qu'outil de partenariat et d'échange. En effet, compte tenu de la jeunesse du domaine mais aussi des chercheurs du domaine et du faible taux d'encadrement au niveau national, nous pensons que la recherche en

informatique doit se mener au Sénégal à travers d'équipes qui transcendent les frontières institutionnelles. Ce n'est que comme ça que nous pourrions atteindre une masse critique de chercheurs dans le pays.

Notre souhait est alors de favoriser la synergie entre les chercheurs et, dans le moyen terme, d'arriver à la mise en place de formations doctorales nationales en informatique avec des pôles spécialisés en fonction des universités.

Nous remercions tous les chercheurs qui ont répondu à notre appel à candidatures avec une mention spéciale pour les conférenciers invités.

Nous félicitons tous les auteurs qui ont vu leur article retenu par le comité scientifique et remercions par la même occasion tous les membres du comité scientifique qui n'ont ménagé aucun effort pour effectuer une relecture des articles qui leur ont été affectés, et ce, dans des délais relativement courts.

Nous terminerons par remercier nos bailleurs, en particulier, les différents Recteurs, la DGES et l'AUF (via le projet KOCC), ainsi que tous ceux qui ont contribué, de près ou de loin, à la réussite de CNRIA'2012.

***Le comité de pilotage du CNRIA***

**Pr. Moussa LO, Pr. Ibrahima NIANG & Dr. Mouhamadou THIAM**

### Comité de pilotage national

Ousmane SALL	Université de Thiès
Dame SAMB	Université de Thiès
Mouhamadou THIAM	Université de Thiès
Cheikh BA	Université de Ziguinchor
Gaoussou CAMARA	Université Alioune Diop de Bambey
Adrien BASSE	Université Alioune Diop de Bambey
Youssou KASSE	Université Alioune Diop de Bambey
Maïssa MBAYE	Université Gaston Berger de Saint-Louis
Jean Marie DEMBELE	Université Gaston Berger de Saint-Louis
Moussa Lo	Université Gaston Berger de Saint-Louis
Ousmane Thiare	Université Gaston Berger de Saint-Louis
Samba Ndiaye	Université Cheikh Anta Diop de Dakar
Ibrahima Niang	Université Cheikh Anta Diop de Dakar
Bamba Gueye	Université Cheikh Anta Diop de Dakar
Idrissa Sarr	Université Cheikh Anta Diop de Dakar
Gervais Mendy	Université Cheikh Anta Diop de Dakar
Khadiatou Wane Keita	Université Cheikh Anta Diop de Dakar
Amadou Gaye	Université Cheikh Anta Diop de Dakar

### Comité scientifique

Arnaud Giacometti	Université de Tours
Alassane DIOP	Télé-Université du Québec
Alassane BA	Ecole Supérieure Polytechnique de Dakar
Almamy KONTE	Université Cheikh Anta Diop de Dakar
Amadou COULIBALY	INSA Strasbourg
Abdourahmane FAYE	SAP Suisse
Amrane HOCINE	Université de Pau et des Pays de l'Adour
Bamba GUEYE	Université Cheikh Anta Diop de Dakar
Béatrice Bouchou	Université de Tours
Borilli M. Jonas SOME	Ecole Polytechnique de Bobodioulasso
Catherine Faron ZUCKER	Université de Nice
Cheikh BA	Université de Ziguinchor
Cheikh SARR	Université de Thiès
Cheikh Talibouya DIOP	Université Gaston Berger de Saint-Louis
Chérif DIALLO	Consultant Sécurité des SI, Paris
Demba COULIBALY	Université de Bamako
Dominique Laurent	Université de Cergy-Pontoise
Fatou Kamara SANGARE	Université Gaston Berger de Saint-Louis

CNRIA 2012

Gayo DIALLO	Université Bordeaux Segalen
Ibrahima NGOM	Ecole Supérieure Polytechnique de Dakar
Ibrahima NIANG	Université Cheikh Anta Diop de Dakar
Idrissa SARR	Université Cheikh Anta Diop de Dakar
Jean Marie DEMBELE	Université Gaston Berger de Saint-Louis
Kader KEITA	Université de Bamako
Khadidiatou Wane KEITA	Ecole Supérieure Polytechnique de Dakar
Khalifa GAYE	Université de Ziguinchor
Maissa MBAYE	Université Gaston Berger de Saint-Louis
Mamadou BOUSSO	Université de Thiès
Mamadou Youry SALL	Université Gaston Berger de Saint-Louis
Marcellin Konan BROU	Institut National Polytechnique de Yamoussoukro
Marie NDIAYE	Université de Ziguinchor
Maurice TCHUENTE	Université de Yaoundé I
Mouhamadou THIAM	Université de Thiès
Moussa LO	Université Gaston Berger de Saint-Louis
Nacera BENACER	Supélec, Paris
Nathalie PERNELLE	LRI Paris
Ndeye Fatou NGOM	Université Cheikh Anta Diop de Dakar
Olivier CORBY	INRIA Sophia Antipolis
Olivier Flouzac	Université de Reims
Olivier MONGA	IRD Yaoundé
Oumar DIOP	Université Gaston Berger de Saint-Louis
Ousmane SALL	Université de Thiès
Ousmane THIARE	Université Gaston Berger de Saint-Louis
Patrick MARCEL	Université de Tours
Rédouane LBATH	Université de Toulouse I
Rim Djedidi	Lim&Bio Paris 13
Sadouaouan MALO	Ecole Polytechnique de Bobodioulasso
Samba DIAW	Université de Ziguinchor
Samba NDIAYE	Université Cheikh Anta Diop de Dakar
Seydina Moussa NDIAYE	Université Gaston Berger de Saint-Louis
Sylvie DEPRES	Lim&Bio Paris 13
Thierno TOUNKARA	Telecom Bretagne
Yahya SLIMANI	Université de Tunis



## Table of Contents

Clustering auto-stabilisant à k sauts dans les réseaux Ad Hoc.....	1
<i>Mandicou Ba, Olivier Flauzac, Bachar Salim Hagggar, Florent Nolot and Ibrahima Niang</i>	
Modélisation ontologique des systèmes complexes : Cas de la propagation de la bilharziose au Sénégal. ....	9
<i>Gaoussou Camara, Sylvie Despres, Rim Djedidi and Moussa Lo</i>	
Un modèle à base d'agents de la propagation de la bilharziose à Ndombo. ....	17
<i>Papa Alioune Cisse, Gaoussou Camara and Moussa Lo</i>	
L'exploration des réseaux de transport : la puissance de l'attente aux stations.....	25
<i>Ilcinkas David and Wade Ahmed Mouhamadou</i>	
ProfMiner:algorithme de construction de profil utilisateur .....	33
<i>Mouhamadou Saliou Diallo, Arnaud Giacometti, Arnaud Soulet, Haoyuan D Li, Cheikh Talibouya Diop and Sandra de Amo</i>	
Choix méthodologiques pour la construction d'une ontologie du domaine de la vulnérabilité urbaine face aux aléas liés aux changements climatiques. ....	41
<i>Ibrahima Diop, Adrien Coly and Moussa Lo</i>	
Modèle de Données multidimensionnel sur les interactions environnement-santé : Cas de la Fièvre de la Vallée du Rift au Sénégal .....	49
<i>Bouba Fanta, Bah Alassane, Jacques André Ndione, Samba Ndiaye and Christophe Cambier</i>	
Optimisation d'un Protocole d'Authentification dans les Réseaux de Capteurs Sans FIL ..	57
<i>Youssou Faye and Ibrahima Niang</i>	
GRAPPS Data Grid Un système Pair-à-Pair de stockage des données.....	65
<i>Olivier Flauzac, Luiz Angelo Steffanel, Diallo Thierno Ahmadou, Ibrahima Niang and Samba Ndiaye</i>	
Organisation virtuelle: autorisation d'accès aux ressources partagées .....	73
<i>Amadou Dahirou Gueye, Alassane Diop and Samuel Ouya</i>	
Services pour les grilles pair-à-pair .....	81
<i>Bassirou Gueye, Olivier Flauzac and Ibrahima Niang</i>	
Passage d'un modèle acteur à un modèle multi-agent pour la gestion des ressources naturelles: Utilisation du méta-modèle d'acteur Actor-In-Context .....	89
<i>Eric Désiré Kameni, Eric Fotsing and De Groot Wouter</i>	
Three-dimensional model for soil pore space using geometrical primitives .....	97
<i>Ndeye Fatou Ngom and Olivier Monga</i>	
Approche pour la gestion de la QoS des applications et services dans les Clouds .....	104
<i>Mohamed Mahmoud Ould Deye and Yahya Slimani</i>	
PFair scheduling on a multiprocessor platform for late released tasks, with constrained deadlines.....	112
<i>Malo Sadouanouan, Choquet-Geniet Annie and Bikienga Moustapha</i>	

## Clustering auto-stabilisant à $k$ sauts dans les réseaux Ad Hoc

Mandicou BA <sup>\*</sup>, Olivier FLAUZAC <sup>\*</sup>, Bachar Salim HAGGAR <sup>\*</sup>,  
Florent NOLOT <sup>\*</sup> — Ibrahima NIANG <sup>\*\*</sup>

<sup>\*</sup> Université de Reims Champagne-Ardenne  
UFR Sciences Exactes et Naturelles  
CReSTIC - SysCom EA 3804  
{ mandicou.ba, olivier.flauzac, bachar-salim.haggar, florent.nolot }@univ-reims.fr

<sup>\*\*</sup> Université Cheikh Anta Diop  
Département de Mathématiques et Informatique  
Laboratoire d'Informatique de Dakar (LID)  
iniang@ucad.sn

**RÉSUMÉ.** Les réseaux *ad hoc* offrent de nombreux domaines d'application du fait de leur facilité de déploiement. La communication qui s'effectue classiquement par diffusion est coûteuse et peut entraîner une saturation du réseau. Pour optimiser ces communications, une approche est de structurer le réseau en *clusters*. Dans cet article, nous présentons un algorithme de *clustering* asynchrone, distribué et auto-stabilisant qui construit des *clusters* à  $k$  sauts. Notre approche ne nécessite aucune initialisation. Elle se base uniquement sur l'information provenant des nœuds voisins à l'aide d'échange périodique de messages. Partant d'une configuration quelconque, le réseau converge à un état stable au bout d'un nombre fini d'étapes. Nous montrons que pour un réseau de  $n$  nœuds, la stabilisation est atteinte au plus en  $n + 2$  transitions et nécessite une occupation mémoire d'au plus  $n * \log(2n + k + 3)$ .

**ABSTRACT.** Most Ad Hoc networks use communications based on diffusion that are typically expensive and may cause network saturation. In order to optimize these communications, one approach is to structure networks into clusters. In this paper, we present a self-stabilizing asynchronous distributed algorithm that builds non-overlapping  $k$ -hops clusters. Our approach does not require any initialization. It is based only on information from neighboring nodes with periodic messages exchange. Starting from an arbitrary configuration, the network converges to a stable state after a finite number of steps. We prove that the stabilization is reached after at most  $n + 2$  transitions and uses at most  $n * \log(2n + k + 3)$  memory space, where  $n$  is the number of network nodes.

**MOTS-CLÉS :** Réseaux *ad hoc*, *clustering*, algorithmes distribués, auto-stabilisation.

**KEYWORDS :** Ad hoc networks, clustering, distributed algorithms, self-stabilizing.

---

## 1. Introduction

Dans les réseaux *ad hoc*, la solution de communication la plus utilisée est la diffusion. C'est une technique simple qui nécessite peu de calcul. Mais cette méthode est coûteuse et peut entraîner une saturation du réseau. Pour optimiser cette communication qui est une importante source de consommation de ressources, une solution est de structurer le réseau en *arbres* [1] ou en *clusters* [5].

Le *clustering* consiste à découper le réseau en groupes de nœuds appelés *clusters* donnant ainsi au réseau une structure hiérarchique [8]. Chaque *cluster* est représenté par un nœud particulier appelé *clusterhead*. Un nœud est élu *clusterhead* selon une métrique telle que le degré, la mobilité, l'identité des nœuds, la densité, etc. ou une combinaison de ces paramètres. Plusieurs solutions de *clustering* ont été proposées. Elles sont classées en algorithmes à *1 saut* [2, 3, 7, 10, 9, 11] et à *k sauts* [4, 12]. Cependant, ces approches génèrent beaucoup de trafic et nécessitent d'importantes ressources.

Dans cet article, partant des travaux de Flauzac et *al.* [7], nous proposons un algorithme de *clustering* à *k sauts* qui est complètement distribué et auto-stabilisant. L'auto-stabilisation [6] a été introduite par Dijkstra en 1974 comme étant un système qui, quel que soit sa configuration de départ, est garanti d'arriver à une configuration légale en un nombre fini d'étapes. Notre approche construit des *clusters* non-recouvrants à *k sauts* et ne nécessite pas d'initialisation. Elle se base sur le critère de l'identité maximale des nœuds et s'appuie seulement sur l'échange périodique de messages avec le voisinage à *1 saut*. Le choix de l'identité comme métrique apporte plus de stabilité par rapport aux critères dynamiques comme la densité, la mobilité ou le poids des nœuds.

La suite de l'article est organisée comme suit. Dans la section 2, nous étudions les solutions de *clustering* existantes. La section 3 présente notre contribution. A la section 4, nous décrivons le modèle sur lequel se base notre approche. Puis à la section 5, nous donnons le principe d'exécution et les détails de notre algorithme. Dans la section 6, nous donnons le schéma de la preuve de convergence de notre approche. Une conclusion et des perspectives sont données dans la section 7.

---

## 2. État de l'art

Plusieurs propositions de *clustering* ont été faites dans la littérature [2, 3, 4, 7, 10, 9, 12, 11].

Les approches auto-stabilisantes [7, 10, 9, 11] construisent des *clusters* à *1 saut*. Mitton et *al.* [11] utilisent comme métrique la *densité* afin de minimiser la reconstruction de la structure en cas de faible changement de topologie. Chaque nœud calcule sa densité et la diffuse à ses voisins situés à distance *k*. Flauzac et *al.* [7], ont proposé un algorithme auto-stabilisant qui combine la découverte de topologie et de *clustering* en une seule phase. Elle ne nécessite qu'un unique message échangé entre voisins. Un nœud devient *clusterhead* s'il possède la plus grande identité parmi tous ses voisins. Johnen et *al.* [9] ont proposé un algorithme auto-stabilisant qui construit des *clusters* de taille fixe. Ils attribuent un poids à chaque nœud et fixe un paramètre *SizeBound* qui représente le nombre maximal de nœuds dans un *cluster*. Un nœud ayant le poids le plus élevé devient

*clusterhead* et collecte dans son *cluster* jusqu'à *SizeBound* nœuds. Dans [10], Johnen et al. ont étendu leur proposition décrite dans [9] pour apporter la notion de *robustesse* qui est la propriété qui assure que partant d'une configuration quelconque, le réseau est partitionné après un round asynchrone. Et durant de la phase de convergence, il reste toujours partitionné et vérifie un prédicat de sûreté. Dans [2, 3], Bui et al. ont proposé un algorithme de *clustering* adaptatif aux changements de topologie mais non auto-stabilisant et non déterministe. Ils utilisent une marche aléatoire pour construire d'abord un cœur de *cluster* composé de 2 à *MaxCoreSize* nœuds. Ce cœur est ensuite étendu aux voisins immédiats, appelés nœuds ordinaires, pour former les *clusters*.

Les approches auto-stabilisantes [4, 12] construisent des *clusters* à *k sauts*.

Dans [12], Miton et al. étendent leurs travaux décrits dans [11] pour proposer un algorithme robuste de *clustering* auto-stabilisant à *k sauts*. Avec cet algorithme, si un nœud est trop mobile, il risque de ne rattacher à aucun *cluster*. Datta et al. [4], utilisant une métrique arbitraire, ont proposé un algorithme auto-stabilisant basé sur un modèle à états nommé *k-Clustering*. Cet algorithme est très lent, il s'exécute en  $O(n * k)$  rounds et nécessite  $O(\log(n) + \log(k))$  espace mémoire par processus, où  $n$  est le nombre de nœuds du réseau.

---

### 3. Contribution

Nous proposons une approche basée sur un modèle à passage de messages contrairement aux solutions proposées dans [9, 10, 4]. Notre solution est distribuée et auto-stabilisante. Elle ne nécessite aucune initialisation. Nous utilisons le critère d'identité maximale qui apporte plus de stabilité par rapport aux métriques variables utilisées dans [12, 11, 9, 10]. Notre algorithme structure le réseau en *clusters* disjoints deux à deux et de diamètre au plus égal à  $2k$ . Cette structuration ne nécessite aucune initialisation. Elle combine la découverte de voisinage et le *clustering* en une seule phase. Elle se base uniquement sur l'information provenant des nœuds voisins situés à distance 1 par le biais d'échange périodique de messages. Notre approche se stabilise au plus en  $n+2$  transitions et nécessite une occupation mémoire d'au plus  $n * \log(2n + k + 3)$ .

---

### 4. Modèle

Le réseau peut être modélisé par un graphe non orienté  $G = (V, E)$ , où  $V$  est l'ensemble des nœuds du réseau et  $E$  représente l'ensemble des connexions existantes entre les nœuds. Une arête  $(u, v)$  existe si et seulement si  $u$  peut communiquer avec  $v$  et vice-versa. Ce qui implique que tous les liens sont bidirectionnels. Dans ce cas, les nœuds  $u$  et  $v$  sont voisins. L'ensemble des nœuds  $v \in V$  voisins du nœud  $u$  situés à distance 1 est noté  $N_u$ . Chaque nœud  $u$  du réseau possède un identifiant unique  $id_u$  et peut communiquer avec tout nœud  $v \in N_u$ . On définit la distance  $d_{(u,v)}$  entre deux nœuds  $u$  et  $v$  quelconque dans le graphe  $G$  comme le nombre d'arêtes minimal le long du chemin entre  $u$  et  $v$ .

Dans notre approche, nous utilisons un modèle *asynchrone à passage de messages*. Pour cela, chaque nœud  $u$  envoie périodiquement à ses voisins  $v \in N_u$  un message contenant les informations sur son état actuel. Un message envoyé est reçu au bout d'un temps fini mais non borné. Nous supposons qu'un message envoyé est correctement reçu. Dès

réception d'un message d'un voisin, chaque nœud  $u$  met à jour sa table de voisinage et exécute l'algorithme de *clustering*.

## 5. Algorithme auto-stabilisant à $k$ sauts

### 5.1. Préliminaires

Nous donnons quelques définitions utilisées dans la suite.

**Définition 5.1** (*Cluster*) : Nous définissons un cluster à  $k$  sauts comme un sous graphe connexe du réseau, dont le diamètre est inférieur ou égal à  $2k$ . L'ensemble des nœuds d'un cluster  $i$  est noté  $V_i$ .

**Définition 5.2** (*Identifiant du cluster*) : Chaque cluster possède un unique identifiant correspondant à la plus grande identité de tous les nœuds du cluster. L'identité d'un cluster auquel appartient le nœud  $u$  est notée  $cl_u$ .

Dans nos clusters, chaque nœud  $u$  possède un statut noté  $statut_u$ . Ainsi, un nœud peut être soit *clusterhead* ( $CH$ ), soit *Simple Node* ( $SN$ ), ou soit *Gateway Node* ( $GN$ ). De plus, chaque nœud choisit un voisin  $v \in N_u$ , noté  $gn_u$ , par lequel il passe pour atteindre son  $CH$ .

**Définition 5.3** (*Statut des nœuds*) :

– *Clusterhead* ( $CH$ ) : un nœud  $u$  a le statut de  $CH$  s'il possède le plus grand identifiant parmi tous les nœuds de son cluster :

$$- statut_u = CH \iff \forall v \in V_{cl_u}, (id_u > id_v) \wedge (dist_{(u,v)} \leq k).$$

– *Simple Node* ( $SN$ ) : un nœud  $u$  a le statut de  $SN$  si tous ses voisins appartiennent au même cluster que lui :

$$- statut_u = SN \iff (\forall v \in N_u, cl_v = cl_u) \wedge (\exists w \in V / (statut_w = CH) \wedge (dist_{(u,w)} \leq k)).$$

– *Gateway Node* ( $GN$ ) : un nœud  $u$  a le statut de  $GN$  s'il existe un nœud  $v$  dans son voisinage appartenant un autre cluster :

$$- statut_u = GN \iff \exists v \in N_u, (cl_v \neq cl_u).$$

**Définition 5.4** (*Nœud cohérent*) : Un nœud  $u$  est cohérent si et seulement si, il est dans l'un des états suivants :

$$- Si statut_u = CH alors (cl_u = id_u) \wedge (dist_{(u,CH_u)} = 0) \wedge (gn_u = id_u).$$

$$- Si statut_u \in \{SN, GN\} alors (cl_u \neq id_u) \wedge (dist_{(u,CH_u)} \neq 0) \wedge (gn_u \neq id_u).$$

**Définition 5.5** (*Nœud stable*) : Un nœud  $u$  est dans un état stable si et seulement si, il est cohérent et satisfait l'une des conditions suivantes :

$$- Si statut_u = CH alors \{\forall v \in N_u, statut_v \neq CH\} \wedge \{\forall v \in N_u \text{ tel que } cl_v = cl_u \text{ alors } (id_v < id_u)\} \wedge \{\forall v \in N_u \text{ tel que } id_v > id_u \text{ alors } (cl_v \neq cl_u) \wedge (dist_{(v,CH_v)} = k)\}.$$

$$- Si statut_u = SN alors \{\forall v \in N_u, (cl_v = cl_u) \wedge (dist_{(u,CH_u)} \leq k)\} \wedge \{\exists v \in N_u, (gn_v = id_u) \wedge (dist_{(v,CH_v)} = dist_{(u,CH_u)} + 1)\}.$$

$$- Si statut_u = GN alors \exists v \in N_u, (cl_v \neq cl_u) \wedge \{(dist_{(u,CH_u)} = k) \vee (dist_{(v,CH_v)} = k)\}.$$

### 5.2. Principe d'exécution

Notre algorithme est auto-stabilisant, il ne nécessite aucune initialisation. Partant d'un état quelconque, avec seulement l'échange périodique de messages *hello*, les nœuds

s'auto-organisent en *clusters* non-recouvrants au bout d'un nombre fini d'étapes. Ces messages *hello* contiennent l'identité du nœud ( $id_u$ ), l'identité du *CH* de son *cluster* ( $cl_u$ ), son statut ( $statut_u$ ), et la distance le séparant du *CH* de son *cluster* ( $dist_{(u,CH_u)}$ ). Ainsi, la structure des messages est la suivante :  $hello(id_u, cl_u, statut_u, dist_{(u,CH_u)})$ . De plus, chaque nœud maintient une table de voisinage ( $StateNeigh_u$ ) contenant l'ensemble des états de ses nœuds voisins.  $StateNeigh_u[v]$  contient les états des nœuds  $v$  voisins de  $u$ .

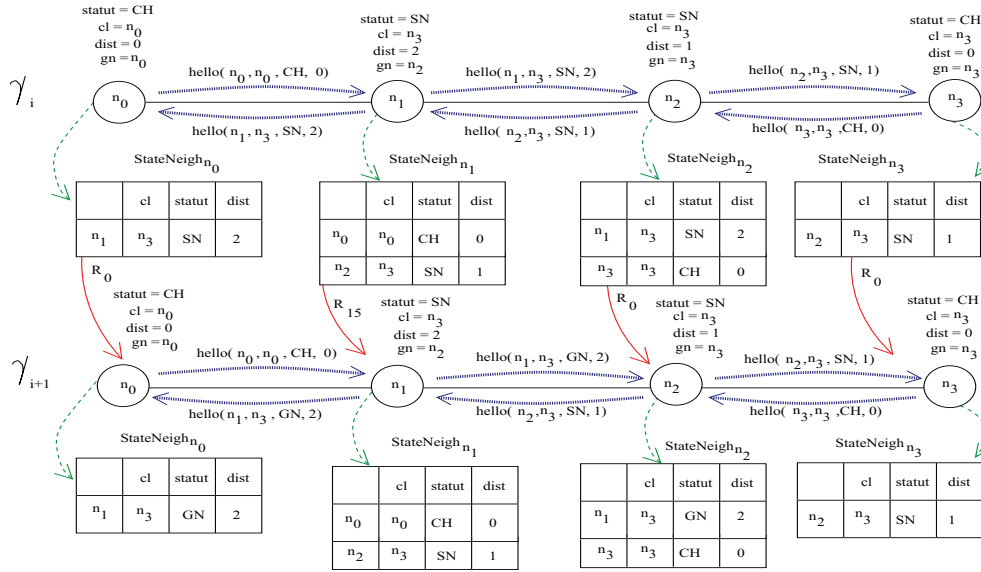


Figure 1. Exemple de clustering à 2 sauts.

La solution que nous proposons se déroule de la façon suivante :

A la réception de messages *hello* de tous ses voisins, chaque nœud met à jour sa table de voisinage puis exécute l'Algorithme 1 et informe tous ses voisins de son éventuel nouvel état. Au cours de l'exécution de l'Algorithme 1, une vérification de la cohérence des informations locales est effectuée à l'exécution de *Cluster-1* puis le traitement de nouvelles informations est effectué à l'exécution de *Cluster-2*. Un nœud  $u$  s'élit *clusterhead* s'il a la plus grande identité parmi tous les nœuds de son *cluster*. Si un nœud  $u$  a un voisin  $v$  *CH*, avec une plus grande identité, alors il devient membre (*SN*) du *cluster* de  $v$ .

Si un nœud  $u$  a un voisin appartenant à un *cluster* d'identité différente à celle de  $u$  et  $v$ . De plus,  $v$  est à distance inférieure à  $k$  de son *CH*.  $u$  en déduit qu'il existe un *CH* à une distance inférieure ou égale à  $k$  et dont l'identité est supérieure à celle de  $v$ . Si de plus l'identité du *cluster* de  $v$  est supérieure à celle de  $u$ , alors  $u$  devient *SN* et appartient au même *cluster* que  $v$ . Un nœud  $u$  devient nœud de passage, statut *GN*, s'il est voisin d'un nœud appartenant à un autre *cluster* ( $cl_v \neq cl_u$ ).

La figure 1 illustre le passage d'une configuration  $\gamma_i$  à  $\gamma_{i+1}$ . A  $\gamma_i$ , chaque nœud envoie à ses voisins un message *hello*.  $\gamma_{i+1}$  est une configuration stable. Dès la réception de messages venant des voisins, chaque nœud met à jour sa table de voisinage puis exécute l'algorithme 1. Dans cet exemple, le nœud  $n_1$  qui est un nœud simple appartenant au

BA et al.

*cluster* de  $n_3$  détecte le nœud  $n_0$  comme un *cluster* voisin. Il devient nœud de passage avec un statut de *GN* et envoie un message *hello* à ses voisins pour une mise à jour. Les nœuds  $n_3$ ,  $n_2$  et  $n_0$ , en fonction de leurs états actuels ainsi que ceux de leurs voisins, ne changent pas d'état.  $\gamma_{i+i}$  correspond à un état stable.

<b>Algorithme 1: Construction de <i>clusters</i> à <math>k</math> sauts</b>
<pre> /* Des la réception d'un message hello d'un voisin */ <b>Prédicats</b> <math>P_1(u) \equiv (statut_u = CH)</math> <math>P_2(u) \equiv (statut_u = SN)</math> <math>P_3(u) \equiv (statut_u = GN)</math> <math>P_{10}(u) \equiv (cl_u \neq id_u) \vee (dist_{(u,CH_u)} \neq 0) \vee (gn_u \neq id_u)</math> <math>P_{20}(u) \equiv (cl_u = id_u) \vee (dist_{(u,CH_u)} = 0) \vee (gn_u = id_u)</math> <math>P_{40}(u) \equiv \forall v \in N_u, (id_u &gt; id_v) \wedge (id_u \geq cl_v) \wedge (dist_{(u,v)} \leq k)</math> <math>P_{41}(u) \equiv \exists v \in N_u, (statut_v = CH) \wedge (cl_v &gt; cl_u)</math> <math>P_{42}(u) \equiv \exists v \in N_u, (cl_v &gt; cl_u) \wedge (dist_{(v,CH_v)} &lt; k)</math> <math>P_{43}(u) \equiv \forall v \in N_u / (cl_v &gt; cl_u), (dist_{(v,CH_v)} = k)</math> <math>P_{44}(u) \equiv \exists v \in N_u, (cl_v \neq cl_u) \wedge \{(dist_{(u,CH_u)} = k) \vee (dist_{(v,CH_v)} = k)\}</math> <b>Règles</b> /* Mise à jour du voisinage */ <math>StateNeigh_u[v] := (id_v, cl_v, statut_v, dist_{(v,CH_v)});</math> /* Cluster-1 : Gestion de la cohérence */ <math>R_{10}(u) : P_1(u) \wedge P_{10}(u) \longrightarrow cl_u := id_u; gn_u = id_u; dist_{(u,CH_u)} = 0;</math> <math>R_{20}(u) : \{P_2(u) \vee P_3(u)\} \wedge P_{20}(u) \longrightarrow</math> <math>statut_u := CH; cl_u := id_u; gn_u = id_u; dist_{(u,CH_u)} = 0;</math> /* Cluster-2 : Clustering */ <math>R_{11}(u) : \neg P_1(u) \wedge P_{40}(u) \longrightarrow</math> <math>statut_u := CH; cl_u := id_v; dist_{(u,CH_u)} := 0; gn_u := id_u;</math> <math>R_{12}(u) : \neg P_1(u) \wedge P_{41}(u) \longrightarrow</math> <math>statut_u := SN; cl_u := id_v; dist_{(u,v)} := 1; gn_u := NeighCH_u;</math> <math>R_{13}(u) : \neg P_1(u) \wedge P_{42}(u) \longrightarrow</math> <math>statut_u := SN; cl_u := cl_v; dist_{(u,CH_u)} := dist_{(v,CH_v)} + 1; gn_u := NeighMax_u;</math> <math>R_{14}(u) : \neg P_1(u) \wedge P_{43}(u) \longrightarrow</math> <math>statut_u := CH; cl_u := id_v; dist_{(u,CH_u)} := 0; gn_u := id_u;</math> <math>R_{15}(u) : P_2(u) \wedge P_{44}(u) \longrightarrow statut_u := GN;</math> <math>R_{16}(u) : P_1(u) \wedge P_{41}(u) \longrightarrow</math> <math>statut_u := SN; cl_v := id_v; dist_{(u,v)} := 1; gn_u := NeighCH_u;</math> <math>R_{17}(u) : P_1(u) \wedge P_{42}(u) \longrightarrow</math> <math>statut_u := SN; cl_u := cl_v; dist_{(u,CH_u)} := dist_{(v,CH_v)} + 1; gn_u := NeighMax_u;</math> /* Envoi d'un message hello */ <math>R_0(u) : hello(id_u, cl_u, statut_u, dist_{(u,CH_u)});</math> </pre>

### 5.3. Algorithme auto-stabilisant de clustering à $k$ sauts

Chaque nœud du réseau connaît le paramètre  $k$  avec  $k < n$ , possède les *macros* suivants et exécute l'algorithme 1.

$$- NeighCH_u = \{id_v/v \in N_u \wedge statut_v = CH \wedge cl_u = cl_v\}.$$

$$- NeighMax_u = (Max\{id_v/v \in N_u \wedge statut_v \neq CH \wedge cl_u = cl_v\}) \wedge (dist_{(v,CH_u)} = Min\{dist_{(x,CH_u)}, x \in N_u \wedge cl_x = cl_v\}).$$

URED

## 6. Schéma intuitif de la preuve de la stabilisation

Dans cette section, nous énonçons les principaux théorèmes et propriétés vérifiés par notre algorithme. Par manque de place, nous ne donnons pas les détails des preuves.

L'état d'un nœud est défini par la valeur de ses variables locales. Une *configuration*  $\gamma_i$  du réseau est une instance de l'état de tous les nœuds. Avec notre approche, les nœuds les plus grands se fixent en premier. Un nœud  $u$  est dit *fixé* à partir de la configuration  $\gamma_i$  si le contenu de sa variable  $cl_u$  ne change plus. L'ensemble des nœuds fixés à  $\gamma_i$  est noté  $\mathcal{F}_i$ . Une *transition*  $\tau_i$  est le passage d'une configuration  $\gamma_i$  à  $\gamma_{i+1}$ . Au cours d'une transition, chaque nœud a reçu un message de tous ses voisins et a exécuté l'Algorithme 1. Ainsi, avec notre approche le nombre de nœuds fixés croît strictement à chaque configuration et tend vers  $n$ ,  $n$  étant le nombre de nœuds du réseau.

**Lemme 6.1** *Soit  $\gamma_0$  une configuration quelconque. A  $\gamma_1, \forall u \in V, u$  est cohérent.*

**Preuve :** à  $\gamma_0$  quelque soit son état, chaque nœud vérifie et corrige sa cohérence par exécution de la règle  $R_{10}$  ou  $R_{20}$  durant la transitions  $\tau_0$ . Ainsi, à  $\gamma_1$  tout nœud est cohérent.

**Corollaire 6.1**  $|\mathcal{F}_1| \geq 1$ .

**Idée de la preuve :** comme tous les nœuds sont cohérents d'après le lemme 6.1. Et  $\exists u$  tel que  $\forall v \in V, id_u > id_v$ . Au moins  $u$  applique  $R_{11}$  durant  $\tau_0$  et est donc fixé à  $\gamma_1$ . D'où  $u \in \mathcal{F}_1$  et  $|\mathcal{F}_1| \geq 1 \Rightarrow |\mathcal{F}_1| > |\mathcal{F}_0|$ . Ce nœud  $u$  est noté  $CH_{Max}$ .

**Théorème 6.1**  $\forall i < k + 1, |\mathcal{F}_{i+1}| > |\mathcal{F}_i|$  et  $\mathcal{F}_i \subset \mathcal{F}_{i+1}$ .

**Idée de la preuve :** d'après le corollaire 6.1,  $|\mathcal{F}_1| > |\mathcal{F}_0|$ . Pour  $i = 0$ , le résultat est vrai. A  $\gamma_2$ , nous pouvons constater que les nœuds situés à distance 1 du  $CH_{Max}$  sont fixés soit par la règle  $R_{12}$  soit par la règle  $R_{16}$  selon que leur statut est  $SN$  ou  $CH$ . Donc,  $|\mathcal{F}_2| > |\mathcal{F}_1|$ . Nous prouvons ensuite par récurrence qu'à  $\gamma_i$ , les nœuds situés à distance  $(i - 1)$  de  $CH_{Max}$  se fixent par la règle  $R_{13}$  ou par la règle  $R_{17}$ . Pour  $i = k$ , nous obtenons par récurrence  $|\mathcal{F}_{k+1}| > |\mathcal{F}_k|$ .

**Théorème 6.2 (Convergence)**

*Partant d'une configuration quelconque, une configuration stable est atteinte au plus en  $n + 2$  transitions.*

**Idée de la preuve :** comme  $|\mathcal{F}_1| \geq 1$ , nous avons  $|\mathcal{F}_{k+1}| > k$ . En réitérant le processus à partir d'un nouveau  $CH_{Max}$  qui est le nœud d'identité maximale  $\notin \mathcal{F}_{k+1}$ , nous prouvons que  $\forall i < n, |\mathcal{F}_{i+1}| > |\mathcal{F}_i|$  et  $\mathcal{F}_i \subset \mathcal{F}_{i+1}$ . Pour  $i = n$ , nous avons  $|\mathcal{F}_{n+1}| > |\mathcal{F}_n|$  d'où  $|\mathcal{F}_{n+1}| = n$ . Il faut ensuite une transition de plus pour que l'état des nœuds ne change plus. Nous obtenons un temps de stabilisation d'au plus  $n + 2$  transitions.

**Théorème 6.3 (Clôture)**

*A partir d'une configuration légale  $\gamma_i$ , sans occurrence de fautes, chaque nœud restera dans une configuration légale.*

**Idée de la preuve :** Soit  $\gamma_i$  une configuration légale,  $\forall u \in V$   $u$  est fixé et seule la règle  $R_0$  s'exécutera. Nous aurons donc  $\forall j > i, \gamma_j$  une configuration légale.

**Lemme 6.2 (Occupation mémoire)**

*L'occupation mémoire nécessaire est d'au plus  $n * \log(2n + k + 3)$ .*

**Idée de la preuve :** Chaque nœud, stocke pour chaque voisin l'identité, l'identifiant de *cluster*, le statut et la distance. L'espace mémoire pour chaque voisin est de  $\log(2n+k+3)$ . De plus, Il stocke aussi les mêmes informations pour lui. Chaque nœud a donc besoin d'un espace mémoire d'au plus  $n * \log(2n + k + 3)$ .



---

## 7. Conclusion

Dans cet article, nous avons présenté un algorithme complètement distribué et auto-stabilisant pour structurer le réseau en *clusters* non-recouvrants à  $k$  sauts. Sans initialisation, notre solution n'utilise que des informations provenant des nœuds voisins. Elle combine la découverte du voisinage et la construction des *clusters*. Nous avons montré que l'état stable est atteint en au plus  $n + 2$  transitions et nécessite au plus une occupation mémoire de  $n * \log(2n + k + 3)$ . Dans nos futurs travaux, nous évaluerons les performances moyennes de notre approche par le biais de simulations, car dans la plus part des cas, nous avons constaté un temps de stabilisation inférieur à  $n + 2$ .

---

## Remerciements

Ces travaux ont été effectués dans le cadre du projet CPER CapSec Rofica co-financé par la région de Champagne-Ardenne et le FEDER. Nous remercions également le Centre de Calcul de Champagne-Ardenne ROMEO et Grid'5000.

---

## Références

- [1] L. Blin, M. G. Potop-Butucaru, and S. Rovedakis. Self-stabilizing minimum degree spanning tree within one from the optimal degree. *Journal of Parallel and Distributed Computing*, pages 438 – 449, 2011.
- [2] A. Bui, A. Kudireti, and D. Sohier. A fully distributed clustering algorithm based on random walks. *International Symposium on Parallel and Distributed Computing*, 2009.
- [3] A. Bui, D. Sohier, and A. Kudireti. A random walk based clustering with local recomputations for mobile ad hoc networks. *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, 2010.
- [4] E. Caron, A. Datta, B. Depardon, and L. Larmore. A self-stabilizing k-clustering algorithm for weighted graphs. *J. Parallel Distrib. Comput.*, pages 1159–1173, 2010.
- [5] A. Datta, L. Larmore, and P. Vemula. Self-stabilizing leader election in optimal space. In *Stabilization, Safety, and Security of Distributed Systems*, pages 109–123. 2008.
- [6] E. W Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, pages 643–644, 1974.
- [7] O. Flauzac, B. S. Haggar, and F. Nolot. Self-stabilizing clustering algorithm for ad hoc networks. *International Conference on Wireless and Mobile Communications*, pages 24–29, 2009.
- [8] C. Johnen and L. H. Nguyen. Self-stabilizing weight-based clustering algorithm for ad hoc sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 83–94. 2006.
- [9] C. Johnen and L. H. Nguyen. Self-stabilizing construction of bounded size clusters. *International Symposium on Parallel and Distributed Processing with Applications*, pages 43–50, 2008.
- [10] C. Johnen and L. H. Nguyen. Robust self-stabilizing weight-based clustering algorithm. *Theoretical Computer Science*, pages 581 – 594, 2009.
- [11] N. Mitton, A. Busson, and E. Fleury. Self-organization in large scale ad hoc networks. 2004.
- [12] N. Mitton, E. Fleury, I. Guerin L., and S. Tixeuil. Self-stabilization in self-organized multihop wireless networks. In *Proceedings of the Second International Workshop on Wireless Ad Hoc Networking - Volume 09, ICDCSW '05*, pages 909–915, 2005.

URED

# Modélisation ontologique des systèmes complexes

## Cas de la propagation de la bilharziose au Sénégal

Gaoussou CAMARA<sup>1,2,3</sup> – Sylvie DESPRES<sup>3</sup> – Rim DJEDIDI<sup>3</sup> –  
Moussa LO<sup>1,2</sup>

<sup>1</sup>LIRIMA, Equipe M2EIPS IRD UMI 209 UMMISCO,

<sup>2</sup>LANI, UFR SAT, Université Gaston Berger, BP 234 Saint-Louis, Sénégal

<sup>3</sup>LIM&BIO, Université Paris 13, 74 rue Marcel Cachin, 93017 Bobigny

[gaoussou.camara@ugb.edu.sn](mailto:gaoussou.camara@ugb.edu.sn), [sylvie.despres@univ-paris13.fr](mailto:sylvie.despres@univ-paris13.fr), [rim.jedidi@univ-paris13.fr](mailto:rim.jedidi@univ-paris13.fr),  
[moussa.lo@ugb.edu.sn](mailto:moussa.lo@ugb.edu.sn)

**RÉSUMÉ.** Dans cette communication, nous proposons de fournir un cadre de modélisation ontologique des systèmes complexes. Un modèle ontologique d'un système complexe permet de fournir une modélisation des connaissances du domaine (aspects statiques) et une spécification des processus (aspects dynamiques). Cette spécification permet une reproduction de macro-processus à partir de la description abstraite de leurs processus internes. L'ontologie permet d'une part, de fournir un vocabulaire commun du domaine facilitant la communication entre les experts du système complexe et une représentation des connaissances favorisant le partage et l'interopérabilité ; et d'autre part, permet d'effectuer des simulations qualitatives sur la spécification des connaissances sur les processus sous-jacents au système complexe et d'inférer les situations possibles sans être contraint par la disponibilité de données numériques.

**ABSTRACT.** In this paper, we propose an ontological modeling framework for complex systems. An ontological model of a complex system provides a domain knowledge model (static aspects) and processes specification (dynamic aspects). This specification allows reproduction of macro-processes from the abstract description of their internal processes. The ontology provides a common vocabulary that facilitates communication between domain experts of the complex system, knowledge sharing and interoperability. Moreover it allows performing qualitative simulation on knowledge specifying processes of the complex system and thus, to infer possible situations without being constrained by numerical data availability.

**MOTS-CLÉS :** Modélisation, ontologie, système complexe, épidémiologie, propagation bilharziose.

**KEYWORDS:** Modeling, ontology, complex system, epidemiology, bilharzia spreading.

Actes du 4e Colloque National sur la Recherche en Informatique et ses Applications (CNRIA'2012)  
Thiès-Bambey, Sénégal, Avril 2012

## 1. Introduction

La modélisation des systèmes complexes s'est longtemps focalisée sur la reproduction de la dynamique du phénomène étudié pour comprendre son évolution et réaliser des prédictions par des simulations numériques. Ces activités produisent des connaissances destinées à être partagées et échangées par les acteurs et organisations intervenant dans l'étude du système complexe. Parmi les représentations possibles figure la modélisation ontologique qui vise à représenter ces connaissances pour faciliter leur exploitation pour des applications diverses.

Un système complexe est constitué d'un grand nombre d'entités en interaction dont la dynamique est représentée par un ensemble de processus. Les modèles systémiques, tels que les équations mathématiques aux dérivées partielles ou les systèmes multi-agents, fournissent une description des propriétés physiques de ces processus. Ces techniques se limitent à la caractérisation numérique des processus et omettent de spécifier leur sémantique. Un modèle ontologique d'un système complexe permet de fournir, en plus des connaissances sur l'aspect statique du domaine, une spécification des processus. Cette spécification permet une reproduction de macro-processus à partir de la description abstraite de leurs processus internes.

La prise en compte de ces particularités nous conduit à décomposer le modèle ontologique en deux parties comportant les connaissances sur l'aspect statique du domaine et les connaissances sur l'aspect dynamique. L'ontologie contribue à un partage des connaissances entre les organisations intervenant dans l'étude du système complexe et une meilleure communication entre les acteurs du système grâce à un vocabulaire commun. Elle permet également de réaliser des simulations qualitatives fondées sur le modèle de spécification des processus et ainsi de ne pas être dépendant des données numériques pour faire de la prédiction.

Dans cet article, nous commençons par présenter les propriétés caractéristiques d'un système complexe puis nous décrivons le modèle ontologique que nous proposons. Ensuite, avant de conclure, nous illustrons notre contribution en donnant un modèle ontologique de la propagation de la bilharziose au Sénégal.

---

## 2. Caractérisation des systèmes complexes

*« Le comportement global d'un système, constitué d'entités multiples et éventuellement hétérogènes, s'explique par le résultat du jeu des interactions entre ses entités constitutives. C'est la possibilité à pouvoir déterminer le comportement global du système qui permet de le caractériser de simple, et dans le cas contraire, de complexe » [1].*

URED

Pour comprendre la dynamique d'un système complexe, il est indispensable de recourir à la modélisation systémique [6]. Elle consiste à donner une représentation simplifiée du phénomène afin de reproduire son comportement dans le cadre d'une simulation. La simulation permet de mettre en œuvre les modèles obtenus et d'observer les propriétés émergentes du système dans le temps. Outre analyser la dynamique du système complexe, la simulation permet d'étudier les solutions opérationnelles à mettre en œuvre pour anticiper sur les risques d'émergence et d'évolution.

Il existe plusieurs approches de modélisation systémique ayant montré tout leur intérêt pour l'étude des systèmes complexes comme les approches mathématiques s'appuyant sur les équations différentielles ordinaires et les approches informatiques fondées sur les systèmes multi-agents [1]. Ces approches sont indispensables pour la compréhension (validation d'hypothèses) du système complexe modélisé, la prédiction et l'exploration. Cependant, ces techniques se limitent à la caractérisation numérique des processus en omettant de spécifier leur sémantique. Ainsi, il devient difficile de partager les modèles obtenus ou de les rendre interopérables. Une autre limite de ces modèles est la nécessité de disposer de données numériques – souvent très difficiles à collecter – pour être exploitables dans le cadre d'une simulation numérique. Il apparaît par conséquent nécessaire de proposer une approche complémentaire permettant la modélisation des connaissances sur le domaine mais aussi sur sa dynamique. Cette approche assurera une interopérabilité des modèles et un partage des connaissances entre les différents acteurs autour du système.

---

### **3. Modélisation ontologique des systèmes complexes**

#### **3.1 Ingénierie des ontologies**

L'ingénierie des ontologies est le domaine de l'informatique qui définit des méthodologies et élabore des outils de construction et de maintenance des ontologies. Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée [9]. Notre approche méthodologique reprend les étapes classiques du cycle de développement d'une ontologie<sup>1</sup> : étude de faisabilité, spécification, conceptualisation, évaluation et maintenance. Dans ce papier, nous nous focalisons sur l'étape de conceptualisation qui consiste en particulier à la modélisation de processus.

La particularité des systèmes complexes est qu'il est nécessaire, lors de la modélisation des connaissances du domaine, de prendre en compte la pluralité de la nature (statique ou dynamique) des entités qui constituent le monde réel [7].

---

<sup>1</sup> <http://www.neon-project.org/nw/Deliverables>

### 3.2 Modélisation des entités constitutives des systèmes complexes

Il existe une distinction fondamentale entre les entités statiques (appelées continuants ou endurants) et les entités dynamiques (appelées occurrents ou perdurants) du monde [7]. Les concepts d'occurrents correspondent aux entités évoluant dans le temps en passant par différentes phases comme les processus, les événements et les états. Les concepts de continuants correspondent aux entités sans partie temporelle comme les objets. Nous spécifions dans les sous sections suivantes les définitions de ces différentes entités avant d'aborder leur modélisation dans les systèmes complexes.

*Processus* : Un processus peut être considéré comme une entité se déroulant dans le temps et l'espace et susceptible de subir des changements [3]. Ainsi, les concepts de *temps*, de *changement* et d'*espace* peuvent être considérés comme typique des processus. Un processus a des *propriétés temporelles* pouvant varier lors de son déroulement dans le temps. Le changement peut intervenir dans un processus mais d'un intervalle de temps à un autre. En se référant à la notion de *fenêtre de temps* telle qu'elle est définie dans [3] le changement intervient sur une propriété (du processus) d'une fenêtre de temps à l'autre (par exemple, le changement de vitesse d'une course qui peut être lente dans une fenêtre de temps et rapide dans la suivante). Dans [6] la définition de la notion de processus prend en compte l'espace : on parle de processus lorsqu'il y a, au fil du *temps T*, la modification de la position dans un référentiel « *Espace-Forme* », d'une collection de « *produits* » quelconques identifiables par leur forme [...]"

*Événement* : Ontologiquement parlant, les événements sont définis comme étant les occurrences de faits qui sont par nature dénombrables [8]. Un événement est instantané ou s'étend dans un intervalle de temps bien défini [*Tinit*, *Tfinal*]. Un événement ne change pas au cours du temps, il ne possède pas de propriétés temporelles.

*Etat* : Un état représente une configuration d'un objet ou d'une collection d'objets à un instant donné. Par exemple, l'état de santé d'une personne ou le taux d'infection d'une population. Un état peut être le résultat d'un changement ou peut fournir un potentiel de changement [8].

*Objet* : Les objets sont des éléments de la réalité que nous sommes capables de percevoir. Les objets peuvent être concrets (physiques) ou abstraits (mentaux ou sociaux). Un objet est persistant et son état peut changer. Ces changements résultent de processus internes et/ou externes à l'objet. Ces visions interne et externe relatives à l'objet résultent de la structure granulaire des systèmes complexes. Par exemple, une *personne* a des processus internes (infection ou digestion) et des processus externes (la propagation d'une maladie). La structure d'un objet est donc, d'une part, maintenue par des processus internes et externes ; et d'autre part, l'objet lui-même est impliqué dans des processus externes de son environnement qui le détermine.

U R E D

### 3.3 Relations entre processus, événements, états et objets

Des relations de haut niveau ont été établies entre « *continuants* » et « *occurrents* ». Elles peuvent être héritées ou spécialisées pour définir des relations plus spécifiques entre les processus, événements, états et objets. Quelques relations, parmi les plus significatives pour les systèmes complexes, sont présentées *infra* :

*La participation* : un continuant *participe* dans un *occurrent* [2]. La participation est générique et doit être spécialisée pour être utilisée dans la modélisation correspondant à une application particulière.

*La méréologie* : traduit la relation de « partie-de » et s'applique à la fois aux *continuants* et aux *occurrents*. Par exemple, un événement est constitué de processus.

*La hiérarchie* : modélise les relations d'héritage, appelées aussi « sorte de ».

*L'internalité* et *l'externalité* : cette relation est proposée par [3] pour mettre en évidence l'importance du « point de vue » dans la modélisation d'une structure complexe et granulaire. En effet, selon le niveau de granularité où se situe l'observation, un processus peut être perçu comme interne ou externe à un objet donné.

*La causalité* : peut être considérée comme l'un des postulats de base des systèmes complexes. En effet, si les systèmes complexes sont caractérisés par leurs émergences de propriétés ou de comportements imprévisibles, ces émergences constituent également l'effet d'une cause. [4] proposent un ensemble de relations causales entre événements et états :

- \* *Causal relation* : un événement *cause* un autre événement.
- \* *Allow, prevent* : un état *autorise* un événement.
- \* *Initiate, terminate* : un événement *initie* un état.
- \* *enable, disable, perpetuate* : un état « *active* » un autre état.

### 3.4 Cadre de modélisation ontologique des systèmes complexes

Une bonne méthodologie de construction d'un modèle ontologique d'un système complexe doit par conséquent distinguer les particularités dynamiques de ces types de systèmes. Les éléments statiques et dynamiques restent néanmoins fortement liés. En effet, les entités statiques participent dans le déroulement des entités dynamiques, et inversement, les entités dynamiques agissent sur les états des entités statiques.

Nous proposons de distinguer deux types de connaissances dans une ontologie dédiée à la modélisation d'un système complexe :

- Les connaissances sur l'aspect statique : les entités statiques du domaine et les relations hiérarchiques, méréologiques et sémantiques les liant.

- Les connaissances sur l’aspect dynamique : les entités ayant une caractéristique temporelle soit du point de vue de leur évolution (les *processus* et les *événements*) ou de leur configuration (les *états* bien qu’ils n’impliquent aucune dynamique comme le fait remarquer [8]). En outre, la dynamique des systèmes complexes est décomposable en des processus plus élémentaires répartis sur plusieurs niveaux de granularité (appelé aussi échelle).

La réutilisation d’ontologies de fondement modélisant les processus est également un point central de la méthodologie que nous avons adoptée. Elle assure l’interopérabilité entre applications et assure la réutilisabilité des modèles construits. L’architecture adoptée distingue trois couches : une ontologie de fondement, une ontologie noyau utilisant l’ontologie de fondement et les ontologies spécifiques modélisant les aspects statique et dynamique du domaine à modéliser. Notre effort de modélisation consiste à identifier les connaissances à réutiliser et à éventuellement spécialiser des notions existantes pour les adapter aux spécificités du domaine considéré.

#### 4. Application : Propagation de la bilharziose au Sénégal

La propagation de la bilharziose est un phénomène épidémiologique. Ces types de phénomènes, par leurs évolutions et leurs émergences issues des interactions des éléments les composant, sont qualifiés de systèmes complexes.

Aspect statique : La bilharziose est une maladie parasitaire touchant les mammifères. L’infection est due à un parasite qui se développe dans l’eau, vecteur de transmission de la maladie. Le processus de transmission implique des mollusques, hôtes intermédiaires des parasites. Les facteurs de risque sont, d’une part, ceux ayant un lien avec la présence de l’eau comme la pluie, les aménagements hydro-agricoles; et d’autre part, ceux favorisant le contact des individus avec l’eau comme la pauvreté, l’agriculture, la pêche. A cela, s’ajoutent les facteurs biologiques, logistiques (les hôpitaux), etc.

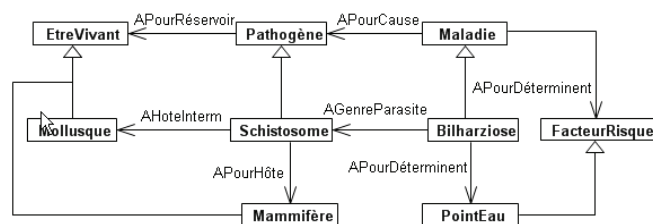
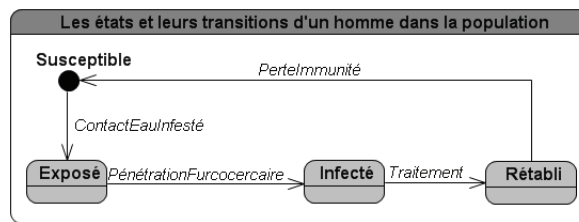


Figure 1. Modèle conceptuel réduit des connaissances sur la bilharziose

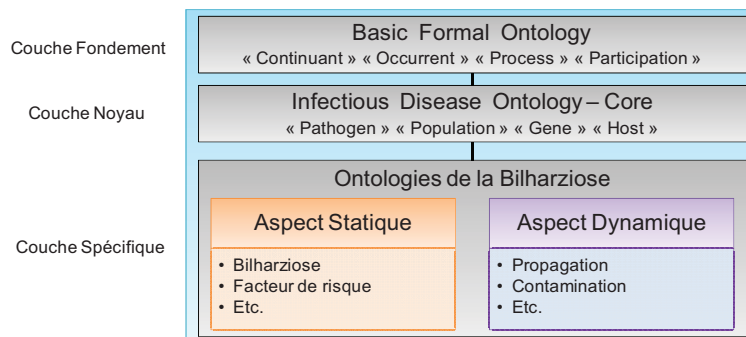
Aspect dynamique : La bilharziose est une maladie infectieuse. Sa dynamique de propagation étend le modèle mathématique standard du processus de la propagation des maladies infectieuses dans une population [5] qui subdivisent la population en trois

groupes d'individus : les *Susceptibles*, les *Infectés* et les *Rétablis* (SIR). Il existe un état intermédiaire entre la Susceptibilité et l'Infection : Exposition (E). Il est également possible pour un malade guéri d'acquérir une immunité temporaire contre la maladie. Ainsi, on distingue par exemple le processus de *contamination* des individus et le processus de *propagation* dans la population.



**Figure 3.** Digramme d'état-transition des états des individus d'une population

Afin de rendre notre ontologie réutilisable et interoperable nous avons adopté l'ontologie de fondement (BFO), l'ontologie noyau (IDO-Core qui utilise BFO). La figure 4 montre quelques concepts (Continuant, Occurrent, Process, etc.) et relation (Participation) de haut niveau modélisés dans BFO et les entités propres à toutes les maladies infectieuses (Pathogen, Host, Gene, Population, etc.). La couche spécifique présente quelques éléments sur les aspects statique et dynamique de la bilharziose à modéliser.



**Figure 4.** Modèle ontologique pour la propagation de la bilharziose

## 5. Conclusion

Les systèmes complexes présentent une structure particulière tant du point de vue statique que du point de vue dynamique. Leurs potentialités de présenter des émergences imprévisibles motivent le recours à la modélisation et à la simulation pour comprendre



leurs dynamiques pour pouvoir prédire leurs évolutions. Le cadre méthodologique que nous avons proposé pour ces systèmes complexes présente un double intérêt. D'une part, la modélisation des connaissances du domaine (aspects statiques) permet de fournir un vocabulaire commun du domaine facilitant la communication entre les experts du système complexe et une représentation des connaissances favorisant le partage et l'interopérabilité. D'autre part, la modélisation des aspects dynamiques fournit une spécification des connaissances sur les processus sous-jacents au système complexe. Cette abstraction faite sur les propriétés physiques des processus permet d'effectuer des simulations qualitatives et déterminer des situations possibles sans être contraint par la disponibilité de données numériques. La mise en relation de ces deux aspects dans un même cadre pour prendre en compte la double dimension d'un système complexe est aussi un apport dans la construction d'une ontologie.

---

## 6. Références

- [1] Drogoul, A., Treuil, J.-P., & Zucker, J.-D. (2008). *Modélisation et simulation à base d'agents : Exemples commentés, outils informatiques et questions théoriques*. Paris: Dunod.
- [2] Grenon, P., & Smith, B. (2004). SNAP and SPAN : Towards dynamic spatial ontology. *Spatial Cognition and Computation* , 4, 69-103.
- [3] Galton, A., & Mizoguchi, R. (2009). The water falls but the waterfall does not fall : New perspectives on objects, processes and events. (I. Press, Éd.) *Applied Ontology* , 4, 71-107.
- [4] Galton, A., & Worboys, M. F. (2005). Processes and events in dynamic geospatial networks. (M. A. Rodriguez, I. F. Cruz, M. J. Egenhofer, & S. Levashkin, Éds.) *Lecture Notes in Computer Science* , 3799, pp. 45–59.
- [5] Kermack, W. O., & McKendrick, A. G. (1927). A Contribution to the Mathematical Theory of Epidemics. *Proceedings of The Royal Society of London. Series A, Containing Papers of A Mathematical and Physical Character (1905-1934)* , 117 (772), 700--721.
- [6] Le Moigne, J.-L. (1990). *La modélisation des systèmes complexes*. Paris: Dunod.
- [7] Lewis, D. (1986). *On the Plurality of Worlds*. Oxford: Blackwell Publishers.
- [8] Mourelatos, A. P. (1978). Events, Processes and Events. *Linguistics and Philosophy* , 2 (3), 415-434.
- [9] Studer, R., Richard Benjamins, V., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *IEEE Transactions on Data and Knowledge Engineering* , 25 (1-2), 161-197.

# Un modèle à base d'agents de la propagation de la bilharziose à Ndombo

## Implémentation et simulation dans GAMA 1.3

Papa Alioune Cisse — Gaoussou CAMARA — Moussa LO

LIRIMA, Equipe M2EIPS IRD UMI 209 UMMISCO,

LANI, UFR SAT, Université Gaston Berger, BP 234 Saint-Louis, Sénégal

[papaaliouneecisse@yahoo.fr](mailto:papaaliouneecisse@yahoo.fr), [gaoussoucamara@gmail.com](mailto:gaoussoucamara@gmail.com), [moussa.lo@ugb.edu.sn](mailto:moussa.lo@ugb.edu.sn)

**RÉSUMÉ.** Dans ce papier, nous proposons un modèle à base d'agents pour la simulation de la propagation de la bilharziose dans la zone de Ndombo située dans la vallée du fleuve Sénégal. L'analyse de ce phénomène révèle des propriétés qui nous ont permis de le caractériser comme un système complexe. Nous avons proposé dans un premier temps un modèle conceptuel à base d'agents de sa propagation. Pour rendre opérationnel ce modèle dans un contexte de veille épidémiologique de cette maladie, nous l'avons implémenté avec la plateforme de simulation GAMA 1.3. Le modèle obtenu permettra de réaliser des simulations pour prédire l'évolution de la maladie et ainsi d'anticiper sur les conséquences (économiques, sociales, etc.) qu'elle engendre.

**ABSTRACT.** In this paper, we propose an agent-based model for simulation of spreading bilharzia in the area of Ndombo located in the Senegal River Valley. The analysis of this phenomenon reveals properties that allowed us to characterize it as a complex system. We initially proposed a conceptual model based on agents of its spread. To help make operational this model in the context of epidemiological monitoring of this disease, we have implemented it in the simulation platform GAMA 1.3. The model obtained will help perform simulations to predict the course of the disease and thus to anticipate the possible consequences (economic, social, academic, etc.) it could cause.

**MOTS-CLÉS :** modélisation multi-agents, bilharziose, simulation, plateforme GAMA

**KEYWORDS:** Agent-based modeling, bilharzias, simulation, GAMA platform

**REMERCIEMENTS :** Gilles RIVEAU (ONG Espoir Pour La Santé) et Idrissa TALLA (Programme National de Lutte contre la Bilharziose)

## 1. Introduction

La modélisation consiste à donner une représentation simplifiée d'un système. Le modèle obtenu peut ensuite être mis en œuvre dans une simulation pour répondre à une question sur le système de référence.

Pour le cas de la propagation de la bilharziose, il s'agit d'un système complexe composé d'un très grand nombre d'entités (hôte, vecteur, agent pathogène, environnement – géographie, climat, végétation, pluie, etc.) dont les interactions donnent lieu à des émergences variant selon les niveaux d'échelles spatiale et temporelle considérés [2]. Ces émergences peuvent avoir des conséquences sanitaires allant jusqu'à la propagation de la maladie dans de vastes zones géographiques et touchant un nombre grandissant d'individus. Pour anticiper sur ces risques, il est donc nécessaire de recourir à la modélisation systémique [4] et à la simulation pour observer les propriétés et comportements émergents du système.

Nous avons utilisé l'approche de modélisation à base d'agents pour concevoir un modèle conceptuel de la propagation de la bilharziose. Le modèle ainsi obtenu est implémenté dans la plateforme de simulation GAMA<sup>1</sup> 1.3 pour la mise en œuvre des simulations.

Dans la section 2, nous présentons la caractérisation de la dynamique de propagation de la bilharziose en tant que système complexe et ses approches de modélisation. La section 3 propose la conception du modèle ProBiNdo de propagation de la bilharziose dans le village de Ndombo situé dans la vallée du fleuve Sénégal. La section 4 donne un aperçu de l'implémentation de ce modèle dans la plateforme GAMA 1.3.

---

## 2. Modélisation de la propagation de la bilharziose

### 2.1. La propagation de la bilharziose : un phénomène complexe

La bilharziose, appelée aussi schistosomiase, est une maladie parasitaire qui est retrouvée dans les zones tropicales et subtropicales et est provoquée par un ver plat appelé schistosome ou bilharzie. Les schistosomes sont des parasites qui présentent deux phases de multiplication, l'une sexuée chez l'hôte définitif, l'homme, et l'autre asexuée chez l'hôte intermédiaire, un mollusque d'eau douce. Entre les deux hôtes, le lien est l'eau douce souvent peu profonde, calme et herbeuse [3].

Pour étudier et comprendre un phénomène, il convient de le caractériser afin de choisir les outils adéquats pour l'appréhender. Pour le cas de la propagation de la bilharziose, on observe au fil du temps l'apparition de comportements globaux inattendus. Or, selon [6], le comportement global d'un système, constitué d'entités multiples et éventuellement hétérogènes, s'explique par le résultat du jeu des

---

<sup>1</sup> <http://code.google.com/p/gama-platform/>.

## **Modèle à base d'agents de la propagation de la bilharziose à Ndombo**

interactions entre ses entités constitutives. C'est la possibilité de déduire le comportement global du système à partir de ceux de ses constituants qui permet de le caractériser de simple, et dans le cas contraire, de complexe. Ainsi, il apparaît que la dynamique de propagation de la bilharziose est un cas typique de système complexe.

Un système complexe ne révèle donc pas un déterminisme latent qu'il est possible de calculer par simple composition de ses parties. Cependant, ce caractère imprévisible des systèmes complexes n'exclut pas leur intelligibilité [4]. Ce qui laisse entendre que l'on peut toujours se ramener, par représentation symbolique, à simplifier un système complexe pour son intelligibilité. Ce procédé qui consiste à produire une représentation simplifiée d'un système est appelé modélisation et son résultat modèle. Le modèle ainsi obtenu peut servir à reproduire, par simulation, le comportement global du système qu'il représente.

### **2.2. Les approches pour modéliser la propagation de la bilharziose**

Il existe deux grandes approches de modélisation des systèmes complexes [6] : une approche mathématique basée sur la résolution d'équations et une approche informatique qui a l'avantage d'offrir un environnement de simulation interactive. Les modèles de type individus-centrés sont les plus utilisés dans la modélisation informatique des systèmes complexes. L'approche individus-centrée pour la modélisation d'un système consiste à représenter explicitement dans le modèle les individus ou les entités qui composent la population considérée. Elle a pour objectif la production d'un modèle de ces entités, de leur environnement, et de leurs interactions entre elles ou avec l'environnement. Parmi les modèles informatiques, on trouve les modèles basés sur les systèmes multi-agents (SMA). Grâce à ces SMA, on peut modéliser des systèmes réels dans lesquels des comportements très complexes émergent d'interactions relativement simples et locales entre de nombreux individus différents.

Dans le cadre de ce travail, nous utilisons l'approche de modélisation à base d'agents qui constitue une approche informatique très adaptée et démontrée [6] pour reproduire la dynamique de propagation de la bilharziose qui est un cas typique de système complexe.

---

### **3. Un modèle agent de la propagation de la bilharziose**

Le modèle proposé (figure 1.) est dénommé « ProBiNdo » (Propagation de la Bilharziose dans le village de Ndombo). Deux mécanismes fondamentaux sont au cœur du modèle : le mécanisme global de contamination qui explique globalement le processus de contamination des individus et le mécanisme individuel d'intervention qui explique la participation individuelle de chacune des entités intervenant dans la propagation de la maladie. Avant de détailler ces deux mécanismes, nous présentons d'abord la sémantique du vocabulaire utilisé.

### 3.1. Vocabulaire du modèle « ProBiNdo »

Tous les agents du modèle sont dotés d'un ensemble d'attributs, de comportements, ainsi que d'un corps et éventuellement d'un mécanisme de contrôle. L'ensemble des valeurs des attributs d'un agent constitue et renseigne sur l'état de cet agent. Les comportements d'un agent sont l'ensemble des actions et interactions qui seront effectuées par le corps de l'agent. Ils assurent l'évolution de l'état de l'agent et celle des autres dont il a la possibilité de modifier les attributs. Le corps d'un agent lui permet d'échanger avec l'extérieur. Les capacités de perception du corps d'un agent lui permettent de consulter les états observables des autres agents du modèle.

### 3.2. Le mécanisme global de contamination

Nous savons qu'un individu attrape la maladie par pénétration de furcocercaires (un parasite dans cet état) dans sa peau lors de ses contacts avec l'eau. Cependant, la probabilité qu'un individu en contact avec l'eau soit infecté dépend de la teneur de l'eau en furcocercaires (*TEF*). Puisque les furcocercaires sont libérés par les mollusques, *TEF* dépend aussi de la population de mollusques (*PM*) se trouvant dans l'eau. Un mollusque *M* libère des furcocercaires à une date *j* grâce aux miracidiums (un autre état du parasite) qui ont réussi à pénétrer *M*. Les miracidiums sont issus de l'éclosion des œufs libérés dans l'eau par les personnes infectées à travers les urines et fèces à la date *i* (*i* inférieur à *j*). Le furcocercaire, n'excédant pas une certaine durée de survie (*DS*), nage dans l'eau à la rencontre d'un individu pour pénétrer sa peau.

Cette vue globale du mécanisme d'infection nous permet de dégager l'ensemble des agents intervenant dans la distribution de la maladie. Il s'agit de la population du village de Ndombo (Agent Individu), des mollusques et des parasites (à l'état miracidium ou furcocercaire).

### 3.3. Le mécanisme individuel d'intervention des agents

Chaque agent est doté d'un ensemble d'attributs, de comportements et éventuellement d'un mécanisme de contrôle qui nous permettront de prendre en compte sa dynamique de population, de gérer ses relations avec son environnement et les autres agents. Par souci de simplicité, la restriction notée dans la définition de certains agents est délibérée.

#### 3.3.1. L'agent Individu

Un individu est susceptible (*S*), infecté (*I*) ou rétabli (*R*) et éventuellement immunisé, d'où l'attribut  $ETATIND = \{S, I, R\}$ . Il a un âge, un sexe, une espérance de vie. S'il est de sexe féminin, l'attribut *NE* représente le nombre d'enfants qu'il peut avoir. Nous lui ajoutons l'attribut *GA* représentant son groupe d'appartenance (par

## Modèle à base d'agents de la propagation de la bilharziose à Ndombo

exemple pêcheur, ménagère, agriculteur, etc.). Ainsi, la probabilité qu'un individu se rende dans les eaux va dépendre de son groupe d'appartenance.

Les comportements *creer\_individu()* et *liberer\_oeufs()* permettent respectivement aux individus de donner naissance à un autre individu et de libérer des œufs dans l'eau. Les déplacements sont gérés par le comportement *se\_deplacer()*.

### 3.3.2. L'agent Mollusque

Nous considérons les attributs âge (*ageMo*), sexe (*sexeMo*) et espérance de vie (*EVPMo*) des mollusques. Un mollusque se déplace dans l'eau, *se\_deplacer()*. Son action principale est la libération de furcocercaires, *liberer\_furcocercaires()*.

### 3.3.3. L'agent parasite

Il convient de distinguer, pour le cas de l'entité parasite, la particularité de la transition entre les états miracidium et furcocercaire. En effet, dans le cycle de vie du parasite, les miracidiums se multiplient dans le mollusque pour donner naissance à plusieurs furcocercaires. Par conséquent, dans la modélisation de l'agent parasite, nous avons deux sous-individus. La multiplication est donc caractérisée par la mort d'un miracidium et la naissance de plusieurs furcocercaires.

Un parasite a une durée de vie (*DVP*) et un état (*ETATPARA*). Le parasite se déplace dans l'eau, *se\_deplacer()*. Quand il rencontre un individu ou un mollusque, il le pénètre, *penetrer()*, et disparaît du système, *disparaitre()*.

### 3.3.4. L'environnement de Ndombo

Il est constitué de deux sous environnements : l'environnement des habitations occupé uniquement par l'agent *Individu* et l'environnement des eaux qui constitue principalement le lieu d'infection des individus. Il peut abriter les agents mollusque et parasite dans une phase de leur cycle de vie, ainsi que les individus venu fréquenter les points d'eau. Un de ses comportements sera de trouver, à tout moment, les agents qui se trouvent dans une même position dans l'eau et, ainsi, autoriser leurs interactions.

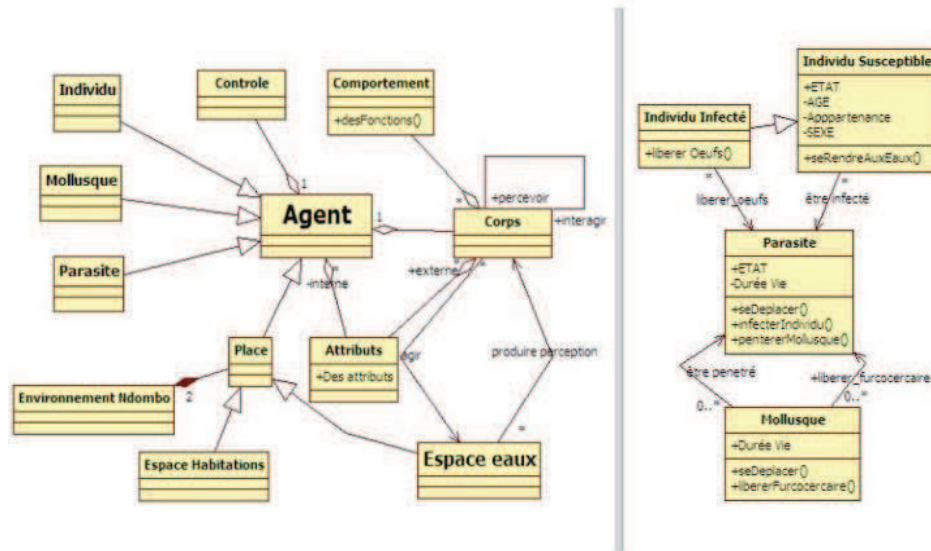


Figure 1. Modèle conceptuel ProBiNdo & Mécanisme global de contamination

#### 4. Implémentation du modèle dans GAMA 1.3

GAMA est une plateforme de simulation à base d'agents. Elle offre un ensemble de fonctionnalités telles que des affichages graphiques et un environnement de simulation supportant les systèmes d'information géographiques (SIG). Elle fournit également une bibliothèque de fonctions réutilisables avec son langage de programmation GAML. Nous présentons ci-après quelques extraits du modèle formel.

##### – Mise en place de l'environnement de simulation

```
<global>
<var type="string" name="shape_file_surfaces" init="surfaces.shp" />
<var type="string" name="shape_file_routes" init="routes.shp" />
<var type="string" name="shape_file_contours" init="contours.shp" />
</global>
```

Dans cet environnement, le fichier *surfaces.shp* contient les maisons (espace des habitations) de la population humaine et l'unique fleuve du village (espace des eaux), le fichier *routes.shp* contient les routes qui mènent des habitations aux points d'eau et le fichier *contours.shp* contient les délimitations de l'environnement du village.

##### – Mise en place du déplacement des individus dans l'environnement

Le code suivant met en œuvre les déplacements des agents *Individu* entre leurs habitations et les points d'eaux.

La figure 2 illustre la distribution des individus dans l'environnement de simulation.

URED

## Modèle à base d'agents de la propagation de la bilharziose à Ndombo

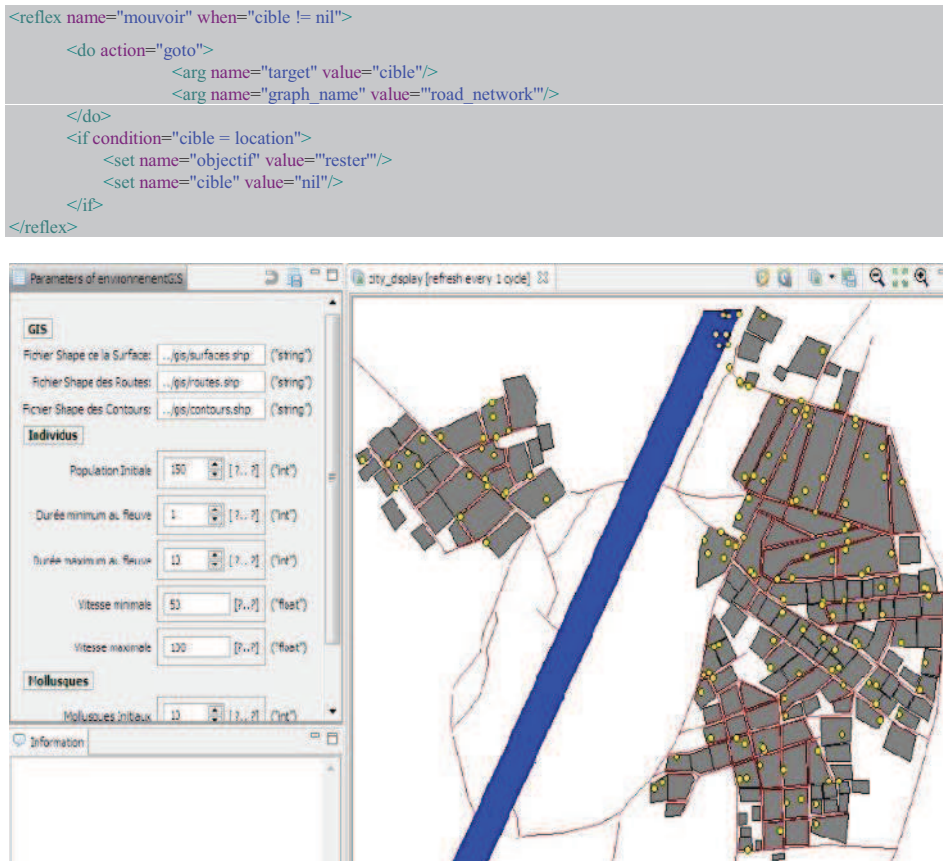


Figure 2. Distribution des individus dans l'environnement de simulation.

### – Mise en place du processus de contamination

Il s'agit de la mise en place de la dynamique d'interactions entre les agents *individu* et parasite (à l'état *furcocercaire*). Au cours de ses déplacements, quand un furcocercaire perçoit la présence d'individus, un nombre compris entre 0 et 1 est généré. Si ce nombre est supérieur à 0.5, le furcocercaire infecte au hasard un individu parmi ceux trouvés.

```

<reflex name="infecter_individu">
  <let var="a_infecter" value="one_of((self neighbours_at (5)) of_species (species individu))">
  <let type="float" name="hasard" value="rnd 1000 / 1000">
  <if condition="((a_infecter != nil) and (hasard > 0.5))">
    <ask target="a_infecter"> <set name="etat" value="I"/> </ask>
    <do action="die"/>
  </if>
</reflex>

```

CNRIA'2012



## 5. Conclusion

Dans ce travail, nous avons analysé la dynamique de propagation de la bilharziose. Il en a résulté que ce phénomène constitue un cas particulier de système complexe. Afin de procéder à sa modélisation, nous avons d'abord identifié toutes les entités intervenant dans sa propagation ainsi que la nature de leurs interactions. Ensuite, nous avons proposé un modèle conceptuel à base d'agents de la propagation de la bilharziose dans le village de Ndombo. Enfin, nous avons implémenté ce modèle dans la plateforme GAMA 1.3.

Cependant, la complexité du phénomène épidémiologique modélisé se répercute sur l'exercice de modélisation lui-même. Malgré cela, il existe des sous-systèmes de ce système global simple à modéliser et qui interviennent dans la dynamique d'autres systèmes. Malheureusement, les approches de modélisation actuelles ne fournissent pas des méthodes élaborées permettant d'assouplir la construction des modèles. Une contribution de nouvelles méthodes de modélisation modulaire et de simulation distribuée serait d'un grand apport dans la communauté. Le principe de la modélisation modulaire consisterait à définir des composants élémentaires du modèle, indépendamment du contexte et donc réutilisables. La simulation distribuée elle, se baserait sur le principe de la modélisation modulaire mais viserait surtout l'interopérabilité de plateformes de simulation réparties à l'échelle du web. Les développements récents des technologies telles que le cloud computing, les ontologies informatiques et les architectures orientées services offrent des bases solides vers une telle approche.

---

## 6. Références

- [1] Camara, G. (2009). Etat de l'art sur la modélisation informatique et la gestion des connaissances en épidémiologie. Rapport de stage de Master 2, Université Gaston Berger de Saint-Louis, UFR SAT, Saint-Louis, Sénégal.
- [2] Camara, G., Despres, S., Djedidi, R., & Lo, M. (2012). Modélisation Ontologique de processus dans le domaine de la veille épidémiologique. RFIA'12. Lyon.
- [3] I. Talla, A. Kongs, P. Verlé. Preliminary study of the prevalence of human schistosomiasis in Richard-Toll (the Senegal river basin). *Trans R Soc Trop Med Hyg*, Vol. 86 (2), pp. 182-191, 1992.
- [4] Le Moigne, J.-L. (1999). La Modélisation des systèmes complexes. Paris, Dunod.
- [5] Région Médicale de Saint-Louis. Les Bilharzioses au cours moyen. Manuel de l'élève. Brochure, Ministère de la santé et de l'action sociale, Sénégal.
- [6] Treuil, J.-P., Drogoul, A., & Zucker, J.-D. (2008). Modélisation et Simulation à base d'Agents. Paris: DUNOD.

.....

# L'exploration des réseaux de transport<sup>1</sup>

## La puissance de l'attente aux stations

David Ilcinkas — Ahmed Mouhamadou Wade

LaBRI  
CNRS & Université de Bordeaux  
France  
{ilcinkas, wade}@labri.fr

.....

**RÉSUMÉ.** Nous étudions le problème d'exploration, par une entité mobile (agent), d'une classe de graphes dynamiques, appelés graphes périodiquement variables (PV-graphes). Ils sont définis par un ensemble de transporteurs suivant infiniment leur route le long des stations du réseau. Flocchini, Mans et Santoro [6] ont étudié ce problème dans le cas où l'agent doit toujours rester sur les transporteurs. Dans ce papier, nous étudions l'impact de la capacité d'attendre sur les stations. Nous prouvons que l'attente sur les stations permet à l'agent, au pire cas, de réduire le nombre de mouvements d'un facteur multiplicatif d'au moins  $\Theta(p)$ , et la complexité en temps de  $\Theta(kp^2)$  à  $\Theta(np)$ , où  $n$  est le nombre de stations,  $k$  le nombre de transporteurs, et  $p$  la période maximale.

**ABSTRACT.** We study the problem of exploration by a mobile entity (agent) of a class of dynamic networks, namely the periodically-varying graphs (PV-graphs). These are defined by a set of carriers following infinitely their prescribed route along the stations of the network. Flocchini, Mans, and Santoro [6] studied this problem in the case when the agent must always travel on the carriers and thus cannot wait on a station. In this paper, we study the impact of the ability to wait at the stations. We prove that waiting at the stations allows the agent to reduce the worst-case optimal number of moves by a multiplicative factor of at least  $\Theta(p)$ , while the time complexity is reduced from  $\Theta(kp^2)$  to  $\Theta(np)$ , where  $n$  is the number of stations,  $k$  is the number of carriers, and  $p$  is the maximum period.

**MOTS-CLÉS :** Exploration, Graphes dynamiques, Agent mobile, PV-graphes

**KEYWORDS :** Exploration, Dynamic graphs, Mobile agent, PV-graphs

.....

Actes du 4e Colloque National sur la Recherche en Informatique et ses Applications (CNRIA) –  
Thiès-Bambey, Sénégal Avril 2012

.....

1. Ce travail a été réalisé avec le soutien du projet ANR ALADDIN, du projet INRIA CEPAGE, et du projet européen EULER. Une version étendue de ces travaux a été publiée dans les actes de la 15th International Conference on Principles of Distributed Systems (OPODIS 2011).

---

## 1. Introduction

L'exploration de graphes est un problème classique très étudié depuis sa formulation initiale en 1951 par Shannon [10]. Explorer un graphe consiste, pour une entité mobile (agent), à visiter tous les sommets au moins une fois puis à quitter le système en un temps fini. Ce problème étant l'un des plus classiques dans le cadre du calcul par agent mobile, il a reçu beaucoup d'intérêt jusqu'ici. La complexité en temps, en espace ou l'impact d'une connaissance à priori ont été largement étudiés au cours des 40 dernières années (voir, par exemple, [4, 8, 9]). Cependant, la plus grande partie de ces travaux concernent les graphes statiques. Les nouvelles générations d'environnements interconnectés sont extrêmement dynamiques et évoluent dans le temps. Pour cela, des chercheurs ont commencé à étudier ces réseaux dynamiques. Dans ce papier, nous étudions le problème d'exploration de graphes dans un modèle de réseaux dynamiques, à savoir les graphes périodiquement variables (PV-graphes).

Grossièrement, un PV-graphe consiste en un ensemble de transporteurs suivant périodiquement leur route respective parmi les sites (sommets) du système. Les PV-graphes modélisent en particulier les systèmes de transport publics, par exemple les systèmes de bus ou de métros. Ils modélisent également les systèmes de satellites en orbite autour de la Terre, ou les systèmes de sécurité composés d'agents de sécurité faisant leur ronde. Explorer ces environnements permet par exemple d'effectuer une opération de maintenance etc. En effet, un agent peut vérifier si tout est en ordre au cours de l'exploration. Cet agent peut être un logiciel, un robot ou un être humain.

Le problème d'exploration de PV-graphes était déjà étudié par Flocchini, Mans et Santoro [6]. Ils ont considéré que l'agent ne peut pas quitter le transporteur pour rester sur un site. Ne pas être capable de rester sur un site est particulièrement légitime dans les systèmes de satellites en orbite autour de la Terre par exemple, ou les sites ne correspondent pas à une quelconque station physique. Cependant, dans la plupart des systèmes de transport public, il est possible pour l'agent (humain ou non) de rester sur un site afin d'attendre un transporteur (éventuellement différent). Dans ce papier, nous considérons le même problème mais dans le cas où l'agent peut quitter les transporteurs pour attendre sur un site. Nous étudions l'impact de cette nouvelle capacité sur la complexité (temps et nombre de mouvements) du problème d'exploration de PV-graphes.

Dans cet article, nous étendons le travail sur les PV-graphes de Flocchini, Mans et Santoro [6] dans le cas où l'agent peut descendre d'un transporteur et rester sur un site. Cette possibilité permet à l'agent de pouvoir explorer même les PV-graphes qui ne sont pas hautement connexes. Nous prouvons que dans le cas général la complexité en mouvements est réduite de  $\Theta(kp^2)$  à  $\Theta(\min\{kp, np, n^2\})$ , et la complexité en temps de  $\Theta(kp^2)$  à  $\Theta(np)$ . (Notons que dans les PV-graphes connexes, nous avons  $n \leq kp$ ). Si le PV-graphe est homogène et hautement connexe, Flocchini, Mans et Santoro prouvent que la complexité en temps est  $O(kp)$ . Dans cet article nous allons prouver que si le PV-graphe satisfait uniquement une des deux restrictions, la complexité en temps est  $\Theta(np)$ . Par ailleurs, il se trouve que notre algorithme effectue non seulement l'exploration, mais réalise également la cartographie, c'est à dire qu'il peut produire une copie isomorphe du PV-graphe. Enfin, notons que notre algorithme n'utilise pas forcément les identifiants des sites.

## 2. Etat de l'art

Ces dernières années, un travail de recherche important a été fait dans les graphes dynamiques, notamment dans l'exploration et la cartographie de ces environnements très dynamiques par des agents mobiles. Motivés par l'exploration robotisée du Web, Cooper et Frieze [3] ont étudié en 2003 la question du temps de couverture minimal d'un graphe qui évolue au cours du temps. Ils ont considéré un modèle particulier de graphes du web et montrent que si après chaque nombre constant de pas de la marche un nouveau sommet apparaît et est raccordé au graphe, une marche aléatoire probabiliste sur le graphe manque de visiter une fraction constante de sommets. C. Avin et al. [1] montrent en utilisant une marche aléatoire que le temps de couverture minimale de certains graphes dynamiques est exponentiel. Dans le même papier, les auteurs montrent que la couverture minimale d'un graphe dynamique connexe est polynomiale en temps, en utilisant la marche aléatoire paresseuse "lazy random walk". Flocchini et al. [5] étudient la cartographie d'un PV-graphe contenant des trous noirs. Pour cela ils considèrent que le système est composé d'un nombre  $k$  d'agents et leur but est de construire la carte sans perdre un grand nombre d'agents. Casteigts et al. [2] intègrent dans un cadre unifié une vaste collection de concepts, de formalismes et de résultats obtenus dans la littérature des graphes variables dans le temps.

En 2009, Flocchini et al. [6] considèrent une famille de graphes variables, les graphes périodiquement variables, et montrent que si les sites du graphe sont étiquetés, la connaissance d'une borne supérieure sur la plus grande période ou du nombre  $n$  de sites est nécessaire pour qu'un agent puisse explorer le graphe. Si les sites du graphe sont anonymes, la connaissance d'une borne supérieure sur la plus grande période est nécessaire. Dans les deux cas, ils prouvent que la complexité en temps et en mouvements de l'agent est, dans le cas général,  $\Theta(kp^2)$ , où  $k$  est le nombre de transporteurs et  $p$  la période maximale des transporteurs. Dans le cas particulier des PV-graphes homogènes (PV-graphes pour lesquels tous les transporteurs ont la même période), ils prouvent que la complexité en temps et en mouvements passe à  $\Theta(kp)$ .

---

## 3. Définitions et modèles

Nous considérons un système  $S = \{s_1, \dots, s_n\}$  de  $n$  sommets (sites) parcouru par  $k$  transporteurs. Chaque transporteur  $c$  a un identifiant  $\text{Id}(c)$  et parcourt une séquence ordonnée  $R(c) = (s_{i_1}, \dots, s_{i_{p(c)}})$  de sites, appelée route, de manière périodique. L'entier positif  $p(c)$  est appelé période du transporteur  $c$ . Plus précisément, le transporteur  $c$  commence à un site  $s_{i_1}$  au temps 0 et à chaque unité de temps, il passe d'un site à son suivant de manière cyclique (quand  $c$  est au site  $s_{i_p}$ , il revient au site  $s_{i_1}$ ).

Un PV-graphe (graphe périodiquement variable) est l'ensemble  $(S, C)$ , où  $S$  est l'ensemble des sites, et  $C$  l'ensemble des transporteurs qui circulent sur les sites. Nous notons par  $n$ ,  $k$  et  $p$ , respectivement, le nombre de sites, le nombre de transporteurs et le maximum sur les périodes des transporteurs. Un PV-graphe est dit homogène si et seulement si tous les transporteurs ont la même période.

Pour tout PV-graphe  $G$ , nous allons définir deux graphes (classiques)  $H_1(G)$  et  $H_2(G)$  comme suit. Les deux graphes ont l'ensemble des transporteurs comme ensemble de sommets. Il existe une arête entre deux transporteurs  $c$  et  $c'$  dans  $H_1(G)$  si et seulement si il

existe un site qui apparaît dans les routes de  $c$  et de  $c'$ . Il existe une arête entre deux transporteurs  $c$  et  $c'$  dans  $H_2(G)$  si et seulement si il existe un site  $s$  et un temps  $t \geq 0$  tel que  $c$  et  $c'$  se rencontrent sur  $s$  au temps  $t$ . Un PV-graphe  $G$  est dit *connexe* si et seulement si  $H_1(G)$  est connexe. Un PV-graphe est dit *hautement connexe* si et seulement si  $H_2(G)$  est connexe. Dans cet article, nous considérons les PV-graphes qui sont au moins connexes. (Les PV-graphes non connexes ne peuvent pas être explorés).

Une entité externe, appelée *agent*, opère sur ces PV-graphes. Cette entité peut voir un transporteur et connaître son identifiant. Elle peut monter avec un transporteur ou changer de transporteur. Contrairement au modèle utilisé dans [6], l'agent a la possibilité de quitter un transporteur et de rester sur le site courant. Il pourra plus tard remonter avec le même transporteur ou un autre. Nous ne faisons aucune restriction sur la taille de la mémoire de l'agent ou sur ses capacités de calcul. Nous considérons deux modèles concernant les identités des sites. Dans le cas des PV-graphes *anonymes*, les sites n'ont pas d'identifiant, ou l'agent ne peut pas les voir. Dans le cas des PV-graphes *étiquetés*, les sites ont des identifiants distincts et l'agent peut les voir et les mémoriser.

Nous disons qu'un agent explore un PV-graphe si et seulement si, en commençant au temps 0 sur le site de départ du premier transporteur (sans perte de généralité), l'agent peut visiter l'ensemble des sites du PV-graphe en un temps fini et se mettre dans un état terminal. Cet état terminal exprime le fait que l'agent détecte que l'exploration est terminée.

De même que dans le cas où l'agent ne peut pas attendre sur un site, un agent sans information sur les PV-graphes à explorer ne peut pas explorer tous les PV-graphes (même si on se limite aux PV-graphes étiquetés homogènes et hautement connexes). Nous supposons donc par la suite que l'agent dispose d'une borne inférieure  $B$  sur la période maximale  $p$ .

---

## 4. Bornes inférieures

Dans cette section, nous ne faisons aucune hypothèse sur les PV-graphes (sauf l'hypothèse de connexité). Nous allons montrer que le fait de descendre permet à l'agent d'explorer l'ensemble des PV-graphes (et pas seulement les PV-graphes hautement connexes). La possibilité de descendre permet aussi une diminution de la borne inférieure sur le nombre de mouvements d'au moins un facteur  $\Theta(p)$ . La complexité en mouvements, dans le cas général, passe de  $\Theta(kp^2)$  à  $\Theta(\min\{kp, np, n^2\})$ . En outre, la complexité en temps passe de  $\Theta(kp^2)$  à  $\Theta(np)$ .

### 4.1. Borne inférieure sur le nombre de mouvements

Flocchini, Mans et Santoro [6] ont prouvé une borne inférieure sur le nombre de mouvements de  $\Omega(kp)$  pour explorer les PV-graphes avec  $k$  transporteurs et une période maximum  $p$  (même si le PV-graphe est homogène, étiqueté et hautement connexe). Cette borne inférieure ne s'applique pas directement dans notre cadre, car l'agent, ayant la possibilité d'attendre, pourrait être en mesure d'explorer en faisant moins de déplacements. Nous démontrons que c'est effectivement le cas : la complexité en mouvements de notre algorithme est bornée par  $O(\min\{kp, np, n^2\})$ . Nous allons prouver ici que cette complexité est optimale.

**Théorème 1** *Pour tout  $n$ ,  $k$  et  $p$ , il existe un PV-graphe  $G_{n,k,p}$  étiqueté, homogène et hautement connexe de  $n$  sites,  $k$  transporteurs et de période  $p$  tel que tout agent (i.e. tout algorithme) a besoin d'au moins  $\Omega(\min\{kp, np, n^2\})$  mouvements pour l'explorer.*

**Éléments de preuve.** Le principe général de la preuve est de construire une famille de PV-graphes dans lesquels la construction des routes obligent l'agent à visiter de nombreuses fois certains sites pour explorer les autres sites. Trois constructions différentes sont utilisées (suivant les valeurs relatives des différents paramètres) qui correspondent aux trois arguments du minimum.  $\square$

## 4.2. Bornes inférieures sur le temps

Dans le cas où l'agent ne pouvait pas descendre, le temps (en unités de temps) qu'il mettait pour explorer le PV-graphe était égal au nombre de mouvements. Le fait de donner à l'agent la possibilité de descendre sur un site et d'attendre un transporteur permet maintenant de faire la différence entre le nombre de mouvements et le temps mis par l'agent pour explorer un PV-graphe. Nous montrons ici une borne inférieure sur le temps plus élevée que celle sur le nombre de mouvements.

**Théorème 2** *Pour tout  $n, k, p$ , il existe une famille  $\mathcal{G}_{n,p,k}$  de PV-graphes hautement connexes et une famille  $\mathcal{G}'_{n,p,k}$  de PV-graphes homogènes telles que chaque PV-graphe  $G$  d'une des deux familles est fait avec  $n$  sites,  $k$  transporteurs et de période maximale  $p$ . Pour tout agent (i.e. tout algorithme), il existe un PV-graphe dans chacune des deux familles où il utilisera au moins  $\Omega(np)$  unités de temps pour l'explorer.*

**Éléments de preuve.** Nous ne donnons l'intuition de la preuve que pour la famille de PV-graphes hautement connexes. La preuve pour la famille de PV-graphes homogènes est similaire.

Les PV-graphes utilisés pour prouver ce théorème sont construits de la manière suivante. Les transporteurs sont numérotés de 1 à  $k$ . Les transporteurs de numéro impair, respectivement pair, sont de période  $p$ , respectivement  $p - 1$ . Un transporteur  $i$  n'a de sites en commun qu'avec les transporteurs  $i - 1$  et  $i + 1$ . (L'alternance des périodes assure donc la haute connectivité.) Plus précisément, le transporteur  $i$  passe une et une seule fois par un unique site visité par le transporteur  $i - 1$ .

La preuve est ensuite basée sur le fait que l'agent ne sait pas précisément dans quel PV-graphe il se trouve. En particulier, l'agent ne sait pas sur quel site du transporteur  $i - 1$  ni à quels temps passera le transporteur suivant (numéro  $i$ ). Il est possible de prouver, grossièrement, que l'agent doit attendre au moins  $p - 1$  unités de temps sur chaque site pour être sûr de trouver le transporteur suivant, d'où la borne obtenue.  $\square$

---

## 5. Notre algorithme

Dans la première partie du papier, nous avons fourni des bornes inférieures sur la complexité en mouvements et en temps. Nous allons maintenant prouver que tous ces résultats sont optimaux en décrivant et en prouvant un algorithme d'exploration de PV-graphes dont les performances correspondent aux bornes inférieures sur la complexité en mouvements et en temps, à condition que l'agent connaisse une borne supérieure, notée

$B$ , sur la période maximale  $p$  du PV-graphe. De ce fait, nous montrons que la possibilité d'attendre permet de diminuer à la fois la complexité en mouvements et en temps, la première par un facteur multiplicatif d'au moins  $\Theta(p)$ .

### 5.1. Principe et performance de notre algorithme

Comme précisé précédemment, notre algorithme utilise une borne supérieure  $B$  sur la période maximale  $p$  du PV-graphe. L'idée principale de l'algorithme consiste à descendre sur chaque site et, pendant  $O(B)$  unités de temps, à noter tous les temps de passages de transporteurs sur ce site. Correctement gérées, ces informations permettent de cartographier le PV-graphe (liste des routes et des horaires de passage de tous les transporteurs).

Plusieurs précautions doivent être prises pour ne manquer aucun site et pour optimiser le nombre de mouvements. Par exemple, après chaque étude d'un site (pendant  $O(B)$  unités de temps), l'algorithme calcule la plus petite période possible de chaque transporteur connu étant donné les informations déjà récoltés. Ceci permet de connaître tous les temps de passage futurs sur les sites étudiés. Afin d'éviter des déplacements inutiles, l'algorithme utilise la notion de transporteur courant. L'agent étudie tous les sites du transporteur courant avant de passer au suivant. Les identifiants des transporteurs rencontrés sont stockés sous la forme d'un arbre, un transporteur  $c'$  étant un fils d'un transporteur  $c$  si  $c'$  est découvert lorsque  $c$  est le transporteur courant. Les transporteurs sont traités en profondeur d'abord pour des raisons de performance.

Une analyse détaillée de l'algorithme complet permet de prouver le résultat suivant.

**Théorème 3** *Un agent exécutant notre algorithme utilise au plus  $O(\min\{kp, np, n^2\})$  mouvements et  $O(nB)$  unités de temps pour explorer et cartographier n'importe quel PV-graphe, où  $n$  est le nombre de sites,  $k$  le nombre de transporteurs et  $B$  une borne supérieure donnée à l'algorithme sur la plus grande période  $p$ .*

Notons que les algorithmes proposés dans [6] ne sont optimaux en temps (et donc en mouvements) que si la borne supérieure sur  $p$  qu'ils utilisent est linéaire en  $p$ . Dans notre cas, d'après les Théorèmes 1 et 2, notre algorithme est de la même façon optimal en temps seulement si la borne supérieure  $B$  est linéaire en  $p$ , mais il est par contre toujours optimal en nombre de mouvements, aussi mauvaise que soit la borne  $B$ .

### 5.2. Description de notre algorithme EXPLORE-AVEC-ARRET

Outre la borne supérieure  $B$  sur la plus grande période  $p$  du PV-graphe exploré, notre algorithme utilise les variables décrites ci-après.

L'algorithme utilise une numérotation propre pour identifier les sites, afin de pouvoir fonctionner même si le PV-graphe est anonyme.

- *numeroCourant* : numéro du site actuellement étudié.

Si le PV-graphe est étiqueté, l'agent maintient un tableau de correspondance entre les numéros donnés par l'agent et les identifiants réels des sites.

- *numeroVersID* : *numeroVersID*[ $j$ ] est l'identifiant du site numéro  $j$ .

L'agent maintient un arbre enraciné dont les différents sommets correspondant aux différents transporteurs rencontrés. Un ordre sur les fils est aussi maintenue.

- *arbre* : l'arbre des transporteurs.

---

**Algorithme** EXPLORE-AVEC-ARRET – Notre algorithme d'exploration de PV-graphes

```

1:  $bFin$  ← false
2:  $numeroCourant$  ← 1
3:  $transporteurCourant$  ← 1
4:  $arbre$  ← arbre réduit à un unique sommet (la racine) correspondant au transporteur 1
5: while ( $bFin = \mathbf{false}$ ) do
6:    $etudierSiteCourant()$ 
7:    $miseAuPropre()$ 
8:   if aucune valeur 0 dans aucun tableau  $route[.]$  then
9:      $bFin$  ← true
10:  else
11:     $trouverEtAtteindreProchainSite()$ 
12:  end if
13: end while
14: Terminer en fournissant la carte du PV-graphe (variables  $route[.]$ ,  $position[.]$  et éventuellement  $numeroVersID$  si le PV-graphe est étiqueté)

```

---

**Procédure**  $etudierSiteCourant()$  – collecte toute les informations possibles sur le site courant

```

1: Rester sur le site courant pendant  $2B$  unités de temps
2: for chaque unité de temps do
3:   for chaque transporteur  $i$  présent sur le site courant à l'instant courant do
4:     if  $i$  absent dans  $arbre$  then
5:       Ajouter  $i$  comme dernier fils de  $transporteurCourant$  dans  $arbre$ 
6:        $route[i]$  ← tableau de longueur  $3B$  rempli de 0
7:        $position[i]$  ← 0
8:     end if
9:      $route[i][position[i]]$  ←  $numeroCourant$ 
10:  end for
11: end for

```

---

**Procédure**  $miseAuPropre()$  – utilise les connaissances acquises pour mettre à jour les variables

```

1: for chaque transporteur  $i$  présent dans  $arbre$  do
2:    $periode[i]$  ← période minimale de  $route[i]$  entre  $position[i] - 2B + 1$  et  $position[i]$ 
3:   Rendre tout le tableau  $route[i]$  périodique de période  $periode[i]$  à partir des valeurs de  $route[i]$  entre  $position[i] - 2B + 1$  et  $position[i]$ 
4:    $position[i]$  ←  $position[i] \bmod periode[i]$ 
5: end for

```

---

**Procédure**  $trouverEtAtteindreProchainSite()$  – déterminer le prochain site à étudier et s'y rendre

```

1:  $transporteurCourant$  ← identifiant  $i$  du premier transporteur suivant l'ordre DFS dans  $arbre$  tel que la valeur 0 apparaît dans  $route[i]$ 
2: Calculer à partir des tables  $route[.]$  le trajet minimum, en nombre de mouvements puis en unité de temps, entre le site courant et un site marqué 0 dans  $route[transporteurCourant]$ 
3: Suivre ce trajet
4:  $numeroCourant$  ←  $numeroCourant + 1$ 
5:  $numeroVersID[numeroCourant]$  ← identifiant du site courant

```

Après chaque mouvement de l'agent le long du trajet :

```

1: for chaque transporteur  $i$  présent dans  $arbre$  do
2:    $position[i]$  ←  $position[i] \bmod periode[i]$ 
3: end for

```

---



– *transporteurCourant* : identifiant du transporteur actuellement étudié.

Pour chaque transporteur d'identifiant  $i$  présent dans *arbre* :

– *route* $[i]$  est un tableau de longueur  $3B$  (indexé de 0 à  $3B - 1$ ) ; il sert à mémoriser la suite des sites visités par le transporteur  $i$ .

– *position* $[i]$  est un entier compris entre 0 et  $3B - 1$  (inclus) ; il indique la position courante du transporteur  $i$  vis-à-vis de *route* $[i]$ .

– *periode* $[i]$  est un entier compris entre 1 et  $B$  (inclus) ; il indique la période minimale du transporteur  $i$  étant donné les connaissances actuelles.

Notons que les compteurs *position* $[i]$  sont incrémenté de 1 à chaque unité de temps. Ceci n'est pas indiqué dans le pseudocode afin de ne pas trop l'allourdir.

---

## 6. Conclusion

Dans cet article, nous avons étudié le problème d'exploration de PV-graphes en donnant à l'agent la possibilité d'attendre sur un site. Nous avons prouvé que cette possibilité permet à l'agent de réduire la complexité en mouvements et en temps de l'exploration.

Il serait intéressant d'approfondir ces résultats par des expérimentations, mieux à même de capturer les performances en moyenne. Ce travail pourrait aussi être étendu au cas probabiliste, pour les transporteurs (site de départ ou route aléatoire) comme pour les agents (utilisation de marche aléatoire).

---

## 7. Bibliographie

- [1] C. AVIN, M. KOUCKY, Z. LOTKER, « How to explore a fast-changing world (cover time of a simple random walk on evolving graphs) », *35th International Colloquium on Automata, Languages and Programming (ICALP)*, vol. LNCS 5125, p. 121–132, 2008.
- [2] A. CASTEIGTS, P. FLOCCINI, W. QUATTROCIOCCI, N. SANTORO, « Time-varying graphs and dynamic networks », *CoRR*, n° 1012.0009, 2010.
- [3] C. COOPER, A. M. FRIEZE, « Crawling on simple models of web graphs », *Internet Mathematics*, vol. 1(1), 2003.
- [4] A. DESSMARK, A. PELC, « Optimal graph exploration without good maps », *Theor. Comput. Sci.*, vol. 326(1-3), p. 343–362, 2004.
- [5] P. FLOCCINI, M. KELLET, P. C. MASON, N. SANTORO, « Mapping an unfriendly subway system », *5th Intl Conference on Fun with Algorithms (FUN)*, vol. LNCS 6099, p. 190–201, 2010.
- [6] P. FLOCCINI, B. MANS, N. SANTORO, « Exploration of periodically varying graphs », *20th Intl Symposium on Algorithms and Computation (ISAAC)*, vol. LNCS 5878, p. 534–543, 2009.
- [8] P. PANAIT, A. PELC, « Exploring Unknown Undirected Graphs », *J. Algorithms*, vol. 33(2), p. 281–295, 1999.
- [9] O. REINGOLD, « Undirected st-connectivity in log-space », *37th ACM Symposium on Theory of Computing (STOC)*, p. 376–385, 2005.
- [10] C. E. SHANNON, « Presentation of a maze-solving machine », *8th Conf. of the Josiah Macy Jr. Found. (Cybernetics)*, p. 173-180, 1951.

.....

## ProfMiner : algorithme de construction de profil utilisateur

Mouhamadou Saliou Diallo<sup>1,2</sup>, Arnaud Giacometti<sup>1</sup>, Arnaud Soulet<sup>1</sup>,  
Haoyuan D. Li<sup>1</sup>, Cheikh Talibouya Diop<sup>2</sup>, Sandra de Amo<sup>3</sup>

<sup>1</sup> Laboratoire d'Informatique, Université François Rabelais Tours, Antenne Universitaire de Blois, 3 place Jean Jaurès, 41000 Blois(France)

[mouhamadousaliou.diallo@etu.univ-tours.fr](mailto:mouhamadousaliou.diallo@etu.univ-tours.fr)  
{[arnaud.giacometti](mailto:arnaud.giacometti@univ-tours.fr), [arnaud.soulet](mailto:arnaud.soulet@univ-tours.fr), [haoyuan.li](mailto:haoyuan.li@univ-tours.fr)}@univ-tours.fr

<sup>2</sup> Laboratoire d'Analyse Numérique et d'Informatique, Université Gaston Berger de Saint-Louis, BP 234 Saint-Louis(Sénégal)

[cheikh-talibouya.diop@ugb.edu.sn](mailto:cheikh-talibouya.diop@ugb.edu.sn)

<sup>3</sup> Universidade Federal de Uberlândia, Brésil  
[deamo@ufu.br](mailto:deamo@ufu.br)

.....

**RÉSUMÉ.** L'utilisation de préférences suscite un intérêt croissant pour personnaliser des réponses aux requêtes dans les domaines du Web ou des bases de données. Pourtant, la construction manuelle de profils de préférences reste à la fois complexe et consommatrice de temps. Dans ce contexte, l'objectif de ce travail est l'élaboration d'une nouvelle méthode automatique d'extraction de préférences où le profil extrait est compréhensible par l'utilisateur. Pour cela, nous avons proposé un algorithme comportant deux phases : extraction des préférences contextuelles et construction du profil. Un test sur une base de données filmographique montre que les profils extraits permettent de prédire avec précision les films préférés d'un utilisateur.

**ABSTRACT.** The use of preferences arouses a growing interest to personalize requests answers in the domains of Web or databases. Nevertheless, manual construction of preferences profiles remains complex and time-consuming. In this context, the objective of this work is the elaboration of a new automatical method of preferences extraction where the extracted profile is understandable by the user. For that purpose, we proposed a two phase algorithm: extraction of the contextual preferences and construction of the profile. A test on a movie database shows that the extracted profiles allow predicting aptly the favorite movies of a user.

**MOTS-CLÉS :** élicitation de préférences, règles de préférence contextuelles, extraction de profils utilisateurs.

**KEYWORDS:** preference elicitation, contextual preference rule, user profile mining

.....

Actes du 4e Colloque National sur la Recherche en Informatique et ses Applications – Thiès-Bambey, Sénégal, Avril 2012

.....

## 1. Introduction

Le développement des systèmes de base de données avec les préférences contextuelles a suscité beaucoup d'intérêt ces dernières années. Cet intérêt s'explique par la variété des applications allant du commerce électronique aux moteurs de recherche personnalisés. Ces systèmes sont conçus essentiellement en tenant compte des préférences des utilisateurs. Plusieurs recherches importantes ont été consacrées à ce sujet, y compris la recherche sur le développement de puissants outils pour la modélisation de préférences tout en permettant d'effectuer des raisonnements sur ces préférences [2, 3] ainsi que le développement de langages de requête permettant la prise en compte de préférences utilisateurs [4, 5].

Cependant, peu de travaux ont été consacrés jusqu'ici à l'extraction de préférences utilisateurs. Dans ce papier, nous nous intéressons à l'extraction de profil de préférences utilisateurs en exploitant des règles de préférence contextuelles [1]. Nous proposons une méthode de construction d'un profil à partir d'un ensemble de préférences automatiquement collectées par observation des comportements de l'utilisateur. Un profil utilisateur est un ensemble de règles de préférence contextuelles qui a été sélectionné avec des critères prédéfinis. Nous montrons ensuite sur des données réelles que les profils construits nous permettent de prédire efficacement les préférences d'un utilisateur. Cette approche a beaucoup d'avantages comparée à celles qui existent dans la littérature. Elle permet de construire un profil compact et facilement compréhensible.

La suite de ce travail sera organisée de la façon suivante : la section 2 introduit les concepts fondamentaux (dont celui de règles de préférence contextuelles). La Section 3 présente l'algorithme de construction de profil utilisateur. Dans la section 4, nous décrivons les expériences effectuées pour évaluer la qualité des profils en termes de taille et de qualité pour effectuer des prédictions. Finalement, dans la section 5, nous dégageons une conclusion et présentons quelques perspectives de recherches.

---

## 2. Formalisation du problème

### 2.1. Règles de préférence contextuelle et base de préférences

Soit  $I$  un ensemble de littéraux appelé items, un itemset est un sous ensemble de  $I$ . Le langage des itemsets correspond à  $L = 2^I$ . Une base de données transactionnelle  $D$  est un multi-ensemble d'itemsets dans  $L$ . Chaque itemset, habituellement appelé transaction est une entrée de la base de donnée. Une base de préférences  $P \subseteq D \times D$  est un ensemble de couples de transactions représentant un échantillon de préférences utilisateur sur

U R E D

l'ensemble des données D. Une paire de transactions  $(t, u) \in P$  signifie que l'utilisateur préfère la transaction t à la transaction u. La table 1 présente une base de données transactionnelle où 5 transactions  $(t_1, \dots, t_5)$  sont décrites par 5 items notés A, ..., E; elle montre aussi 5 préférences utilisateur identifiées par  $p_1, \dots, p_5$ .

D				P	
Tid	Items			Pid	Préférences
$t_1$	A	C	D	$p_1$	$(t_1, t_3)$
$t_2$	A	B	D	$p_2$	$(t_2, t_3)$
$t_3$	A	B	C	$p_3$	$(t_2, t_4)$
$t_4$		C	D	$p_4$	$(t_3, t_4)$
$t_5$	A	B		$p_5$	$(t_4, t_5)$

Table 1. Exemple de données et de préférences utilisateurs

**Définition (Règles de préférence contextuelles).** Une règle de préférence contextuelle est de la forme  $i^+ > \bar{i} \mid X$  où X est un itemset de L,  $i^+$  et  $\bar{i}$  des items de  $\mathcal{I}X$ . L'itemset X est le contexte de la règle tandis que l'expression  $i^+ > \bar{i}$  exprime le fait que  $i^+$  est préféré à  $\bar{i}$ . L'ensemble des règles de préférence contextuelle est noté CP.

La sémantique derrière une règle de préférence contextuelle induit un ordre partiel strict sur les transactions. Une règle  $\pi : i^+ > \bar{i} \mid X$  induit qu'une transaction t est préférée à une transaction u, notée  $t \succ_{\pi} u$  si  $(Xi^+ \subseteq t) \wedge (Xi^- \subseteq u) \wedge (\bar{i} \notin t) \wedge (i^+ \notin u)$ . Par exemple la règle  $B > D \mid C$  induit que la transaction ABCE est préférée à ACD. Une règle de préférence contextuelle  $\pi$  peut être en accord avec  $(t, u) \in P$  (i.e.  $t \succ_{\pi} u$ ) ou en désaccord avec  $(t, u) \in P$  (i.e.  $u \succ_{\pi} t$ ). Dans les deux cas, on dit que la règle couvre la paire de transaction  $(t, u)$ . Par exemple,  $D > E \mid A$  couvre la paire  $p_1 = (t_1, t_3)$  et  $B > D \mid C$  couvre aussi  $p_1$ ; mais  $B > D \mid E$  ne couvre pas  $p_1$  parce que le contexte E n'appartient pas à  $t_1$ .

Nous adaptons dans notre approche les concepts de support et de confiance des règles d'association pour écarter les règles de préférence contextuelles qui ne satisfont pas un seuil minimum de support  $\sigma$  et un seuil minimum de confiance  $\kappa$  fixés.

Le support d'une règle de préférence contextuelles  $\pi$  dans P est défini par :  $supp(\pi, P) = |\{(t, u) \in P \mid t \succ_{\pi} u\}| / |P|$ . Par exemple, comme  $ACD \succ_{D > E \mid A} ABCE$  et  $ABD \succ_{D > E \mid A} ABCE$ , nous obtenons  $supp(D > E \mid A, P) = |\{p_1, p_2\}| / |P| = 0.4$ . La confiance évaluée si une règle de préférence contextuelle contredit beaucoup de préférences utilisateur. Elle est définie par :  $conf(\pi, P) = |\{(t, u) \in P \mid t \succ_{\pi} u\}| / |\{(t, u) \in P \mid t \succ_{\pi} u \vee u \succ_{\pi} t\}|$ . Cette mesure montre que la règle  $D > E \mid A$  a plus de valeur que la règle  $D > E \mid \emptyset$  parce que  $conf(D > E \mid A, P) = 2/2 = 1$  et  $conf(D > E \mid \emptyset, P) = 2/3$ . L'ensemble des règles excédant le

seuil minimum de support  $\sigma$  et le seuil minimum de confiance  $\kappa$  est appelé l'ensemble des règles intéressantes, il est noté  $CP_{\sigma,\kappa}(L,P)$  (ou simplement  $CP_{\sigma,\kappa}$ ). Dans notre proposition, on dit qu'une règle de préférence  $\pi$  est meilleure qu'une règle  $\pi'$  si les conditions suivantes sont satisfaites :

$$\pi >_{\text{csc}} \pi' \Leftrightarrow \begin{cases} \text{conf}(\pi) > \text{conf}(\pi') \text{ ou} \\ \text{conf}(\pi) = \text{conf}(\pi') \text{ et } \text{supp}(\pi) > \text{supp}(\pi') \text{ ou} \\ \text{conf}(\pi) = \text{conf}(\pi') \text{ et } \text{supp}(\pi) = \text{supp}(\pi') \text{ et } \pi <_{\text{CP}} \pi' \end{cases}$$

Cette définition montre que nous favorisons d'abord les règles ayant la meilleure confiance, ensuite à confiance égale on choisit celle qui a le plus grand support et en cas d'égalité de support et de confiance, on choisit la règle ayant la cardinalité du contexte la plus petite. Nous avons nommé cette façon de choisir une meilleure règle, l'ordre *confiance-support-cardinalité* noté  $>_{\text{csc}}$ . L'ordre  $<_{\text{CP}}$  est un ordre favorisant d'abord les règles dont le contexte est le plus court; puis, en cas d'égalité,  $<_{\text{CP}}$  classe les règles selon un ordre lexicographique arbitraire.

## 2.2. Construction du profil utilisateur

Nous modélisons les préférences d'un utilisateur en partant des règles de préférence contextuelles intéressantes  $CP_{\sigma,\kappa}$ . Pour extraire l'ensemble  $CP_{\sigma,\kappa}$ , nous utilisons *Cont-PrefMiner* : un algorithme que nous avons proposé et que nous ne présentons pas dans ce papier faute de place. Les règles de préférence contextuelles extraites par cet algorithme constituent le point de départ pour la construction du profil.

Nous considérons qu'un ensemble de règles de préférences contextuelles constitue le profil de l'utilisateur s'il est concis et consistant. Le profil construit sera ensuite utilisé pour faire de la prédiction ; supposons que nous voulons utiliser le profil  $\Pi = \{D>E|A, A>C|D, A>D|C, D>B|\emptyset\}$  pour prédire quelle est la transaction préférée entre CDE et AC (notez que ces transactions ne sont pas présentes dans la table initiale table 1). Une façon très naturelle de déterminer si une transaction  $t$  est préférée à une transaction  $u$  est d'utiliser la meilleure règle de préférence contextuelle  $\pi \in \Pi$  couvrant  $(t, u)$ , qui garantit que  $t >_{\pi} u$  ou  $u >_{\pi} t$ . On note  $t >_{\Pi} u$ , si la meilleure règle de  $\Pi$  couvrant la paire  $(t, u)$  induit  $t >_{\pi} u$ .

Pour évaluer la concision du profil extrait, nous utilisons simplement sa cardinalité. D'autre part, la précision et le rappel nous permettent d'évaluer la consistance du profil. Étant donné une base de préférences  $P$  et un ensemble de règles de préférence contextuelles  $\Pi$ , la précision de la relation de préférence  $>_{\Pi}$  noté  $\text{Prec}(>_{\Pi}, P)$  est définie par  $\text{Prec}(>_{\Pi}, P) = |\{ (t, u) \in P \mid t >_{\Pi} u \}| / |\{ (t, u) \in P \mid t >_{\Pi} u \vee u >_{\Pi} t \}|$  (3). De même le rappel de  $>_{\Pi}$ , noté  $\text{Rec}(>_{\Pi}, P)$  est défini par :  $\text{Rec}(>_{\Pi}, P) = |\{ (t, u) \in P \mid t >_{\Pi} u \}| / |P|$  (4).

L'objectif de la construction du profil est de sélectionner le profil utilisateur  $\Pi$  (inclus dans un ensemble des règles de préférence contextuelles  $S$ ) qui maximise la précision et le rappel et qui est aussi concis que désiré. En pratique, avec des données volumineuses,

URED

la construction d'un tel profil est un problème difficile. Pour faire face à cette difficulté, nous proposons dans la section 3 ProfMiner : un algorithme inspiré de CBA [6] qui maximise la précision et le rappel.

### 3. Contributions

Dans cette section, nous présentons un algorithme, de type glouton, pour construire un profil utilisateur à partir d'un ensemble de règles de préférence intéressantes et d'un ensemble de couples de préférences utilisateurs. La construction du profil utilisateur se fait selon deux principes répétés à chaque itération :

- Choisir la meilleure règle de préférence contextuelle
- Supprimer les règles de préférence contextuelles inutiles

Ces deux étapes sont réitérées jusqu'à ce qu'il ne reste plus de règles de préférence à sélectionner ou qu'il ne reste plus de préférences utilisateurs non couvertes.

La table 2 illustre, sur notre exemple jouet (voir table 1), l'ordre  $>_{CSC}$  sur les règles de préférence extraites avec  $\sigma = 0.2$  et  $\kappa = 0.6$ . Étant donné une base de préférence  $P$ , un

Cont.Pref	Supp	Conf
$r_1 : D > E   A$	0.4	1
$r_2 : D > C   \emptyset$	0.2	1
$r_3 : A > C   D$	0.2	1
$r_4 : A > D   C$	0.2	1
$r_5 : B > C   D$	0.2	1
$r_6 : D > B   A$	0.2	1
$r_7 : D > E   B$	0.2	1
$r_8 : D > E   AC$	0.2	1
$r_9 : D > B   \emptyset$	0.4	2/3
$r_{10} : D > E   \emptyset$	0.4	2/3

**Table 2.** Règles de préférences  
 $CP_{0.2, 0.6}$

ensemble de règles de préférence contextuelle  $S$  et un seuil d'accord minimal  $k$ , l'algorithme ProfMiner retourne le profil utilisateur  $\Pi$  en sélectionnant un sous-ensemble de règles de préférence contextuelles de  $S$ . Le seuil d'accord  $k$  nous permet d'ajuster la taille du profil utilisateur. Plus il est grand, plus le profil de l'utilisateur est petit. Après initialisation du profil (ligne 1), la boucle principale (ligne 2-7) choisit la meilleure règle de préférences contextuelle en se basant sur l'ordre  $>_{CSC}$  (ligne 3) et l'ajoute dans le profil (ligne 4) jusqu'à ce que  $S$  soit vide (ligne 3). Cette condition est assurée par la réduction de  $P$  (ligne 5) et la réduction de  $S$  (ligne 6). En effet, une règle de préférence contextuelle  $\pi$

ne figurera pas dans le profil en construction si elle n'est pas en accord avec au moins  $k$  préférences utilisateurs restantes (qui ne sont pas en accord avec les autres préférences utilisateurs). La table 3 illustre le résultat d'exécution de *ProfMiner* sur notre exemple jouet avec  $S = CP_{0.2, 0.6}$  et  $k = 1$ . Ce profil est obtenu au bout de 4 itérations qui se sont effectuées de la façon suivante : à la première itération, la ligne 3 sélectionne la règle  $r_1$  comme la meilleure règle en se basant sur  $>_{CSC}$  (voir la Table 2). La ligne 5 supprime les préférences utilisateurs  $p_1$  et  $p_2$  et ensuite, la ligne 6 supprime 5 règles de préférence contextuelle de  $S$  à savoir :  $r_2, r_6, r_7, r_8, r_{10}$ . Les règles  $r_3, r_4, r_5$  et  $r_9$  sont préservées, à la première itération, parce qu'elles couvrent (chacune) des préférences utilisateurs autre que  $p_1$  et  $p_2$  déjà couvertes par  $r_1$ .

La deuxième itération ajoute  $r_3$  au profil parce que c'est la meilleure règle de préférence contextuelle restante. Ce processus s'arrête à la fin de la 4<sup>ème</sup> itération parce que  $S$  est vide (voir ligne 3 de *ProfMiner*). Le profil final obtenu est composé de seulement 4 règles qui sont : ( $r_1, r_3, r_4, r_9$ ). Les règles barrées (sur la table 3) représentent celles qui ne sont pas retenues pour le profil final.

<i>Algorithm ProfMiner</i>	<b>Cont.Pref</b>	<b>accord</b>
<i>Input: A preference database P, a set of contextual preference rules S, a minimal agreement threshold k</i>	$r_1 : D \succ E   A$	$p_1, p_2$
<i>Output: A user profile <math>\Pi</math></i>	<del><math>r_2 : D \succ C   \emptyset</math></del>	$p_2$
1: $\Pi := \emptyset$	$r_3 : A \succ C   D$	$p_3$
2: while $S \neq \emptyset$ do	$r_4 : A \succ D   C$	$p_4$
3: $\pi_{best} = \max_{csc} S$	<del><math>r_5 : B \succ C   D</math></del>	$p_3$
4: $\Pi := \Pi \cup \{\pi_{best}\}$	<del><math>r_6 : D \succ B   A</math></del>	$p_1$
5: $P := \{ \langle t, u \rangle \in P \mid t \not\prec_{\pi_{best}} u \}$	<del><math>r_7 : D \succ E   B</math></del>	$p_2$
6: $S := \{ \pi \in S \mid \text{supp}(\pi, P) \geq k/ P  \}$	<del><math>r_8 : D \succ E   AC</math></del>	$p_1$
7: od	$r_9 : D \succ B   \emptyset$	$p_1, p_5$
8: return $\Pi$	<del><math>r_{10} : D \succ E   \emptyset</math></del>	$p_1, p_2$

**Table 3.** Exécution de *ProfMiner* avec  $S = CP_{0.2, 0.6}$  et  $k = 1$ .

## 4. Expérimentations

Une étude de l'efficacité de notre approche est effectuée sur des données réelles. Ces données sont basées sur APMD-workbench [7] qui est construit automatiquement de Movielens<sup>1</sup> et IMDB<sup>2</sup>. Les données utilisées ainsi que la procédure détaillée de préparation des données sont disponibles sur le dépôt du projet CPrefMiner<sup>3</sup>. Ces données sont composées, essentiellement, de 6 bases de préférences utilisateurs sur des films (une base par utilisateur). Chaque base est nommée par le nombre de paires de transactions quelle contient (par exemple la base P301 contient 301 paires de transactions, P3000 contient 3000 paires de transactions...). Dans chaque base, une préférence est représentée par un couple de films ( $m_1, m_2$ ) signifiant que l'utilisateur préfère le film  $m_1$  au film  $m_2$ . Chaque film est caractérisé par les attributs : *Genre, Directeur, Année, Acteur*, etc. En raison de la limitation d'espace sur ce papier, nous présentons seulement les résultats obtenus sur les fichiers P301, P3000 et P30000, les résultats sur les trois autres bases sont similaires. L'extraction de profil utilisateur et la prédiction est effectuée, dans chaque base, en utilisant 10 itérations de validation croisée. Ensuite la moyenne des

<sup>1</sup> [www.movielens.org](http://www.movielens.org)

<sup>2</sup> [www.imdb.com](http://www.imdb.com)

<sup>3</sup> [www.lsi.ufu.br/cprefminer](http://www.lsi.ufu.br/cprefminer)

métriques (précision et rappel) sur ces différentes itérations est utilisée pour devenir la précision et le rappel final du modèle sur ces données. D'abord les règles de préférence contextuelles sont extraites avec  $\sigma = 0.001$  et  $\kappa = 0.5$ . Notez que d'autres seuils minimaux de support et de confiance ont été testés (ces tests ne sont pas reportés sur ce papier faute de place) montrant que la hausse du support endommage la qualité du profil utilisateur tandis que la hausse de la confiance  $\kappa$  a un faible impact sur le profil extrait. Les règles de préférence contextuelles sont ensuite utilisées pour construire le profil avec ProfMiner et enfin la prédiction est effectuée en utilisant l'ordre induit par le profil construit.

Les expériences effectuées sur ces données vont montrer, dans un premier temps, que la concision du profil utilisateur est contrôlée par le seuil d'accord minimal (voir figure 3) puis dans un second temps que même avec une forte réduction du seuil d'accord la consistance du profil (précision et rappel) reste à un niveau acceptable (voir figure 4).

La figure 3 montre le nombre de règles de préférence dans le profil quand le seuil d'accord varie de 1 à 90. Même avec  $k = 1$ , le nombre de règles de préférence contenues dans le profil utilisateur est considérablement réduit comparé au nombre de règles de préférence contextuelles initial : de 5319.4 à 108.7 pour P301 ; de 4833.9 à 432.9 pour P3000 et de 4913.3 à 925 pour P30000. La figure 3 montre qu'on peut avoir la taille du

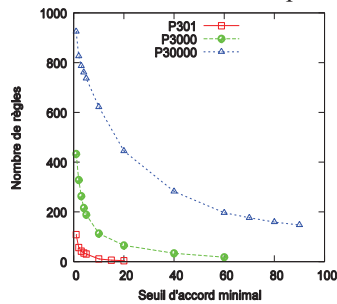


Figure 3. Nombre de règles de préférence contextuelles par rapport à  $k$

profil que l'on veut en variant le seuil d'accord.

La figure 4 montre la précision et le rappel du profil utilisateur par rapport au taux de réduction des règles dans le profil. Le taux de réduction des règles dans le profil est défini par :  $(|\Pi_{k=1}| - |\Pi|) / |\Pi_{k=1}|$ . Cette définition constitue une normalisation de la taille du profil utilisateur  $|\Pi|$  permettant de faciliter la comparaison entre les différentes bases de préférences.  $|\Pi_{k=1}|$  est la cardinalité du profil obtenu pour  $k = 1$ . On constate qu'en moyenne, les résultats obtenus sur la figure 4 sont intéressants (en particulier si le taux de réduction des règles est inférieur à 0.8). La précision reste toujours très élevée tandis que le rappel dépend de la taille du profil

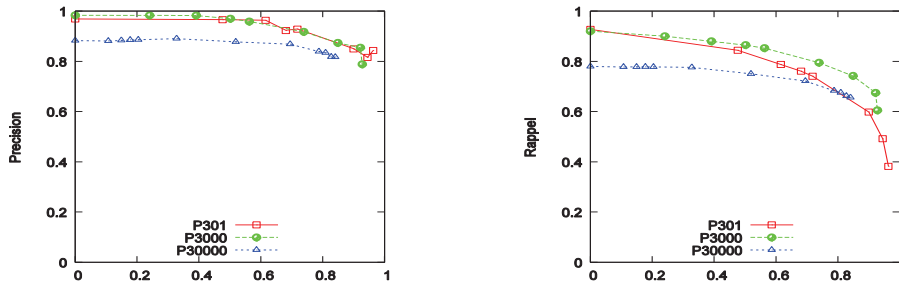


Figure 4. Performances de l'approche sur une base de données réelles



---

## 5. Conclusion et perspectives

Dans ce papier, nous avons considéré le formalisme de [1] pour exprimer des préférences utilisateurs par des règles de préférence contextuelles. Nous avons défini un profil utilisateur comme un ensemble concis et consistant de règles de préférence contextuelles et nous avons proposé un algorithme pour le construire. Nous avons également proposé une stratégie efficace de prédiction de préférences utilisateurs à partir du profil. Cette stratégie a été validée par une expérimentation sur des données réelles. L'évaluation basée sur la précision et le rappel garantit l'efficacité de notre méthode d'extraction de profils utilisateurs.

Notre méthode a été conçue pour l'extraction de préférences entre des ensembles d'items, chaque item étant identifié par son nom. Nous avons l'intention de généraliser notre proposition pour l'extraction de préférences contextuelles entre des données plus complexes comme des ensembles de tuples [8] aussi bien qu'entre des séquences d'items.

**Remerciements.** Nous remercions tous les organismes qui ont supportés ce travail à savoir : le Service de la Coopération et de l'Action Culturelle (SCAC Dakar), l'agence de recherche Brésilienne CNPq, CAPES (projet SticAmSud 016/09) et FAPEMIG.

---

## 6. Bibliographie

- [1] R. Agrawal, R. Rantzaou, and E. Terzi. Context-sensitive ranking. In *SIGMOD Conference*, pages 383–394. ACM, 2006.
- [2] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.*, 21:135–191, 2004.
- [3] R.I.Brafman, C. Domshlak, and S. E. Shimony. Ongraphical modeling of preference and importance. *J. Artif. Intell. Res.*, 25:389–424, 2006.
- [4] J. Chomicki. Preference formulas in relational queries. *ACM Trans. Database Syst.*, 28(4):427–466, 2003.
- [5] S. de Amo and F. Pereira. Evaluation of conditional preference queries. *Proceedings of SBBD 2010. Journal of Information and Data Management (JIDM)*, 1(3):521–536, 2010.
- [6] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, pages 80–86, 1998.
- [7] V. Peralta, D. Kostadinov, and M. Bouzeghoub. APMD-workbench: A benchmark for query personalization. In *Proceedings of the CIRSE Workshop*, 2009.
- [8] X. Zhang and J. Chomicki. Preference queries oversets. In *Proc. 27th International Conference on Data Engineering (ICDE)*, pages 1019–1030, 2011.

URED

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

## **Choix méthodologiques pour la construction d'une ontologie du domaine de la vulnérabilité urbaine face aux aléas liés aux changements climatiques.**

Ibrahima DIOP<sup>1,3</sup>, Moussa LO<sup>1,3</sup>, Adrien COLY<sup>2,3</sup>

<sup>1</sup>Laboratoire d'Analyse Numérique et d'Informatique (LANI),  
UMMISCO UGB – Equipe LIRIMA UGB

<sup>2</sup>Laboratoire LEIDI «Dynamiques des territoires et développement»

<sup>3</sup>Projet CLUVA (CLimate change and Urban Vulnerability in Africa)

Université Gaston Berger de Saint-Louis - BP 234, Sénégal

E-mail: [ibadopfr@yahoo.fr](mailto:ibadopfr@yahoo.fr), [moussa.lo@ugb.edu.sn](mailto:moussa.lo@ugb.edu.sn), [adrien.coly@ugb.edu.sn](mailto:adrien.coly@ugb.edu.sn)

.....

**RÉSUMÉ.** Cet article présente l'approche adoptée pour la construction d'une ontologie, nommée OntoCLUVA, du domaine de la vulnérabilité urbaine face aux changements climatiques. Avec cette approche, l'ontologie est modélisée à partir d'un corpus de textes et de ressources termino-ontologiques et de plusieurs expertises individuelles. Elle s'appuie sur la modélisation systémique et les approches classiques de modélisation du domaine à partir de textes et d'expertises individuelles.

**ABSTRACT.** This paper presents the approach to the construction of an ontology, named OntoCLUVA, the field of urban vulnerability to climate change. With this approach, the ontology is modeled from a corpus of texts and resources terminology and ontology of several individual expertise. It relies on the systemic modeling and classical approaches to domain modeling from texts and individual expertise.

**MOTS-CLÉS :** approche de construction d'ontologie, méta-modèle, ontologie, changement climatique, vulnérabilité urbaine, projet CLUVA, OntoCLUVA.

**KEYWORDS:** approach to construction of ontology, meta-model, ontology, climate change, urban vulnerability, project CLUVA, OntoCLUVA.

.....

## 1. Introduction

Le projet CLUVA<sup>1</sup> (CLimate Change and Urban Vulnerability in Africa), soutenu par l'Union Européenne dans le cadre du FP7-ENV-2010, a pour objectif de développer des méthodes et des connaissances pour la gestion des risques climatiques, la réduction de la vulnérabilité, l'amélioration de la capacité d'adaptation et de la résilience des villes africaines, face aux changements climatiques. Cinq villes ont été sélectionnées pour servir de cas d'étude : Saint-Louis (Sénégal), Ouagadougou (Burkina Faso), Douala (Cameroun), Adis Abeba (Éthiopie), Dar-es Salam (Tanzanie).

Ce domaine étudié par le projet CLUVA est complexe et est au carrefour de plusieurs autres domaines comme le climat, l'urbanisme, la gouvernance des risques, etc. Par conséquent, il fait intervenir plusieurs experts : géographes, urbanistes, hydrologues, climatologues, économistes, sociologues, mathématiciens, etc. Aussi, chacun de ces sous-domaines dispose de plusieurs corpus de textes et de ressources termino-ontologiques.

Notre équipe, CLUVA-Saint-Louis, a été amenée à proposer une plate-forme de gestion des connaissances du domaine de la vulnérabilité urbaine face aux changements climatiques, en vue de capitaliser les connaissances de ce domaine pour gérer leur partage par les experts engagés dans le projet et leur réutilisation ultérieure. Cette plate-forme se base sur une ontologie de ce domaine, nommée OntoCLUVA.

Pour construire cette ontologie, nous avons adopté une approche qui réutilise les ressources textes et termino-ontologiques existantes et permettant à des experts, non pas d'un domaine, mais de plusieurs domaines de collaborer dans cette tâche. Cette approche s'inspire de la modélisation systémique [7] et d'approches classiques de modélisation (méthode MERISE<sup>2</sup>, langage UML<sup>3</sup>) du domaine à partir de textes et d'expertises individuelles.

Cet article se compose principalement de deux parties. La première partie présente une plate-forme de gestion des connaissances de la vulnérabilité urbaine face aux changements climatiques. La deuxième partie présente l'approche adoptée pour construire une ontologie du domaine.

---

<sup>1</sup> [www.cluva.eu](http://www.cluva.eu)

<sup>2</sup> Méthode française de modélisation par les étapes MCD, MLD et MPD

<sup>3</sup> UML, Unified Modeling Language pour dire langage de modélisation unifié

## 2. Vers une plate-forme de gestion des connaissances de la vulnérabilité urbaine face aux changements climatiques

Un des objectifs spécifiques du projet CLUVA consiste à développer une plate-forme de gestion des connaissances du domaine de la vulnérabilité urbaine face aux changements climatiques (cf. fig. 1).

Cette plate-forme qui s'appuie sur une ontologie (OntoCLUVA) du domaine de la vulnérabilité urbaine face aux changements climatiques permettra le partage, l'échange et l'exploitation des sources de données et des modèles utilisés dans le cadre du projet pour le cas de la ville de Saint-Louis.

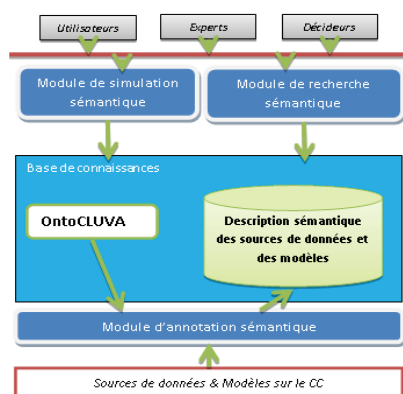


Fig. 1 : Architecture de la plate-forme de gestion des connaissances de CLUVA

OntoCLUVA joue un rôle central dans cette plate-forme.

Dans le module d'annotation, elle est utilisée pour effectuer la description sémantique des sources de données et des modèles ; elle forme ensuite avec cette description des sources de données et des modèles une base de connaissances.

Cette base de connaissance est utilisée par les modules de simulation sémantique et recherche sémantique. Ces deux derniers modules sont accessibles aux utilisateurs de la plate-forme pour entre autres aider à la prise de décision.

Dans la partie qui suit, nous présentons l'approche que nous avons utilisée pour la construction de cette ontologie.

## 3. Approche de construction de l'ontologie ONTOCLUVA

Le domaine de la vulnérabilité urbaine face aux changements climatiques dispose de corpus de textes et de ressources termino-ontologiques. Elle fait intervenir plusieurs experts (géographes, urbanistes, hydrologues, climatologues, économistes, sociologues, mathématiciens, etc.) amenés à collaborer.

Il existe plusieurs méthodologies et approches de construction d'ontologie : KOD [10], TERMINAE [1], approche différentielle [2], etc.

Pour construire ONTOCLUVA, nous avons d'abord utilisé une approche de modélisation systémique, qui est une démarche générale de réduction de la complexité du système objet d'étude par parcellisation de la difficulté.

Ensuite, compte tenu de l'existence de plusieurs ressources textes et terminologiques pour les sous-domaines considérés et la diversité des experts qui interviennent ainsi que leur disponibilité, nous avons choisi d'opter pour une méthode qui exploite à la fois :

- les approches classiques de modélisation du domaine à partir d'expertises individuelles (ex. la méthode MERISE et le langage UML): cela permet de tenir compte des connaissances individuelles des experts et de les faire collaborer en vue d'obtenir un modèle conceptuel commun ;
- et les approches basées sur l'analyse de corpus de textes et de ressources termino-ontologiques existantes dans un domaine [3] : cela en vue d'exploiter toute la richesse terminologique du domaine.

Ainsi notre approche se décline en trois grandes étapes.

La première étape, que l'on pourrait considérer comme préliminaire, consiste à la constitution d'un corpus de textes et de ressources termino-ontologiques existantes dans le domaine. La seconde étape consiste à définir un modèle systémique de la vulnérabilité urbaine face aux changements climatiques à partir de corpus de textes et d'expertises individuelles. La troisième étape consiste à élaborer un modèle conceptuel pour chacun des systèmes identifiés dans le modèle systémique.

### 3.1. Corpus de textes et de ressources termino-ontologiques

Ce corpus a été créé à partir de ressources disponibles sur la vulnérabilité urbaine face aux changements climatiques. Il s'agit :

- de la terminologie de *la Stratégie internationale de prévention des catastrophes des Nations Unies* [9] ;
- du glossaire *Géo confiance* permettant de questionner, problématiser, identifier et comprendre des mots-clefs en géographie ;
- de revues d'articles scientifiques : [6], [5] ;
- du cours sur les risques naturels [4] ;
- et du modèle conceptuel des risques et des catastrophes dans [8].

L'analyse de ce corpus et une discussion avec les experts nous a permis d'identifier les concepts centraux du domaine, à savoir : **Changement climatique, Vulnérabilité, Aléa, Enjeu, Risque et Catastrophe.**

Dans la partie qui vient, nous allons trouver des relations entre ces concepts et proposer un modèle systémique pour le domaine.

### 3.2. Un modèle systémique pour la vulnérabilité urbaine face aux changements climatiques

Dans cette partie, nous proposons un modèle systémique du domaine de la vulnérabilité urbaine face aux changements climatiques. Ce modèle peut être vu comme un ensemble de systèmes (Changement climatique, Vulnérabilité, Aléa, Enjeu, Risque et Catastrophe) qui interagissent et communiquent entre eux par leurs relations. Chacun de ces systèmes constitue un modèle en soi.

Il ressort de notre analyse les relations suivantes (cf. table.1) entre les concepts identifiés.

Table. 1: Extrait du corpus de textes

- *Le changement climatique **augmente la fréquence** des aléas climatiques.*
- *Ces aléas **composés** avec les enjeux (territoire, des biens et des sociétés) qui **ont** des vulnérabilités donnent le concept de risque qui est **concrétisé** par la catastrophe.*

Nous en avons déduit le modèle systémique de la vulnérabilité urbaine face aux changements climatiques présenté dans la figure 2 suivante.

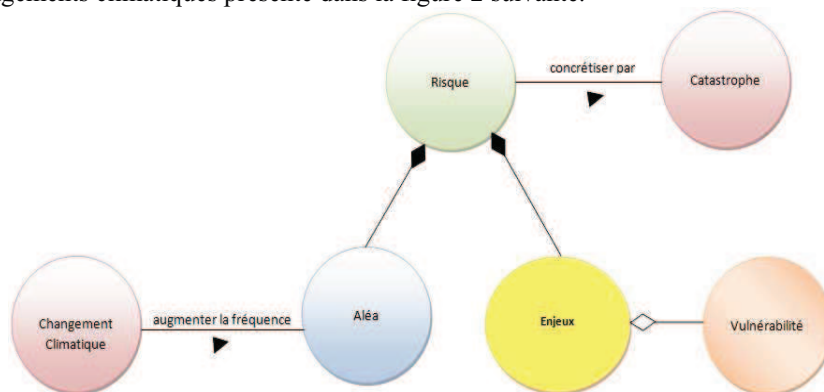


Fig. 2 : Un modèle systémique du domaine CLUVA

Pour la validation de ce modèle, nous avons organisé une séance de présentation et de discussion avec des experts du domaine (urbanistes, climatologues, experts en gouvernance de risques, modélisateurs informatiques et mathématiciens).

### 3.3 Modélisation conceptuelle

La phase suivante consiste, à partir du modèle systémique, d'effectuer une modélisation conceptuelle de chaque sous-domaine : Changement climatique, Aléa, Enjeux, Vulnérabilité, Risque et Catastrophe.

Pour chacun d'entre eux, nous avons analysé le corpus, effectué des interviews des experts et réutilisé aussi des modèles pour proposer un modèle conceptuel. Le modèle ainsi obtenu est ensuite validé par les experts pour enfin être opérationnalisé en OWL<sup>4</sup> (Ontologies Web Language).

Un extrait du modèle conceptuel du sous-domaine *Changement climatique* est représenté ci-dessous (cf. fig.4). Dans cet extrait, on peut noter que le concept de *Risque* est composé des concepts d'*Aléa* et d'*Enjeux*. Il généralise les concepts *Désertification*, *Sécheresse*, *Salinisation* et *Inondation*. L'*Enjeux* dans ce modèle est un *Système* composé de concept *Elément* et admettant une *Vulnérabilité*. Cette figure décrit aussi les types d'*Aléas* (*Aléa climatique*, *Aléa volcanique*, *Avalanche*, *Tempête*,...).

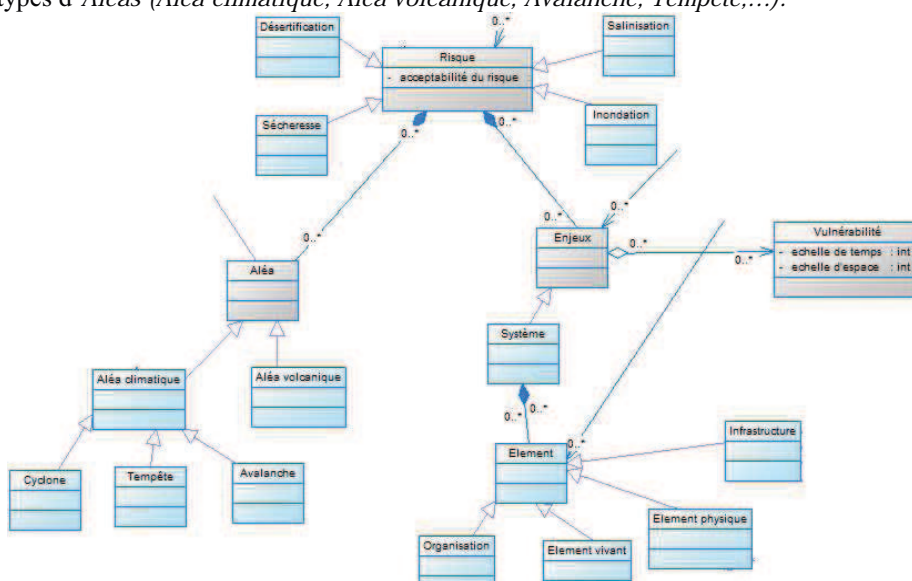


Fig. 3 : Extrait 1 du modèle conceptuel d'OntoCLUVA (sous-domaine changement climatique)

Un autre extrait du modèle conceptuel, celle du sous-domaine *Vulnérabilité* est représenté ci-dessous (cf. fig.5). On peut noter, dans cet extrait, que la vulnérabilité est étudiée sous deux angles ou approches: la Vulnérabilité analytique et la Vulnérabilité systémique. La vulnérabilité analytique est décomposée en vulnérabilité corporelle, vulnérabilité structurelle, vulnérabilité humaine ou sociale, vulnérabilité environnementale, vulnérabilité structurelle et vulnérabilité fonctionnelle. Quant à la vulnérabilité systémique voit la vulnérabilité comme un système complexe et donc non décomposable. Et ce concept de *Vulnérabilité* a une relation *porter sur* avec le concept *Enjeu*.

<sup>4</sup> <http://www.w3.org/2001/sw/WebOnt/>

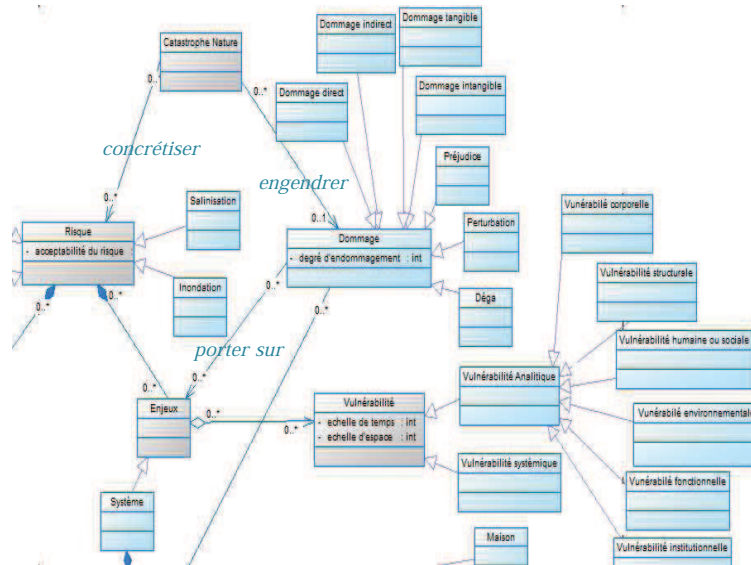


Fig. 4: Extrait 2 du modèle conceptuel d'OntoCLUVA (sous-domaine vulnérabilité)

Les modèles conceptuels obtenus peuvent ensuite être mis à jour et/ou enrichis par un processus d'extraction de concepts spécifiques à un sous-domaine (ex. climat) à partir de corpus de textes proposés par les experts (ex. climatologues dans le cas du sous-domaine climat).

#### 4. Conclusion

Nous avons décrit l'approche que nous avons proposée pour construire une ontologie de la vulnérabilité urbaine face aux changements climatiques. Compte tenu du contexte dans lequel cette ontologie est construite, nous avons d'abord proposé un modèle systémique du domaine. Nous avons ensuite développé un modèle conceptuel pour chacun des concepts fondamentaux du domaine présent dans le modèle systémique.

Dans la suite, nous comptons compléter cette approche par une étape d'enrichissement d'ontologie par un outil d'enrichissement à partir des corpus de textes et ressources termino-ontologiques. Pour enrichir l'ontologie, nous envisageons que chaque expert (ou groupe d'experts) d'un domaine puisse créer son corpus de textes et disposer d'un outil d'extraction de concepts.

Par ailleurs, avec cette ontologie, le domaine de la vulnérabilité dispose d'un vocabulaire facilitant le partage et l'exploitation des connaissances du domaine de la



Diop et al.

vulnérabilité aux changements du climat. Dans notre contexte, l'ontologie obtenue va être utilisée dans une plate-forme de gestion de connaissances du domaine de la vulnérabilité urbaine face aux changements climatiques en vue de fournir aux décideurs les connaissances et les outils nécessaires dans le cadre de l'intégration de la dimension Climat lors des prises de décisions.

---

## 5. Références

- [1] Nathalie Aussenac-Gilles, Brigitte Biébow et Sylvie Szulman, Modélisation du domaine par une méthode fondée sur l'analyse de corpus, 9e Conférence Francophone d'Ingénierie des Connaissances IC 2000, Toulouse : France (2000)"
- [2] Bruno Bachimont, Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. In : Ingénierie des Connaissances, évolutions récentes et nouveaux défis. Paris:Eyrolles, 2000.
- [3] Brigitte Biébow et Sylvie Szulman, Terminae : une approche terminologique pour la construction d'ontologies du domaine à partir de textes. Actes de RFIA2000, Reconnaissances des Formes et Intelligence Artificielle, Paris (F), 2000.
- [4] Adrien Coly, notes de cours 2011, les risques naturels : les risques géomorphologiques, les risques climatiques et hydrologiques, les risques sismiques et risques volcaniques, Université Gaston Berger de Saint-Louis.
- [5] La revue GÉORISQUES, La vulnérabilité des sociétés et des territoires face aux menaces naturelles, (2005).
- [6] Info-Resource, la revue focus N°2/09, « Réduction des risques de catastrophe : une approche axée sur l'égalité hommes-femmes et les moyens d'existence », (2009).
- [7] Lou Bertalanffy, Théorie Générale des Systèmes, nouvelle édition, Dunod, Paris, 1993.
- [8] Provitolo D., Dubos-Paillard E., Müller J.-P., Vers une ontologie des risques et des catastrophes : le modèle conceptuel, Dans XVI èmes rencontres de Rochebrune, Du 19 au 23 janvier 2009, Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels - Ontologie et dynamique des systèmes complexes, perspectives interdisciplinaires, Rochebrune (Megève) : France (2009) - <http://halshs.archives-ouvertes.fr/halshs-00643597/fr/>.
- [9] UNISDR (2009) La terminologie de la Stratégie internationale de prévention des catastrophes des Nations Unies (UNISDR), version anglaise de la terminologie 2009, [www.unisdr.org](http://www.unisdr.org).
- [10] VOGEL C., Génie cognitif. Paris : Masson, 1988.

URED

.....

## Modèle de Données multidimensionnel sur la Fièvre de la Vallée du Rift au Ferlo (Sénégal)

Fanta Bouba<sup>1</sup> et Alassane Bah<sup>1</sup> et Jacques-André Ndione<sup>2</sup> et Samba Ndiaye<sup>3</sup> et Christophe Cambier<sup>4</sup>

<sup>1</sup> UMI 209, UMMISCO-UCAD et ESP/UCAD, Dakar, Sénégal

<sup>2</sup> CSE et LPA/UCAD, Sénégal

<sup>3</sup> UMI 209, UMMISCO-UCAD et FST/UCAD, Dakar, Sénégal

<sup>4</sup> UMI 209, UMMISCO-IRD et Université Pierre et Marie Curie, France

boubafanta@gmail.com ; alassane.bah@gmail.com ; jacques-andre.ndione@cse.sn ;  
samba.ndiaye@ucad.edu.sn ; christophe.cambier@upmc.fr

.....

**RÉSUMÉ.** La maladie de la Fièvre de la Vallée du Rift (FVR) est une anthro-zoonose répandue dans les régions tropicales. Dans le cadre des études menées sur les facteurs environnementaux qui favorisent son déclenchement et sa propagation, nous travaillons sur la fouille de données qui offre des outils de reporting et d'analyse nécessaires au suivi et la prise de décision. Nous proposons, dans cet article, un modèle de représentation de données multidimensionnel prenant en compte la complexité de la maladie au cœur des systèmes agroécologique, pathogène et économique, sanitaire et social [2].

**ABSTRACT.** The Rift Valley Fever (RVF) is an anthro-zoonosis widespread in tropical regions. As part of studies on environmental factors that promote the initiation and propagation, we are working on datamining that provides tools for reporting and analysis for monitoring and decision making. This article propose a multi-dimensional model for data representation that taking into account the complexity of the disease in the heart of 3 systems : agroecological, pathogen and economic, health and social [2].

**MOTS-CLÉS :** modélisation multidimensionnelle, entrepôt de données, données complexes, extraction de connaissances, Fièvre de la Vallée du Rift, multi échelle, hiérarchie.

**KEYWORDS:** Multi-dimensional modeling, datawarehouse, complex data, datamining, Rift Valley Fever, multi-scale, hierarchy.

.....

*Actes du 4<sup>e</sup> Colloque National sur la Recherche en Informatique et ses  
Applications – Thiès-Bambey, Sénégal, Avril 2012*

.....

## 1. Introduction

Les indicateurs environnementaux influent considérablement sur la santé des populations humaines et animales. C'est pourquoi la recherche sur les facteurs de propagation des maladies à transmission vectorielle, telle que la Fièvre de la Vallée du Rift (FVR), prend en compte ces indicateurs. La FVR est une maladie virale, plus exactement une zoonose due à un arbovirus de la famille des *Bunyaviridæ*, genre *Phlebovirus*. Elle est transmise par des moustiques [4, 5, 6]. Même si une large gamme d'animaux est susceptible de développer la maladie, les petits ruminants domestiques et les bovins y sont les plus sensibles. La contamination de l'homme peut également se faire par aérosols ou par contact direct avec le sang ou autres liquides provenant d'un animal infecté [1]. Le virus de cette maladie a été isolé dans le Département de Kedougou, au Sénégal en 1974 sur des moustiques (*Aedes*). Plus tard, des enquêtes sérologiques ont décelé des affections chez des ruminants dans plusieurs zones du Sénégal (Haute Casamance, Ferlo, Vallée du Fleuve Sénégal) [7, 8, 12]. En 1987, des cas humains ont été identifiés à la frontière mauritanienne du Fleuve du Sénégal [11, 12]. L'étude [3] montre que la propagation de la FVR, identifiée dans la zone du Ferlo, est liée au cycle de vie des moustiques et au cycle d'évolution des mares (voir *figure 1*). En effet, les mares du Ferlo constituent à fois les sources d'approvisionnement en eau potable pour les populations et le bétail tout en étant le gîte larvaire des vecteurs potentiels de la FVR.

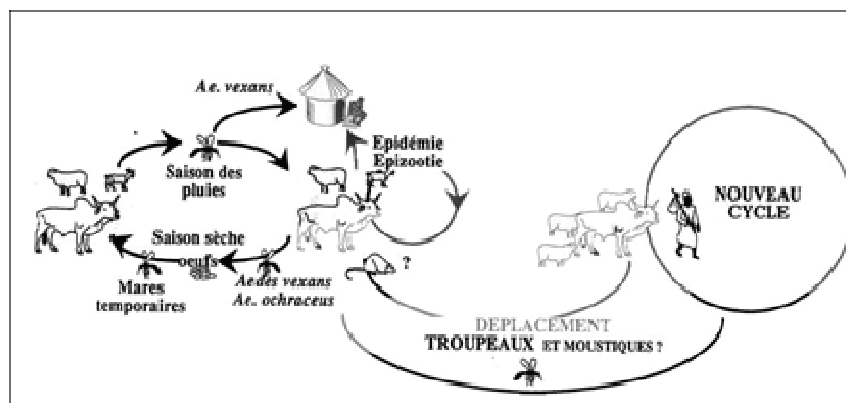


Figure 1. Cycle épidémiologique de la FVR au Ferlo [2]

Dans cette zone du Ferlo, la variabilité intra-saisonnière de la pluviométrie, la dynamique de la végétation et la turbidité des mares temporaires sont les principaux facteurs qui expliquent la forte concentration des moustiques [9]. La complexité de la

URED

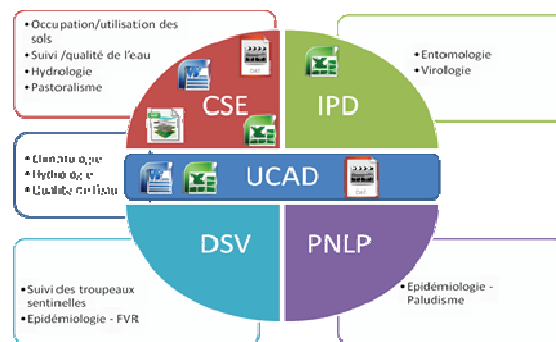
FVR [2] implique nécessairement la prise en compte de plusieurs paramètres dans l'étude de cette maladie. Ceci justifie le fait qu'au Sénégal plusieurs équipes de diverses disciplines (climatologie, entomologie, virologie, environnement, etc.) travaillent en synergie afin de mieux comprendre le climat, le fonctionnement des mares et la qualité de l'eau qui s'y trouve, la dynamique des populations des vecteurs et le déplacement des hôtes (troupeaux et population) ainsi que les pratiques pastorales.

Ainsi, nous nous proposons de leur offrir un système fédéré pour le stockage des données et l'analyse des données par couplage, corrélation ... Dans cet article, nous aborderons, dans un premier temps, les objectifs de nos travaux de recherche puis nous présenterons le modèle de données et les rapports d'analyse qui en résultent.

## 2. Position du problème

De ce fait, notre objectif est d'offrir des outils d'extraction des connaissances, à priori non évidentes, basées sur la mise en place de divers scénarii en interrogeant l'entrepôt. Cela permettrait de répondre à certaines questions soulevées par les recherches sur la propagation de la FVR. La complexité des systèmes étudiés (environnement, climat, hydrologie, qualité de l'eau, entomologie, épidémiologie, virologie, socio-économie...) a rendu nécessaire la prise en compte dans un même environnement de phénomènes multiples et variés, mais aussi à différentes échelles (temporelle, spatiale et organisationnelle).

Comme le montre la figure 2, les partenaires impliqués au Sénégal nous ont fourni des données provenant des enquêtes de terrain (données observées ou données générées par les équipements) ou des tests de laboratoire (données d'analyse). Ces données se présentent sous des formats « bureautique » (tableur, traitement de texte), des formats « données » (data) ou des données géographiques (shape file).



**Figure 2.** Complexité pluridisciplinaire des acteurs autour de la problématique de la FVR à Barkédji

### 3. Résultats

La modélisation multi dimensionnelle offre un formalisme dédié aux systèmes décisionnels. Elle est décrite comme une démarche de représentation qui consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions [10]. Les données sont organisées de manière à mettre en évidence les données quantitatives (faits) et les données qualifiantes (dimensions). « Le fait modélise le sujet de l'analyse ; il est formé de mesures (dimensions) correspondant aux informations de l'activité analysé » [13].

Les entrepôts de données « classiques » utilisent des techniques d'analyse qui traitent seulement des données simples (chaînes de caractères, nombres, etc). Les données complexes engendrent plus d'informations nécessaires à l'analyse de données. Ainsi, pour représenter une donnée complexe, il est nécessaire de représenter, en plus des descripteurs de bas niveau, des connaissances et des métadonnées associées.

Nous utilisons le modèle en constellation pour représenter nos données.

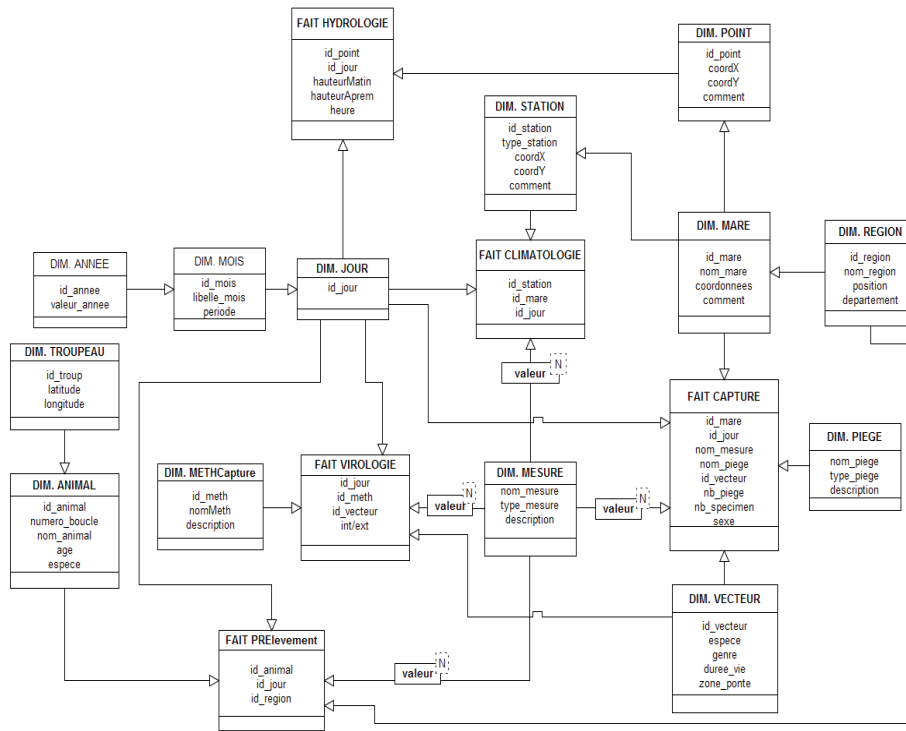


Figure 3. Modèle de données

URED

Nous avons identifié les tables de faits suivantes (voir Figure 3) :

- Hydrologie : pour recenser les niveaux d'eau des mares ;
- Climatologie : pour les caractéristiques climatiques ;
- Capture : pour les données sur les captures entomologiques ;
- Virologie : pour les tests virologiques effectués sur les vecteurs capturés ;
- Prélèvement : pour les prélèvements sérologiques (sang, avorton ...) effectués.

Les faits sont analysés suivant différentes perspectives. Ces perspectives correspondent à une catégorie utilisée pour caractériser les mesures d'activité analysées ; on parle de dimensions. Par exemple, dans le fait « Hydrologie », nous pouvons avoir la mesure "hauteur pluviométrique" suivant l'axe « lieu » et/ou l'axe « date » - qui représentent des dimensions. Les dimensions ont été identifiées en fonction des différents critères d'analyse ; on retrouve ainsi (voir Figure 3) :

- les vecteurs (moustiques) ;
- les stations (pluviomètre ou station installé dans les sites) ;
- les pièges posés pour capturer les moustiques (appât de mouton, CO2, ...) ;
- les mesures qui recensent l'ensemble des mesures prévisibles ou non ;

Lors du processus d'analyse, les données sont généralement analysées en partant d'un faible niveau de granularité vers des données plus détaillées pour "forcer vers le bas". Pour définir ces différents niveaux de détail, chaque dimension est munie d'une (ou plusieurs) hiérarchie(s) de paramètres. A travers le même exemple cité ci-dessus, nous pouvons être plus précis en affinant le lieu en « point d'eau → mare → région » ou la date « en jour → mois → année ».

Pour la visualisation, nous avons utilisé la suite décisionnelle Pentaho. Les données proviennent d'une étude faite par l'Institut Pasteur de Dakar en 1991. Ainsi, nous avons alimenté notre base avec des données entomologiques en utilisant l'outil Pentaho Analyzer Report. Après modélisation du cube, nous avons obtenu un tableau d'analyse qui regroupe des statistiques sur les insectes recensés dans la région de Barkédji. Ce tableau d'analyse nous a permis de générer plusieurs types de rapport en utilisant divers formats. Le diagramme, figure 5, résume la répartition des vecteurs (genre) suivant le type de moyen de capture utilisé sous divers format.

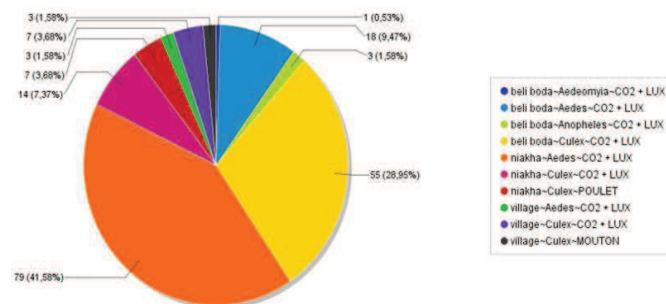


Figure 4. Répartition des genres / mode de capture

Bouba et al.

Dans la figure 6, nous comparons cette répartition à celle d'une analyse plus affinée (en augmentant le niveau de détail par l'intégration de la dimension « espèce »).

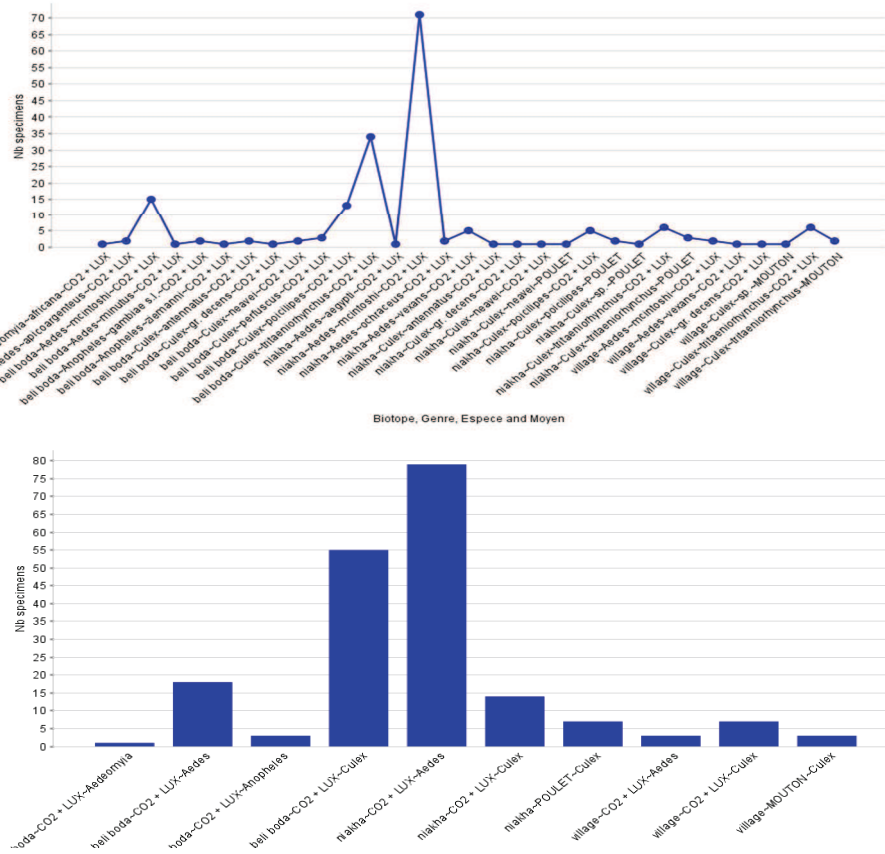


Figure 5. Comparaison des résultats de répartition des genres vs espèces

Le modèle de données obtenu, qui représente une vue globale des données utilisées par les différentes équipes de chercheurs, nous a permis de mettre en exergue les caractéristiques et les objets communs à la climatologie, l'hydrologie, l'épidémiologie la virologie. La pertinence de ce modèle se justifie dès lors car il nous donne les principales indications voulues et les liens qui en découlent. En effet, les concepts de « fait » et de « dimension » du modèle de données permettent d'intégrer des données hétérogènes en exhibant leurs interactions.

URED

Pour résoudre le problème relatif à la non exhaustivité des caractéristiques des examens effectués sur les eaux de mares ou sur les vecteurs, nous avons introduit le concept de « cardinalités », emprunté au modèle entité-association.

D'une part, à travers cette représentation des données, nous pouvons constater les effets de la répartition des pluies dans une saison sur la qualité des mares ou encore sur les types de virus identifiés dans une région (cf Faits « Climatologie » et « Virologie » de la Figure 3). D'autre part, en intégrant ce modèle, à travers les différents diagrammes (Figures 4 et 5), nous constatons que les dimensions définies correspondent à des axes de mesures qui facilitent la génération de rapports d'analyse interactifs suivant les objectifs recherchés.

---

## 5. Conclusion et perspectives

Notre recherche vise à apporter des solutions dans la mise en place d'un système décisionnel en environnement-santé. L'objectif est de concevoir et de valider des solutions robustes en termes de fonctionnalités et de performances ; les données du processus de simulation doivent être modélisées, stockées, traitées et manipulées par des algorithmes robustes, performants, et adaptés aux supports décisionnels.

Ainsi, dans ce papier nous avons présenté le modèle de données qui nous sert de support pour la mise en place de notre entrepôt de données. L'implémentation de ce modèle, nous a permis de générer différents rapports sur la base des enquêtes de nos partenaires scientifiques. Le modèle actuellement établi peut répondre aux questions liées aux activités hydrologiques, climatologiques, virologiques et entomologiques de façon indépendantes mais également sur les interactions en découlant.

Dans la suite de nos travaux, nous comptons étendre ce modèle en y intégrant les indicateurs nécessaires aux activités épidémiologiques, à la génération de cartes d'émission de moustiques et à la maîtrise du déplacement des hôtes qui représente toute la composante socio-économique du pastoralisme dans la zone d'étude du projet. D'autre part, nous travaillons sur la définition d'algorithme d'extraction de données orientées analyse pour offrir des vues d'informations et de connaissances aux scientifiques et aux utilisateurs finaux.

---

## 6. Bibliographie et biographie

[1] Ba Y., Diallo D., Dia I., Diallo M., 2006, *Comportement trophique des vecteurs du virus de la fièvre de la vallée du Rift au Sénégal : implications dans l'épidémiologie de la maladie*. Bulletin de la Société Pathologique Exotique, 99 (4), 283-289.



Bouba et al.

- [2] De La Rocque S., 2004, *Des maladies qui entrent en scène : Homme, plante, animal : tous exposés*, In : CEMAGREF, CIRAD, IFREMER, INRA, IRD, Museum national d'histoire naturelle, Agropolis. Montpellier
- [3] Diallo M., 1995, Dynamique comparée des populations de Culicidae à Kédougou (zone soudano-guinéenne) et Barkédji (zone de savane sahélienne) : conséquences dans la transmission des arbovirus, Mémoire DEA, UCAD, 87 p.
- [4] Diallo M., Lochouarn L., Ba K., Sall A.A., Mondo M., Girault L., Mathiot C., 2000, *First isolation of the Rift Valley fever virus from Culex poicilipes (Diptera: Culicidae) in nature*. American Journal of Tropical Medicine and Hygiene, 62, 702-704.
- [5] Fontenille D., Traore-Lamizana M., Diallo M., Thonnon J., Digoutte J-P., Zeller H. G., 1998, *Nouveaux vecteurs de la fièvre de la vallée du Rift en Afrique de l'Ouest*. Emerging Infectious Diseases, 4, 289-293.
- [6] Fontenille D., Traore-Lamizana M., Zeller H.G., Mondo M., Diallo M., Digoutte J.P., 1995, *Rift valley fever in western Africa : isolations from Aedes mosquitoes during an inter epizootic period*. American Journal of Tropical Medicine and Hygiene, 5, 403-404
- [7] Guillaud M., Le Guenno B., Wilson M.L., Desoutter D., Gonzalez J.P., Digoutte J.P., 1988, *Prévalence en anticorps contre la Fièvre de la Vallée du Rift chez les petits ruminants du Sénégal*. Institut Pasteur, Virol., 139 : 455-459
- [8] Jouan A., Le Guenno B, Digoutte JP., Philippe B., Riou O. et Adam F. - 1988 - *An RVF epidemic in Southern Mauritania*. Annales de Virologie Institut Pasteur., 139: pp. 307-308
- [9] Ndione JA., Lacaux JP., Tourre Y., Vignolles C., Fontanaz D. et Lafaye M., 2009, *Mares temporaires et risques sanitaires au Ferlo : contribution de la teledetection pour l'étude de la fièvre de la vallée du Rift entre aout 2003 et janvier 2004*
- [10] Site officiel de Pentaho, <http://www.pentaho.com/>
- [11] Ravat F., Teste O. et Zurfluh G., 2001, *Modélisation multidimensionnelle des systèmes décisionnels*, <http://www.irit.fr/recherches/IRI/SIG/personnes/teste/pdfs/eca01.pdf>
- [12] Saluzzo JF., Anderson GW., Hodgson LA., Digoutte JP. et Smith JF., 1989 - *Antigenic and biological properties of Rift Valley fever virus isolated during the 1987 Mauritanian epidemic*. Research in Virology, pp. 155-164
- [13] Sarr J., Diop M., Dieme Y., 1988, *La Fièvre de la Vallée du Rift au Sénégal données épizootiologiques dans le triangle Dagana- Podore et Niassante entre 1982 et 1984*, LNERV, 9p
- [14] Teste O., 2000, *Modélisation et manipulation d'entrepôts de données complexes et historisées*, Thèse de Doctorat de l'Institut de Recherche en Informatique - Université Paul Sabatier de Toulouse France.

URED

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

## Optimisation d'un Protocole d'Authentification dans les Réseaux de Capteurs Sans Fil

Youssou Faye\* —Ibrahima Niang\*\* —Hervé Guyennet\*

\* Département d'informatique  
Université de Franche Comté  
Besançon Cedex  
FRANCE

\*\* Département d'informatique  
Université Cheikh Anta Diop  
Dakar  
SENEGAL

yfaye@femto-st.fr, herve.guyennet@femtost.fr iniang@ucad.sn

.....

**RÉSUMÉ.** Pour plusieurs types d'applications des Réseaux de Capteurs Sans Fil (RCSFs), produire une variété de fonctions de sécurité avec une faible consommation de ressources devient un vrai défi. Récemment, Wong et al., puis Vaidya et al. ont proposé un protocole qui permet aux utilisateurs d'un RCSF de s'authentifier avant d'accéder aux services fournis par le RCSFs. Dans ce papier, après avoir montré les vulnérabilités lors d'attaque par Déni de Service (DoS) et par falsification sur le protocole proposé par Vaidya et al.. Nous proposons une solution efficace dans l'environnement des capteurs, qui non seulement conserve tous les avantages du protocole proposé par Vaidya et al., mais améliore sa sécurité. Nous montrons par la suite que notre solution est sécurisée contre les attaques mentionnées, et consomme moins d'énergie dans certains cas.

**ABSTRACT.** For various Wireless Sensor Networks (WSNs) applications, providing a variety of security functions with limited energy resources and low power capabilities is a very challenge. Recently, Vaidya and al. proposed an Improved Robust Dynamic User Authentication for WSNs that allows legitimate users to query sensor data at every sensor node of the network. In this work, we show that Vaidya and al.'s scheme suffers from the risk of forgery attacks and denial of-service (DoS) attacks. To cope with them, we propose a new solution which is quite adequate for power and resource constrained sensor networks. The proposed scheme not only retains all the advantages in Vaidya and al.'s scheme but also enhances its security. After an overhead coast energy comparison, we show that our scheme is protected against above attacks and justify better energy consumption.

MOTS-CLÉS: Réseaux de Capteurs Sans Fil, contrôle d'accès, authentification, mot de passe

KEYWORDS: Wireless Sensor Networks, access control, authentication, password.

.....

## 1. Introduction

Le RCSF est souvent destiné au contrôle d'espaces géographiquement limités. En général, les données récoltées par les capteurs comme par exemple la température ne sont pas confidentielles. Dans certaines applications, les requêtes sont envoyées à une station de base ou à la passerelle du réseau. Cependant, pour les applications temps réel ou critiques, comme par exemple les applications militaires, les données critiques doivent être protégées contre toute utilisation frauduleuse, et accessibles en temps réel non seulement depuis la station de base ou la passerelle du réseau, mais parfois aussi depuis n'importe où dans le réseau à travers les capteurs en mode ad hoc.

Dans ce contexte, le contrôle d'accès au réseau devient nécessaire [1]. L'authentification des utilisateurs est la solution la plus utilisée dans les réseaux traditionnels. Cependant, dans les RCSFs, elle reste moins étudiée à cause des contraintes énergétiques, de mémorisation, de calcul et de transmission.

Vaidya et al.[2], une des dernières solutions basée sur un mot de passe fort est divisée en quatre phases: une phase d'enregistrement, une phase de login, une phase d'authentification et une phase de changement de mot de passe. Dans cet article, nous montrons que, la solution de Vaidya et al., de par son manque de vérification du mot de passe lors de la phase de login, est vulnérable lors d'attaque par DoS. De même, elle manque de protection contre la falsification des estampilles temporelles. Nous proposons une solution qui maintient tous ses avantages et améliore sa sécurité par l'intégration d'un mécanisme de vérification de mot de passe et de chiffrement des estampilles basés sur les fonctions de hachage et l'opérateur OU-exclusif

Le reste du papier est organisé comme suit. La section 2 présente l'état de l'art. Une description de la solution de Vaidya et al. [2] est fournie à la section 3. Une analyse de leur solution et une présentation de celle proposée sont respectivement décrites dans les sections 4 et 5. La sécurité du nouveau protocole est analysée à la section 6. La section 7 décrit une implémentation de notre solution et celle de Vaidya et al.[2]. Enfin, nous concluons le papier et identifions des perspectives de recherche à la section 8.

---

## 2. Etat de l'art.

Le contrôle d'accès a toujours été un problème classique dans beaucoup d'applications et systèmes informatiques existants. L'authentification à distance des utilisateurs a été depuis longtemps la solution de base la plus utilisée. Initialement, les solutions d'authentification d'utilisateurs [3,4,5,6,7 ] proposées dans l'environnement des cartes à puce étaient inspirées de Lamport(1981) [8], à la différence qu'aucune table de vérification n'était stockée dans le système distant pour la validité du login de l'utilisateur. Ces solutions utilisent une approche par mot de passe avec un login statique. Certaines d'entre elles [4] utilisent la techniques d'un mot de passe faible. Elles présentent l'avantage d'une mémorisation facile du mot de passe. Cependant la cryptographie à clé publique utilisée reste le principal inconvénient pour une application dans l'environnement des

URED

capteurs. Par contre, d'autres solutions [3,5,6,7 ] utilisent la technique d'un mot de passe fort et sont basées uniquement sur les fonctions de hachage et l'opérateur OU-exclusif, ce qui facilite leur implémentation dans les RCSFs. Leur inconvénient réside dans la difficulté de mémorisation du mot de passe. Des solutions comme [9] implantent la technique du mot de passe fort avec un login dynamique afin de se protéger contre l'usurpation de login. Ce qui permet le libre changement de login et de mot de passe. Cependant, Lee et al.[10] ont montré que ces protocoles basés sur un login dynamique sont vulnérables aux attaques telles que la répétition, la contrefaçon de login et la fabrication. Leur solution proposée pour les cartes à puce utilise la même technique des fonctions de hachage et du OU-exclusif. Pour une adaptation dans les RCSFs, Wong et al. [11] proposent une solution basée sur Lee et al.[10] moins coûteuse en calcul. La solution de Tseng et al. [13] montre les insuffisances de Wong et al. [11] et améliore sa sécurité. C'est ainsi que Vaidya et al., dans [2], proposent une version plus robuste de [12] basée sur Wong et al. [11].

---

### 3. Passage en revue du protocole de Vaidya et al..

Dans les prochaines sections, nous allons décrire les trois premières phases de la solution de Vaidya et al. [2] en utilisant les notations du Tableau 1. Seule la phase de changement de mot de passe n'est pas décrite car elle rest inchangée dans les deux solutions.

#### ***Phase d'Enregistrement (PE)***

*PE1*-L'utilisateur (UD) au voisinage de la passerelle (GW) choisit librement un mot de passe PW et calcule le haché  $vpw = H(PW)$ .

*PE2*-Au temps TS, UD envoie son identité UID et vpw à la GW en mode sécurisé.

*PE3*-La GW calcule  $X = H(UID||x)$ , stocke (UID, vpw, X, TS), puis répond à UD que l'enregistrement est effectué avec succès (Succ\_Reg (X)), et distribue ensuite les paramètres (UID, X, TS) aux capteurs de login (LN) capables de fournir des interfaces à l'UD pour se logger.

#### ***Phase de Login (PL)***

*PL1*-L'UD au voisinage d'un LN calcule  $A = H(vpw||t)$  et soumet (UID, A, t) au LN.

*PL2*- Après réception de la requête à  $T_0$ , le LN vérifie la validité du UID et si  $T_0 - t \geq \Delta T$ .

*PL3*-Si les conditions sont vérifiées, le LN récupère le paramètre A correspondant et calcule  $C_K = (X \oplus A \oplus T_0)$ , puis il envoie (UID,  $C_K$ ,  $T_0$ , t) à la GW.

#### ***Phase d'Authentification (PA)***

*PA1*- A la réception du message de login, la GW vérifie la validité de UID et t. S'ils sont valides, alors la GW vérifie si  $T_1 - T_0 \geq \Delta T$  et  $T_0 - t \geq \Delta T$  afin d'éviter la répétition de la requête de login. S'ils sont valides, la GW récupère le vpw et le paramètre A correspondant, puis calcule  $A' = H(vpw|| t)$  et  $C_K' = (X \oplus A' \oplus T_0)$ . Si  $C_K \neq C_K'$ , le message de login est rejeté, sinon, la GW calcule  $V_M = H(X||A'||T_1)$ .

*PA2*- La GW envoie un message d'acceptation (Acc\_login,  $V_M$ ,  $T_1$ ) au LN.

**Table 1: Notations**

Symboles	Description
UD	Dispositif de l'utilisateur (PDA, PC etc..)
GW, LN	Respectivement, le nœud passerelle et le nœud login
H()	Une fonction de hachage à sens unique
$\oplus, \parallel$	Respectivement l'opérateur OU-exclusif et de concaténation
Succ_Reg	Message d'enregistrement avec succes
Acc_login	Message d'acception du login
Succ_Change	Message de changement avec succes
$x, \Delta T$	Respectivement la clé de la passerelle et le délai de transmission
UID, PW	Respectivement l'identité de l'utilisateur et son mot de passe
TS	Temps d'enregistrement d'un noeud
T, T0, T1, T2, T3	Temps actuels enregistrés par un des noeuds

PA3- Le LN calcule  $V_M' = H(X \| A \| T_1)$ , et si  $V_M = V_M'$ , il calcule  $Y_K = H(V_M' \| T_2)$ . PA4-  
 Le LN envoie (Acc\_login,  $Y_K, T_1, T_2$ ) à l'UD. PA5-  
 Après réception du message au temps  $T_3$ , l'UD vérifie si  $T_1 - T_0 \geq \Delta T$  et  $T_0 - t \geq \Delta T$ . Si ces conditions  
 vérifiées, l'UD calcule  $V_M'' = H(X \| A \| T_1)$  et  $Y_K' = H(V_M'' \| T_2)$ , si  $Y_K = Y_K'$ , l'UD commence à  
 obtenir les données, sinon le message d'acceptation de login (Acc\_login) est rejeté.

#### 4. Vulnérabilités du protocole de Vaidya et al..

Nous allons démontrer que Vaidya et al.[2] est vulnérable lors d'attaque par DoS et par falsification d'estampilles temporaires. Durant la phase de login, seuls le UID et le temps  $t$  sont vérifiés par le LN, le DoS peut survenir de deux façons. Premièrement l'intrus peut intercepter ou écouter un UID valide puis le soumet avec un faux mot de passe. Puisque le LN vérifie seulement le UID, il va retransmettre ce message via les capteurs à la GW située à plusieurs sauts. Deuxièmement, ce scénario peut arriver autrement. Généralement, les PWs sont masqués lors de la saisie, si un UD se trompe de saisie de mot de passe, le même effet se reproduit. La transmission étant généralement l'opération la plus coûteuse en énergie, la propagation de fausses requêtes doit être évitée.

Vaidya et al.[2] ont supposé qu'un attaquant ayant capturé un LN, obtient UID, X, TS, et qu'il lui est aussi possible d'écouter le message (UID, A, t) afin de démontrer une attaque par falsification sur Wong et al. [11]. Partant de ces suppositions, leur protocole, de façon différente, sera vulnérable à l'attaque par falsification des temps transmis. Puisque seul le délai de transmission est vérifié, l'attaquant peut créer deux faux temps  $T_0'$  et  $t'$  dont la différence respecte le délai de propagation en y ajoutant un petit nombre  $\xi$  partout sur les deux temps  $T_0$  et  $t$  déjà transmis en clair dans le réseau.

URED

Ainsi il effectue  $T_0' = T_0 + \xi$ ,  $t' = t + \xi$ , ensuite calcule  $C_K' = H(X \parallel A \parallel T_0')$  puis envoie le message (UID,  $C_K', T_0', t'$ ). Puisque  $(T_1 - T_0) \leq \Delta T$  et  $(T_0' - t') \leq \Delta T$ , le message passe.

## 5. Solution proposée.

Dans cette section, nous proposons une nouvelle solution afin de résoudre les faiblesses notées dans Vaidya et al.[2]. Notre solution comporte les mêmes phases que Vaidya et al.

### *Phase d'Enregistrement (PE)*

*PE1*- Dans Vaidya et al., l'UD choisit un mot de passe PW, puis calcule  $vpw = H(PW)$  et envoie vpw avec son identité pour se loguer. Et pour le reste du protocole, le PW n'est plus utilisé. Il n'est pas nécessaire de calculer vpw, qui est autant vulnérable que PW. C'est ainsi que dans la solution proposée, l'UD choisit librement son mot de passe PW.

*PE2*- Au temps TS, l'UD soumet son UID avec PW à la GW en mode sécurisé.

*PE3*- La GW calcule  $X = H(\text{UID} \parallel x)$  puis répond à l'UD le message Succ\_Reg(X). Elle stocke les paramètres (UID, H(PW), X, TS), et distribue (UID, X, H(PW), TS) aux LNs.

### *Phase de Login (PL)*

*PL1*- L'UD calcule  $A = H(H(PW) \parallel t)$  et soumet (UID, A, t) au LN.

*PL2*- Après réception de la requête au temps  $T_0$ , le LN vérifie : si l'identité UID est non valide ou  $A \neq H(H(PW) \parallel t)$ , ou  $T_0 - t \geq \Delta$  alors le message de login est rejeté.

*PL3*- Le LN calcule  $C_K = (X \oplus A \oplus T_0)$  et  $t' = H^2(PW) \oplus t \oplus x$ , et envoie (UID,  $C_K, T_0, t'$ ) à GW.

### *Phase d'authentification*

*PA1*- La GW vérifie si le UID et le temps t sont valides, puis elle calcule  $t = t' \oplus H^2(PW) \oplus x$ , ensuite vérifie si  $T_1 - T_0 \geq \Delta T$  et  $T_0 - t \geq \Delta T$ . Si ces conditions sont valides, la GW récupère H(PW) et le paramètre A puis calcule  $A' = H(H(PW) \parallel t)$  et  $C_K' = (X \oplus A' \oplus T_0)$ . Le message de login est rejeté si  $C_K \neq C_K'$ , sinon la GW calcule  $V_M = H(X \parallel A' \parallel T_1)$ .

*PA2*- La GW envoie le message (Acc\_login,  $V_M, T_1$ ) au LN et stocke t.

*PA3*- Le LN calcule  $V_M' = H(X \parallel A \parallel T_1)$ , et si  $V_M = V_M'$ , il calcule  $Y_K = H(V_M' \parallel T_2)$ .

Le LN envoie le message (Acc\_login,  $Y_K, T_1, T_2$ ) à l'utilisateur UD.

*PA4*-

*PA5*-

Après réception du message au temps  $T_3$ , l'UD vérifie si  $T_1 - T_0 \geq \Delta T$  et  $T_0 - t \geq \Delta T$ . Si les conditions sont valides, l'UD calcule,  $V_M'' = H(X \parallel A \parallel T_1)$  et  $Y_K' = H(V_M'' \parallel T_2)$ , puis vérifie si  $Y_K = Y_K'$ . Si la condition est vraie, l'UD commence à obtenir les données, sinon il rejette le message d'acceptation de login.

## 6. Sécurité de la solution proposée.

Dans cette partie, nous allons démontrer que notre solution est résistante aux attaques mentionnées. Cette section se termine par une étude comparative avec d'autres solutions.

### 6.1. Gestion des attaques par Déni de Service et par falsification

*Le Deni de Service* : en donnant aux LNs la possibilité de vérifier le mot de passe, notre solution se protège contre le DoS. Puisque le LN stocke  $H(PW)$ , après réception du message (UID, A,t), il calcule  $A'=H(H(PW)||t)$ . Si  $A'=A$ , alors le mot de passe est correct.

*La falsification* : pour éviter la falsification des estampilles temporaires entre le LN et la GW, nous les envoyons en mode sécurisé avec une utilisation du OU-exclusif (XOR). Ainsi le LN, après avoir reçu le message de login, calcule une fausse estampille  $t'=H^2(PW)\oplus t\oplus x$  puis envoie le message (UID, $C_K$ , $T_0$ , $t'$ ) au lieu de (UID, $C_K$ , $T_0$ , $t$ ) à GW. Ce qui rend l'estampille t confidentiel. Puisque  $H^2(PW)\oplus t\oplus H^2(PW)\oplus x\oplus x=t$ , la GW calcule  $t=t'\oplus H^2(PW)\oplus x$

### 6.2. Etude comparative

Le Tableau 2, donne une comparaison en termes d'opérations de hachage, de OU-exclusif et de nombre de communications multi-sauts de quelques solutions.

**Tableau2: Nombre d'opérations effectuées**

Protocoles	Nombres total d'opérations
Wong et al.[11]	$7T_H+4T_{XOR}+3C_{MH}$
Tseng et al.[13]	$5T_H+4T_{XOR}+3C_{MH}$
Vaidya et al.[12]	$8T_H+4T_{XOR}+3C_{MH}$
Vaidya et al.[2]	$11T_H+4T_{XOR}+3C_{MH}$
Solution proposée	$15T_H+9T_{XOR}+3C_{MH}$

**Tableau 3: Nombre d'opérations effectuées**

Protocoles	Nombres total d'opérations
Vaidya et al.[12]	$4T_H+2T_{XOR}+2C_{MH}$
Vaidya et al.[2]	$4T_H+2T_{XOR}+2C_{MH}$
Solution proposée	$3T_H$

$T_H$  : temps pour exécuter la fonction de hachage H().

$T_{XOR}$  : temps pour exécuter l'opération XOR

$C_{MH}$ : Délai de communication multi-sauts entre le LN et la GW.

ND: Ces opérations peuvent aussi être traduites en énergie

D'après le Tableau 2, on peut remarquer que la solution proposée a un coût supérieur de quatre opérations de hachage ( $T_H$ ) et de cinq opérations OU-exclusif ( $T_{XOR}$ ) à celle de Vaidya et al.[2]. Notons que le coût en énergie de l'opération OU-exclusif est largement inférieur à celui du hachage.

Par ailleurs, si on considère les requêtes avec une UID valide mais avec un mot de passe erroné qui proviendrait de l'intrus ou d'une erreur de saisie de la part d'un utilisateur légitime. Pour notre solution, cette requête sera freinée au niveau du LN, et pour les autres solutions, il sera propagé jusqu'à la GW. Une comparaison est effectuée sur le Tableau 3.

U R E D

## 7. Validation expérimentale.

Nous rapellons qu'il y a deux protocoles de Vaidya et al.[2, 12]. Le but de notre implémentation est d'estimer la consommation énergétique en fonction du nombre de sauts entre le LN et la GW en s'appuyant sur les comparaisons effectuées dans les deux tableaux de la précédente section, . Nous avons fait une implémentation de notre solution et celle de Vaidya et al. [2] avec TinyOS. Le programme est testé sur la plateforme MicaZ, et le simulateur Avrora est utilisé pour mesurer la consommation d'énergie. Nous avons évalué la consommation d'énergie en fonction du nombre de sauts entre le LN et la GW en se basant sur le Tableau 2 où les deux protocoles sont considérés sans attaques. Pour chaque paramètre de données comme (UID,A,PW,  $C_k$  etc.) on a utilisé des données de 16 bits. Pour le hachage, nous avons utilisé une implémentation de la fonction de hachage universelle PolyR décrite dans le papier de Ted and al.[14], comme une interface TinyOS. La Figure 1, montre la consommation d'énergie de chaque solution.

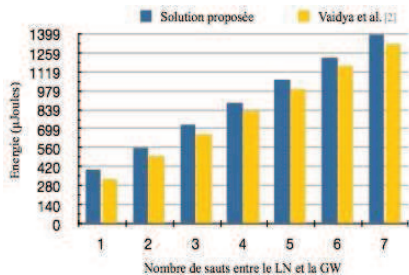


Figure 1 : Consommation énergétique basée sur le Tableau 2

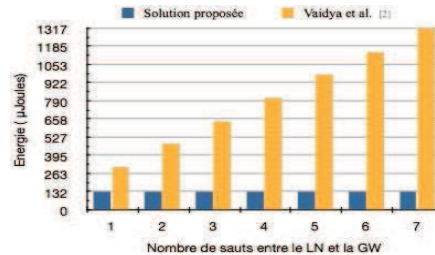


Figure 2 : Consommation énergétique basée sur le Tableau 3

A la deuxième étape, nous nous intéressons à l'effet de propagation d'une fausse requête sur la consommation énergétique. Ainsi nous avons utilisé le Tableau 3 pour évaluer l'énergie consommée en fonction du nombre de sauts entre le LN et la GW. La Figure 2, montre la consommation d'énergie de chacune des solutions. Ainsi nous pouvons voir que l'énergie reste constante pour notre solution car la fausse requête ne se propage pas. Elle augmente en fonction du nombre de sauts dû aux transmissions des capteurs assurant la propagation de la fausse requête.

## 8. Conclusion et perspectives.

Dans ce papier, nous avons proposé d'optimiser le protocole de Vaidya et al.[2] afin d'avoir un protocole beaucoup plus sûr dans l'environnement des capteurs. La solution proposée conserve tous les avantages de celle de Vaidya et al. et améliore sa sécurité par la protection contre le DoS et la



falsification. Dans certains cas, elle justifie une meilleure sécurité. Et dans d'autres cas, nous avons aussi montré dans l'implémentation, qu'elle est bien meilleure en consommation énergétique. Nos travaux futurs vont dans le sens d'introduire la probabilité de risque à partir de laquelle le comportement de chaque solution sera étudié

---

## 9. Bibliographie

- [1] Y. FAYE, I. NIANG, T. Noël « A Survey of Access Control Schemes in Wireless Sensor Networks », World Academy of Science, Engineering and Technology,, no Issue 59 : 2011, Paris, France, Pages 814-823, November 2011.
- [2] BINOD VAIDYA, M. Chen , J. Rodrigues « Improved Robust User Authentication Scheme for WirelessSensorNetworks,WirelessCommunicationandSensorNetworks(WCSN)», 2009 Fifth IEEE Conference : December 15-19,
- [3] A. K. AWASTHI,S. LAL, « A remote user authentication scheme using smart cards with Forward Secrecy, », IEEE Transactions on Consumer Electronics,, vol. 49, no 4 pp.1246-1248, Nov. 2003.
- [4] M. HWANG, C. Chang, K. Hwang, « An ElGamal-like cryptosystem for enciphering large messages », IEEE Trans .on Knowledge and Data Engineering, vol. 14, no 2, pp.445-446, 2002..
- [5] C. C. LEE, L. H. Li, M. S. Hwang « A remote user authentication scheme using hash functions », ACM Operating Systems Review, vol. 36, no 4, pp.23-29, 2002.
- [6] J. J. SHEN, C. W. LIN, M. S. Hwang « A modified remote user authentication scheme using smart cards », IEEE Trans. on Consumer Electron., vol. 49,no 2, pp.414-416, May 2003.
- [7] H.M.SUN,«AnEfficientremoteseuserauthenticationschemeusingsmartcards»,IEEETrans. on Consumer Electron., vol. 46, no 4, pp. 958-961, Nov. 2000.
- [8] L.LAMPORT«Passwordauthenticationwithinsecurecommunication,Communicationsofthe ACM, 1981. », Commun ACM , no 1981 ; vol.24, no.11, pp.770-772.
- [9] M.L. DAS, A. SAXENA, V.P. Gulati « A Dynamic ID-based Remote User Authentication Scheme », IEEE Transactions on Consumer Electronics, vol. 50, no 2, 2004.
- [10] C. LEE, C.H. Lin,, C. Chang « An Improved Low Communication Cost User Authentication Scheme for Mobile Communication », Proceedings of the IEEE 19th International Conference on Advance Information Networking and Applications (AINA 2005), no Taiwan, 2005.
- [11] K.WONG,Y.ZHENG,J.Cao,SWang«Adynamicuserauthenticationscheme for wireless sensor networks », In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 06), vol. 1, no Jun. 2006, pp. 244-251.
- [12] B. VAIDYAJ.S. SILVA, J.J. Rodrigues, « Robust Dynamic User Authentication Scheme for Wireless Sensor Networks », In Proc. of the 5th ACM Symposium on QoS and Security for wireless and mobile networks (Q2SWinet 2009), Tenerife, Spain, Oct., no 2009, pp 88-91.
- [13] H. R. TSENGJAN, R. H., W. Yang « An improved dynamic user authentication scheme for wireless sensor networks. », In Proceedings of the IEEE Global Communications Conference (GLOBECOM07), Nov. 2007, no 2007 ;986-990.
- [14] TEDKROVETZ,P.ROGAWAY,«Fast Universal Hashing with Small Keysand No Preprocessing : The PolyR Construction », D. Won (Ed.) : ICISC 2000, LNCS 2015, pp. 73-89, 2001. Springer-Verlag Berlin Heidelberg 2001,

U R E D

=====

# GrAPP&S Data Grid

## Une approche de type grille et système pair à pair pour le stockage de données

Thierno Ahmadou Diallo\*‡ – Olivier Flauzac\* – Luiz Angelo Steffene†  
– Samba N'Diaye‡

‡ Département d'Informatique  
LMI, Université Cheikh Anta Diop  
5005 Dakar-Fann  
SENEGAL

thierno.diallo.sn@gmail.com  
samba.ndiaye@ucad.edu.sn

\* Département de Mathématique et Informatique  
CReSTIC, Université de Reims Champagne-Ardenne  
BP 1039, F-51687 Reims Cedex  
FRANCE

{olivier.flauzac, Luiz-Angelo.Steffene} @univ-reims.fr

=====

**RÉSUMÉ.** Dans cette article nous présentons **GrAPP&S (Grid APPLICATIONS & SERVICES)**, un intergiciel pair à pair (P2P) pour le stockage de tous types de données. En effet, GrAPP&S a pour vocation de gérer tous les types de données, soient ils des fichiers, bases de données, de documents structurés ou semi-structurés, des flux vidéo/Audio/VoIP, des systèmes de stockage *cloud* ou encore d'autres réseaux P2P. L'architecture GrAPP&S est très modulaire et fondée sur des protocoles ouverts, ce qui simplifie son déploiement dans n'importe quelle type d'environnement. Ceci permet une gestion totalement décentralisée, une haute disponibilité des données, et une réactivité face aux pannes de ses différents composants.

**ABSTRACT.** In this paper we present **GrAPP&S(Grid APPLICATIONS & SERVICES)**, a peer-to-peer (P2P) middleware designed to deal with any kind of data type. Indeed, data managed by GRAPP&S include files, databases, structured or semi-structured documents, data streams (video/audio/VoIP), cloud storage or even other P2P networks. GrAPP&S architecture is modular and relies on open protocols, which simplifies its deployment on different computational environments. This modularity allows a fully decentralized management, with high data availability and reactivity in the case of components' failures.

**MOTS-CLÉS :** Intergiciel, système pair à pair, grille de données.

**KEYWORDS:** Middleware, peer-to-peer system, data grid.

Actes du 4<sup>e</sup> Colloque National sur la Recherche en Informatique et ses Applications -  
Thiès-Bambey, Sénégal, Avril 2012

=====

## 1. Introduction

Le stockage de grande quantité de données est devenu crucial. De nombreux efforts ont été réalisés envers la proposition de solutions de stockage adaptées. Alors que les clusters, les SAN ou les grilles sont des alternatives pour la résolution de problèmes ponctuels nécessitant une grande capacité de stockage, ces solutions sont limitées par leur caractère fortement centralisé.

Pour résoudre ce problème, certaines approches s'appuient sur des réseaux pair à pair (P2P), qui offrent des propriétés intéressantes pour la tolérance aux pannes et le passage à l'échelle. Les architectures P2P désignent un ensemble d'utilisateurs, appelés nœuds, ainsi que les protocoles utilisés par ces nœuds pour communiquer entre eux. Plus précisément, les nœuds communiquent d'égal à égal, au contraire d'autres protocoles hiérarchiques type client/serveur. Ces architectures P2P permettent non seulement aux utilisateurs de mettre en commun des ressources, mais elles sont intrinsèquement distribuées, passent à l'échelle et conservent des bonnes performances en termes de temps d'accès à l'information. Il existe différents types d'architecture P2P : centralisées, structurées et non structurées [7].

Dans les architectures P2P centralisées, les pairs qui se connectent au réseau envoient une copie de leurs indexes au serveur central. Quand une requête est émise depuis un pair, elle est acheminée jusqu'au serveur qui traite cette requête puis retourne au pair émetteur une liste des pairs qui contiennent l'information recherchée. Le pair émetteur contacte ensuite directement les pairs qui possèdent les informations correspondantes aux critères de recherche afin de les récupérer. L'accès aux informations est donc décentralisé alors que la gestion des requêtes reste centralisée, ce qui rend le système fragile vis-à-vis d'une panne du serveur et pose des problèmes de passage à l'échelle (ce mécanisme limite la taille maximale du réseau). Napster [12], eDonkey [4] et Bittorrent [1] sont des exemples d'architecture P2P centralisée.

Les architectures structurées ont la particularité d'organiser le réseau P2P en une topologie routable, c'est-à-dire que chaque pair dispose d'un identifiant permettant de le localiser en suivant un chemin déterministe parmi les pairs, et nécessitant un minimum de messages. Chaque ressource partagée au sein du réseau possède également un identifiant, qui est le résultat d'une fonction de hachage et qui permet de la localiser rapidement. L'indexation des ressources dans les réseaux P2P structurées utilise le principe des tables de hachage distribuées (DHT), où chaque pair devient responsable des entrées de la table égales ou proches de son identifiant. Les DHT permettent une localisation des pairs et des données très performante. De même, le routage de proche en proche permet de limiter le nombre de messages à  $O(\log N)$ ,  $N$  étant le nombre de pairs dans le réseau. Les DHT sont implémentées dans bon nombre de systèmes comme Chord [13], Pastry [11] ou Can [10]. L'avantage des réseaux P2P structurées est le

**URED**

passage à l'échelle et son efficacité dans la recherche d'information. Cependant ils montrent leur limite dans les environnements dynamiques à cause de l'arrivée ou départ de bon nombre de nœuds. Un autre inconvénient vient du fait que les P2P structurées sont limitées à des données de même nature (fichier, par exemple) et ne permettent pas des recherches multiparamétriques, notamment à cause des fonctions de hachage utilisées.

Finalement, les architectures P2P non structurées sont basées sur des graphes aléatoires, utilisant une méthode de recherche par inondation. Ces solutions sont résistantes dans les environnements dynamiques du fait des faibles contraintes imposées sur la topologie virtuelle. On peut citer quelques exemples de systèmes non structurés, dont Gnutella 6.0 [12], Fast Track avec son client Kazaa [6] ou FreeNet [1]. Toutes ces solutions P2P sont des architectures planes, i.e., ne sont pas hiérarchiques. Pour améliorer les performances des réseaux P2P, des architectures hiérarchiques ont été proposées dans la littérature [9][5][3][10]. Cependant la majeure partie d'entre elles se focalisent uniquement sur l'amélioration du temps de latence lors d'une recherche. D'ailleurs, les architectures hiérarchiques existantes ne gèrent que des données de même type, ce qui limite leur domaine d'application.

Dans cet article, nous proposons une architecture hiérarchique nommée GrAPP&S, structurée sur deux niveaux : un niveau d'interconnexion et un niveau de communautés. Le niveau d'interconnexion est utilisé pour relier les différentes communautés, chacune composée par des nœuds qui peuvent communiquer mutuellement et qui partagent les mêmes propriétés. Cette approche permet la diversification des types de données stockées par GrAPP&S, qui permet ainsi la gestion de différentes sources de données, selon les propriétés de chaque communauté, tout en gardant l'interconnexion des sources grâce à la couche d'interconnexion.

Cet article est organisé comme suit : la section 2 est consacré à l'état de l'art et la section qui suit présente l'architecture GrAPP&S. La section 4 est consacrée à la gestion des nœuds et nous présentons les premières conclusions de ce travail en section 5.

---

## 2. Etat de l'art

Différentes architectures hiérarchique P2P ont été proposées dans [9][5][3][10]. Dans [5], l'auteur propose une architecture hiérarchique à deux niveaux. Au niveau inférieur il regroupe les pairs d'une même région, organisés sur un anneau de Chord et coordonnés par un super nœud. Au niveau supérieur se trouvent les supers nœuds de chaque région. [5] propose un algorithme de recherche régionale basé sur les super-nœuds, qui gardent une table de routage bidirectionnelle dans le but de réduire efficacement la redondance de la table de routage originale de Chord. Toutefois, même si l'architecture passe à l'échelle, elle ne résiste pas dans un environnement dynamique. Dans [3] les auteurs ont

proposé un modèle hiérarchique de DHT (HDHT), où les pairs sont organisés en groupes. L'objectif des HDHTs est d'améliorer l'architecture plane DHT conventionnelle, par une exploitation ressources hétérogènes des pairs, la mise en cache des infrastructures, la transparence et l'autonomie de différentes parties du système, afin de rendre la recherche de clés plus efficace tout en générant moins de trafic.

Les auteurs de [10] proposent l'architecture hiérarchique CBT pour améliorer le protocole de téléchargement de fichier de Bittorrent dans un réseau de grande échelle. Il utilise trois types de pairs : les sources, les téléchargeurs et les super pair. Tous les supers pairs sont connectés au *torrent tracker* (gérant de l'index des ressources) pour former un réseau dédié. La disponibilité des informations est fortement liée au *torrent tracker*, car si ce dernier tombe en panne tout le service disparaît.

Les travaux effectués dans [5][3][10] reposent sur une architecture hiérarchique à deux niveau. Leur objectif est d'améliorer efficacement les performances tels que la latence lors d'une recherche et générer moins de trafic réseau. Cependant, ces architectures montrent leur limite dans les environnements dynamiques à cause de l'instabilité des nœuds. Une alternative est proposée dans [9], où les auteurs proposent HP2P, une architecture hiérarchique hybride à deux niveaux, combinant DHT et systèmes P2P non structuré. HP2P utilise dans son premier niveau Chord et au deuxième niveau Kazaa, et procède par inondation lors d'une recherche. Ceci fait de HP2P un système robuste car elle combine les avantages des DHT et des systèmes non structurés, mais reste aussi limité par leurs inconvénients, comme la dépendance aux ressources de même type ou le nombre de messages lors d'une recherche. Face à ces travaux, nous sommes motivés à proposer la spécification d'une architecture hiérarchique plus générique, qui permet le stockage de tous les formats de données tout en conservant les avantages vis-à-vis de la performance réseau.

---

### 3. Présentation de l'architecture GrAPP&S

#### 3.1. Description

Afin de présenter notre architecture, nous introduisons dans un premier temps quelques notations. Une **communauté** (*C*) est une entité autonome, qui regroupe des nœuds qui peuvent se communiquer et qui partagent une propriété définie : même localisation, même autorité d'administration (par exemple, des serveurs distants appartenant à la même entreprise) ou même domaine d'application (base de données métier, par exemple).

Le **nœud communicateur** (*c*) joue un rôle essentiellement lié au transport d'informations et à l'interconnexion entre différentes communautés, comme par exemple

URED

lors du passage de messages à travers des pare-feux. C'est le point d'entrée de la communauté, et il assure sa sécurité vis-à-vis de l'extérieur. Un nœud *c* peut être par exemple un serveur php, un serveur J2EE, un serveur .Net. Il dispose d'un identifiant unique (ID) à partir duquel on construit les identités des autres nœuds de la communauté. Ce nœud ne stocke de données et ne fait pas d'indexation.

Les nœuds **ressource manager (RM)** assurent l'indexation et l'organisation des données dans la communauté. Ils reçoivent les requêtes des utilisateurs et assurent leur prétraitement. Les nœuds **RM** participent à la recherche de données dans la communauté.

Les nœuds **data manager (DM)** interagissent avec les sources de données, qui peuvent être dans différents supports tels que les bases de données (objet ou relationnelle), les documents (texte/XML/multimédia), des flux (vidéo, audio, VoIP), des données issus de capteurs ou encore une service *cloud*. Un nœud **DM** est un service qui dispose des composants suivants : (i) une interface (*proxy*) adaptée aux différentes sources de données (disque dur, serveur WebDAV, FTP, base de données, stockage sur *cloud* type Dropbox, etc.) ; (ii) un gestionnaire de requêtes qui permet d'exprimer des requêtes locales ou globales et (iii) un gestionnaire de communication qui permet au nœud **DM** de communiquer avec le nœud **RM** auquel il est connecté. Cette communication est assurée par des services tels que RMI, WSDL et des protocoles réseaux TCP, HTTP. Le nœud **DM** dispose d'un protocole de connexion spécifique au type de donnée, par exemple JDBC, ODBC, FTP, etc.

Les nœuds de GRAPP&S sont organisés de façon hiérarchique dans une communauté. Dans chaque communauté, il y'a un seul nœud *c* sur lequel est connecté un ou plusieurs nœuds **RM**. Sur chaque nœud **RM** est connecté un ou plusieurs nœuds **DM**. Les nœuds communiquent par message. Les communautés *C* sont reliées par un réseau d'interconnexion.

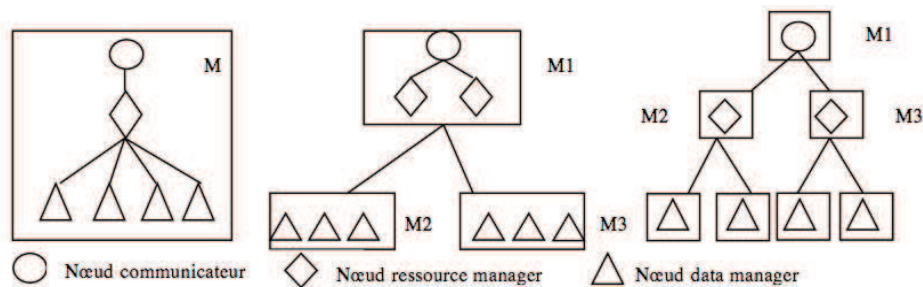


Figure 1 - Organisation des nœuds (a) dans une machine, (b) dans un cluster et (c) dans un réseau

### 3.2. Formation des communautés

GrAPP&S peut être déployé dans plusieurs types d'architecture selon le placement des nœuds. *(i)* les nœuds peuvent être regroupés dans une seule machine physique (voir Figure 1a). C'est l'exemple typique d'une machine d'un particulier, qui souhaite héberger une communauté de l'architecture. Le placement des nœuds sous cette forme peut être justifié par sa simplicité à mettre en œuvre lors de sa phase d'implémentation, en utilisant les concepts d'héritage et de polymorphisme. Les nœuds sont interconnectés par des sockets, des solutions RPC pour qu'ils puissent communiquer par message dans les deux sens entre deux nœuds. Dans *(ii)* les nœuds sont organisés dans une ferme de serveurs telle qu'un cluster, ce qui est caractéristique des réseaux HPC (Figure 1b). Finalement, *(iii)* les nœuds peuvent être regroupés s'ils partagent une même propriété de localisation ou d'administration. Ils sont proches géographiquement les uns des autres (voir Figure 1c). Ceci est l'exemple d'un réseau formé par les nœuds d'une entreprise ou d'un département.

### 3.3. Adressage des nœuds de GrAPP&S

Chaque nœud de GrAPP&S a un identifiant (ID) unique. L'adresse IP ou MAC ne peuvent pas régler l'identification des nœuds car ils peuvent être déployés dans seule machine. La solution la plus intéressante est celle proposée par JXTA [8] pour identifier ses pairs et qui utilise une chaîne de 128 bits. Chaque nœud dispose ainsi d'une chaîne unique  $ID_{local}$ , sous la forme «  $urn:nom\_communaute:uuid:chaîne-de-bit$  », de taille 128 bits. L'expression de l'adressage hiérarchique se fait par la concaténation des IDs sous forme de préfixe, i.e., l'ID du nœud  $c_i$  est équivalent à son  $ID_{local}$ , l'ID du nœud  $RM_i$  est formé par la «  $ID_{c_i}/ID_{RM_i}$  », et l'ID du nœud  $DM_i$  présente la forme «  $ID_{c_i}/ID_{RM_i}/ID_{DM_i}$  ».

---

## 4. Gestion des nœuds

La topologie du réseau change fréquemment à cause de la mobilité des nœuds. Nous travaillons dans l'hypothèse où les nœuds  $c$  et  $RM$  sont fixes dès la construction de la communauté ; ainsi, tout autre nœud qui arrive dans le réseau est un nœud  $DM$ .

### Arrivé d'un nœud

Quand un nœud  $DM$  arrive dans le réseau il dispose de deux moyens pour trouver un nœud  $RM$  sur lequel il peut se connecter. *(i)* Si le nœud  $DM_i$  connaît un ou plusieurs nœuds  $RM$ , il envoie un message de diffusion  $REQ()$  et collecte toutes les identités des nœuds  $RM$ , qu'il garde dans un tableau ordonné par l'identifiant. Il peut ainsi se connecter au nœud  $RM$  qui a l'identifiant le plus grand. Si ce dernier se déconnecte,

U R E D

alors le  $DM_i$  le supprime du tableau et se connecte au nœud  $RM$  suivant ; **(ii)** Si par contre le nœud  $DM$  ne connaît aucun nœud  $RM$ , il doit effectuer une découverte sur le réseau local (par exemple, grâce à un multicast) ou contacter un service d'annuaire qui peut indiquer l'identifiant d'un nœud  $RM_i$ . Comme la manière de trouver le nœud  $RM_i$  dépend de l'implémentation, elle n'est pas précisée dans notre architecture.

### Panne d'un nœud

Les nœuds peuvent subir deux types de pannes : des pannes volontaires ou des pannes involontaires. Les pannes sont détectées soit par des messages périodiques de type *Pull* (aussi connu comme *heartbeat*), soit à la demande, par des messages *Push* (*ping-pong*).

**Dans le cas d'une déconnexion involontaire**, les nœuds  $RM_i$   $\{RM_1, \dots, RM_n\}$  envoient des messages  $req_i(ping)$  respectivement  $\{req_1, \dots, req_n\}$  au nœud  $c$  et initialisent des temporisateurs d'attente  $\{ta_1, \dots, ta_n\}$ . **(i)** Si le nœud  $RM_i$  ne reçoit pas de message  $req_i(pong)$  du nœud  $c$  jusqu'à l'expiration du temps  $ta_i$ , alors le nœud  $RM_i$  interroge son voisin directe  $RM_j$  avec un message jeton initialisé à faux. **(ii)** Si  $RM_j$  reçoit le message  $req_j(pong)$  du nœud  $c$  avant l'expiration de son temps d'attente  $ta_j$ , alors  $RM_j$  modifie la valeur du jeton à **vrai** et retourne le message jeton à son émetteur  $RM_i$ . Ceci signifie (indirectement) que le nœud  $c$  n'est pas déconnecté. Alors le nœud  $RM_i$  peut envoyer à nouveau un message au nœud  $c$ . **(iii)** Sinon  $RM_j$  fait suivre le message jeton à son voisin directe avec la valeur **faux**. **(iv)** Si le jeton revient avec la valeur **faux** à  $RM_i$  alors tous les nœuds  $RM$   $\{RM_1, \dots, RM_n\}$  constatent le départ du nœud  $c$  et exécutent la procédure d'élection d'un nouveau nœud  $c$ .

**Dans le cas d'une déconnexion volontaire**, **(i)** un nœud  $DM$  qui décide de quitter le réseau il informe son nœud  $RM$ , qui va mettre à jour son indexe. Durant la déconnexion le nœud  $DM$  en question ignore les messages entrants. **(ii)** Si par contre il s'agit d'un nœud  $RM$  (ou  $c$ ) qui décide de quitter le réseau, il doit informer ses voisins  $DM$  (respectivement  $c$ ). Ces derniers exécutent un algorithme d'élection entre eux pour élire un nouveau  $RM$  (respectivement  $c$ ) qui prendra en charge les éventuels  $DM$  (ou  $RM$ ) orphelins.

---

## 5. Conclusion

Nous avons présenté les premiers éléments de la spécification de GrAPP&S, notre architecture de gestion de données. Cette architecture est flexible et modulaire, et adapte une approche orientée services permettant de gérer l'hétérogénéité des données. La validation de cette spécification nous permettra de définir les lignes qui guideront la suite de nos travaux. Nous souhaitons définir des stratégies pour le stockage pour le stockage des données, la réplication des données pour assurer leur disponibilité et aussi le routage des requêtes de recherche.



---

## 6. Bibliographie

- [1] Clarke, O. Sandberg, B. Wiley, and T. W. Hong. "Freenet: A distributed anonymous information storage and retrieval system". In *Workshop on Design Issues in Anonymity and Unobservability*, ICSI, Berkeley, CA, USA, July 2000, 311–320, *Lecture Notes in Computer Science* No 2009, 2001.
- [2] B. Cohen, "Incentives build robustness in Bittorrent". In *Workshop on Economics of Peer to Peer Systems*, Berkeley, USA, May, 2003.
- [3] L. Garcés-Erice, P. Felber, E. W. Biersack, G. Urvoy-Keller and K. W. Ross: "Data Indexing in Peer-to-Peer DHT Networks". ICDCS 2004, p. 200-208, 2004.
- [4] O. Heckmann, A. Bock, A. Mauthe, R. Steinmetz. "The eDonkey File-Sharing Network" in *Proceedings of Workshop on Algorithms and Protocols for Efficient Peer-to-Peer Applications*, Informatik 2004, September 2004.
- [5] M. Ji. "Hierarchical Bidirectional Chord". In *2010 International Conference on Educational and Information Technology (ICEIT 2010)*, p. 486-489. IEEE Xplore, 2010.
- [6] J. Liang, R. Kumar, and K.W. Ross, "The Kazaa Overlay: A Measurement Study", *Elsevier Computer Networks Journal (special issue on Overlays)*, 2005
- [7] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim. "A survey and comparison of peer-to-peer overlay network schemes". *IEEE Communication Survey and Tutorial*, march 2004.
- [8] S. Oaks, B. Traversat and L. Gong. "JXTA in a Nutshell", O'Reilly, 2002.
- [9] Z. Peng, Z. Duan, J.-J. Qi, Y. Cao, and E. Lv. "HP2P: A hybrid hierarchical P2P network". *ICDS '07*, pages 18–, Washington, DC, USA, 2007. IEEE Computer Society.
- [10] S Ratnasamy, P Francis, M Handley, R Karp, S Shenker, "A Scalable Content-Addressable Network", *ACM SIGCOMM Computer Communication Review* 31 (4), 161-172
- [11] Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, pages 329-350, November, 2001.
- [12] S. Saroiu, P. K. Gummadi, S. D. Gribble, "Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts". In *Multimedia Systems Journal*, Volume 9, Number 2, pp. 170-184 August 2003, Springer-Verlag.
- [13] Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. Frans Kaashoek, F. Dabek and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications". *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17-32, February 2003.
- [14] Yu and M. Li: "CBT: A proximity-aware peer clustering system in large-scale BitTorrent-like peer-to-peer networks". *Computer Communications* 31:(3). p. 591-602, 2008.

URED

## Organisation virtuelle

### Autorisation d'accès aux ressources partagées

Amadou Dahirou Gueye (1)

UFR Sciences Appliquées et Technologies  
de l'Information et de la Communication  
(SATIC), Université Alioune Diop de  
Bambey, Sénégal  
dahirou.gueye@uadb.edu.sn

Alassane Diop (2)

Centre de Recherche Licef, Télé-  
Université du Québec (Téluq)  
diop.alassane@teluq.ca

Samuel Ouya (3)

Ecole Supérieure Polytechnique de Dakar,  
Université Cheikh Anta Diop de Dakar  
(UCAD)  
samuel.ouya@gmail.com

**RÉSUMÉ.** Ce travail s'effectue dans le cadre des recherches sur les organisations virtuelles, particulièrement dans ses composants d'authentification, d'autorisations et de collaboration lors des activités de formation et de laboratoire. Dans ce papier, nous considérerons une organisation virtuelle formée d'une part, d'entités de plateformes de collaboration pédagogiques avec des ressources électroniques; d'autre part, d'entités de laboratoires à distance avec des dispositifs fixes ou mobiles. Notre article propose une architecture logicielle multithreadée (a-thread) conçu pour assurer la gestion des autorisations et des accès simultanés. Suite à la définition de cette architecture, un algorithme d'exécution et de gestion des threads est proposé. Les résultats obtenus sont aussi présentés.

**ABSTRACT.** This work is in research on virtual organizations, particularly in components for authentication, authorization and collaboration in learning activities and laboratory. In this paper, we consider a virtual organization formed on the one hand, entities online platforms with electronic resources; on the other hand, entities of remote laboratories with fixed or mobile devices. Our paper proposes multithreaded software architecture (a-thread) designed to manage authorisation and concurrency. Following the definition of this architecture, algorithm execution and thread management is proposed. The results are also presented.

**MOTS-CLÉS:** organisation virtuelle, apprentissage, dispositif, laboratoire distant, autorisation.

**KEYWORDS:** virtual organization, learning, device, remote laboratory, authorisation.

Actes du 4<sup>e</sup> Colloque national sur la recherche en informatique et ses applications Thiès-Bambey,  
Sénégal, Avril 2012

## 1. Introduction

Les besoins de collaboration pédagogiques et de développement des capacités en recherche, au moyen des technologies de l'information et d'internet, amènent les universités et laboratoires de recherche à établir une nouvelle forme d'organisations que l'on nomme organisation virtuelle (OV). Cette dernière prend forme quand des organisations décident de mettre en place un réseau d'échanges collaboratifs dans l'objectif de répondre à de nouvelles opportunités en exploitant leurs compétences et ressources respectives. Le type d'organisation virtuelle abordé dans ce papier met en jeu, à travers des environnements distribués informatisés, des ressources de plateformes collaboratives et des dispositifs réels de laboratoires à distance disposant d'interfaces logicielles. Dans ce papier, nous proposons un modèle d'infrastructure d'autorisation centralisé avec la création de tâches automatisées ou « automated-threads » et un algorithme d'exécution et de gestion des threads.

## 2. Organisation virtuelle : Métadonnées existantes

Une organisation virtuelle (OV) est un ensemble d'organisations partenaires reliées par les technologies de l'information et de la communication, coopérant à des projets communs tout en harmonisant leurs compétences. Dans l'état de l'art, nous trouvons plusieurs définitions de l'organisation virtuelle qui s'appuient sur des évaluations différentes de la notion de virtualisation. Manjouk A. et Kathleen M. Carley définissent l'organisation virtuelle comme « une organisation géographiquement distribuée dont les membres sont liés par un objectif ou un intérêt commun à long terme, qui communiquent et coordonnent leurs travaux à travers les technologies de l'information » [1]. D'un autre côté, Camarhina-Mathos et Lima considèrent l'organisation virtuelle comme « une alliance temporaire d'organisations (entreprises, institutions, industries, etc.) qui se réunissent pour partager leurs compétences et leurs ressources afin de mieux répondre aux opportunités métiers qui se présentent. La coopération et la coordination sont soutenues par des réseaux informatiques » [2]. Par ailleurs, Sowa et Sniezynski considèrent que « la construction d'une organisation virtuelle nécessite l'interconnexion des domaines de sécurité des organisations membres et la collaboration des autorités administratives gérant ces domaines et assurant les services de sécurité de base (authentification et autorisation des entités communicantes) dans ces domaines » [3]. Kamel Michel, dans sa thèse, considère que du point de vue organisationnel, une organisation virtuelle est une structure réseau dans laquelle des organisations peuvent jouer le rôle d'un fournisseur d'identité et/ou d'un fournisseur de services [4]. Par ailleurs, notre revue des principales recherches publiées sur l'organisation virtuelle nous permet de constater différentes formes d'OV parmi lesquelles on peut citer : les « organisations étendues » [5], les « grilles de calcul » [6], les « campus numériques ».

URED

### 3. Problématique

Les OV sont des environnements ouverts où les contraintes de sécurité pèsent énormément sur les organisations partenaires. Parmi les contraintes, notre domaine d'intérêt porte sur les processus d'authentification, d'autorisation et de collaboration lors des activités de formation et de laboratoire. L'article aborde d'abord un problème central des organisations virtuelles dans les structures de formation et de laboratoires à distance, puis traite la problématique de gestion aux données tant du point de vue des autorisations et des droits d'accès que des moyens de gérer les accès concurrents.

### 4. Scénarios collaboratifs d'organisation virtuelle

Le scénario collaboratif met en jeu deux types d'organisation virtuelle : l'une portant sur les laboratoires à distance et l'autre portant sur les applications de plateformes pédagogiques en ligne comme l'illustre la figure 2.

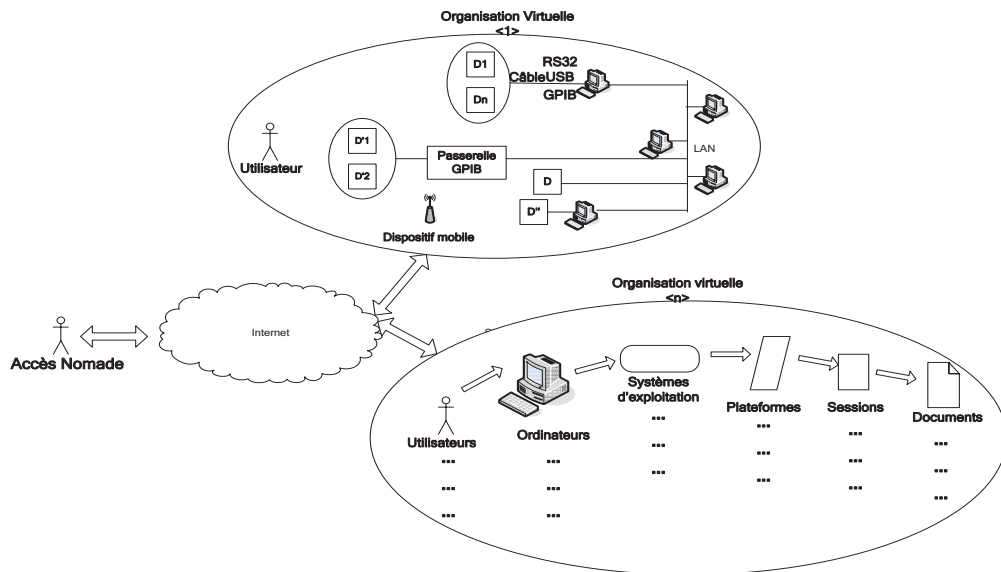


Figure 2 : Scénarios d'organisation virtuelle

Un laboratoire à distance permet de réaliser par les réseaux informatiques et en collaboration, des expérimentations en interagissant avec des dispositifs réels qui sont soit des instruments de mesures et/ou des mécanismes réels distants [7].

Les dispositifs  $\langle D_1, \dots, D_n \rangle$  peuvent être connectés chacun à un ordinateur via un câble avec un protocole de communication qui dépend de la nature de l'équipement. Ces derniers sont raccordés par l'intermédiaire de leurs ports autorisés (RS232, GPIB, USB, etc.). Ces mêmes dispositifs peuvent être dans un réseau public ou privé.

Des dispositifs  $\langle D'_1, \dots, D'_n \rangle$  pouvant être des instruments de mesures comme un oscilloscope, un générateur de fonction, un multimètre, un analyseur de spectre, reliés entre eux par une passerelle GPIB connectée au réseau local donc disposant d'une adresse IP. Un utilisateur peut utiliser les instruments de mesures à distance par l'intermédiaire de la passerelle GPIB via un serveur d'accès.

Les dispositifs  $\langle D \rangle$  sont directement connectés au réseau local et accessible via une interface graphique. Un utilisateur du réseau local peut accéder directement au dispositif en utilisant l'adresse IP ou le nom de ce dernier.

Les dispositifs  $\langle D' \rangle$  sont accessibles par le web et reliés à un ordinateur du réseau. Ces dispositifs peuvent être une plateforme d'apprentissage à distance ou un système offrant des services comme la téléphonie. Les dispositifs mobiles accessibles à distance peuvent être une caméra ou un robot mobile.

L'organisation virtuelle  $\langle n \rangle$  regroupe des plateformes pédagogiques. Nous considérons que ces plateformes sont hébergées dans des machines (ordinateurs, serveurs) avec un système d'exploitation. Chaque plateforme possède des sessions pour les utilisateurs et, à partir d'une session, on accède aux documents cibles.

## 5. Architecture de gestion des autorisations

Un certain nombre de questions se pose dès lors qu'on veut mettre en place un mécanisme d'autorisation : il ne suffit pas seulement de répondre à la question « qui peut accéder à la ressource ? » mais en plus, il faut préciser « une fois le système ouvert, ce que l'utilisateur est autorisé à faire ? ». Les problématiques de contrôle d'accès distribué dans les organisations virtuelles sont devenues plus complexes, où les données sont externalisées dans les organisations partenaires. (Agrawal D. et Al., 2009) [8] décrivent l'externalisation de données comme étant une préservation d'une vie privée évolutive. Ils expliquent qu'au lieu de crypter les données, qui sont coûteuses en calcul, qu'il vaut mieux utiliser la distribution sur des bases de données multiples hébergées dans des sites différents. Le modèle proposé par (Agrawal D. et Al., 2009), contribue à la création et à l'élaboration d'un cadre efficace pour la tolérance aux pannes modulable et la confidentialité des données théoriquement sécurisées préservant l'externalisation et la diversité des opérations de base de données exécutées sur différents types de données.

U R E D

Ces derniers peuvent s'appuyer sur des données publiquement disponibles. Le modèle que nous proposons utilise une approche similaire mais basée sur des systèmes hétérogènes déjà existants (Machines, Instruments, Systèmes de sécurité, Système d'exploitation, etc.). Vu que nous cherchons à mettre en place une solution de gestion des autorisations et des accès simultanés pour les organisations virtuelles, nous nous sommes intéressés à une architecture logicielle multithreadée (a-thread) permettant aux organisations de partager leurs ressources tout en gardant la main ferme sur leurs actifs. Cette architecture utilise deux composants principaux : une infrastructure d'autorisation (IA) centralisée et un gestionnaire de ressources (GR) implanté dans chaque site partenaire. Chaque composant comporte un thread mère et des threads clients comme l'illustre la figure 3.

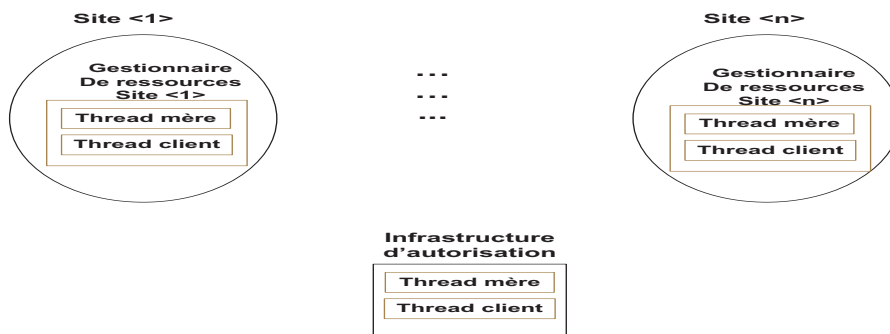


Figure 3 : architecture logicielle

Le GR sera géré par l'administrateur du site local et aura une base de données qui va contenir l'ensemble des ressources disponibles. L'IA dispose d'une base des politiques qui va modéliser les accès. Pour cela, quand un administrateur local veut partager une ressource, il va l'enregistrer dans l'IA qui contient l'ensemble des ressources partagées ainsi que les sites. Les interactions entre l'utilisateur, l'infrastructure d'autorisation (IA) et le gestionnaire de ressources (GR) suivent les étapes suivantes : (1) l'utilisateur qui désire accéder à une ressource, s'authentifie d'abord auprès de son site d'origine. (2) si l'utilisateur est reconnu, son GR envoie une requête d'accès, contenant l'id de la ressource demandée et l'id du site cible, au GR du destinataire. (3) Ce dernier envoie une requête à l'IA qui prendra la décision d'autorisation après consultation de sa base de politiques. (4) IA informe le gestionnaire de la prise de décision. (5) le GR vérifie la disponibilité de la ressource. (6) puis envoie un message de disponibilité ou non au GR de l'utilisateur qui, enfin, (7) transmet le message final à l'utilisateur ou l'application. Le message va aussi contenir le mode d'accès de la ressource (IP ou non)

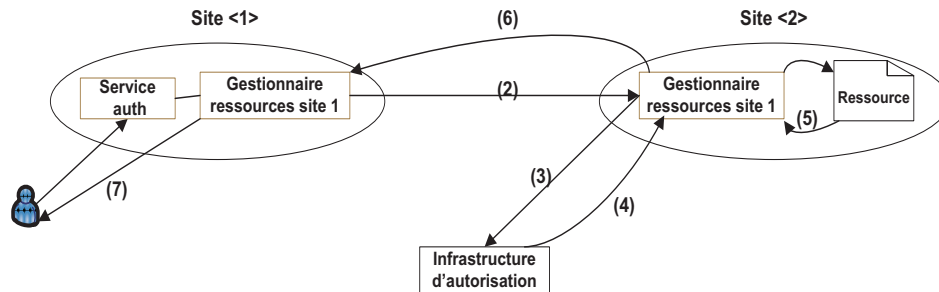


Figure 4 : chaîne d'autorisation d'accès aux ressources

Ce modèle proposé offre une autorisation de niveau de granularité plus fin dans la mesure où les seules communications possibles sont : (application, gestionnaire de ressources), (entre gestionnaire de ressources) et (gestionnaire de ressources, infrastructure d'autorisation).

## 6. Solution logicielle multithreadée

Pour communiquer avec une application ou un utilisateur, le gestionnaire de ressources va initialiser la connexion à l'aide d'un processus serveur appelé thread mère. Ce dernier va écouter sur un port pour récupérer l'ensemble des requêtes qui lui seront envoyées. Une application ou un utilisateur qui veut communiquer avec le thread mère va envoyer une requête à partir de ce port. Le thread, une fois récupéré la requête va créer un autre processus appelé thread client. Ce dernier va traiter toutes les requêtes provenant de l'application; ce qui va permettre au thread mère d'être libre et de pouvoir écouter une autre application. En somme, le thread mère initialise la connexion avec l'application et crée un thread client qui va continuer la communication avec l'application. De ce fait, une application qui demande l'accès à une ressource sera vu attribuée un thread client qui va gérer ses requêtes. Les threads mère et client auront le même comportement au niveau de l'infrastructure d'autorisation.

## 7. Réserveation d'accès par système de verrouillage de la ressource

URED

Pour gérer l'accès aux ressources partagées, nous proposons une synchronisation des threads clients qui activent ou désactivent l'état d'une ressource à travers une variable d'accès nommée « accès ». A chaque fois que des threads s'exécutent en même temps, il faut prendre des mesures adéquates pour qu'ils n'accèdent pas simultanément à une même variable. En fait, la synchronisation va permettre de gérer les accès concurrents concernant la manipulation simultanée de ressources partagées. Elle utilise la notion de verrou c'est-à-dire à un instant donné, une seule méthode synchronisée peut accéder à une ressource donnée. Pour chaque ressource de l'espace de la collaboration, on va créer un objet qui va la référencer. L'objet sera défini par son état et son comportement. Parmi les champs de l'objet, nous notons un champ particulier nommé « accès », une variable booléenne qui, lorsqu'elle prend la valeur « true » signifie que la ressource est disponible et non disponible dans le cas contraire. Ce champ « accès » sera verrouillé par le thread client du gestionnaire de ressources si ce dernier trouve la ressource disponible c'est-à-dire à « true ». La gestion des accès simultanés suit les étapes suivantes (1) le thread client au niveau du gestionnaire de ressources consulte le champ « accès » de la ressource pour déterminer sa disponibilité ou non. (2) Si le champ « accès » prend la valeur « false » le thread est mise en attente sinon (3) le thread client informe d'abord l'utilisateur ou l'application de la disponibilité de la ressource puis met le champ à « false » pour éviter l'accès à la ressource à un autre thread client. (4) L'utilisateur accède à la ressource, et une fois terminée, il remet à « true » la variable « accès » pour permettre l'accès à un autre thread client.

## 8. Algorithmes des threads

Pour le thread serveur ou mère, on crée d'abord une classe Thread\_Mere. L'algorithme de ce thread suit les étapes suivantes : (1) il écoute sur un port en attente de connexion. (2) Si une application lance une connexion vers ce port, il initialise la connexion et récupère les paramètres de connexion (socket). (3) Il crée un thread client pour l'application en lui fournissant comme paramètre l'objet socket. (4) Il démarre le thread client, puis (5) continue l'écoute pour une nouvelle connexion.

Pour le thread client on crée aussi une classe Thread\_Client qui étend la classe Thread. L'algorithme du thread client obéit aux étapes suivantes : (1) le constructeur prend en paramètre un socket qui est le socket responsable des paramètres de la connexion. (2) Le thread client écoute l'application pour récupérer les requêtes. (3) Après récupération d'une requête, on fait appel à la méthode « executer » de la classe Traitement pour traiter la requête. Cette classe permet d'interpréter toute nature de données et de les traiter. (4) le thread client continue l'écoute en attente d'autres requêtes provenant d'une application.



## 9. Conclusion

Ce papier propose une architecture logicielle, ainsi que les méthodes et moyens de la chaîne d'autorisation d'accès aux ressources partagées. Cette dernière présentée dans la figure 4 adopte le modèle *Pull*, une approche existante de système d'autorisation à base politiques, mais n'autorise pas l'accès direct d'un utilisateur ou d'une application vers un gestionnaire de ressources d'une organisation partenaire. Cette approche apporte un niveau de sécurité plus fin où les seules communications possibles sont (entre gestionnaires de ressources), ou (gestionnaire de ressources, infrastructures d'autorisation). De plus, l'authentification de l'utilisateur ou de l'application s'effectue auprès de son organisation d'origine fournissant ainsi une solution à l'hétérogénéité des systèmes d'authentification.

## 10. REFERENCES

- [1]. Ahuja, Manju K. and Carley Kathleen, «Network Structure in Virtual Organizations», *Journal of Computer-Mediated Communication* 3 (4):0-0, June 2008.
- [2]. Camarhina-Matos L. and Pantoja-Lima C., « Cooperation Coordination in Virtual Enterprises », *Journal of Intelligent Manufacturing* (12), Apr 2001.
- [3]. Sowa G. Sniezynski T., « Configurable multi-level security architecture for CNOs » ECOLEAD project, Deliverable D63.1, COMARCH, April 2005.
- [4]. Kamel M., Patrons organisationnels et techniques pour la sécurisation des organisations virtuelles, Thèse Université de Toulouse III, 198p, 29 septembre 2008.
- [5]. Benzekri A. et Barriere F., « CASH D32V3.080403.IRIT.CASH » a CASH project document, 23 May 2003.
- [6]. Ian Foster, Carl Kesselman, « The grid : blueprint for a new computing infrastructure », ISBN 1558609334, 9781558609334, 2004
- [7]. Diop A., « Etude et réalisation d'un laboratoire réel distant de télématique et réseaux pour l'enseignement en ligne », Atelier de reprod. de la Section de physique, 2005. – V, 137 p. : ill. ; 30 cm. – Th. sc. Genève, 2005.
- [8]. Agrawal D., Abbadi E., Emekci F., Metwally A., « Database Management as a Service: Challenges and Opportunities », *IEEE International Conference on Data Engineering*, 2009.

U R E D

=====

## C N R I A' 2012

### Services pour les grilles pair-à-pair

Bassirou Gueye\*<sup>#</sup> — Olivier Flauzac<sup>#</sup> — Ibrahima Niang\*

\* LID, Département de Mathématiques et Informatique  
Université Cheikh Anta Diop de Dakar  
SENEGAL

[bassirou.gueye@ucad.edu.sn](mailto:bassirou.gueye@ucad.edu.sn) — [iniang@ucad.sn](mailto:iniang@ucad.sn)

<sup>#</sup> CReSTIC, UFR Sciences Exactes et Naturelles  
Université de Reims Champagne Ardenne  
FRANCE

[bassirou.gueye@ucad.edu.sn](mailto:bassirou.gueye@ucad.edu.sn) — [olivier.flauzac@univ-reims.fr](mailto:olivier.flauzac@univ-reims.fr)

=====

**RÉSUMÉ.** Les grilles basées sur des architectures pair-à-pair (P2P) ont été utilisées soit dans le cadre du stockage et du partage de données ou de la gestion de calculs. En ce qui concerne la mise en place de grilles de services, les solutions proposées sont basées sur des grilles hiérarchiques qui présentent un fort degré de centralisation. Cette centralisation induit une gestion unifiée des ressources mais aussi des difficultés de réagir face à des pannes et fautes. Dans ce papier, nous proposons une spécification de grille P2P de services auto-adaptatifs. En effet, le déploiement et l'invocation sont totalement délégués à la plateforme et se feront de manière transparente pour l'utilisateur. La spécification se voudra la plus générique possible, non liée à un réseau P2P particulier ou à un protocole de gestion de services défini à l'avance.

**ABSTRACT.** The grid-based peer-to-peer (P2P) architectures were used either for storage and data sharing or computing. So far, the proposed solutions with respect to grid services are based on hierarchical topologies, which present a high degree of centralization. The main issue of this centralization is the unified management of resources and the difficult to react rapidly against failure and faults that can affect grid users. In this paper, we propose a new specification that enables self-adaptive service. Therefore, the deployment, and invocation are completely delegated to the platform and are done a transparent manner with respect to the end user. The specification should be generic will as much as possible, and not related to a given P2P services or a particular management protocol services defined in advance.

**MOTS-CLÉS :** Réseaux pair-à-pair, Grilles de calcul, Services web.

**KEYWORDS:** Peer to peer networks, Grid computing, Web services.

=====

Actes du 4<sup>e</sup> Colloque National sur la Recherche en Informatique et ses Applications – Thies-Bambey, Sénégal, Avril 2012

=====

## 1. Introduction

Dès leur création, les grilles de calcul [4, 3, 5] ont été conçues pour stocker des données ou effectuer des calculs scientifiques. Une nouvelle génération de grilles est apparue par la suite ; les grilles de services [6, 7], permettant à l'instar de ce qui est offert sur le Web avec les services web [9], de mettre en place l'accès à des traitements, tout en exploitant le paradigme de la grille. Ces grilles de services ont permis de distribuer les traitements, pour obtenir une utilisation optimale des ressources. Un effort particulier de normalisation est actuellement fait, afin d'apporter toutes les ressources et les moyens nécessaires au développement d'applications dans ces grilles de services. Cependant, les grilles qui exploitent la notion de services sont basées sur des architectures hiérarchiques fortement centralisées [4, 11]. Cette centralisation implique une gestion unifiée des ressources, mais aussi des difficultés à réagir vis à vis des pannes et des fautes qui impactent la communauté.

Dans cet article, nous présentons une spécification originale de gestion des services dans un environnement de grilles de calcul basé sur une architecture pair-à-pair. Le déploiement, comme l'invocation sont totalement délégués à la plate-forme et se feront de manière transparente pour l'utilisateur.

La suite de ce document est organisée ainsi. Dans un premier temps, nous allons présenter l'originalité de notre solution, puis étudier les travaux connexes. Ensuite, nous présenterons le modèle que nous utilisons, ainsi que les différents aspects de notre spécification. Par la suite, nous proposerons quelques solutions d'implémentations. Nous terminerons cet article par une conclusion, ainsi que des perspectives futures.

---

## 2. Contribution

La spécification que nous proposons présente l'originalité de ne pas lier l'infrastructure pair-à-pair à la plate-forme d'exécution de services. En effet, contrairement à des solutions de type Pastry [1] nous proposons de séparer la couche de gestion de la grille pair-à-pair, de la couche de localisation et d'exécution de services.

Cette spécification tend à être applicable sur toute architecture P2P. Comme nous l'avons précisé, nous ne nous focalisons pas sur la gestion de l'architecture de la plate-forme pair-à-pair, mais sur l'aspect implantation, déploiement et exécution des requêtes. Nous faisons, dans le cadre de la modélisation, le choix de ne pas nous aligner sur une architecture spécifique d'exécution, mais de spécifier les opérations de manière détachée. Il sera donc tout à fait possible d'associer cette spécification en exploitant les différents environnements de contrôle, d'échange et d'exécution : .Net, J2EE, Web Services SOAP, Corba, ...

**U R E D**

Toute combinaison de systèmes pair-à-pair respectant les contraintes de notre modèle pourra ainsi être composée avec toute plate-forme d'exécution de services.

---

### 3. Travaux connexes

L'exécution de code distant a été définie de différentes façons. L'appel distant de code a donné lieu au schéma général des RPC (Remote Procedure Call) qui décrit les échanges entre un client et un serveur. Différentes implémentations qui mettent en œuvre différents langages, protocoles ou systèmes ont été proposées. Parmi celles-ci, on citer les implémentations basées sur des bibliothèques systèmes (ONC RPC), celles basées sur des objets à distance (Java RMI, Corba [2]), celles basées sur l'utilisation de composants (J2EE, .Net, Mono) et sur les services web [12] (Xml-RPC, SOAP).

Le concept de RPC a été transféré dans les grilles, soit à partir des bibliothèques permettant la mise en place de solutions techniques, comme Globus [3] ; soit à partir des solutions directement conçues pour assurer le GRID RPC [10, 11]. Les solutions proposées sont hiérarchiques, permettent des exécutions distantes mais nécessitent des points de centralisation de la connaissance. C'est le cas par exemple de Ninf-G [10] et de DIET [11] dans lesquelles les capacités d'exécution s'enregistrent auprès d'un point de centralisation. Comme dans le cas de l'exécution de services sur le Web, des solutions d'implantation de services dans les grilles ont été proposées sur de telles architectures [6, 7, 8]. L'ensemble de ces propositions constitue des solutions hiérarchiques dont les architectures présentent un degré de dynamisme réduit, et dont les services d'infrastructure, comme par exemple la localisation de services, sont des services eux même centralisés.

Dans le cadre d'architecture P2P, on note une gestion décentralisée de la topologie. C'est le cas par exemple de Pastry [1] qui propose une organisation en anneau. Par conséquent, les spécifications de la localisation des ressources ainsi que de la gestion de la topologie sont liées : une seule spécification est donnée pour ces deux charges.

---

### 4. Modélisation

Afin de définir notre spécification, nous décrivons notre modèle d'application basé sur 3 composants : le réseau, les nœuds qui composent le réseau et la notion de service.

#### 4.1. Le réseau

Le réseau que nous considérons est le réseau overlay. Il s'agit en fait, non pas du réseau physique d'interconnexion, mais du réseau logique qui est mis à notre

**Gueye et al.**

disposition par la plate-forme pair-à-pair. Afin d'être exploitable dans notre spécification, on modélise le réseau sous forme d'un graphe  $G = (V; E)$  où  $V$  est l'ensemble de nœuds et  $E$  l'ensemble des liens de communications. Pour chacun des nœuds on définit la notion de voisinage ainsi : deux nœuds  $i$  et  $j$  sont voisins si :

- 1)  $i$  peut communiquer avec  $j$  et  $j$  peut communiquer avec  $i$
- 2)  $i$  connaît  $j$  dans sa liste de voisins et  $j$  connaît  $i$  dans sa liste de voisins

Ces définitions permettent la prise en compte des différentes spécificités du réseau : éléments de sécurité comme des firewalls qui limitent la possibilité de communication ; éléments de configuration et d'implémentation du réseau comme du NAT.

Le réseau offre au minimum les primitives de communication suivantes :

- 1) émission d'un message
  - envoi d'un message à un voisin *send (Neigh, message)*
  - envoi d'un message à un nœud de la communauté *route (id, message)*
- 2) réception d'un message
  - réception bloquante d'un message *receive ()*
  - réception non bloquante d'un message *async\_receive (callback)*, ou callback est une procédure à exécuter à la réception d'un message.

## 4.2. Les nœuds

Chacun des nœuds détient un identifiant unique dans le réseau pair-à-pair. Les nœuds sont en charge de la gestion locale du réseau, ils assurent collectivement les tâches décrites précédemment. Outre ces tâches, ils assurent le réceptacle des services, c'est à dire la plate-forme d'exécution. Les principales charges de cette plate-forme sont la gestion du déploiement, le cycle de vie des services, la gestion des requêtes et des exécutions. Les nœuds détiennent de plus une table des services, qui, comme nous le verrons par la suite, répertorie au minimum l'ensemble des services détenus localement, et au maximum l'ensemble de tous les services présents sur la grille.

## 4.3. Les services

Les services sont des objets intégrables aux plates-formes d'exécution détenues en chaque nœud. Un service est caractérisé par :

- la plate-forme d'exécution pour laquelle il a été connu ;
- les ressources nécessaires à son exécution (CPU, mémoire, données, ...)
- le format et contenu nécessaires lors de l'invocation du service et les contraintes des données résultat.

**U R E D**

---

## 5. Spécification des services en environnement pair-à-pair

Nous allons définir l'ensemble des primitives et opérations spécifiques à l'exécution de services sur notre grille pair-à-pair.

### 5.1. Déploiement de services : « deploy »

Le déploiement d'un service nécessite une première phase de détection des nœuds candidats de la plate-forme en mesure d'héberger et d'exécuter le service. Parmi ces nœuds candidats, il faudra donc en appliquant une stratégie particulière, désigner celui qui va héberger le service. Nous proposons trois stratégies de placement :

- 1) une stratégie aléatoire qui consistera à tirer au sort le nœud sur lequel sera déployé le service.
- 2) une stratégie équilibrée qui consistera à collecter les statistiques des nœuds et à tenter d'équilibrer la charge entre les candidats.
- 3) la stratégie "premier nœud" qui consistera à déployer le service sur le premier nœud détecté capable de supporter son exécution. Soulignons que cette stratégie limite la connaissance nécessaire, mais peut clairement mener à une surcharge.

### 5.2. Localisation de services : « lookup »

La localisation de services est la première étape d'une chaîne d'exécution de service. Chaque nœud de la grille P2P détient un registre de services. *lookup* est considéré comme un service local. En effet, il est disponible sur chacun des nœuds de la communauté ; mais ne peut pas être invoqué depuis une machine distante. A la réception d'une requête d'exécution de service par un nœud de la plate-forme, ce dernier fait appel à ce service de localisation qui exécute les opérations suivantes :

- si le service est présent sur le nœud, celui-ci est invoqué ;
- si le service n'est pas présent sur le nœud, si le nœud connaît la localisation du service, il route la requête vers le nœud qui l'héberge ;
- si le service n'est pas présent sur le nœud, si le nœud ne connaît pas la localisation du service, il fait suivre la requête vers ses voisins.

REMARQUE. 1. Le service requis peut être absent de la grille, soit suite à une panne, ou car il n'a pas encore été déployé. Dans ce cas, on pourra utiliser un algorithme de type *PIF* (*propagation of information and Feedback*), qui retournera au point d'invocation un code d'erreur qui sera transmis à l'émetteur de la requête.

REMARQUE. 2. Dans le cas d'une architecture basée sur un anneau orienté ou une chaîne, la procédure *forward* ne permet d'exploiter la communauté que de manière

**Gueye et al.**

sérielle. Par contre, dans le cadre d'un anneau non orienté ou d'un arbre, la procédure d'exploration initiée par *forward* se fera en parallèle.

### **5.3. Invocation de services : « invoke »**

L'invocation d'un service peut se faire sur n'importe quel nœud de la communauté. Cette invocation peut se faire soit à partir d'un nœud de la grille, soit par un nœud extérieur à la communauté. Dans ce second cas, tout nœud de la communauté peut être le «point d'invocation», qui sera en charge de faire l'invocation sur la communauté, de récupérer le résultat et de retourner ce dernier, ou une erreur, à la source de la requête.

### **5.4. Exécution de services et retour des résultats: « exec »**

Une fois que la requête a atteint le nœud qui détient le service, ce dernier l'exécute. La première étape va consister à identifier les paramètres de l'invocation avec ceux de l'appel. Cette exécution peut provoquer la génération d'un résultat ou d'une erreur. Une fois le résultat produit, ce dernier est routé vers le «point d'invocation».

### **5.5. Enregistrement de services : « save »**

Il est possible d'exploiter la plate-forme par l'intermédiaire de recherche sans mémoire des services à chaque invocation. Comme nous l'avons précisé précédemment, nous pouvons augmenter les performances de notre plate-forme en exploitant un registre de services. Nous envisageons la construction de ce registre de services selon deux principales stratégies :

1) la diffusion au déploiement : dans cette première stratégie, la diffusion de l'information sur la localisation du service se fera sous la forme d'une inondation auprès de toutes les nœuds de la plate-forme

2) la diffusion à l'appel : dans cette seconde stratégie l'information de localisation sera diffusée lors de la réponse après invocation. Tous les sites relayant la réponse vers le point d'invocation seront alors informé de la localisation du service.

Afin d'éviter la surcharge des mémoires de sites par le registre de service, il est possible de répartir ce registre sur plusieurs sites consécutifs. Ce qui aura l'intérêt de réduire la charge des sites, et de limiter la perte d'information dans le cas de disparition de site.

---

## **6. Propositions d'implémentation**

Pour implémenter notre solution, nous proposons deux approches décrites ci-après.

**U R E D**

## 6.1. Topologie P2P en anneau

Cette topologie adopte une architecture P2P distribuée qui essaie de répartir la totalité des fonctions du système entre les nœuds à savoir : la recherche, le routage et la récupération des services dans le réseau logique.

Nous utilisons une approche similaire à celle de Pastry [1] basée sur les tables de hachage distribuées (DHT). Ainsi, lorsqu'un message est routé vers un destinataire, le préfixe commun entre les nœuds intermédiaires et la destination augmente à chaque saut. Chaque nœud maintient une table de routage contenant  $(2^b - 1) * \log_2^b N$  voisins où  $2^b - 1$  représente le nombre d'entrées de chaque ligne.

Lorsqu'un nœud cherche un service dans la grille P2P, il regarde d'abord si ce dernier est détenu par un de ses voisins. Si oui, le service est exécuté par ce nœud voisin. Sinon, il forward la requête de recherche/localisation à son voisin qui a le plus long préfixe dans sa table. Ce nœud voisin exécute le même algorithme, et ainsi de suite. De ce fait, le service est absent si la requête revient au nœud source. Par contre, si un nœud localise le service, il l'exécute et renvoie le résultat qui prendra le chemin inverse de celui de localisation. Dans ce chemin, tous les nœuds intermédiaires ainsi que le nœud source mettrons à jour leur registre de services ( $ID_{Service}, ID_{Node}$ ).

## 6.1. Topologie P2P en arbre

Cette topologie adopte une architecture P2P décentralisée qui repose sur des interconnexions de super nœuds sur le niveau haut de la hiérarchie.

Pour implémenter cette approche de topologie P2P en arbre, nous nous basons principalement sur le protocole Gnutella2 [12] qui propose d'organiser le réseau logique selon une architecture décentralisée sur deux niveaux. Les nœuds simples ou feuilles se rattachent à un super nœud et les super nœuds entre eux sont reliés dans un réseau P2P distribué. Avec cette approche, les services partagés par une feuille sont enregistrés sur le super nœud responsable de cette feuille. Un nœud peut être nommé super nœud suivant plusieurs critères : cela peut dépendre de son adresse IP (publique ou privée), de son système d'exploitation, de sa bande passante, de l'instant depuis lequel il est connecté au réseau, ou de ses ressources matérielles (CPU, capacité mémoire, ...).

Lorsqu'un nœud recherche un service, il envoie la requête de localisation à son super nœud. Celui-ci exécute la fonction *lookup* (recherche locale). Si le service est localisé, alors il l'exécute et renvoie le résultat. Sinon, la requête de localisation est diffusée aux autres supers nœuds. Ainsi, le service est absent si aucune réponse n'est retournée après un temps TTL. Sinon, le super nœud détenant le service recherché l'exécute et renvoie le résultat au super nœud source. Ce dernier renvoie le résultat à son nœud feuille et met à jour son registre de service ( $ID_{Service}, ID_{Node}$ ).



## 7. Conclusion

Au constat que les grilles exploitant la notion de services sont basées sur des architectures hiérarchiques fortement centralisées, nous proposons une solution originale de spécification en couche qui tente de séparer la gestion de l'architecture et du réseau P2P de la couche d'exécution. La spécification est non liée à un réseau pair-à-pair particulier ou à un protocole de gestion de services défini à l'avance.

Comme perspective immédiate, nous mettrons en œuvre nos deux solutions d'implémentation. Ce travail sera suivi par une étude de solution de tolérance aux fautes dont le redéploiement dynamique de services en cas de détection de requêtes non résolues. Parallèlement, le développement des démonstrateurs et une comparaison des stratégies de déploiement et de localisation des services sont envisagés.

---

## 8. Bibliographie

- [1] A. Rowstron, P. Druschel, « Pastry : Scalable, distributed object location and routing for large-scale peer-to-peer systems », In Proc. IFIP/ACM Middleware, Germany, 2001.
- [2] S. Vinoski. CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments. IEEE Communications Magazine, pp 46-55. Feb. 1997.
- [3] I. Foster « Globus Toolkit version 4 : software for service oriented stsems », JSCT, vol. 21, no 4, Juillet 2006.
- [4] I. Foster, C. Kesselman, S. Tuecke « The anatomy of the grid : enabling scalable virtual organizations », IJSA'01, vol. 3, 2001.
- [5] O. Flauzac, M. Krajecki, L.A. Steffene « CONFIT a middleware for peer-to-peer computing », Journal of supercomputing, vol. 53, no 1, Juillet 2010.
- [6] S. Tuecke, K. Czaikowski, I. Foster, J. Frey, S. Graham, C. Kesselman, D. Snelling, P. Vanderbilt « Open grid services infrastructure »,
- [7] S. Tuecke, K. Czaikowski, I. Foster, J. Frey, G. Carl « Grid service specification », 2002.
- [8] D. Talia « The open grid services architecture : Where the grid meets the web », IEEE Internet Computing, vol. 6, no 6, 2002.
- [9] W.A. Naguy, F. Curbera, S. Weerawarana « Web services : Why and how? », ACM OOPSLA, Workshop on Object-Oriented Web Services, 2001.
- [10] Y. Tanaka, H. Nakada, S. Sekiguchi, T. Suzumura, S. Matsuoka « Ninf-G : A Reference Implementation of RPC-based Programming Middleware for Grid Computing », Journal of Grid Computing, 2003.
- [11] E. Caron, F. Desprez « DIET : A Scalable Toolbox to Build Network Enabled Servers on the Grid », International Journal of High Performance Computing Applications, vol. 20, no 3, 2006.
- [12] Gnutella 0.6 « The Gnutella Protocol Development. [En ligne] [http ://rfcgnutella.sourceforge.net/developer/index.html](http://rfcgnutella.sourceforge.net/developer/index.html) », July 2003.

U R E D

.....

# Passage d'un modèle acteur à un modèle multi-agent pour la gestion des ressources naturelles

## Utilisation du méta-modèle d'acteur Actor-in-Context

Eric Désiré Kameni\*, Eric Fotsing\*\*, Wouter De Groot\*\*\*

\* LIRIMA, Equipe IDASCO, Faculté des Sciences, Département d'Informatique BP 812 Yaoundé, Cameroun

Université de Yaoundé I, Ecole Normale Supérieure, Département d'Informatique, Cameroun, [ericokameni@gmail.com](mailto:ericokameni@gmail.com)

\*\* GeoInformatic and Environmental Modeling Laboratory, Department of Computer Science, IUT FV at Bandjoun & Centre for Environment and Development Studies in Cameroon at Maroua, University of Dschang, Cameroun, [efotsing@gmail.com](mailto:efotsing@gmail.com)

\*\*\* CML, Leiden University, Institute for Science, Innovation & Society, Radboud University Nijmegen P.O. Box 9010, 6500 GL Nijmegen, The Netherlands. [deGroot@cml.leidenuniv.nl](mailto:deGroot@cml.leidenuniv.nl)

.....

**RÉSUMÉ.** : Ce travail propose une méthodologie renforçant les capacités de collecte, d'analyse, et d'exploitation des informations liées aux causes des problèmes environnementaux en collaboration avec les populations concernées, afin de les intégrer dans un modèle multi-agent en utilisant un méta-modèle d'acteur Actor-in-Context (AiC). Cette méthodologie nous permettra d'expliquer l'impact de l'activité humaine sur le processus de déboisement de la réserve forestière de LAF située dans la partie Nord du Cameroun et ses retombés sur l'environnement afin de proposer des mesures de gestion plus rationnelle.

**ABSTRACT.** This paper introduces a methodology to improve the capacity to collect, analyse and exploit information linked to the causes of environmental problems in a multiagent model, using the Actor in Context meta model. The methodology is used to explain the impact of human activity on the deforestation process of Laf forestry reserve located in the Northern part of Cameroon.

**MOTS-CLÉS** : Environnement, Ressources naturelles, Modèle acteur, Modèle multi-agent,

**KEYWORDS:** Environment, Natural resources, Actor model, Agent-based model.

.....

Actes du 4<sup>e</sup> Colloque National sur la Recherche en Informatique et ses Applications – Thiès-Bambey, Sénégal, Avril 2012

.....

## 1. Introduction

Les problèmes posés dans les systèmes complexes exigent de nos jours des approches pluridisciplinaires combinant à la fois l'informatique, les mathématiques, les sciences sociales et la géographie. Les problèmes environnementaux tels que la gestion des ressources naturelles, la pollution de l'air, le changement climatique s'inscrivent dans cette catégorie. L'analyse des problèmes liés à l'environnement tels que la gestion des ressources naturelles présente des causes qui sont étroitement liées tant sur le plan physique que sur le plan social. L'exploration des causes physiques montre que celles-ci sont dues à des facteurs écologiques dont les paramètres peuvent être par exemple les sécheresses, les pluies abondantes, etc. Ces paramètres sont plus ou moins statiques et varient difficilement avec le temps. Quant aux causes sociales, celles-ci sont directement liées à la population. Elles résultent des actions conjuguées des différents utilisateurs sur le processus de gestion d'un ensemble de ressources telles que l'eau, la terre, la monnaie, la force de travail, la disponibilité de travail et la technique de travail. On note ainsi dans cette situation une forte dynamique des paramètres liés aux causes. La collecte, l'analyse et l'exploitation des informations dans ce cas afin de comprendre des phénomènes et de proposer une solution au problème environnemental ne sont pas toujours faciles. La modélisation se présente de nos jours comme un moyen approprié pour apporter des solutions à ces types de problème. La modélisation étant une représentation de la réalité que l'on élabore, nécessite une bonne compréhension et une analyse de cette réalité selon une méthodologie bien appropriée. Une extrapolation de cette méthodologie peut produire des modèles à conséquence néfaste, au niveau des cadres de concertation et de prise de décision (mauvaise prédiction, persistance du problème à résoudre). Suite à ce constat nous proposons une approche méthodologique afin de renforcer les capacités de collecte et d'analyse des informations environnementales en collaboration avec les populations concernées afin de les intégrer dans un modèle multi-agent en utilisant un méta-modèle d'acteur (AiC).

Le travail proposé se déroule en trois étapes. (1) La collecte des informations et l'identification du problème à l'aide de la méthodologie Problem- in – Context ([3]). Cette méthode permet de construire un aperçu des problèmes constatés avec des liens cause à effet. (2) La mise sur pied d'un modèle acteur en utilisant le méta-modèle d'acteur, Actor – in – Context (AiC) ([3]). L'AiC permet d'expliquer plus clairement les causes sociales des problèmes environnementaux à travers les actions des acteurs (3) L'intégration du méta-modèle AiC obtenu dans une architecture multi-agents adéquate.

## 2. Méthodologie PiC et AiC

### 2.1 Méthodologie PiC

L'étude de la gestion des problèmes environnementaux par les occupants du milieu et les acteurs extérieurs n'est généralement pas rationnelle car seules les préoccupations de l'exploitant comptent ; les impacts des actions de ce dernier étant dans la plupart des cas mal ou pas du tout perçus. La méthodologie Problem-in-Context (PiC) en connexion avec Actor-in-Context permet ainsi d'analyser les problèmes environnementaux en y intégrant en même temps les préoccupations des acteurs exploitants et l'impact de leurs actions sur l'environnement.

Le point d'entrée pour la méthodologie Problem-in-Context (figure 1) est focalisé sur l'activité ou l'action. Cette action conduit à une chaîne d'effets sur l'environnement. Ces effets sont déterminés à partir des études faites sur les impacts environnementaux afin de déterminer par exemple des variables finales des impacts. Le niveau de l'impact dépend en grande partie des normes (fixées) sur les variables finales, ceci grâce à la chaîne des normes environnementales. La capacité de charge de l'environnement dépend du problème à traiter ou de l'action à entreprendre. Cette première partie du cadre correspond à ce que l'on appelle l'analyse du problème environnemental.

### 2.2 Méthodologie AiC

Pour l'explication du problème environnemental, le bloc inférieur de la figure 1 est utilisé à cet effet. En d'autres termes, ce bloc permet de donner une explication sur des causes sociales du problème. La méthodologie Actor-in-Context (AiC) offre les facultés permettant d'étudier et d'expliquer les causes sociales des problèmes environnementaux en se basant sur les acteurs.

L'approche AiC (figure 2) utilisé comme le méta-modèle d'acteur source de notre approche intègre des études de cas tels que les études sur des facteurs (politique, économique), et connecte des acteurs locaux et internationaux à des contextes à grande échelle. Cette jonction permet de faciliter l'identification et la connexion des options des politiques locales, régionales et mondiale en ce qui concerne le problème environnemental à étudier. Cette méthodologie souligne la causalité empirique portant sur les décisions concrètes des acteurs et peut se subdiviser en cinq (05) étapes.

*L'étape 1* consiste à définir les actions pertinentes à l'étude. Une action est une unité de comportement sur laquelle un acteur a pris une décision. La méthodologie AiC prescrit l'identification claire des actions pertinentes dans un premier temps avant de se concentrer sur les acteurs où sur les données qui sont influencées par ces actions.

*L'étape 2* consiste à identifier les acteurs ayant une capacité de prise de décision importante. Les acteurs sont toutes entités sociales qui agissent individuellement ou collectivement. Une entité sociale ne peut être acteur à l'égard de certaines activités que

si elle exerce une importante capacité de prise de décision à l'égard de ces activités. Cette importance se définit en fonction de l'objectif de l'étude.

*L'étape 3* consiste à ressortir la liste des actions possibles de chaque acteur identifié précédemment, cette liste d'actions sera appelée les options de l'acteur. Les options se réfèrent essentiellement à la liste des actions possibles envisagées par l'acteur comme étant liées à un de ses objectifs. Ces options peuvent être connues ou inconnues à l'acteur, elles peuvent aussi être positivement ou négativement appréciées par l'acteur, ainsi qu'elles peuvent être disponibles ou indisponibles pour l'acteur. L'approche AiC distingue deux types d'options : les options possibles et les options réalisables.

*L'étape 4* consiste à identifier l'ensemble des facteurs dynamiques qui orientent l'action de l'acteur vers l'objectif qu'il s'est fixé. Ces ensembles de facteurs seront nommés motivations. La motivation rapporte ce qui est important par rapport à l'action de l'acteur et aussi pourquoi l'acteur accorde une importance sur une option dans la gamme de ses valeurs.

Enfin *l'étape 5* qui est une analyse globale du champ d'acteurs. Une analyse plus approfondie du champ d'acteurs permet de connecter l'action de l'acteur à son environnement ou à son contexte. Cette analyse va d'un acteur à l'autre en se concentrant sur l'identification, l'étude et l'interconnexion des facteurs culturels et structurels qui sous-tendent les choix des acteurs.

### **3. Passage du modèle acteur à l'architecture d'agent**

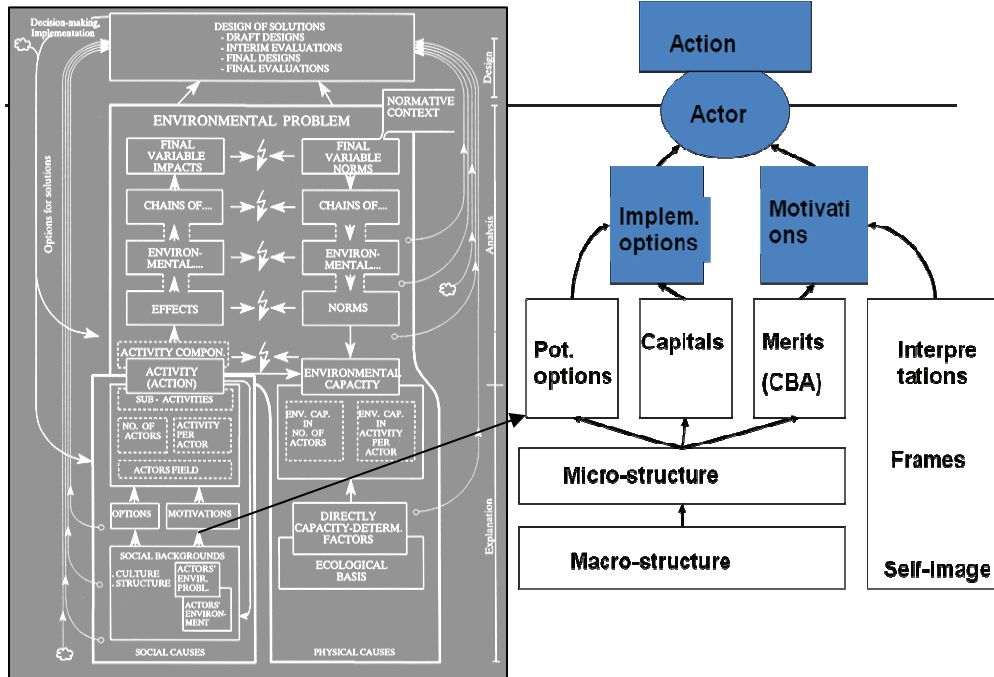
Selon [1] un agent est un système informatique, situé dans un environnement, qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

Il existe plusieurs architectures d'agents, mais peu importe l'architecture adoptée, un agent est toujours vu comme une fonction liant ses perceptions à ses actions. Plus précisément, un agent perçoit l'environnement à l'aide de ses capteurs et il agit sur son environnement à l'aide de ses effecteurs. Ce qui différencie les architectures d'agents, c'est la manière dont les perceptions sont liées aux actions.

Ainsi on distingue trois grandes architectures d'agent à savoir :

-les agents réactifs : qui ne réagissent qu'aux changements qui surviennent dans leur environnement. Ils ne font ni délibération ni planification, ils se contentent simplement d'acquiescer des perceptions et de réagir à celles-ci en appliquant certaines règles prédéfinies. Ils n'ont pas de mémoire de leur historique ni de représentation explicite de leur environnement.

- Les agents délibératifs/cognitifs : qui effectuent une délibération en se basant sur leur but pour choisir leurs actions. Ils se posent des questions du genre « Qu'est-ce qui va arriver si je fais telle ou telle action ? » et « Est-ce que je serai satisfait si cela se produit ? ».



**Figure 1.** The Problem-in-Context general framework: the analysis, explanation and solution of environmental problems (de Groot, 1992)

**Figure 2 -** Action-in-Context framework (méta-modèle d'acteur source)

- les agents hybrides : qui intègrent les deux caractéristiques précédentes.

Dans notre travail nous nous intéressons à l'architecture délibérative et plus précisément l'architecture d'agents basés sur l'utilité.

A la figure 3 nous avons une architecture d'agent basé sur l'utilité selon RUSSEL ([2]), et à la figure 4, l'architecture d'agent cible intégrant le méta-modèle d'acteur dans l'architecture d'agent basé sur l'utilité. Il est à noter que l'utilité est le paramètre qui permettra à un agent intelligent de faire le meilleur choix pour atteindre son but. La figure 4 a été obtenue en définissant et en faisant un rapprochement entre les concepts définis dans l'architecture de RUSSEL et ceux du méta-modèle d'acteur.

Pour rendre plus explicite le passage du modèle acteur au modèle multi-agent, nous allons utiliser la plateforme d'implémentation NetLogo. Le choix de cette plate forme a été fait par rapport à sa compréhension facile et son langage moins complexe. Elle permet entre autre de mieux expliquer aux utilisateurs les différentes étapes de la transformation. Ainsi partant du méta-modèle d'agent dans NetLogo (figure 5), nous pouvons définir les concepts et établir les correspondances de la figure 6

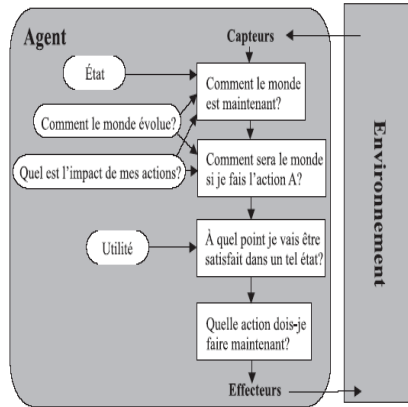


Figure 3 : Architecture d'agent basé sur l'utilité (RUSSEL [2])

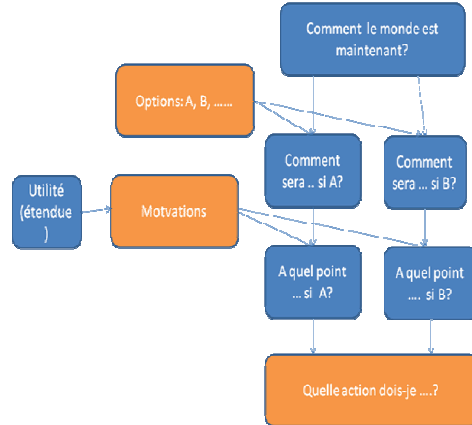


Figure 4 : Architecture d'agent cible intégrant le méta-modèle d'acteur source et l'architecture d'agent basé sur l'utilité

*Perception* : évènement cognitif pouvant survenir dans l'environnement et dont un agent ou un objet de l'environnement peu en bénéficier de son résultat.

*Croyance* : représente l'ensemble des informations que l'agent détient à propos de son environnement.

*Interface* : représente la liste des actions à effectuer par l'agent

*Environnement* : représente l'espace dans lequel l'agent évolue.

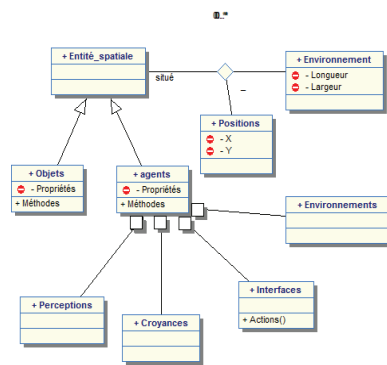


Figure 5. Méta-modèle d'agent dans NetLogo

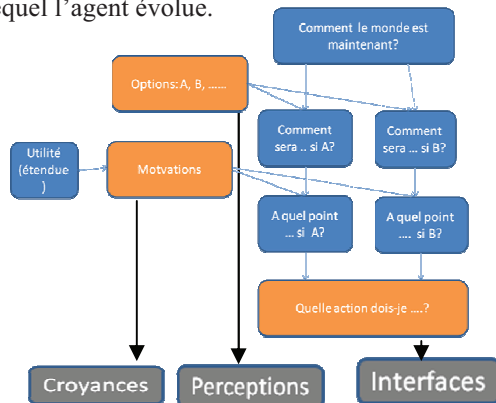


Figure 6. Construction du modèle d'agent dans NetLogo à partir d'architecture d'agent basé sur l'utilité

Il apparait clairement que dans la construction du modèle agent dans NetLogo, les perceptions et les croyances de l'agent sont construites respectivement par les motivations et les options de l'acteur (figure 6). Il en est de même pour l'interface qui est construite grâce aux actions réalisables de l'acteur. Les concepts environnements et

croissance restent inchangés dans la construction. Il est à noter qu'une fois le modèle agent obtenu dans NetLogo il est plus aisé de déduire le code.

#### 4. Application

Dans cette partie nous appliquons la méthodologie développée sur le processus de déboisement de la réserve forestière de Laf située dans la partie Nord du Cameroun.

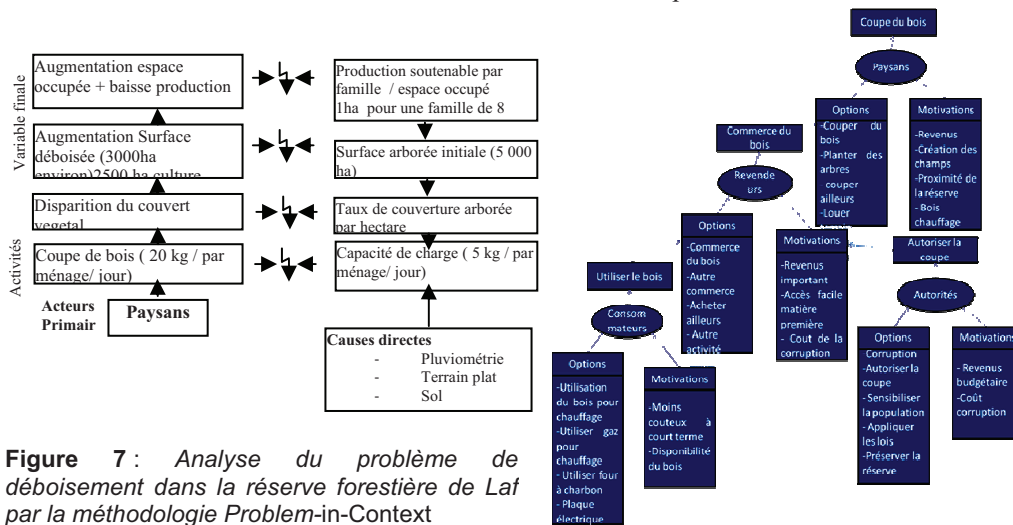


Figure 7: Analyse du problème de déboisement dans la réserve forestière de Laf par la méthodologie Problem-in-Context

Figure 8: Modèle acteur du processus de déboisement dans la réserve de Laf

Sur la figure 7 on observe la chaîne d'effets produits par l'activité principale qui est la coupe de bois et dont son impact conduira à la variable finale qui est une augmentation de l'espace occupé contre une baisse de production. Dans la partie supérieure droite nous avons une chaîne de normes calculées sur la réserve permettant de justifier l'impact de la chaîne d'effets de la partie gauche, ceci par comparaison.

La figure 8 propose une analyse des causes sociales du processus de déboisement tout en mettant les actions des acteurs au centre. Sur cette figure, l'action principale est la coupe de bois dans la réserve. Cette option choisie par le paysan donne naissance au commerce du bois qui est motivé par le coût de corruption (3000/mois) versé aux autorités et le revenu important. Ce commerce est également motivé par les autorisations des coupes non contrôlées établies par les autorités.

Il est à noter que pour chaque activité correspond une catégorie d'acteur qui l'exerce (paysans, revendeurs, consommateurs, autorités).



Comme présenté dans la section précédente la méthodologie appliquée au problème nous donne le modèle agent de NetLogo de la figure 8.

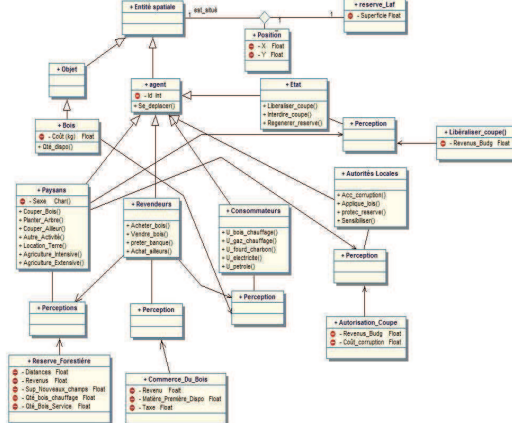


Figure 9: Diagramme de classe du processus de déboisement de la

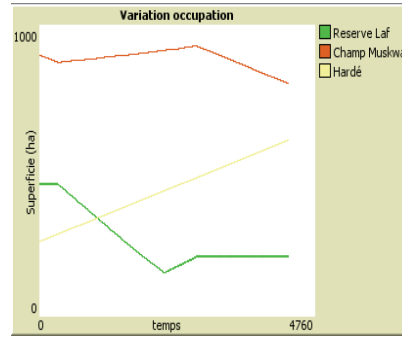


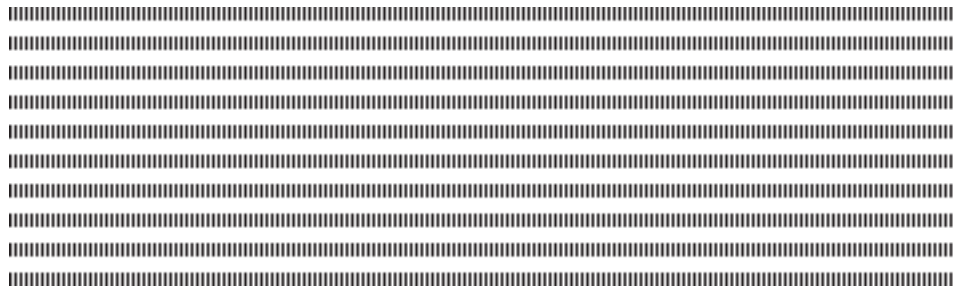
Figure 10 : Résultat simulation du processus de déboisement de la réserve de Laf pendant 50 ans dans

## 5. Conclusion et perspectives

Il ressort au terme de notre travail que le processus de déboisement (réserve en vert) est motivé par la création des nouveaux champs (en orange) qui remplacent progressivement l'espace boisé (figure 10). On note également une croissance de la dégradation des terres (en jaune) suite aux techniques pratiquées (mal adopté) dans les champs (figure 10). Cette méthodologie offre une faculté pour les chercheurs en science environnementale d'intégrer aisément l'approche multi-agent aux problèmes aux quels ils s'attaquent tout en mettant l'acteur au centre. Par ailleurs il sera intéressant d'adapter cette méthodologie sur les autres plates-formes de simulation multi-agent.

## 6. Bibliographie

- [1] Jennings N., Sycara K., Wooldridge M., « A Roadmap of Agent Research and Development », *Autonomous Agents and Multi-Agent Systems*, vol. 1, n°1, p. 7 - 38, July 1998.
- [2] Russell S. J., Norvig P., *Artificial Intelligence. A Modern Approach*, Prentice-Hall, Englewood Cliffs, 1995.
- [3] Wouter T. de Groot, 1992. *Concepts and methods in One-World Problem-Oriented Paradigm*. Thèse de Doctorat, Université de Leiden – Pays-bas, 415 pages.



Actes du 4e Colloque National sur la Recherche en Informatique et ses Applications, Thies-Bambey, Senegal, Avril 2012

## Three dimensional model for soil pore space

### Using geometrical primitives

Ndeye Fatou NGOM\* — Olivier MONGA\*\*

\* Unité de Modélisation Mathématique et Informatique des Systèmes Complexes  
UMMISCO-IRD Université Cheikh Anta Diop de Dakar  
1386 Dakar Fann  
SENEGAL  
ngomnf@gmail.com

\*\* UMMISCO-IRD  
Institut de Recherche pour le développement  
FRANCE  
olivier.monga@ird.fr



**RÉSUMÉ.** Nous nous intéressons à la modélisation de la structure des milieux poreux à l'aide de technologies d'imagerie non destructives. Nous proposons une représentation de la géométrie de l'espace poral basée sur un assemblage de primitives géométriques. Notre modèle calcule d'abord, à partir de données images, un recouvrement du squelette par des boules maximales. Ensuite, ce recouvrement est segmenté en chaînes de boules simplement connexes qui donnent une première description des différentes galeries de l'espace poral. Puis chaque chaîne de boules est approchée par des primitives géométriques simples ou des cylindres généralisés circulaires uniformes finis. Nous validons le modèle sur différents échantillons d'images tomographiques de sols réels. Des travaux futurs s'appuieront sur ce modèle pour simuler des dynamiques spatio-temporelles du sol et à terme rendre compte de propriétés émergentes du végétal et simuler des interactions entre la plante et son environnement.

**ABSTRACT.** We deal with the modeling of soil porous media micro-structures based on quantitative analysis of synchrotron X-ray micro tomography data. We have represented pore space using a compact, stable and robust piece wise approximation by means of balls and circular generalized cylinder. Our model calculate the minimal set of balls that recover the skeleton of the pore space using Delaunay tessellation, then the simply connected sets of balls that could be considered as potential soil pore channels, and after an approximation by a generalized cylinder of each simply chain is calculated. This technique was applied to data sets on real volume Computed Tomography soil sample. During future studies, this geometric pore space representation will be used to simulate soil biological dynamics and link it with root penetration and plant growing.

**MOTS-CLÉS :** Traitement d'images, morphologie mathématique, Images tomographiques, Modélisation géométrique, géométrie algorithmique, segmentation, espace poral, sol;

**KEYWORDS :** image processing, Soil micro-structures, Computed Tomography images, 3D geometric modeling, pore space, Computational geometry, Volume segmentation.



---

## 1. Introduction

Soil structure is a key factor in the ability of soil to support microbial activity and root penetration, and to modulate and control the distribution of the flow and retention of water, solutes, gases and biota in agricultural and natural ecosystems (Young et al, 2001). The geometric modeling of these micro-structures is made possible by the improved performance of computed tomography techniques which provide very high resolution 3D volume images of soil samples in which pore spaces can be extracted by thresholding. Spheres have been used to fill the pore space in order to calculate the skeleton, pore size and retention curve (Monga et al, 2007). This is beneficial from a computational point of view since it reduces the amount of data to be stored in the memory. Despite considerable advances in our understanding of soil systems in recent years, the interaction between soil structure and soil functioning remains poorly described by models. In this study, we propose an improved representation that enables more robust results. It is based on Delaunay triangulation used in (Monga et al, 2007 ; Monga et al 2008, Ngom et al, 2011) to extract the soil structure from computed tomography images of soil aggregates. Indeed, we calculated the minimal set of balls that recovered the skeleton of the pore space using Delaunay tessellation, then the simply connected sets of balls that could be considered as potential soil pore channels, and after an approximation by a generalized cylinder of each simply chain is calculated.

In the sequel, we present the soil samples in section 2, our approach to geometrical modeling with the algorithms that were developed in section 3 and a conclusion in section 5.

---

## 2. Extraction of soil sample

The computed tomography image datasets on soil aggregates are obtained using synchrotron-based X-ray-tomography, were analyzed during this study. The soil was classified as Alfisol, with the following texture : 10% sand, 70% silt and 20% clay. The aggregates have been sampled from soil blocks (collected at 15 – 30 cm depth) by manually breaking the blocks along planes of weakness. Soil aggregates (of 3.3 mm diameter for CTT and 4.6mm diameter for GL) were scanned at the Synchrotron radiation microtomography facility operated by the GKSS research centre at HASYLAB (Hamburger Synchrotron Strahlungslabor) belonging to the DESY (Deutsches Elektronen Synchrotron) in Hamburg, Germany. A  $400 \times 400 \times 400$  voxel size cube, with a resolution of  $5.403 \mu\text{m}$ , was extracted from the original microtomograms, from near the centre of the aggregate (Figure 1 and Figure 2). More details can be found in (Peth et al, 2008).

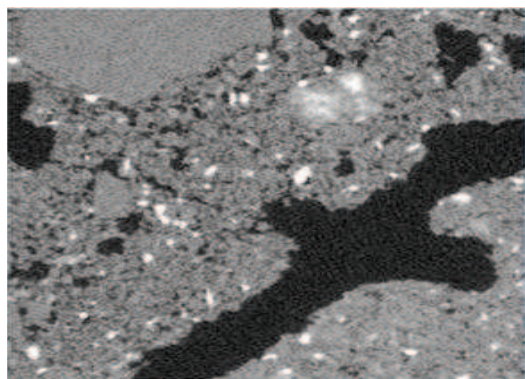


Figure 1 – A 2D CT image

---

### 3. The Model

#### 3.1. Computation of Delaunay Triangulation

We computed the pore space envelope from the 3D image using a simple thresholding algorithm to select the boundary points because in X-ray scanner images, gray levels are approximately proportional to the density of the material being studied. OpenDX and OpenGL were used as visualization softwares. We thus obtained a set of voxels (Figure 2) that defined the indicator function of the volume defining the pore space. We computed 3D Delaunay triangulation of the boundary points (Figure 3). A simplified version of the Tetmesh-GHS3D code was used to obtain (3D) Delaunay triangulation (George, 2004) and the free software MEDIT was employed to visualize Delaunay tetrahedra (Frey, 2001).

#### 3.2. Extraction of minimal balls recovering the skeleton

In the specific case where sampling of the shape border is optimal (Schmitt and Mattioli, 1994), which is the situation under consideration here, maximal balls correspond to Delaunay spheres in a 3D Delaunay triangulation of shape boundaries . The following

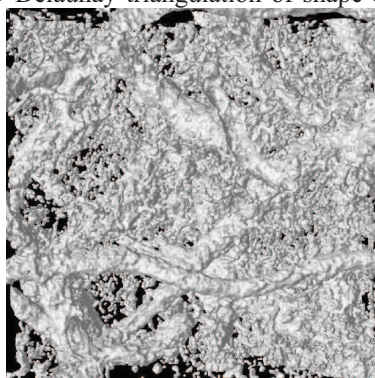


Figure 2 – Perspective view of the 3D voxel representation of the soil pore space

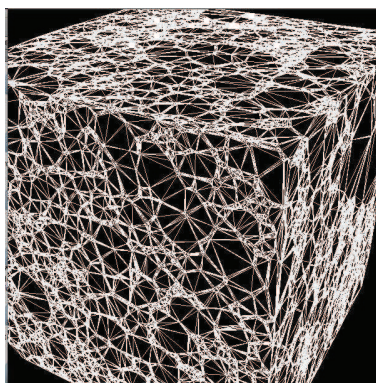


Figure 3 – Perspective view of the Delaunay Triangulation of the soil pore space algorithm can therefore be used to extract the maximal balls :

- 1) Computation of the volume shape border using the 26-connectivity.
- 2) Computation of 3D Delaunay Triangulation of the volume shape border using the method developed by the GAMMA project (Frey, 2001 ; George, 2004).
- 3) Computation of Delaunay spheres from Delaunay tetrahedral. The Delaunay sphere attached to a tetrahedron is the sphere that passes through its four vertices.
- 4) Computation of the skeleton which is the set of all centers of Delaunay spheres within the shape (Chazal, 2005).

With the algorithm described below, we have obtained a representation (Figure 4) with 170250 balls for the soil sample. The maximum ball radius was  $26.94\mu m$  and the minimum ball radius  $3.83\mu m$ . The total porosity was 6.69%. These correspond to soil pores whose diameters were larger than  $3.83\mu m$ .

### 3.3. Extraction of the optimal connected set of balls

The aim is to approximate the balls using more sophisticated geometrical primitives like circular generalized cylinder. Our first stage consists in linking these balls into optimal simply connected chains by optimizing a linking criterion, highlighted by Figure

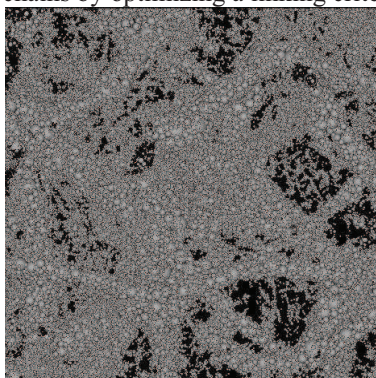


Figure 4 – Ball representation of the soil pore space

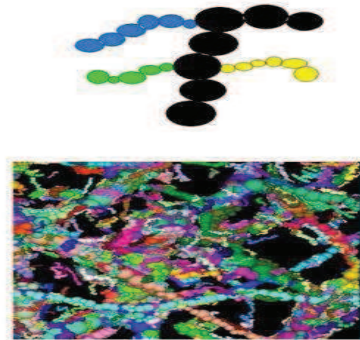


Figure 5 – Segmentation in balls chains  
5, to favors the formation of homogeneous chains in terms of ball radii. Let  $C$  be a simply connected set of balls. We will say that  $C$  is a simply connected ball chain if and only if :

- 1) Each ball of  $C$  is connected to no more than two balls of  $C$ ,
- 2) Exactly two balls of  $C$  are connected with only one other ball of  $C$ ,
- 3) For each ball of  $C$  that is connected to two different balls of  $C$ , if the ball is removed from  $C$  then  $C$  is split into connected component.

We have obtained 14151 balls chains from the initial set of balls which contained 170250 balls. Figure 5 presents the experimental results by displaying the simply connected chain in different colors.

---

#### 4. APPROXIMATION BY A GENERALIZED CYLINDER

We compute from each connected balls chain a uniform generalized cylinder (UGC). A UGC is a surface of revolution around a curve with a fixed radius. For each chain of balls, we compute a UGC as a circle moving along a path which is a Catmull-Rom which have the centers of the balls chain as control points. This method provides a reasonable

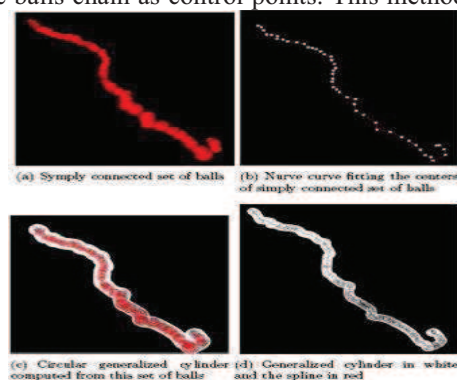


Figure 6 – From a connected ball chain to a generalized cylinder representation

Three dimensional model for soil pore space

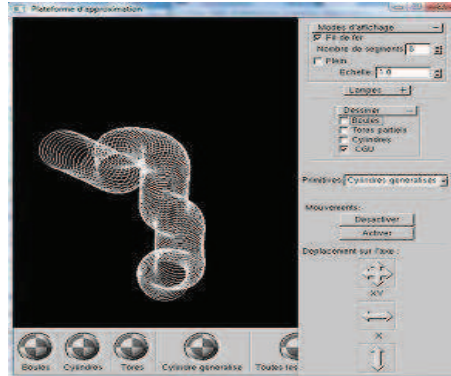


Figure 7 – General view of the Model degree of continuity and flexibility that can be generated and evaluated efficiently.

Figure 6 and Figure 7 illustrate the spline computed from a set of control points and the generalized cylinder derived from the spline. Given the positions values of  $P_{m-1}$ ,  $P_m$ ,  $P_{m+1}$ ,  $P_{m+2}$  at the control points, the Catmull-Rom spline extends from  $P_m$  to  $P_{m+1}$ , according to the following equation :

$$pos_m(t) = \begin{pmatrix} -0.5 & 1.5 & -1.5 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_{m-1} \\ P_m \\ P_{m+1} \\ P_{m+2} \end{pmatrix} (t^3 \quad t^2 \quad t \quad 1) \quad (4.1)$$

---

## 5. Conclusion

We have presented a way to model complex 3D geometrical micro-structures in soil using a piece wise approximation by means of generalized cylinders. Our initial data consist in high resolution 3D Computed Tomography images of soil samples obtained using an X ray scanner. The method is applied to the modeling of pore space extracted by thresholding the CT volume images. However, it can also be applied to other types of tomography images likes the one of medicine. In future work, we will show how a biological simulation can be performed in the pore space using this geometrical representation.

---

## Acknowledgements

The authors thank Dr Patricia Garnier (INRIA, Grignon, France), Dr Felix Beckmann (GKSS Research Centre at Hamburger HASYLAB, Germany) and Dr. S. Peth (Institute of PlantNutrition and Soil Science, Christian-Albrechts-University of Kiel, Germany) for the provision of the microtomography datasets of the soil aggregates.

---

## 6. Bibliographie

- Chazal, F., Lieuthier, A., 2005. *The lambda medial axis*. *Graph Models*, 67, 668 304-331, 2005.
- Delerue, J.F, Perrier, E., 2002. *Dxsoil a library for 3d image analysis in soil science*. *Computers and Geosciences*. 28, 1041-1050.
- Frey, P.J., 2001. *An interactive Mesh visualization software*. INRIA , *Technical Report-0253*, 41 pp, 2001.
- George, P.L., 2004. *Tetmesh-ghs3d -tetrahedral mesh generator*. INRIA , *Technical Report-0253, Report-78153, INRIA User's manual, Le Chesnay c ´edex : Institut national de Recherche en Informatique et Automatique*, 13 pp.
- Monga, O., Ngom, N.F., Delerue, J.F., 2007. *Representing geometric structures in 3D tomography soil images : application to pore space modeling*. *Computers and Geosciences*, 33(9), 2007.
- Monga, O., Bousso, M., Garnier, P., POT, V., 2008. *3d geometrical structures and biological activity : application to microbial soil organic matter decomposition in pore space*. *Ecological Modelling*, pages 291–302.
- Ngom, N.F, Garnier, P., Monga, O., Peth, S., 2011. *Extraction of three dimensional pore space from microtomography images using a geometrical approach*. *Geoderma*, doi : 10.1016, *J. geoderma*.
- Peth, S., Horn, R., Beckmann, F., Donath, T., Fischer, J., Smucker, A.J.M., 2008. *Three dimensional quantification of intra aggregate pore space features using synchrotron radiation based microtomography*. *Soil Science Society. Am.* 72, 897-907.
- Schmitt, M., Mattioli, J. 1994. *Morphologie mathématique*. Masson, Paris, 687 211pp.
- Young, N.M, Crawford, J.W., Rappold, C., 2001. *New methods and models for characterising structural heterogeneity of soil*. *Soil Tillage Ressource*, 61, 33-45.



.....

## Approche pour la gestion de la QoS des applications et services dans les Clouds

Mohamed Mahmoud OULD DEYE\* — Yahya SLIMANI\*\*

Université Cheikh Anta Diop de Dakar, Sénégal\*  
mohamed.oulddeye@ucad.edu.sn

Université de Tunis El-Manar, Tunisie \*\*  
yahya.slimani@fst.rnu.tn

.....

**RÉSUMÉ.** La technologie du Cloud Computing permet de définir de nouveaux systèmes informatiques possédant une très grande capacité de calcul et de stockage. Sa réussite est due surtout à son modèle économique où l'utilisateur ne paye que les ressources qu'il a réellement consommé et à l'illusion de l'existence d'une infinité de ressources pouvant satisfaire dynamiquement l'évolution de ses besoins. Malgré ses nombreuses promesses, l'adoption de cette technologie par les entreprises est encore freinée par quelques difficultés. L'un des problèmes qui intéresse à la fois les chercheurs et les utilisateurs concerne la QoS dans les Clouds. Nous proposons, dans ce papier, une approche pour la gestion de la QoS des applications et services dans les Clouds. Cette approche est basée sur un répartiteur de charge qui maintient une liste d'instances par application. La liste des instances d'une même application est gérée et maintenue par des composants logiciels s'exécutant au niveau de machines virtuelles. Ces Composants Logiciels sont déployés à travers les fichiers images de machines virtuelles et sont indépendants de l'infrastructure sous-jacente du Cloud.

**ABSTRACT.** Cloud computing is a new technology that provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. A parallel to this concept can be drawn with the electricity grid, wherein end-users consume power without needing to understand the component devices or infrastructure required to provide the service. QoS within Grids has been a major topic of interest and continues to be actively researched in Clouds. As a cloud must provide services to many users at the same time and different users have different QoS requirements, the scheduling strategy should be developed for multiple workflows with different QoS requirements. In this paper, we propose an approach to manage QoS for applications and services in Clouds. This approach is based on a load balancer that maintains a list of instances for each application. The list of instances for the same application is updated by Software Components running on virtual machines. The Software Components are deployed in the image files of virtual machines and are independent of the underlying infrastructure of the Cloud.

**MOTS-CLÉS :** Cloud Computing, QoS, Application, Instance, Composant Logiciel

**KEYWORDS :** Cloud Computing, QoS, Application, Instance, Software Component

.....

---

## 1. Introduction

Les architectures de type Clouds représentent l'une des plus récentes évolutions des systèmes informatiques [1]. C'est un nouveau modèle de systèmes informatiques où les données et les calculs sont localisés et traités quelque part dans un réseau qui constitue le nuage (cloud). Celui-ci fait référence à un ensemble de Datacenters appartenant et maintenus par des tiers. Ce sont des systèmes très dynamiques accessibles sur Internet et ils sont caractérisés par une très grande capacité de calcul et de stockage. Du point de vue architectural, ils ressemblent à un nouveau type de Mainframe accessible sur Internet, mais ils se distinguent essentiellement par le fait que leurs capacités ne sont pas limitées comme pour le cas des Mainframes, mais plutôt ils donnent l'illusion de l'existence d'une infinité de ressources à travers des mécanismes de scalabilité et des infrastructures distribuées à l'échelle mondiale. Ils se distinguent aussi par la diversité de services et ressources fournis, allant du calcul simple jusqu'à la mise à disposition de toute une infrastructure informatique.

Les Clouds représentent une technologie qui tire profit de l'avancement de technologies de l'information et de la communication. Ils constituent une évolution et une exploitation des capacités de systèmes parallèles et distribués tels que les Clusters et les Grilles [2]. Ils sont le résultat de l'évolution de toute une panoplie de technologies [3] : les technologies liées aux architectures matérielles, les technologies de virtualisation, les technologies des architectures distribuées et les technologies de l'Internet. Ils sont considérés par certains chercheurs comme une étape vers la concrétisation de l'idée d'avoir de la prestation informatique comme un service public au même titre que l'eau, l'électricité et le téléphone. L'idée du service public informatique est de fournir du calcul et/ou du stockage selon les besoins des utilisateurs, qui ne sont plus concernés par l'emplacement où seront stockées leurs données ou exécutées leurs applications. Du point de vue économique, les Clouds permettent de payer uniquement les ressources qui sont réellement consommées contrairement aux grilles de calcul.

Il n'existe pas une définition unique pour les Clouds, mais la plupart des définitions renvoient à la fourniture de ressources matérielles, de systèmes logiciels et applications sous forme de services. Certaines caractéristiques sont essentielles pour une offre Cloud [3] : (1) un Cloud doit permettre un accès auto-service, ce qui signifie que les utilisateurs doivent pouvoir accéder, personnaliser, payer et utiliser les services sans interventions d'opérateurs humains ; (2) les services de Cloud doivent être tarifés sur une base à court terme (par exemple, à la minute), permettant aux utilisateurs de ne pas payer des ressources dès lors qu'elles ne sont plus utilisables ; (3) l'élasticité, qui veut dire que les applications peuvent solliciter automatiquement des ressources supplémentaires en cas de montée de charge et aussi libérer ces ressources si la charge démunie ; (4) Une offre Cloud doit être hautement personnalisable pour permettre à tout type d'utilisateur de pouvoir choisir les ressources qui lui conviennent. Les offres de Clouds sur Internet sont

nombreuses, mais malgré leurs nombreuses promesses et leur croissante popularité, leur adoption dans le monde des entreprises est encore freinée par quelques difficultés. L'un des problèmes qui intéresse à la fois les chercheurs et les utilisateurs concerne la QoS dans les Clouds que nous discutons dans la section 3 de ce papier. Dans la section 4, nous présentons l'architecture d'une approche pour la gestion de la QoS des applications et des services dans les Clouds. La section 2 résume les approches liées et la section 5 conclue ce papier.

---

## 2. Approches liées

La gestion de la QoS des applications et services dans les Clouds a connu ces derniers temps un intérêt particulier de la part des chercheurs comme l'atteste le nombre croissant de papiers sur le sujet [7, 8, 9, 10, 11, 12, 13]. Dans cette section, nous présentons, très brièvement, les approches les plus similaires à notre proposition.

Li et al. [7] ont proposé une méthode d'optimisation de ressources au moment de l'exécution par l'utilisation de modèles de performances dans le développement et le déploiement des applications des Clouds. Leur approche est basée sur un modèle de réseaux de files d'attente appelé LQM (ou Layered Queueing network performance Model) où pour chaque nouvelle application déployée, un modèle LQM doit être généré et maintenu pour prévoir les effets des changements au niveau des allocations des ressources. Toutefois, cette approche ne peut pas être transparente aux applications existantes par le fait qu'elle est basée sur des mesures qui doivent être faites en amont par le développeur pour la génération des modèles LQM. Nathuji et al. [8] ont démontré que les performances d'une application, s'exécutant dans un environnement Cloud, peuvent être sensiblement affectées par l'existence d'autres machines virtuelles partageant le même serveur physique. Ils présentent un framework "Q-Cloud" qui doit assurer que les performances expérimentées par les applications sont les mêmes que celles réalisées dans un environnement sans interférences des performances. Leur approche consiste à maintenir une quantité des ressources libres (appelée "head-room") sur chaque serveur physique pour compenser ses machines virtuelles affectées sur la base d'un modèle MIMO (Multi-Input, Multi-Output) capturant les relations entre les allocations de ressources et la QoS expérimentée par les machines virtuelles. Cette approche, à la différence de la notre, n'est pas indépendante vis-à-vis de la politique de placement de machines virtuelles au niveau des serveurs physiques de l'infrastructure Cloud. Calheiros et al. [9] ont proposé une technique de provision basée sur un contrôleur d'admission de requêtes des utilisateurs. Ce contrôleur peut accepter ou refuser des requêtes, suivant le nombre de requêtes déjà soumises au niveau des instances de chaque application. Leur proposition est la plus similaire à la notre à la différence que nous effectuons un traitement par des composantes des requêtes afin de prendre en compte, en plus des applications de type BoT, les applications dont les requêtes ne sont pas indépendantes. Aussi, nous effectuons, à l'aide de nos composants logiciels,

**URED**

un rééquilibrage de charge en redistribuant des requêtes non encore traitées au niveau de certaines instances qui commencent à présenter le risque de violation de la QoS.

---

### 3. Problématique de QoS

La gestion de la QoS dans les Clouds permet d'obtenir des gains pour les différents acteurs de Clouds. Pour les utilisateurs, le gain est de s'assurer de la qualité de l'exécution de leurs tâches au sein des infrastructures Clouds, qui sont souvent partagées à large échelle. Du point de vue des fournisseurs des infrastructures, le gain est de rester compétitif en offrant la garantie d'une certaine QoS, mais aussi en optimisant l'utilisation des ressources en n'allouant que celles qui sont nécessaires pour un utilisateur donné. Actuellement dans les Clouds, les utilisateurs sont facturés sur la base de ressources utilisées ou réservées et aucune garantie n'est donnée pour la QoS que ces ressources vont offrir. Aussi, dans les cas de contrats SLA (Service-Level Agreements), la non-violation de la QoS est souvent assurée par une politique de sur-provision en ressources qui est déterminée pour le pire de cas [4], d'où un double déficit : une inefficacité d'utilisation de ressources de l'infrastructure et une augmentation de coûts facturés aux clients.

Les principales difficultés pour la gestion ou la garantie de la QoS au sein de Clouds, sont : (1) la nature très dynamique de ces infrastructures où l'avènement d'un certain comportement imprévu n'est pas exclu, tel que par exemple la panne d'un serveur ou la perte d'une liaison réseau ; (2) les charges extrêmement variables qui caractérisent les applications de ces systèmes (réseaux sociaux, hébergement Web, etc.) rendent impossibles le maintien d'une allocation statique de ressources et nécessitent la mise en place d'une politique d'adaptation de ces allocations pour pouvoir maintenir un niveau de performance acceptable ; (3) les erreurs d'estimation de quantités de ressources physiques ou virtuelles qui sont nécessaires pour accomplir un niveau de qualité applicatif donné. En effet, avec les nombreux paramètres (temps CPU, ressources mémoire, flux d'I/O) qui sont utilisés au moment de l'exécution d'une application, il est difficile, voire impossible, de prévoir exactement la combinaison optimale de ressources pour accomplir l'exécution sous une contrainte de qualité comme le temps de réponse par exemple ; (4) le phénomène des interférences de performances qui résulte du partage de certaines ressources physiques de l'infrastructure Cloud et qui entraîne des interférences entre les machines virtuelles qui partagent les mêmes serveurs physiques. En effet, la virtualisation offre un certain nombre d'avantages, tels que la cohabitation de plusieurs systèmes sur une seule machine physique et l'isolation de défaillances à l'intérieur de chaque machine virtuelle garantissant ainsi la protection des autres machines du même serveur. Par contre, la virtualisation ne garantit pas l'isolation de performances [6], ce qui veut dire que les performances des applications d'une machine virtuelle peuvent changer en raison de l'existence d'autres machines virtuelles sur le même serveur physique.

Dans cet environnement distribué et dynamique, une solution de gestion de la QoS doit

permettre une efficacité dans l'utilisation de ressources de l'infrastructure en s'appuyant d'une part, sur des mécanismes efficaces d'adaptation des allocations de ressources, et d'autre part sur un modèle de prévision permettant à tout moment de calculer les ressources qui sont à réserver pour répondre à des changements dans les charges de services et applications existants ou pour assurer la QoS des nouveaux services déployés.

#### 4. Architecture proposée

Nous nous plaçons dans le cas d'un fournisseur d'applications (Application Service Provider ou ASP) qui voudrait profiter des capacités des Clouds de type IaaS (Infrastructure as a Service) pour l'exécution de ses applications, tout en offrant un niveau de QoS à ses utilisateurs et en même temps optimiser les coûts de ressources qui lui sont facturées par les fournisseurs des infrastructures Clouds. Cependant, les Clouds actuels n'offrent pas la garantie de performances, mais ils peuvent fournir à tout moment toutes les ressources nécessaires aux applications hébergées. Dans ce sens, il appartient à l'ASP de prévoir toutes les ressources nécessaires pour atteindre les performances désirées. Ce choix, qui est illustré par la figure 1, a l'avantage de permettre une abstraction des difficultés de la gestion interne des infrastructures Clouds, comme la gestion de placement de machines virtuelles au niveau des serveurs physiques ou la gestion des effets des interférences au niveau des ressources physiques partagées.

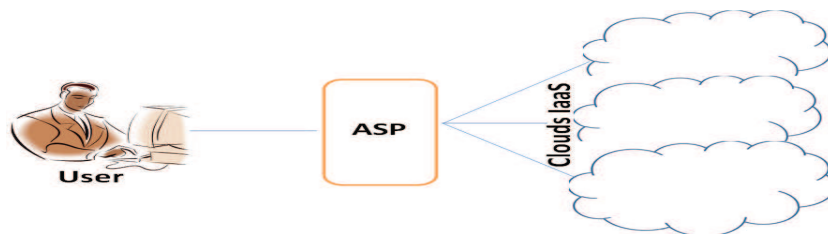


Figure 1. Acteurs de l'architecture proposée

Dans le cadre de la gestion de la QoS dans les Clouds, nous proposons une solution qui soit portable et automatiquement déployable. Le choix que nous avons fait est d'utiliser des Composants Logiciels qui sont implémentés au niveau de machines virtuelles. Ces composants sont pré-installés au niveau des fichiers images utilisés pour la création de machines virtuelles et sont donc automatiquement déployables et indépendants des infrastructures Clouds, à partir du moment où la plupart des Clouds existants autorisent les utilisateurs à déployer leurs machines virtuelles à travers leurs fichiers images. Nous considérons aussi que chaque instance d'une application donnée tourne seule avec son composant logiciel au niveau d'une machine virtuelle. L'architecture proposée est illus-

URED

trée par la figure 2. L'idée principale de cette approche est de mettre en place une politique de régulation des trafics des différentes instances des applications, afin d'atténuer ou compenser les dégradations des performances des machines virtuelles qui sont engendrées le plus souvent par l'existence d'autres machines virtuelles partageant les mêmes infrastructures physiques. La régulation des trafics est effectuée à travers la création de nouvelles instances et d'une politique d'équilibrage de charge réalisée en deux temps : dans un premier temps, au niveau du Répartiteur lors de la réception des requêtes et en deuxième temps, au niveau des composants logiciels pour redistribuer des requêtes non encore traitées au niveau de certaines instances qui commencent à présenter le risque de violation de la QoS.

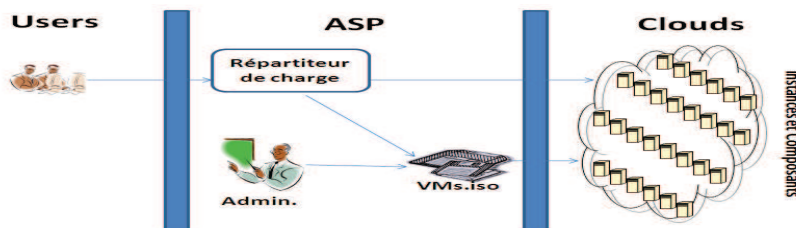


Figure 2. Les composants de l'architecture proposée

Notre proposition est une architecture distribuée composée de deux principaux composants : un *Répartiteur de charge* et des *Composants Logiciels* qui s'exécutent au niveau de machines virtuelles. Ces deux composants interagissent entre eux pour maintenir un niveau de QoS pour chaque instance.

#### 4.1. Le Répartiteur de charge

Le rôle du *Répartiteur* est la redirection de requêtes sur la base d'une liste d'instances par application qui est actualisée par les *Composants Logiciels*. Dans le cas où aucune instance n'est disponible à un instant donné, le *Répartiteur* crée une nouvelle instance de machine virtuelle à l'aide du fichier image de l'application concernée. Il existe au niveau du *Répartiteur* deux types de listes : (1) une liste simple pour les applications qui sont de type BoT (Bag of Tasks) où les requêtes sont indépendantes les unes des autres. Pour ce type d'applications, la stratégie de répartition de requêtes aux instances est de type Rond-robin ; (2) une liste étendue pour les applications ayant un mode de session ou des dépendances entre leurs requêtes. Nous désignons l'ensemble de requêtes non-indépendantes d'un utilisateur par *composante de requêtes*. Pour ce type d'application, le *Répartiteur* gère une liste d'instances étendue par la liste de composants déjà affectés à des instances. La répartition des charges pour ces applications commence par la vérification de l'appartenance de chaque nouvelle requête à l'une des composantes exist-

tantes. Si la requête appartient à une composante déjà existante, la requête sera transmise à l'instance correspondante. Sinon, une nouvelle composante sera créée au niveau de l'instance la moins chargée selon la technique Rond-Robin parmi les instances de l'application concernée. Le *Répartiteur* peut suspendre une instance à la réception d'une demande de suspension de la part de son composant logiciel. Il peut aussi détruire la machine virtuelle correspondante à une instance déjà suspendue lors de la réception d'un message de fin d'exécution de la dernière requête au niveau de cette instance. Il peut remettre dans la liste des instances une instance préalablement suspendue en cas de surcharge signalée par les instances actives.

## 4.2. Les Composants Logiciels

Chaque *composant logiciel* s'exécute avec son instance sur la même machine virtuelle. Il intercepte toutes les requêtes qui sont destinées à son instance. Si la requête fait partie d'une composante déjà hébergée au niveau de l'instance, la requête est acceptée. Si la requête est la première d'une nouvelle composante, un calcul de la charge de l'instance est nécessaire pour décider de l'acceptation ou non de cette nouvelle requête. Ce calcul est réalisé à l'aide d'une file d'attente dont les paramètres sont actualisés en permanence et aussi en considérant une taille pour une composante de requêtes. Si la requête est refusée, le composant la transfère au *Répartiteur* pour une nouvelle redirection. Un composant logiciel, qui ne maintient pas de composantes au niveau de son instance peut, sur la base d'une diminution de charge, demander sa suspension de la liste d'instances au niveau du *Répartiteur*. Cette demande, si elle est acceptée par le *Répartiteur*, sera suivie par une demande de suppression de l'instance après la terminaison de la dernière requête, sauf dans le cas où l'instance est de nouveau rajoutée par le *Répartiteur* dans la liste des instances actives.

---

## 5. Conclusion

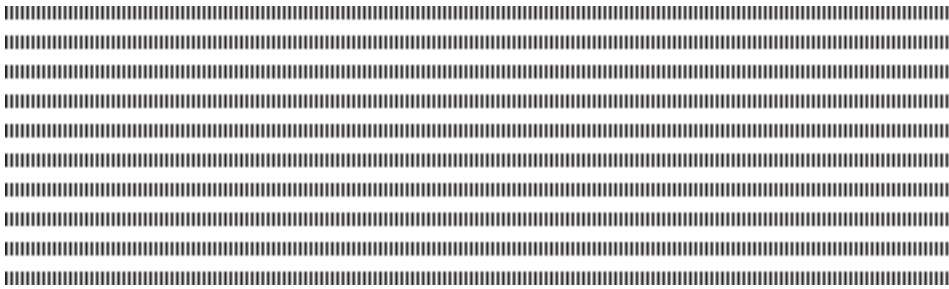
Nous avons présenté les grandes lignes d'une architecture distribuée pour la gestion de la QoS des applications et des services s'exécutant dans des infrastructures de type Clouds. Notre proposition a l'avantage d'être indépendante de toutes technologies sous-jacentes au niveau des Clouds. Elle est basée sur un répartiteur de charge et des composants logiciels s'exécutant sur les machines virtuelles qui hébergent les applications dont la QoS doit être garantie. Le maintien de la QoS dans notre proposition repose sur un double équilibrage de la charge en premier temps au niveau du répartiteur de charge et en deuxième temps au niveau des composants logiciels pour prendre en compte les dégradations de performances et pour mieux exploiter les ressources des machines virtuelles. Toutefois, des travaux d'expérimentations sont en cours pour démontrer l'efficacité de cette approche d'adaptation au moment de l'exécution.

---

## 6. Bibliographie

- [1] BORKO FURHT, ARMANDO ESCALANTE, « Handbook of Cloud Computing », *Springer*, 2010.
- [2] IAN T. FOSTER, YONG ZHAO, IOAN RAICU, SHIYONG LU, « Cloud Computing and Grid Computing 360-Degree Compared », *Grid Computing Environments Workshop, Austin, Texas, USA, 16 November 2008*.
- [3] RAJKUMAR BUYYA, JAMES BROBERG, ANDRZEJ GOSCINSKI, « Cloud Computing : Principles and Paradigms », *Wiley Series on Parallel and Distributed Computing*, vol. 81, 2011.
- [4] STEFANO FERRETTI, VITTORIO GHINI, FABIO PANZIERI, MICHELE PELLEGRINI, ELISA TURRINI, « QoS - Aware Clouds », *IEEE 3rd International Conference on Cloud Computing, Miami, Florida, USA, 5-10 July 2010*.
- [6] YOUNGGYUN KOH, ROB KNAUERHASE, PAUL BRETT, MIC BOWMAN, ZHIHUA WEN, CALTON PU, « An analysis of performance interference effects in virtual environments », *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS-2007)*, San Jose, California, USA, 25-27 April 2007.
- [7] JIM LI, JOHN CHINNECK, MARIN LITOIU, GABRIEL ISZLAI, « Performance model driven qos guarantees and optimization in clouds », *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, Vancouver, British Columbia, Canada, 23-23 May 2009.
- [8] RIPAL NATHUJI, AMAN KANSAL, ALIREZA GHAFKARHAH, « Q-clouds : managing performance interference effects for QoS-aware clouds », *Proceedings of the 5th European conference on Computer systems (EuroSys 2010)*, Paris, France, 13-16 April 2010.
- [9] RODRIGO N. CALHEIROS, RAJIV RANJAN, RAJKUMAR BUYYA, « Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments », *International Conference on Parallel Processing (ICPP 2011)*, Taipei City, Taiwan, 13-16 Sept. 2011.
- [10] QIAN ZHU, GAGAN AGRAWAL, « Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments », *Proceedings of the 19th (ACM) International Symposium on High Performance Distributed Computing (HPDC 2010)*, Chicago, Illinois, USA, 20-25 June, 2010.
- [11] VINCENT C. EMEAKAROHA, IVONA BRANDIC, MICHAEL MAURER, IVAN BRESKOVIC, « SLA-Aware Application Deployment and Resource Allocation in Clouds », *35th IEEE Annual Computer Software and Applications Conference Workshops*, Munich, Germany, 18-22 July 2011.
- [12] TOMMASO CUCINOTTA, DHAVAL GIANI, DARIO FAGGIOLI, FABIO CHECCONI, « Providing Performance Guarantees to Virtual Machines using Real-Time Scheduling », *Euro-Par Workshops*, Ischia-Naples, Italy, August 31st - September 3rd, 2010.
- [13] ZHILIN WANG, XINHUI TANG, XIANGFENG LUO, « Policy-Based SLA-Aware Cloud Service Provision Framework », *Seventh International Conference on Semantics, Knowledge and Grids (SKG)*, Beijing, China, 24-26 Oct. 2011.





## PFair scheduling of late released tasks with constrained deadlines

Sadouanouan Malo<sup>1</sup> and Annie Choquet-Geniet<sup>2</sup> and Moustapha Bikienga<sup>3</sup>

<sup>1</sup>Polytechnic University of Bobo Dioulasso. Information Technology High School. 01 BP 1091 Bobo Dioulasso 01, Burkina Faso.

<sup>2</sup>University of Poitiers. Laboratory of Applied Computer Science. 1 Av. Clément Ader BP 40109 - 86961 Futuroscope Chasseneuil-France.

<sup>3</sup>University of Koudougou. 01 BP 376 Koudougou, Burkina Faso.

E-mail: <sup>1</sup>(\*)sadouanouan.malo@ensma.fr, <sup>2</sup>annie.geniet@univ-poitiers.fr, <sup>2</sup>bmoustaph@yahoo.fr

\*Corresponding author



**RÉSUMÉ.** Les ordonnancements P-équitable ont jusque là été étudiés pour des systèmes de tâches à départs simultanés et à échéances sur requêtes. Notre objectif dans ce papier est d'étendre la définition de la P-équité aux systèmes de tâches à départs différés et à échéances contraintes puis de dégager une condition suffisante d'ordonnabilité. Enfin, des simulations que nous avons effectuées ont permis d'illustrer la pertinence de nos résultats et de mesurer l'impact de la variation des facteurs d'utilisation et de charge du système sur l'ordonnabilité du système.

**ABSTRACT.** Pfair scheduling has usually been applied in the context of synchronous periodic task systems with implicit deadlines. This paper addresses the problem of scheduling asynchronous hard real-time tasks with constrained deadlines using a Pfair strategy on multiprocessor systems. First, we extend the notion of Pfairness to the context of asynchronous tasks with constrained deadline. Then we investigate feasibility conditions, we propose a rather efficient one and we illustrate the relevance of our criteria through some simulations.

**MOTS-CLÉS :** Ordonnement - Équité - Ordonnabilité P-équitable

**KEYWORDS :** Scheduling - Fairness - Pfair feasibility



## 1. Introduction

In this paper we investigate the extension of Pfairness [1][2] in order to design Pfair scheduling schemes for asynchronous tasks with deadlines less than or equal to periods. Indeed, Pfair scheduling algorithms have been widely investigated last years [6][5]. In all these works, the notion of Pfairness is defined in the context of synchronous tasks with implicit deadlines. In our study, we first extend and adapt Pfairness definition to the new context and then we propose a sufficient but rather effective feasibility condition and then we present some simulation results.

### 1.1. Basic definitions

**Definition 1.** A task  $\tau_i = \langle r_i, C_i, D_i, T_i \rangle$  is characterized by four parameters : its first release time  $r_i$ , its per period worst case execution time  $C_i$ , its relative deadline  $D_i$  and its period  $T_i$ . The weight of task  $\tau_i$  is  $U_i = \frac{C_i}{T_i}$  and its density is  $CH_i = \frac{C_i}{D_i}$ .

We have  $r_i \geq 0$  and we assume that  $C_i, D_i$  and  $T_i$  are integral and verify  $C_i \leq D_i \leq T_i$ . Thus we have  $0 < U_i \leq 1$ . In this section we consider a set  $\Gamma$  of tasks such that  $r_i = 0$  and  $D_i = T_i$ .

**Definition 2.** A schedule for a task set  $\Gamma$  is a function  $S : \Gamma \times \mathbb{N} \rightarrow \{0, 1\}$ , such that  $\forall t \in \mathbb{N} : \sum_{\tau_i \in \Gamma} S(\tau_i, t) \leq m$  with  $S(\tau_i, t) = 1$  if task  $\tau_i$  is scheduled in slot  $[t, t + 1)$  and  $S(\tau_i, t) = 0$  otherwise. A schedule is then valid iff  $\forall \tau_i \in \Gamma, \sum_{t=0}^{t=r_i-1} S(\tau_i, t) = 0$  and

$$\forall k \in \mathbb{N}^*, \quad \sum_{t=0}^{t=r_i+(k-1)T_i+D_i-1} S(\tau_i, t) = \sum_{t=0}^{t=r_i+kT_i-1} S(\tau_i, t) = kC_i$$

### 1.2. Pfair scheduling

**Definition 3.** A schedule  $S$  is said to be **Pfair** iff

$$\forall \tau_i, t : \tau_i \in \Gamma, t \in \mathbb{N}, -1 < \omega_i(t) - \sum_{j=0}^{j=t-1} S(\tau_i, t) < 1.$$

Three Pfair scheduling algorithms,  $PF$ ,  $PD$  and  $PD^2$  [6], are known to be optimal on an arbitrary number of processors as stated in the theorem 1. They differ in the choice of tie-breaking rules.

URED

**Theorem 1.** [6] *The algorithms PF, PD and PD<sup>2</sup> are optimal for sets of synchronous independent tasks with implicit deadlines. Moreover, such a task set  $\Gamma$  has a Pfair schedule on  $m$  processors if and only if  $\sum_{\tau_i \in \Gamma} \frac{C_i}{T_i} \leq m$ .*

The remainder of the paper is organized as follows. In section 2, we prove that Pfair algorithm exists for any periodic task set and we give sufficient feasibility conditions. In section 3, we present the simulations results. Conclusions and perspectives are given in section 4.

---

## 2. Extension of Pfairness

Our aim is now to prove that fairness can be extended to all independent periodic task sets. We thus consider a periodic task set  $\Gamma$  such that  $\forall \tau_i \in \Gamma, \tau_i = \langle r_i, C_i, D_i, T_i \rangle$  with  $r_i \geq 0$  and  $D_i \leq T_i$ . We first extend the notion of Pfairness in the following way : in an ideal schedule of a periodic task set, each task  $\tau_i$  must have received at time  $t$ ,  $\omega_i(t)$  processor time units.  $\omega_i(t)$  is such that :

$$\omega_i(t) = \begin{cases} t \in [0, r_i) \Rightarrow \omega_i(t) = 0 \\ t \in [kT_i + r_i, kT_i + D_i + r_i) \Rightarrow \omega_i = kC_i + \frac{C_i}{D_i}(t - kT_i - r_i) \\ t \in [kT_i + D_i + r_i, (k+1)T_i + r_i) \Rightarrow \omega_i = (k+1)C_i \end{cases}$$

where  $k = \lfloor \frac{t}{T_i} \rfloor$  represents the instance number of the pending instance of the task.

We first have carried out some simulations using the algorithm PF. These simulations lead to two constataions : firstly, if deadlines are less than or equal to periods, if  $\sum_{i=1}^{i=n} \frac{C_i}{D_i} \leq m$ , the task set is feasible on  $m$  processors and secondly if asynchronous tasks with implicit deadlines are considered, Baruah's condition still holds. This leads us to infer the following result :

**Theorem 2.** *Given a periodic task set  $\Gamma$ , if  $\sum_{i=1}^{i=n} \frac{C_i}{D_i} \leq m$ , then  $\Gamma$  has a valid Pfair schedule on  $m$  processors over any time interval  $[0, t)$ .*

The proof of the theorem is an adaptation of the proof presented in [2] to prove theorem 1 for PF. It is based on the graph theory. We prove that a Pfair schedule exists on any time interval  $[0, L)$ . In the further,  $CTR_{\tau_i}(t)$  denotes the complete processor demand of all sub-tasks of task  $\tau_i$  whose feasibility intervals are included in  $[0, t]$ . Thus,  $CTR_{\tau_i}(t)$  is equal to  $j$  where  $j$  is such that  $d_i^j \leq t < d_i^{j+1}$ . We first define a weighted digraph  $G$  and prove that if  $G$  has an integral flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$  then the task set  $\Gamma$  has a Pfair schedule.

**Definition 4.** The Pfair-graph is the weighted digraph  $G(L)$  is defined as  $G(L) = (V, E)$  with  $V = V_0 \cup V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_5$  and  $E = E_1 \cup E_2 \cup E_3 \cup E_4$ ;  $\mathbf{V}_0 = \{\langle \text{Source} \rangle\}$ ;  $\mathbf{V}_1 = \{\langle 1, \tau_i \rangle, \tau_i \in \Gamma\}$ ;  $\mathbf{V}_2 = \{\langle 2, \tau_{i0}, 0 \rangle, \tau_i, i = 1 \dots n \text{ s.t. } r_i > 0\} \cup \{\langle 2, \tau_{i0}, j \rangle, (i, j) \text{ s.t. } i = 1 \dots n, j \in [1, \lfloor \frac{L-r_i}{T_i} \rfloor]\}$ ;  $\mathbf{V}_3 = \{\langle 3, \tau_{i0}, t \rangle, (i, t) \text{ s.t. } i = 1 \dots n, t \in [0, r_i]\} \cup \{\langle 3, \tau_i, t \rangle, (i, t) \text{ s.t. } i = 1 \dots n, t \in [kT_i + r_i, tT_i + r_i + D_i), \text{ with } 1 \leq k \leq \lfloor \frac{L-r_i}{T_i} \rfloor\} \cup \{\langle 3, \tau_{i0}, t \rangle, (i, t) \text{ s.t. } i = 1 \dots n, t \in [kT_i + D_i, (k+1)T_i), \text{ with } 1 \leq k \leq \lfloor \frac{L-r_i}{T_i} \rfloor\}$ ;  $\mathbf{V}_4 = \{\langle 4, t \rangle, t \in [0, L)\}$  and  $\mathbf{V}_5 = \{\langle \text{sink} \rangle\}$ . Edges and capacities are defined by:  $\mathbf{E}_0 = \{\langle \langle \text{source} \rangle, \langle 1, \tau_i \rangle, CTR_{\tau_i}(L) \rangle, i = 1 \dots n\}$ ;  $\mathbf{E}_1 = \{\langle \langle 1, \tau_i \rangle, \langle 2, \tau_{i0}, 0 \rangle \rangle, i = 1 \dots n \text{ s.t. } r_i > 0\}$ ;  $\mathbf{E}_2 = \{\langle \langle 2, \tau_{i0}, 0 \rangle, \langle 3, \tau_{i0}, t \rangle, 0 \rangle, (i, t) \text{ s.t. } i = 1 \dots n, t \in [0, r_i)\} \cup \{\langle \langle 2, \tau_i^j \rangle, \langle 3, \tau_i, t \rangle, 1 \rangle, (i, j, t) \text{ s.t. } i = 1 \dots n, j \in [0, CTR_{\tau_i}(L)), t \in [r_i^j, d_i^j)\} \cup \{\langle \langle 2, \tau_{i0}, j \rangle, \langle 3, \tau_i, t \rangle, 0 \rangle, (i, j, t) \text{ s.t. } i = 1 \dots n, j \in [1, \lfloor \frac{L-r_i}{T_i} \rfloor], t \in [kT_i + r_i + D_i, (k+1)T_i + r_i) \text{ with } 1 \leq k \leq \lfloor \frac{L-r_i}{T_i} \rfloor\}$ ;  $\mathbf{E}_3 = \{\langle \langle 3, \tau_{i0}, t \rangle, \langle 4, t \rangle, 0 \rangle, t \in [0, r_i)\} \cup \{\langle \langle 3, \tau_i, t \rangle, \langle 4, t \rangle, 1 \rangle, t \in [kT_i + r_i, kT_i + r_i + D_i) \text{ with } 1 \leq k \leq \lfloor \frac{L-r_i}{T_i} \rfloor\} \cup \{\langle \langle 3, \tau_{i0}, t \rangle, \langle 4, t \rangle, 0 \rangle, (i, t) \text{ s.t. } i = 1 \dots n, t \in [kT_i + r_i + D_i, (k+1)T_i) \text{ with } 1 \leq k \leq \lfloor \frac{L-r_i}{T_i} \rfloor\}$ ;  $\mathbf{E}_4 = \{\langle \langle 4, t \rangle, \langle \text{sink} \rangle, m \rangle, t \in [0, L)\}$ .

In order to prove the theorem, we first establish the following lemma.

**Lemma 1.** If the Pfair-graph  $G(L)$  has an integral flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$ , then  $\Gamma$  has a Pfair schedule on  $[0, L]$ .

*Démonstration.* Let us assume that such an integral flow exists. We first define a schedule  $S$  deduced from the Pfair -graph as :

**Definition 5.** Let  $f$  be an integral flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$  of  $G$ , we define  $SG$  as follows. For  $\tau_i \in \Gamma, t \in \mathbb{N}$ ,

$$SG(\tau_i, t) = \begin{cases} 1 & \text{if } t \in [0, L) \wedge (\exists j \in [0, CTR_{\tau_i}(L)) \text{ s.t. } f(\langle \langle 2, \tau_i^j \rangle, \langle 3, \tau_i, t \rangle, 1 \rangle) = 1) \\ 0 & \text{otherwise} \end{cases}$$

We show that  $SG$  is Pfair over the time interval  $[0, L)$ .

The size of the flow is  $\sum_{i=1}^{i=n} CTR_{\tau_i}(L)$ , thus each node of  $V_1$  is filled to capacity, i.e.

the flow carried by each link from source to  $\langle 1, \tau_i \rangle$  is equal to  $CTR_{\tau_i}(L)$ . Each node in  $V_1$  has exactly  $CTR_{\tau_i}(L)$  outgoing edges of capacity 1, the other outgoing edges have a null capacity thus they receive a flow equal to 0. Thus each node  $\langle 2, \tau_i^j \rangle$  of  $V_2$

**URED**

receives a flow equal to 1. Then, each node  $\langle 2, \tau_i^j \rangle$  has one single outgoing edge which carries a flow equal to 1, the other outgoing edges carry a flow equal to 0. Now, each node of  $V_3$  has exactly one outgoing edge. This edge carries a flow equal to the flow carried by its incoming edge. In the same way, the nodes in  $V_4$  have one single outgoing edge, which carries the cumulated flow carried by their incoming edges. Since the capacity of this outgoing edge is  $m$ , at most  $m$  incoming edges carry a flow equal to 1. Thus at most  $m$  sub-tasks are processed. Now, the potential exists for a task  $\tau_i$  to get scheduled twice at the same time (if  $r_i^j = d_i^{j-1} - 1$ ). But since the edge from  $\langle 3, \tau_i, t \rangle$  to  $\langle 4, t \rangle$  has a capacity equal to 1, this situation is avoided : if this edge carries a flow equal to 1, only one incoming edge carries a non-null flow. Thus two different processed sub-tasks cannot belong to the same task. Thus at any time  $t$  in  $[0, L)$ , there exists at most  $m$  tasks such that  $SG(\tau_i, t) = 1$ . Furthermore,  $f(\langle \langle 2, \tau_i^j \rangle, \langle 3, \tau_i, t \rangle, 1 \rangle) = 1$  implies that  $r_i^j \leq t < d_i^j$  thus, each processed sub-task is processed in its feasibility window. Finally, each sub-task is effectively processed. Indeed, there exists  $\sum_{i=1}^{i=n} CTR_{\tau_i}(L)$  sub-tasks in  $[0, L)$ . Following the definition of  $SG$ , the number of processed sub-tasks is equal to the global incoming flow of vertices of  $V_3$ . And the global input flow of  $V_3$  is constant, equal to  $\sum_{i=1}^{i=n} CTR_{\tau_i}(L)$  by definition of the flow. Thus each sub-task is processed. The schedule  $SG$  is thus Pfair on  $[0, L)$ .  $\square$

We now prove the existence of an integer flow of size  $\sum_{i=1}^{i=n} CTR_{\tau_i}(L)$ . We use the following flow assignments :

**Definition 6.** Let  $f$  be the flow defined as :

$$\begin{aligned}
f(\langle source \rangle, \langle 1, \tau_i \rangle, (L)) &= CTR_{\tau_i}(L), f(\langle 1, \tau_i \rangle, \langle 2, \tau_{i0}, 0 \rangle, 0) = 0, \\
f(\langle 1, \tau_i \rangle, \langle 2, \tau_i, j \rangle, 1) &= 1, f(\langle 1, \tau_i \rangle, \langle 2, \tau_{i0}, j \rangle, 0) = 0, \\
f(\langle 2, \tau_{i0}, 0 \rangle, \langle 3, \tau_{i0}, t \rangle, 0) &= 0, f(\langle 2, \tau_{i0}, j \rangle, \langle 3, \tau_{i0}, t \rangle, 0) = 0, \\
f(\langle 2, \tau_i^j \rangle, \langle 3, \tau_i, r_i^j \rangle, 1) &= CH_i - j + r_i^j CH_i \\
d_i^j - 1 = r_i^{j+1} \Rightarrow f(\langle 2, \tau_i^j \rangle, \langle 3, \tau_i, d_i^j - 1 \rangle, 1) &= j + 1 - r_i^{j+1} CH_i \\
otherwise \Rightarrow f(\langle 2, \tau_i^j \rangle, \langle 3, \tau_i, t \rangle, 1) &= CH_i \\
f(\langle 3, \tau_{i0}, t \rangle, \langle 4, t \rangle, 0) &= 0, f(\langle 3, \tau_{i0}, t \rangle, \langle 4, t \rangle, 1) = CH_i, \\
f(\langle 4, t \rangle, \langle sink \rangle, m) &= \sum_{\substack{\tau_i \in \Gamma \\ r_i + kT_i \leq t < r_i + kT_i + D_i}} CH_i
\end{aligned}$$

**Lemma 2.**  $f$  is a flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$  of  $G(L)$ .

*Démonstration.* We first prove that the capacity constraints are met. Edges in  $E_0, E_1$  are filled to capacity, and edges in  $E_3$  carry flows either equal to 0 or to the density  $CH_i$  which is less than or equal to 1, thus capacity constraints are met. If an edge in  $E_4$  is

Malo et al.

considered, it carries a flow  $\sum_{\substack{\tau_i \in \Gamma \\ r_i + kT_i \leq t < r_i + kT_i + D_i}} CH_i \leq \sum_{\tau_i \in \Gamma} CH_i$  now by assumption

we have  $\sum_{\tau_i \in \Gamma} CH_i \leq m$ , so the capacity constraint is met. Finally, for edges in  $E_2$ , we

must prove that  $CH_i - (jr_i^j CH_i) \leq 1$  and  $(j+1) - r_i^{j+1} CH_i \leq 1$  if  $d_i^j - 1 = r_i^{j+1}$ .

We have  $r_i^j = \lfloor \frac{j}{CH_i} \rfloor$  thus  $\frac{j}{CH_i} - 1 < r_i^j \leq \frac{j}{CH_i}$  thus  $CH_i < r_i^j CH_i - j \leq 0$  thus

$0 < CH_i - (j - r_i^j CH_i) \leq CH_i \leq 1$ . We prove that  $(j+1) - r_i^{j+1} CH_i \leq 1$  using

similar arguments. Thus capacity constraints are all met. We must then show that the flow

is preserved at every inner vertex. For null capacity nodes, the flow is clearly preserved.

For any node  $\langle 1, \tau_i \rangle$  of  $V_1$ , the incoming flow is  $CTR_{\tau_i}(L)$ , and the outgoing flow

is  $CTR_{\tau_i}(L)$  too. Each vertex  $\langle 2, \tau_i, j \rangle$  has an incoming flow of 1. Each vertex  $\langle 2, \tau_i, j \rangle$

has  $d_i^j - 1 - r_i^j$  outgoing edges. Then the flow out of  $\langle 2, \tau_i, j \rangle$  is, if  $d_i^j - 1 = r_i^{j+1}$ ,

$CH_i - (j - r_i^j CH_i) + CH_i (d_i^j - r_i^j - 2) + (j+1) - r_i^{j+1} CH_i$  which simplifies to 1.

Otherwise, we have  $d_i^j - 1 \neq r_i^{j+1}$  thus  $\lfloor \frac{j+1}{CH_i} \rfloor - 1 \neq \lfloor \frac{j+1}{CH_i} \rfloor$  which means that  $\frac{j+1}{CH_i}$  is

integral thus  $d_i^j = \frac{j+1}{CH_i}$ . Now, the flow out is  $CH_i - (j - r_i^j CH_i) + CH_i (d_i^j - r_i^j - 1)$

which then simplifies to 1. There is only one outgoing edge leaving any vertex  $\langle 3, \tau_i, t \rangle$

of  $V_3$ , which carries a flow equal to  $CH_i$ . If  $d_i^j - 1 = r_i^{j+1}$ , then there are two incoming

edges which carry a flow of size  $(j+1) - r_i^{j+1} CH_i + CH_i - ((j+1) - r_i^{j+1} CH_i) =$

$CH_i$ . Otherwise there is only one incoming edge which carries a flow equal to  $CH_i$ .

We consider finally a vertex  $\langle 4, t \rangle$  of  $V_4$ . Its incoming edges with non zero capacity are

edges  $(\langle 3, \tau_i, t \rangle, \langle 4, t \rangle, 1)$  with  $r_i + kT_i \leq t < r_i + kT_i + D_i$ . Thus the incoming flow

is  $\sum_{\substack{\tau_i \in \Gamma \\ r_i + kT_i \leq t < r_i + kT_i + D_i}} CH_i$ , which is thus equal by definition to the flow of the unique

outgoing edge. Thus, we proved that the flow is preserved at any inner node. Thus  $f$  is a

flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$ .  $\square$

Now Lemma 2 implies the existence of a fractional flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$  for

the Pfair-graph  $G(L) = (V, E)$ . Since capacities are integral, this implies the existence

of an integral flow of size  $\sum_{\tau_i \in \Gamma} CTR_{\tau_i}(L)$  in  $G(L)$  [3]. Then Lemma 1 proves that a

Pfair schedule can be constructed. This proves the theorem 2.

Then we extend the algorithm *PF* to periodic task sets ( $r_i \leq 0, D_i \leq T_i$ ). Here, a

task can be *Urgent*, *Tnegru* or *Contending* if  $t \in [kT_i + r_i, kT_i + r_i + D_i)$  and is *Idle*

if  $t < r_i$  or  $t \in [kT_i + r_i + D_i, (k+1)T_i)$ . The extension is then straightforward : at

each time  $t$ , *Urgent* tasks are scheduled, *Contending* tasks are sorted and the firsts of them

are allocated to the remaining processors.

### 3. Simulation results

The next point of interest is to determine whether this condition is efficient. We thus investigate the soundness of our bound. For that purpose, we have carried out some simulations. We have first implemented a task set simulator and a scheduler based on our extension of *PF*. We have then generated a significant number of task sets with different

values of either  $U = \sum_{i=1}^{i=n} \frac{C_i}{T_i}$  or  $CH = \sum_{i=1}^{i=n} \frac{C_i}{D_i}$ . In order to limit the scheduling step, we

have generated periods according to Goossens methodology [4], which permits to have a bound for the hyperperiod (the LCM of the task periods). For our simulations the upper bound of hyperperiods is set to 210. Offsets, constrained deadlines and WCET are chosen uniformly within respectively the intervals  $[0, T_i]$ ,  $[1, T_i]$  and  $[1, D_i - 1]$ . Then we have scheduled them with our adapted *PF* policy, in order to estimate the ratio of feasible sets among them. All simulations have been carried out over a time interval included in

$\left[0, \max_{i \in [1, n]} (r_i) + 2T\right)$ . Then we have considered several cases. Task sets can be synchronous or asynchronous, with implicit or constrained deadlines. They are characterized by either  $U \leq m$  or  $CH \leq m$  or  $CH > m$ . For each case, we have generated a sample of 5000 tasks sets for simulations. As expected, we find a Pfair feasibility rate of 100% for systems with implicit deadlines : for synchronous systems, it corresponds to Baruahs theorem (theorem 1), and for asynchronous systems, it comes from our result (theorem 2). For the other cases, we conclude that for constrained task sets,  $U \leq m$  is no more a sufficient condition, since there exists Pfair unfeasible task sets with a utilization factor less than  $m$  and  $CH \leq m$  is not a necessary condition since there exist Pfair feasible task sets with  $CH > m$ .

Then we refined our simulations in order to determine the incidence of  $U$  or  $CH$  on Pfair feasibility. We have considered systems  $m$  processors. For each case, we generate samples for different values of  $CH$  between  $m$  and  $m + 1$ , namely  $CH = m + \frac{k}{10}$  ( $0 \leq k \leq 10$ ). For each value of  $CH$ , we again determine the ratio of Pfair feasible task sets. We have seen that this rate decreases rather quickly when  $CH$  increases. If  $CH$  remains close to  $m$ , the rate of valid system remains high, but the slope of the curve is high and consequently the rate becomes very small if  $CH$  approaches  $m + 1$ . We can conclude from these results that our bound is rather good in the sense that only few systems rejected by our test are in fact Pfair feasible.

We also investigated the correlation between  $U$  and the Pfair feasibility. We consider systems with  $m$  processors, and synchronous task systems with constraint deadlines. We see that if  $U$  is close to  $m$ , then quite no systems are Pfair feasible. But if  $U$  is less than  $\frac{m}{2}$ , we have 100% of Pfair feasible systems. Further investigations must be done here.

---

## 4. Conclusions

We have considered real-time applications running on a multiprocessor platform. For such systems, we have extended the notion of Pfairness to any set of periodic independent task set with implicit or constrained deadlines. We have considered as well late released tasks as constraint deadlines. We have proposed a sufficient condition and given an adapted version of  $PF$ . We proved the existence of a Pfair schedule if  $CH = \sum_{i=1}^{i=n} \frac{C_i}{D_i} \leq m$ .

Then we have presented simulation results, which illustrate the soundness of our sufficient conditions. These simulations have shown that if  $CH$  increases from  $m$ , the rate of Pfair feasible systems decreases quickly. We also present some results about the incidence of

the utilisation factor  $U = \sum_{i=1}^{i=n} \frac{C_i}{T_i}$ . We have speculated about the existence of a lower  $U$ , which can be used for any periodic task set with implicit or constrained deadlines.

---

## 5. Bibliographie

- [1] S. Baruah, J. Gehrke, and C.G. Plaxton. Fast scheduling of periodic tasks on multiple resources. In *Proceedings of the 9<sup>th</sup> International Parallel Processing Symposium*, pages 280–288, April 1995.
- [2] S.K. Baruah, N.K. Cohen, C.G. Plaxton, and D.A. Varvel. Proportionate progress : a notion of fairness in resource allocation. *Algorithmica*, 15 :600–625, 1996.
- [3] L.R. Ford Jr and D.R. Fulkerson. *Flows in networks*. Princeton University Press, 1962.
- [4] C. Macq and J. Goossens. Limitation of the hyper-period in real-time periodic task set generation. In Teknea, editor, *Proceedings of the 9th international conference on real-time systems*, pages 133–148, Paris France, March 2001. ISBN 2-87717-078-0.
- [5] S. Ramamurthy. Scheduling periodic hard real-time tasks with arbitrary deadlines on multiprocessors. In *IEEE Real-Time Systems Symposium*, 2002.
- [6] A. Srinivasan, P. Holman, and J.H. Anderson. Integrating aperiodic and recurrent tasks on fair- scheduled multiprocessors. In *14th Euromicro Conference on Real- Time Systems*, pages 189–198, Vienna, Austria, June 2002. IEEE Computer Society.



## Keyword Index

Actor model	89
Agent-based model	89
Algorithmes distribués	1
Application	104
apprentissage	73
approche de construction dontologie	41
Authentification	57
Auto-stabilisation	1
autorisation	73
bilharziose	17
changement climatique	41
Cloud Computing	104
Clustering	1
Composant Logiciel	104
Computational geometry	97
Computed Tomography images	97
contextual preference rule	33
Contrôle d'Accès	57
data	65
dispositif	73
données	65
données complexes	49
Dynamic graphs	25
entrepôt de données	49
Environment	89
Exploration	25
extraction de connaissances	49
Fairness	112
Fièvre de la Vallée du Rift	49
geometric modeling	97
Grid computing	81
hiérarchie	49
Image processing	97
Instance	104
KEYWORDS : grid	65

laboratoire distant	73
Mobile agent	25
Modélisation	9
modélisation multi-agents	17
modélisation multidimensionnelle	49
Mot de Passe	57
MOTS-CLÉS : grille	65
multi échelle	49
méta-modèle	41
Natural resources	89
ontologie	9, 41
organisation virtuelle	73
Peer to peer networks	81
Pfair feasibility	112
plateforme GAMA	17
pore space	97
preference elicitation	33
projet CLUVA	41
propagation bilharziose	9
PV-graph	25
QoS	104
Réseaux ad hoc	1
Réseaux de Capteurs Sans Fil	57
Scheduling	112
simulation	17
Soil micro-structures	97
système complexe	9
système pair-à-pair	65
système peer to peer	65
user profile mining	33
Volume segmentation	97
vulnérabilité urbaine	41
Web services	81
épidémiologie	9

## Author Index

Ahmed Mouhamadou, Wade	25
Alassane, Bah	49
Annie, Choquet-Geniet	112
Ba, Mandicou	1
Camara, Gaoussou	9, 17
Cambier, Christophe	49
Cisse, Papa Alioune	17
Coly, Adrien	41
David, Ilcinkas	25
de Amo, Sandra	33
Despres, Sylvie	9
Diallo, Mouhamadou Saliou	33
Diop, Alassane	73
Diop, Cheikh Talibouya	33
Diop, Ibrahima	41
Djedidi, Rim	9
Fanta, Bouba	49
Faye, Youssou	57
Flauzac, Olivier	1, 65, 81
Fotsing, Eric	89
Giacometti, Arnaud	33
Gueye, Amadou Dahirou	73
Gueye, Bassirou	81
Haggar, Bachar Salim	1
Kameni, Eric Désiré	89
Li, Haoyuan D	33
Lo, Moussa	9, 17, 41
Monga, Olivier	97
Moustapha, Bikienga	112
Ndiaye, Samba	49, 65
Ndione, Jacques André	49
Ngom, Ndeye Fatou	97
Niang, Ibrahima	1, 57, 65, 81
Nolot, Florent	1

Ould Deye, Mohamed Mahmoud	104
Ouya, Samuel	73
Sadouanouan, Malo	112
Slimani, Yahya	104
Soulet, Arnaud	33
Steffenel, Luiz Angelo	65
Thierno Ahmadou, Diallo	65
Wouter, De Groot	89

## INSTRUCTIONS AUX CONTRIBUTEURS

1. Le texte, saisi au format Word .doc ou .rtf, doit nous parvenir sous forme électronique (fichier) et sur papier (3 exemplaires). Les contributeurs qui se trouvent hors du Sénégal peuvent envoyer leur article par courrier électronique seulement (fichier attaché) à : ured@ugb.sn.
- 2.- Tous les contributeurs sont tenus d'indiquer leur adresse électronique.
3. Les titres des ouvrages sont en italiques (et ne sont ni soulignés, ni mis en gras), tandis que les titres des articles sont mis entre guillemets sans aucune mise en forme particulière.
4. La présentation bibliographique doit respecter les normes en vigueur. Pour les études littéraires, il est préférable d'adopter le format suivant :  
GENET, Jean. *Lettres à Roger Blin*. Paris : Gallimard, 1966.  
DALMON, Vincent. « Théâtre et société ». *Convergences Culturelles*, n° 12, 2003, pp. 31-46.  
Pour les études linguistiques et apparentées, les contributeurs peuvent utiliser le format suivant :  
ADAM, J.M. & GOLDENSTEIN J. P. (1976). *Linguistique et discours littéraire*. Paris : Larousse.  
LAROCHE, Jean-Michel (1964a). « Les couloirs du bonheur ». *Reflets Philosophiques*, n° 21, pp. 89-104.
5. Les citations de plus de trois lignes doivent être isolées (marge de 1,5 à 2 cm des quatre côtés).
6. Un texte français doit être accompagné d'un résumé en anglais et un texte anglais d'un résumé en français. Pour d'autres langues, le résumé doit être en français ou en anglais.
7. L'article doit être accompagné d'une bibliographie.
8. Les opinions émises dans les articles n'engagent que leurs auteurs.
9. Tout texte est soumis à l'appréciation de deux lecteurs. Ne sont publiés que les textes qui ont recueilli deux avis favorables

## APPEL D'ARTICLE

*Université, Recherche et Développement* est une revue dont l'un des deux numéros annuels porte sur les arts, les lettres, les sciences humaines, sociales, juridiques et économiques. Outil de réflexion et d'actualité bannissant le sectarisme et le parti pris, *Université, Recherche et Développement* ambitionne de contribuer au développement de l'Afrique et de se faire l'écho des préoccupations individuelles ou communautaires.

*Université, Recherche et Développement* accueille tous articles ou comptes rendus (publications, colloques) susceptibles d'intéresser la communauté des chercheurs et des décideurs. Elle porte un intérêt particulier aux travaux émanant de chercheurs isolés ou d'universitaires qui, faute de support éditorial, demeurent souvent contraints au silence.